

UNIVERSITÄT HEIDELBERG

DOCTORAL THESIS

**Metrics of Graph-Based Meaning
Representations with Applications from
Parsing Evaluation to Explainable NLG
Evaluation and Semantic Search**

Juri Opitz

This thesis is submitted for the degree of Doctor of Philosophy

First examiner: Prof. Dr. Anette Frank
Second examiner: Prof. Dr. Sebastian Padó
Date of final version: January 18, 2024.

Acknowledgements

First and foremost I would like to thank Anette Frank for her invaluable guidance during the last years. Working together has greatly helped me grow, as a researcher, and as a person.

I am also grateful to Sebastian Padó for agreeing to be the second reviewer of this thesis, and to a few persons at the computational linguistics department for their great and uplifting company that I gladly enjoyed over the last years: Maria Becker, Esther van den Berg, Angel Daza, Xyian Fu, Éva Mújdricza-Maydt, Todor Mihaylov, Debjit Paul, Letitia Parcalabescu, Moritz Plenz, Julius Steen, and Philipp Wiesenbach (special thanks to Julius and Moritz for their feedback on a draft of this thesis).

Abstract

“Who does what to whom?” The goal of a *graph-based meaning representation* (in short: MR) is to represent the meaning of a text in a structured format. With an MR, we can explicate the meaning of a text, describe occurring events and entities, and their semantic relations. Thus, a *metric of MRs* would measure a distance (or similarity) between MRs. We believe that such a meaning-focused similarity measurement can be useful for several important AI tasks, for instance, testing the capability of systems to produce meaningful output (system evaluation), or when searching for similar texts (information retrieval). Moreover, due to the natural explicitness of MRs, we hypothesize that MR metrics could provide us with valuable explainability of their similarity measurement. Indeed, if texts reside in a space where their meaning has been isolated and structured, we might directly see in which aspects two texts are actually similar (or dissimilar).

However, we find that there is not much previous work on MR metrics, and thus we lack fundamental knowledge about them and their potential applications. Therefore, we make first steps to explore MR metrics and MR spaces, focusing on two key goals: 1. Develop novel and generally applicable methods for conducting similarity measurements in the space of MRs; 2. Explore potential applications that can profit from similarity assessments in MR spaces, including, but (by far) not limited to, their ‘classic’ purpose of evaluating the quality of a text-to-MR system against a reference (aka parsing evaluation).

We start by analyzing contributions from previous works that have proposed MR metrics for parsing evaluation. Then, we move beyond this restricted setup and start to develop novel and more general MR metrics based on i) insights from our analysis of the previous parsing evaluation metrics and ii) our motivation to extend MR metrics to similarity assessment of natural language texts. To empirically evaluate and assess our generalized MR metrics, and to open the door for future improvements, we propose the first benchmark of MR metrics. With our benchmark, we can study MR metrics through the lens of multiple metric-objectives such as sentence similarity and robustness.

Then, we investigate novel applications of MR metrics. First, we explore new ways of applying MR metrics to evaluate systems that produce i) text from MRs (MR-to-text evaluation) and ii) MRs from text (MR parsing). We call our new setting *MR projection-based*, since we presume that one MR (at least) is unobserved and needs to be approximated. An advantage of such projection-based MR metric methods is that we can ablate a costly human reference. Notably, when visiting the MR-to-text scenario, we touch on a

much broader application scenario for MR metrics: explainable MR-grounded evaluation of text generation systems.

Moving steadily towards the application of MR metrics to general text similarity, we study MR metrics for measuring the meaning similarity of natural language arguments, which is an important task in argument mining, a new and surging area of natural language processing (NLP). In particular, we show that MRs and MR metrics can support an explainable and unsupervised argument similarity analysis and inform us about the quality of argumentative conclusions.

Ultimately, we seek even more generality and are also interested in practical aspects such as efficiency. To this aim, we distill our insights from our hitherto explorations into MR metric spaces into an explainable state-of-the-art machine learning model for semantic search, a task for which we would like to achieve high accuracy *and* great efficiency. To this aim, we develop a controllable metric distillation approach that can explain how the similarity decisions in the neural text embedding space are modulated through interpretable features, while maintaining all efficiency and accuracy (sometimes improving it) of a high-performance neural semantic search method. This is an important contribution, since it shows i) that we can alleviate the efficiency bottleneck of computationally costly MR graph metrics and, vice versa, ii) that MR metrics can help mitigate a crucial limitation of large ‘black box’ neural methods by eliciting explanations for decisions.

Contents

Acknowledgements	iii
Abstract	v
1 Introduction	1
1.1 Motivation	1
1.2 Research questions	5
1.3 Contributions of this thesis	6
1.4 Thesis overview	7
1.5 Generated papers and resources	11
2 Background	13
2.1 Graphs and graph-based MRs	13
2.1.1 Meaning representations as graphs	13
2.1.2 Same meaning, different structure: graph translations	15
2.2 Abstract Meaning Representation (AMR)	18
2.2.1 Pillars of AMR	18
2.2.2 A deeper look into the AMR toolbox	21
2.2.3 From text to AMR and from AMR to text	23
2.3 Graph metrics and metric space	24
2.3.1 Graph similarity measures	24
2.3.2 Discussion	25
2.4 Measuring MR similarity: can we use a graph measure off the shelf?	26
2.5 Metric performance evaluation	28
2.6 Unsupervised versus supervised metrics	29
3 Related work	31
3.1 MR metrics	31
3.1.1 AMR metrics	31

3.1.2	Discussion	35
3.2	Meaning focused evaluation of natural language generation	37
3.3	Metric extrapolation: Quality estimation of predicted structures	38
3.4	Semantic textual similarity	39
3.4.1	Data sets of human text similarity	39
3.4.2	Automatic methods for rating text similarity	43
3.4.3	Explainability of decisions	44
I	MR metric analysis and development	47
4	MR metrics: assessment and development	49
4.1	Chapter outline	49
4.2	Assessment of MR metrics through eight principles	50
4.3	Using our principles to assess AMR metrics	55
4.3.1	AMR metric principle analysis I–VII	55
4.3.2	Towards enabling principle VIII with a novel metric: S ² MATCH	64
4.4	Summary of our metric analyses	66
4.5	Discussion: Limits of principle-based metric analysis and outlook	68
4.5.1	On dueling principles	68
4.5.2	Parallels from machine translation evaluation research	69
4.5.3	We need better MR metrics: Making the case with an example	70
4.6	Building novel MR metrics from Weisfeiler-Leman	71
4.6.1	Basic Weisfeiler-Leman Kernel (WLK)	71
4.6.2	Wasserstein Weisfeiler-Leman (WWLK)	73
4.7	WWLK _θ with 0 th -order optimization	76
4.8	Taking a step back: principle analysis of WLK and WWLK	78
4.9	Discussion	79
5	Extended empirical studies on MR metrics	81
5.1	Chapter outline	81
5.2	BAMBOO _{MR} : A first benchmark for MR metrics	82
5.2.1	Human similarity objectives	82
5.2.2	Robustness challenges	84
5.3	Experimental insights from BAMBOO _{MR}	89
5.3.1	BAMBOO _{MR} studies previous metrics	91

5.3.2	BAMBOO ₂ assesses MR metrics	93
5.3.3	Analyzing hyper-parameters of our novel metrics WLK and WWLK	94
5.3.4	Revisiting the data quality in BAMBOO ₂	95
5.3.5	Alignment discussion	96
5.3.6	Conclusions from BAMBOO ₂ results	98
5.4	Evaluating strong parsers with automatic and human AMR metrics	99
5.4.1	Study Setup: Data creation and MR metric setup	100
5.4.2	Study I: System-level scoring	104
5.4.3	Study II: Metric accuracy on parse level	107
5.4.4	Metric specificity	110
5.5	Discussion of limitations and recommendations for evaluating strong AMR parsers	112
5.5.1	Limitations	112
5.5.2	Recommendations for parser selection and metric improvement perspectives	113
5.6	Discussion	114
5.7	Creating a live benchmark with metric versioning	115

II MR metrics for novel evaluation applications 117

6	MRs in NLG evaluation	119
6.1	Chapter outline	119
6.2	Motivation: Why standard metrics are insufficient for MR2text evaluation	120
6.3	Casting meaning and form into a metric: \mathcal{MF}_β	122
6.3.1	From principles to \mathcal{MF}_β	122
6.3.2	REMATCH: Measuring meaning with MR and MR metrics	124
6.3.3	Parameterizing form with LMs	125
6.3.4	Goals of our pilot studies	127
6.4	Study I: Assessing potential for enhanced evaluation interpretability	127
6.4.1	Interpretability of system rankings	129
6.4.2	MR distance via REMATCH explains (re-)rankings	130
6.4.3	MR distance via REMATCH explains negation error	131
6.4.4	MR distance via REMATCH explains SRL error	132
6.4.5	Assessing aspectual text quality using fine-grained MR distances	132
6.5	Study II: Probing vulnerabilities of our approach	134

6.5.1	The parser: Achilles' heel of $\mathcal{M}\mathcal{F}_\beta$?	134
6.5.2	The <i>Form</i> component of $\mathcal{M}\mathcal{F}_\beta$	137
6.6	Discussion	139
7	AMR quality estimation	141
7.1	Chapter outline	141
7.2	Motivation: rating MR quality in the absence of human reference	142
7.3	Task formalization	143
7.4	Model I: LSTM on enriched linearized graphs	143
7.5	Model II: Novel MR-as-image encoding with CNN	145
7.5.1	MR as image with latent channels	145
7.5.2	A lightweight CNN to rate AMR quality	147
7.6	Multi-quality dimensions	150
7.7	Experimental data construction	150
7.8	Experiments on MR quality prediction	152
7.8.1	Results	153
7.8.2	Analysis	155
7.9	Discussion	157
III	MR metrics for effective semantic similarity	159
8	Exploring argumentation with MR metrics	161
8.1	Chapter outline	161
8.2	Research questions	162
8.3	Hypotheses	162
8.4	Argument Similarity through MR Metrics	164
8.4.1	Models	164
8.4.2	Implementation	165
8.4.3	AMR metric variants for exploring argument similarity	166
8.5	Argument Similarity Prediction with MR Metrics: Experiments	166
8.5.1	Setup	166
8.5.2	Results	167
8.6	Analyses & Explainability	169
8.6.1	Fine predictors of argument similarity	169
8.6.2	Example case with alignment	170

8.6.3	Investigations of conclusion quality	172
8.6.4	Can we predict conclusion quality?	173
8.6.5	Conclusion usefulness	174
8.7	Discussion	176
9	Building efficient and effective similarity models from MR metrics	179
9.1	Chapter outline	179
9.2	Pilot study: Fast similarity with learned SMATCH	180
9.2.1	Learning NP-hard graph metric: problem definition and models	181
9.2.2	Evaluation	184
9.2.3	Discussion	186
9.3	<i>Efficient, explainable and effective</i> similarity metrics	188
9.4	Structuring embedding spaces with graph metric guidance	189
9.4.1	Structured embedding spaces: Formal problem definition and objective	189
9.4.2	Learning to partition the semantic space	190
9.4.3	Preventing catastrophic forgetting	191
9.4.4	Global objective	192
9.5	AMR metrics and data construction	192
9.5.1	Global AMR similarity	192
9.5.2	Aspectual AMR similarity	192
9.5.3	Data setup	193
9.6	Evaluation Study Setup	194
9.7	Evaluation of S ³ BERT space partitioning	194
9.8	Correlation with human judgements	196
9.8.1	Sentence semantic similarity	197
9.8.2	Argument similarity	199
9.8.3	Ablation and parametrization experiments	199
9.8.4	AMR metric approximation inspection	200
9.9	Data analyses with S ³ BERT	201
9.9.1	Studying S ³ BERT predictions	201
9.9.2	Studying predictors of human scores	203
9.9.3	Evaluation with a CheckList	204
9.10	Discussion	204

IV	Conclusions and future work	207
10	Conclusions and outlook	209
10.1	Conclusions	209
10.2	Decision guide for MR metric application	211
10.3	Outlook and future work	213
A	Appendix	217
A.1	On the soundness of comparing MR-generated sentences in the MR domain	217
A.2	Form predictor selection experiment	218
A.3	Fine-tuning the conclusion generator	219
A.4	Hyper-parameter setups	219
A.4.1	Sequence-to-sequence network parameters	219
A.4.2	CNN network parameters	219
A.5	S3BERT Hyper-parameters and training	221
A.6	Results on semantic CheckList	221
	Bibliography	231

Chapter 1

Introduction

After discussing the main motivation that underlies this thesis (1.1), we outline its key research questions (Section 1.2) and summarize its contributions (Section 1.3). Finally, we give an overview of the thesis' structure (Section 1.4) and point to the papers and resources that have provided a main proportion of the fabric from which this thesis is woven (Section 1.5).

1.1 Motivation

“Who does what to whom?” The main goal of Meaning Representations (MRs) is to crystallize the meaning of a text in a structured and explicit format. Therefore, MRs lie at the heart of *semantics* (from Ancient Greek: *sēmantikós*, “significant”), which is the study of reference, meaning, or truth. Interestingly, graphs provide us with a powerful and intuitive means for describing MRs: We simply represent occurring events and entities as nodes and connect them with labeled semantic edges to express their relationships, e.g., to represent an event (“what?”) that involves an *agent* (“who?”), a *patient* (“whom?”) and possibly other items such as *location* (“where?”), *time* (“when?”), or *cause* (“why?”).

In this thesis, we focus on metrics for measuring *distances between MRs*. Given some objects in some space, the space becomes a *metric space*, when we have a measure that shows us a distance (or equivalently: a similarity) between any two objects, informing us about how close (distant) the two objects are. So we want to measure distance in a *explicit semantic space*, where meaning is captured in a more isolated and understood form.

What could we gain from a metric that measures distance (or similarity) in an explicit semantic space? To better understand our general motivation, let us first discuss the importance of assessing text similarity for natural language processing (NLP). In fact, the task of assessing textual similarity bridges all kinds of different NLP areas. For instance,

similarity metrics are found at the heart of search engines for text and image retrieval (e.g., ‘Google’), where we need to calculate similarities between a query document and a other documents of a corpus to select the documents that will be returned to a user. As an integral part of evaluation and benchmarking, metrics inform us about the selection of NLP systems and drive NLP system developments. Furthermore, texts can be clustered and classified using metrics, both being very general problems that have countless applications.

Of course, also depending on the concrete application, we could adopt all kinds of different views to measure the similarity of texts.¹ However, generally, it is desirable that a metric is a *semantic* metric, i.e., a metric that considers meaning before form. Let us think about a popular text metric that, up until today, constitutes the backbone of many text search engines (Beel et al., 2016): computing the amount of overlapping tokens from two text documents. For instance, consider an image retrieval system, where we would like to detect which two of the four images/captions are most likely to describe the same scene:

1. *The dog runs after the cat.*
2. *The cat runs after the dog.*
3. *A kitten is chased by a pupper.*
4. *The ketchup runs from the hot dog after I took a bite.*

The examples 1. and 2. are considered equivalent by a metric that computes the token overlap between two texts. But they are not equivalent at all, despite their perfect token overlap. This is because the semantic roles of the event’s participants are reversed and thus our hypothetical user would be provided with pictures that depict significantly different event dynamics. On the other hand, the text that is semantically most similar to 1. is 3. In fact, it is true that all images that show 3. are fully contained in the set of images that show 1., since *kitten* and *pupper* are hyponyms of *cat* and *dog*. Therefore, a user would be happy with high probability, if they are provided with any image that shows 3., given that they searched with 1. However, 1. and 3. have zero token overlap whatsoever, which makes these images highly unlikely to ever be returned to them. Instead, due to significant token overlap, it is more likely that images are returned that depict 4., an event that has

¹For instance, to evaluate a machine translation systems, we would want to measure *adequacy* of generations, where a polarity error is strongly penalized, while for document search without any further specifications, we would be less interested in strict adequacy, but more in a general form of relatedness.

nothing to do with the query, at all. In sum, the ranking that simple token-overlap metrics would provide us with is 2., 4., 3.; while actually we would be more happy when provided with the ranking 3., 2., 4.

So let us confidently conclude from our considerations that

We need a *semantic* similarity metric.

A semantic metric should compare the *meaning* of two texts. It should be able to, e.g., filter out ‘noise’ from different surface realization choices such as lexical similarity (*kitten*, *cat*) and/or active vs. passive voice (see above, 1. vs. 3.), and assess that different words or phrases can be more or less similar (e.g., *dog*, *pet*, *ketchup*).² But the insight that we need a semantic metric is not new and probably has been made countless times. It led to the emergence of text metrics that do not directly compare whether tokens are the same, but instead consider the degree of difference using word embeddings (Padó and Lapata, 2007; Pennington et al., 2014b; Mikolov et al., 2013), sometimes also taking their context into account to be able to provide different vectors to the same word to sensibly model phenomena such as polysemy (Erk and Padó, 2008; Peters et al., 2018; Devlin et al., 2019).³ But it remains unclear how these methods differentiate (superficial) textual *structure* from *meaning* and how they modulate these two aspects. Indeed, this is a long-standing problem in natural language processing (NLP) that has not decreased in its significance (Bender and Koller, 2020).

Therefore, we want to explore whether we can take this idea (to measure *semantic* similarity) a whole step further and conduct the similarity measurement *directly* in the space of meaning, based on a key hypothesis that is underlying this work:

There should exist an MR metric for effective semantic similarity.

Based on the fact that an MR makes the *meaning* of a text explicit, looking at the MRs of two texts should give us valuable information on how and why two texts are, or are not, similar. The space of explicit meaning also differentiates it from other semantic metrics that perform the comparison in a less-defined and/or hardly-comprehensible space (such

²There are also more abstract form variations that may – or may not – be considered a part of the meaning. E.g., it is known that different social groups choose to express the same or similar meaning with different structure (Kroch, 1978). Thus, there may be cases where a semantic metric should allow document retrieval that is invariant to different structures that arise from social classification.

³When inspecting meaning similarity of single words without context, *dog* in 4. will be considered same as *dog* in 1. and 2., resulting in too high overall similarity. Word embeddings that take context into account promise to mitigate such issues.

as the space of texts, or a basic high-dimensional vector space). Indeed, we want to have a representation that explicitly shows what happens in a text: E.g., going back to the example above, in 1., we would like to explicate that there is an *event* (run) in the sense of *running behind something*, with two actors (*dog, cat*) that exhibit different roles (the runner that follows something and the thing that is followed).

Not only can we use a metric between MRs to focus more on explicit semantics, but it can potentially directly explain to us the semantic aspects in which two texts bear similarities or dissimilarities (due to the explicitness of (graph-based) MRs). For instance, considering the example above, an explanation could look as follows: ‘*Yes, 1. and 2. are quite similar, but the roles of the two event participants are switched*’. Indeed, since MRs explicitly capture different semantic and linguistic phenomena – negation, semantic roles, and so on – aspect-targeted measurements between two MRs can show us in which ‘dimensions’ two texts actually are similar, or dissimilar. By contrast, when relying on a metric that is not grounded in explicit meaning representations (where we could draw from a myriad of already existing methods) we would not be able to explicitly differentiate between structure and meaning, and even less so between other and more finer ‘dimensions’ or aspects of meaning. They also often calculate a dot-product similarity on high dimensional real-valued vectors, so with this it can be hard to trace the influence of different features on final metric judgments, a problem that is further aggravated by the seemingly ever-increasing number of parameters of recent models. Therefore, *transparency of similarity assessment* is another reason that makes the exploration of distance analysis in the *explicit* MR space attractive. For instance, extracting subgraphs that relate to *polarity* would allow us to conduct controlled distance measurements with regard to the polarity aspect. More generally, testing whether a meaning representation is a subgraph of another, could inform us about semantic entailment, that is the question of whether a specific hypothesis can be derived from a premise. So, while MR metrics may be useful for any system applications that are meaning-focused, they may also help safety-critical applications, since their way of measurement is transparent.

Finally, MR-based graph metrics might also extend to more machine learning tasks, since the problem of measuring the similarity of graphs is far-reaching is “core to learning on graphs” (Shervashidze et al., 2011). In that aspect, we hope that research into MR metrics may open new views, or offer new and useful tools, to build meaning-focused machine-learning systems on top of MR metrics. Of course, working towards this goal, we may have to go beyond basic graph metrics and develop generalizable MR metrics that respect their specific topology and design.

In conclusion, motivated by the generality and explicitness of MRs, the main goal of our thesis is to learn about MR metric spaces, develop novel MR metrics, and explore potential application cases by touching on an exemplary selection of topics and tasks where we believe that MRs and corresponding metrics can be useful. For instance, we would like to explore the potential of MR metrics to provide us with explanations for why an automatically generated text deviates from a reference, and we would like to figure out ways to enhance semantic search with MR metrics, to increase its explainability and semantic accuracy.

1.2 Research questions

For better overview, we group our main research questions into four research question sets (RQS1-4):

- **RQS1: Theoretical assessment of MR metrics and development of enhanced MR metrics:** Although there are MR metrics that have been previously proposed (for parser evaluation), we know only little about their properties and use-cases. Therefore, we will first investigate whether we can design and apply intuitive theoretical criteria to distinguish previous MR metrics and detect their strengths and weaknesses. Based on our insights from this assessment, we ask ourselves: Can we design enhanced and generalizable MR metrics that mitigate weaknesses in previous metrics and extend the MR metric applicability to different tasks beyond parsing evaluation?
- **RQS2: Empirical assessment of MR metrics:** How can we empirically evaluate MR metrics? Can we define measurable objectives that we would like an MR metric to fulfill? And if we could define such criteria and build an MR metric benchmark thereupon, what conclusions can we make regarding previous and our novel MR metrics? These and similar empirical questions will be targeted throughout our thesis.
- **RQS3: Novel MR-task related applications: System evaluation with partially available MR:** An MR metric usually receives two MR inputs. But sometimes, one MR may be hidden. We find such a situation in two MR-related generation evaluation tasks: i) Evaluation of text generation from MR, and ii) evaluation of MR generation from text (parsing evaluation), without a costly reference. In both cases,

we would like to determine the quality of the generation in the MR space. Therefore, we want to assess whether we can develop strategies to cope with situations where we are missing input pieces.

- **RQS4: Novel and extended MR metric applications:** Based on our insights from our previous explorations into MR metrics, we want to know whether we can demonstrate their usefulness in extended application cases that are of broader interest to the NLP community, such as sentence and argument similarity. Besides accuracy, in practice, these tasks impose – or increase the importance of – an additional desideratum: efficiency. This is because in key applications such as document search, similarity metrics tend to be executed a lot of times. Since efficiency is a notorious problem in graph metrics, we would also like to know whether we can successfully mitigate an efficiency bottleneck.

1.3 Contributions of this thesis

In sum, our main contributions are:

- **Addressing RQS1:** *We introduce criteria for MR metric analysis and demonstrate that they can reveal strengths and drawbacks of previous MR metrics, helping researchers to make more informed decisions when selecting MR metrics and showing us perspectives for improvement. Based on our insights from the MR metric analysis, we contribute novel MR metrics that combine strengths of previous metrics while alleviating their weaknesses. Our metrics are prepared for general application cases by providing many-to-many node alignments for subgraph similarity and assessment of meaning composition.*
- **Addressing RQS2:** *We propose the first benchmark for empirical assessment of MR metrics, containing different objectives such as sentence similarity and robustness checks. We evaluate MR metrics on our benchmark, investigate their empirical trade-offs, and gather information for hyper-parameter recommendation. Similar evaluations will be conducted as part of RQS4.*
- **Addressing RQS3:** *We propose the first systems for addressing novel MR metric application tasks where one MR input is hidden. To the best of our knowledge, we propose the first metric for NLG evaluation grounded in MR, and the first system*

for MR quality evaluation in the absence of a costly reference. We show that for NLG evaluation, we can use a reliable parser to infer the hidden MR from the generated sentence, which facilitates execution of an MR metric in the MR space. For evaluating MR parses in the absence of a reference, we show that we can predict a MR metric, treating the reference MR as a latent variable. In our experiments, we show that both approaches can provide a multi-aspect assessment of system quality. In particular, we find that MR-based NLG evaluation metrics can help us to explain errors and provide fine-grained system diagnostics, as well as more meaningful and discriminative scores, due to their capability to detach meaning from form.

- **Addressing RQS4:** *We show how to use MR for exploring relations between natural language arguments, enhancing the accuracy and explainability in automatic argument similarity rating. We further show that the efficiency bottleneck can be mitigated, by proving that NP-complete graph alignment can be efficiently approximated with neural networks, opening the door to large-scale use-cases of MR metrics, e.g., for pattern-based search or MR clustering. Finally, we show that MR metrics can enhance a state-of-the-art semantic search engine by inducing an MR-metric based feature-structuring that helps explain similarity prediction. This is a contribution with broad application, since semantic search is an important problem in industry and research, with state-of-the-art methods lacking transparency and interpretability. Our proposed method retains (or sometimes improves) on the accuracy of the state-of-the-art method while keeping its full efficiency, allowing for very fast, aspectual and semantically targeted text clustering.*

1.4 Thesis overview

The main content of this thesis follows after discussing background and related work in the next two chapters. It is distributed over three parts. In Part I (Chapter 4 and 5), we study, develop and test MR metrics. In Part II (Chapter 6 and 7), we study novel tasks for MR metrics where (at least) one MR is hidden and needs to be projected: cost-efficient MR quality evaluation and multi-dimensional evaluation of text generated from systems. In Part III (Chapter 8 and 9) we assess the general problem of studying text similarity with MR metrics, to address broader NLP tasks such as semantic search.

Preliminaries: background and related work

In Chapter 2 we will discuss necessary background information that will enable us to better understand the main content of this thesis. In particular, we start by introducing some basic concepts: In Section 2.1, we introduce our definition of graph-based MRs and discuss translations between different graph formats. Then, in Section 2.2 we will introduce pillars of Abstract MR (AMR, (Banarescu et al., 2013)). Due to its popularity in the current NLP landscape and the fact that other MRs are often closely related to (and sometimes inspired from) AMR, the space of AMRs is the space where we will mostly work in. We will then touch on graph metrics, discuss their general desiderata (Section 2.3), and a coarse practical categorization of graph metrics into supervised and unsupervised metrics (Section 2.6).

In Chapter 3 we visit work that is related to the subject of this thesis. In Section 3.1 we discuss related work on MR metrics. Then, we discuss related works on evaluating systems that target the generation of natural language text (Section 3.2) and systems for estimating the quality of predicted structures without relying on a costly reference (Section 3.3). In Section 3.4 we discuss work on data sets that contain measurements of human text similarity assessments and automatic methods that have been developed to approximate such assessments. Finally, in Section 3.4.3, we discuss explainability in text similarity.

Part I: MR metric analysis and development

In Chapter 4 we develop analysis tools for MR metric analysis. Based on insights from the analyses, we begin to develop novel MR metrics. In particular, in Section 4.2 we construct a toolbox for metric analysis that views metrics through the lens of 8 principles that express different desiderata. In Section 4.3, we use these principles to analyze two previously proposed metrics, finding out that both share a common weakness. This instigates us to build our first novel metric. After an intermediate summary of our analyses (Section 4.4), we discuss learnt lessons and steps to go next (Section 4.5). Based on these lessons, we build novel and generalizable MR metrics from Weisfeiler-Leman graph kernel in Section 4.6, and show how we can optimize our metrics from direct human feedback (Section 4.7). Finally, we conclude this chapter with a discussion (Section 4.9).

In Chapter 5 we perform empirical studies on MR metrics. In Section 5.2, we propose BAMBOO_{MR} (Benchmark for A)MR Metrics Based on Overt Objectives), the first benchmark for empirically assessing MR metrics, consisting of human similarity objectives and robustness challenges. In Section 5.3, we employ BAMBOO_{MR} to assess the empirical performance and behavior of an array of different MR metrics. In the second part of this chapter, beginning in Section 5.4 we study a classical use-case of MR metrics and gain new insights about strong MR parsers across different domains through automatic MR metric assessment and human assessment, resulting in several recommendations (Section 5.5). Again, we conclude this chapter with a discussion (Section 5.6).

Part II: MR metric projection for system evaluation

In Chapter 6 we study an important problem of natural language processing through the lens of MRs: evaluation of automatically generated text. Here, we focus on a particular setup: evaluation of text that is automatically generated from MRs, by matching the MR of the generated text against its input. After discussing our motivation in more detail (Section 6.2), we formalize the problem and build a composite metric that assesses generated text from two crucial perspectives: its form, and its meaning (Section 6.3). Then we conduct two pilot studies: In Section 6.4, we re-rank NLG systems using our novel metric, assessing its discriminatory power and potential for providing us with coarse and fine-grained explanations for system quality.⁴ Then, in our second pilot study (Section 6.5), we probe weaknesses of our metric, such as its dependence on an automatic MR parser or language model. Finally, we conclude the chapter with a discussion (Section 6.6).

In Chapter 7 we study a novel application of *quickly* assessing the performance of MR parsers, in the absence of a costly human reference. We provide a more detailed motivation in Section 7.2 and provide formal definition of the problem in Section 7.3. We propose two neural graph encoding strategies that can be trained to solve the problem: A structure-enriched LSTM that processes the graph as a serialized string with alignment information (Section 7.4); And a CNN that is inspired by simplicity and human annotator view, by exploiting a structured and concise multi-line graph string serialization (Section 7.5). We set up the experiment by defining MR quality dimensions of interest (Section

⁴Coarse explanation, e.g.: is a system better at preserving meaning, or generating well-formed and grammatical sentences? Fine-grained explanation, e.g.: what specific linguistic error types does a system tend to produce?

7.6), and construct training and testing data (Section 7.7). We test our proposed systems against baselines in Section 7.8 and ablate various system parts. We conclude the chapter again with a discussion (Section 7.9).

Part III: MR metrics for effective semantic similarity

In Chapter 8 we investigate MRs and MR metrics for measuring the similarity of natural language arguments, and assessing argument quality. After providing more background (Section 8.2), we introduce two MR-argument similarity hypotheses (Section 8.3). Then we describe the implementation of our MR-metric based approach in Section 8.4 and conduct experiments on argument similarity through the lens of MR and MR metrics (Section 8.5). Finally, we conduct extensive analysis in Section 8.6, investigating potential explainability advantages of our approach and the usefulness of MR metrics for rating the quality of automatically generated conclusions, which is an upcoming topic in the argument mining community. We conclude the Chapter with a discussion (Section 8.7).

In our last (main) Chapter 9 we test whether we can leverage insights from our studies on MR metric spaces to empower methods for semantic search, which is of very broad interest for NLP research and industry: semantic search. For this, a crucial bottleneck of MR metrics has to be addressed, namely their costly graph-similarity computation and graph generation. More precisely, when we execute MR metrics, we normally need at least a parser, and possibly a metric that calculates a costly graph alignment. First, in a pilot study (Section 9.2), we show that we can mitigate the second issue, and develop strategies for approximating an NP-complete graph alignment problem, finding that we can efficiently and accurately approximate it with neural networks. However, we also note that *accurate graph alignment* doesn't imply *a similarity rating that resembles that of a human*, the latter being crucial for semantic search. So, finally, to alleviate these MR-metric related issues, we show how MR metrics can be used to improve a state-of-the-art neural sentence embedding model, by providing MR explanation while preserving, or improving its performance. After discussing some background (Section 9.3), we describe our model and the training data setup (Sections 9.4 and 9.5). Then we conduct extensive intrinsic and extrinsic evaluation starting in Section 9.6. In Section 9.7, we assess explainability performance, and in Section 9.8 we examine performance on three diverse downstream similarity rating tasks. We conclude with data explainability analysis (Section 9.9) and a discussion (Section 9.10).

Finally, we conclude the thesis in Chapter 10, where we summarize the thesis' main results (Section 10.1), give a short guide for selecting MR metrics for applications (Section 10.2), and elaborate on perspectives for future work (Section 10.3).

1.5 Generated papers and resources

The following list contains works that provide the main fabric of this thesis.

- “AMR metrics from principles” (Opitz et al., 2020). In: Transactions of the Association for Computational Linguistics. Code: <https://github.com/Heidelberg-NLP/amr-metric-suite>. C.f.: Chapters 4 and 5.
- “Weisfeiler-Leman in the BAMBOO: Novel AMR metrics and a benchmark for AMR graph similarity” (Opitz et al., 2021a). In: Transactions of the Association for Computational Linguistics. Code: <https://github.com/Heidelberg-NLP/weisfeiler-leman-bamboo>. C.f.: Chapters 4 and 5.
- “Better Smatch = Better Parser? AMR evaluation is not so simple anymore” (Opitz and Frank, 2022a). In: Proceedings of the 3rd Workshop on Evaluation and Comparison of NLP Systems. Code: <https://github.com/Heidelberg-NLP/AMRParseEval>. C.f.: Section 5.4.
- “Towards a decomposable metric for explainable evaluation of text generation from AMR” (Opitz and Frank, 2021). In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics. Code: <https://github.com/Heidelberg-NLP/MFscore>. C.f.: Chapter 6.
- “Automatic accuracy prediction for AMR parsing” (Opitz and Frank, 2019b). In: Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics. Code: <https://gitlab.cl.uni-heidelberg.de/opitz/quamr>. C.f.: Chapter 7
- “AMR quality rating with a light-weight CNN” (Opitz, 2020). In: Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics. Code: <https://github.com/Heidelberg-NLP/amr-quality-rater>. C.f.: Chapter 7.

- “Explainable Unsupervised Argument Similarity Rating with Abstract Meaning Representation and Conclusion Generation” (Opitz et al., 2021b). In: Proceedings of the 8th Workshop on Argument Mining. Code: <https://github.com/Heidelberg-NLP/amr-argument-sim>. C.f.: Chapter 8.
- “SBERT studies meaning representations: Decomposing Sentence Embeddings into Explainable Semantic Features.” (Opitz and Frank, 2022b) In: Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics. C.f.: Chapter 9.
- “SMARAGD: Synthesized sMatch for Accurate and Rapid AMR Graph Distance” (Opitz et al., 2023a). In: Proceedings of the 15th International Conference for Computational Semantics (IWCS 2023). URL: <https://arxiv.org/abs/2203.13226>. C.f.: Chapter 9.

Other related works where the author of this thesis participated in and on which we will touch in this thesis:

- “A Dynamic, Interpreted CheckList for Meaning-oriented NLG Metric Evaluation — through the Lens of Semantic Similarity Rating” (Zeidler et al., 2022). In: Proceedings of the 11th Joint Conference on Lexical and Computational Semantics. Code: <https://github.com/Heidelberg-NLP/NLG-CHECKLIST>.
- “Translate, then parse! A strong baseline for cross-lingual AMR parsing” (Uhrig et al., 2021). In: Proceedings of the 17th International Conference on Parsing Technologies. Code: <https://github.com/Heidelberg-NLP/simple-xamr>.

Chapter 2

Background

In this chapter, we introduce basic concepts that will help us better understand the topics and methods later developed in this thesis. We make one exception, and assume that the reader already has some knowledge about neural networks such as LSTMs (Hochreiter and Schmidhuber, 1997) or transformers (Vaswani et al., 2017) and their associated machine learning tasks ranging from classification to regression and sequence-to-sequence modeling.¹

2.1 Graphs and graph-based MRs

2.1.1 Meaning representations as graphs

A graph is a powerful means for representing all sorts of things. Formally, a graph $G = (V, E)$ consists of a node set V and an edge set $E = V \times V$. MR graphs typically have (at least) three special properties: They have a root, and their edges are directed and labeled, i.e., there exists a (surjective) function $el : V \times V \rightarrow P(\Sigma^E)$, where $P(\Sigma^E)$ denotes a powerset of a set of edge labels Σ^E . This essentially means that the graph allows multiple (distinctly labeled) edges between nodes.

With the type of graph described above, we can express several powerful meaning representations, e.g., Discourse Representation Structures (DRS, (Kamp, 1981)) or Abstract Meaning Representation (AMR, (Banarescu et al., 2013)). Likewise, other linguistic structures such as syntactic structures (dependency trees, constituency trees) and rhetorical discourse structure (Mann and Thompson, 1988) can also be captured with such

¹Several parts of this thesis can be understood without knowledge on neural networks. Also there exist abundant material of introductions, online lectures, papers, blogs, etc., where the reader may educate themselves about neural networks in a way that best suits their individual learning style. For instance, a useful online course on neural networks can be found here: <https://cs230.stanford.edu/>.

graphs. Typically, the vocabulary of semantic edge-labels is rather small and similar for different MR types; within it we tend to find ‘meaning-heavy’ labels such as `cause`, `instrument`, or `location`. Also common to many MR types is a specific terminology of *variable*. Essentially a *variable* has the same function as a node index v , and therefore each variable can be used to identify a node (more background on *variables* will follow below in Section 2.2).

Graph formats. Graphs are powerful but also complex objects. Graph libraries such as `networkx`² define specific data structures that enable us to access the graph efficiently in computer RAM, e.g., to apply general graph algorithms for traversing, searching, restructuring, and other objectives. However, it is also important to think of graph formats for other purposes, such as disk-storage of a graph in a string format (also sometimes called *serialization*) or visualization for visual inspection.

To this aim, one option is to store MR graphs as a set of (‘rdf’) subject–predicate–object triples, where a triple has the form $\langle u, \text{edgeLabel} \in \Sigma^E, v \rangle$ with $u, v \in V$ two nodes (or variables) and *edgeLabel* is the label of the edge that starts from source u and ends at target v . An example of an ‘rdf-MR’ of the sentence *A person works on their laptop* is displayed in the top right of Figure 2.1 (*RDF*)

Obviously, while a set of triples allows easy disk-storage, it is not very appealing for visual inspection: the *visual* graph structure is almost completely lost and we will struggle to comprehend non-atomic structures such as node neighborhoods. One option would be to simply use a library for graph plotting. But interestingly, for our targeted type of graph there is a format that can combine the best of the two worlds: The Penman notation (Mann, 1983).

The Penman format allows us to represent any directed graph as a string that is also accessible for visual inspection. An example of such a string structure is provided in Figure 2.1 (e.g., top, middle, *Penman format*). After specifying one node as root (a root is presupposed by many MR-types) we can traverse the graph with a depth-first traversal using `/` to indicate node labels, and brackets or indentation for broader graph structure. Note that inverting edges, such as $\langle x, \text{:edgeLabel-of}, y \rangle \equiv \langle y, \text{:edgeLabel}, x \rangle$ allow us to preserve the full graph, even though the string de-facto represents a tree-structure. In sum, we can say that the Penman notation allows easy disk-storage – but additionally enhances visual presentation by exploiting a compact multi-line indent-structured format. On a

²<https://networkx.org/>

theoretical side note, observe that for representing the graph structure, either brackets or indentation would be sufficient (however, both tend to be used simultaneously).

2.1.2 Same meaning, different structure: graph translations

Meaning preserving translations. As we have already seen above, graphs can be expressed in different ways. Besides representing them in different formats, we can also apply controlled structural changes that do not change the meaning of a graph, but change its structure. Graph translations can aid us with tailoring graphs to different applications and requirements. Since we will access such translations at some points in this thesis, and to warm up a bit more to MR graphs in general, we visit an exemplary selection of graph translations below (we use $f^{(-1)}$ to denote the option of inverting translation f):

- **string**⁽⁻¹⁾, as described above (Section 2.1.1), we can use this function to translate a string into a graph (and vice versa).
- **attr**⁽⁻¹⁾ takes as input a graph in which some edges carry *instance*-labels, where we can view the targets as labels of source nodes. We remove each instance edge, gather the target node that explicitly captures the concept in its index, and then add this concept/index as a label/attribute to the (former) source node. On top of the edge label projection, we then assume a node label projection: $nl : V \rightarrow \Sigma^V$, that projects a $v \in V$ onto a label from an (open) vocabulary of node labels Σ^V , often coined ‘concepts’ in the world of MRs. Note that this translation converts a graph with unlabeled nodes to a graph with node labels and a reduced number of nodes. An example is shown in Figure 2.1 (top, right).
- **reify**⁽⁻¹⁾ requires specific rules tailored to a certain graph-type. Depending on the vocabulary of edge labels, we define a subset of labels where we say that they are *canonical*. E.g., for Abstract MR graphs that are used in the example, canonical edge labels include arg_n and op_n ; $n = 1, \dots, N$. Since any edge can be viewed as a predicate with two arguments, e.g., $poss(x,y) \equiv \langle x, :poss, y \rangle$ indicating that x is possessed by y , we can map via a manual rule to an equivalent structure $arg_1(p,x) \wedge arg_2(p,y) \wedge label(p, possession)$, where the relation is now expressed by a new node p labeled with *poss*, connected to its arguments via the canonical edges (an example is shown in Figure 2.1, middle). In our rule-book, we would see that the arg_1 of a node labeled with *possession* is the thing that is owned by the owner indicated with arg_2 .

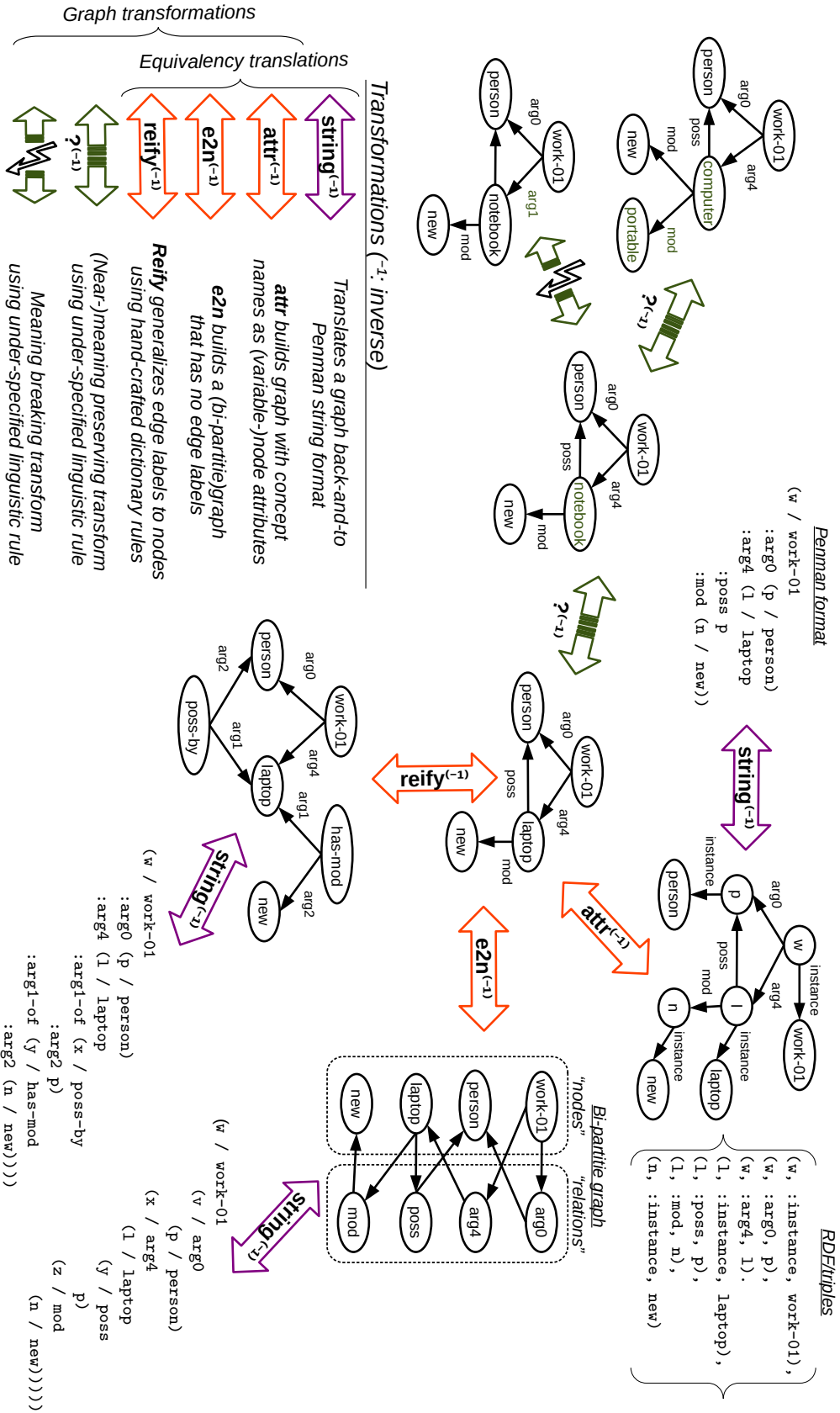


Figure 2.1: Meaning preserving, quasi meaning preserving, and meaning breaking (A)MR translations.

Essentially, after reification, a graph expresses the same content but it has changed its structure and only contains edges that have canonical labels. A possible advantage of reification is that we can represent more information about the non-canonical relations, treating them as events/nodes. For instance, in addition to who possesses what, we could model the time when the possession-state occurred, attaching a *time*-edge to the *possession* node.

- $e2n^{(-1)}$: More generally, an *edge-labeled graph* can be translated to a graph with unlabeled edges, sometimes also called Levi Graph (Levi, 1942). As indicated in Figure 2.1 (right), this graph is bi-partite, containing ‘original’ nodes on the left, and relation labeled nodes on the right. A potential advantage of such a graph is that the relation can be treated in the same way as nodes, removing the requirement to define potential extra-steps for edge-label processing.³

Touching on linguistics: almost meaning preserving and meaning breaking transformations. MR graphs aim at capturing the *meaning* of text. Therefore, we could also map between graphs using *linguistic* rules. This is indicated in Figure 2.1 on the left side with green arrows. Using a near-meaning preserving transformation, we could map a node labeled *laptop* onto its (near-)synonym *notebook*, which preserves (almost) all of the meaning that the original graph expresses. We can also think of larger structural changes in a similar way: For instance, a node labeled *laptop* can be mapped onto two nodes *computer* and *portable*, that are connected with a *modifier* relation. By contrast, the meaning of the original graph might break if other manipulations are performed, e.g., changing the node *notebook* to the node *car*, now indicating that a *person* is working on their *car* instead on their *computer*, which would induce a significantly different meaning.

Note that any *linguistic* transformation, be it largely meaning preserving or not, is under-specified and we require linguistic knowledge for full specification (as opposed to the fully meaning preserving structural translations detailed above).

³Note that such a ‘Levi’-graph is a by-product of building a *line graph* (Whitney, 1932; Krausz, 1943; Harary and Norman, 1960) that shows neighborhood-relations of relations.

2.2 Abstract Meaning Representation (AMR)

Most of our experiments will be based on a particular type of a Meaning Representation: Abstract Meaning Representation (AMR), proposed by Banarescu et al. (2013). To represent meaning, AMR uses a rooted, directed, acyclic graph format with labels on edges (relations). Together with the theory of AMR (that we will describe next in more detail), a large dataset of several thousand manually crafted AMRs was released. Therefore, we can count on the availability of strong parsers that allow us to robustly project meaning representations for new sentences.

The large size of manually created data and its effects such as robust parsers are not the only reason why we prefer AMR as a basis for our studies. Additionally, it bears significant similarities to many other meaning representations, such as DRS (Kamp, 1981), Universal Meaning Representation (Stengel-Eskin et al., 2020), Uniform Meaning Representation (Van Gysel et al., 2021) and BabelNet Representation (Lorenzo et al., 2022). Some MRs are extensions or/and strongly inspired from AMR, e.g., to better address challenges like cross-linguality (Lorenzo et al., 2022), and most of them can be expressed in an AMR-like rooted DAG graph format as defined above, like the compositional and logics-flavored DRS representation or Universal Meaning Representation that practically stacks, in one MR, individual meaning structures that are deemed complementary.⁴ So with the knowledge that MRs tend to bear strong similarities, specifically with regard to their structure, and MR metrics promise to generalize well over different MR types due to their shared graph format (DAG), we can confidently proceed with using AMR as the main testing substrate for our MR metrics. Of course, this does not preclude that future research may find a need to further tailor our metrics to non-AMR MRs, or build new MR metrics, to best take any of their possible specifics into account (e.g., logics and compositionality of DRS).

2.2.1 Pillars of AMR

From Neo-Davidsonian event semantics to graph (triples). MRs such as AMR have roots in Neo-Davidsonian semantics (Davidson and Rescher, 1967; Lasersohn, 2016; Wang et al., 2020d) that seek to flexibly and comprehensively model complex events and states. These semantics model the relation between events and their arguments (entities) by specified semantic roles (e.g., the *agent* [usually *arg0*] of an event, the *patient* [usually

⁴I.a., universal dependency structures (Nivre et al., 2016) and semantic proto-roles (Dowty, 1991; Reisinger et al., 2015; Teichert et al., 2017; Opitz and Frank, 2019a; Spaulding et al., 2023).

argI] of an event, and other key semantic relations such as *location*, etc.). This can be formalized as a conjunction of two-place (binary) predicates:

$$\{pred(x,y) \mid pred \in PRED; x,y \in ENT\} \quad (2.1)$$

where we use *PRED* to denote a set of descriptive binary predicates and *ENT* a set of semantic entities in a text (referring to objects, events, states,...) typically denoted by variables such as *x,y* that can only be understood through their context. For instance, *A person is working on a laptop* would trigger a structure similar to:

$$\exists p,l,w : person(p) \wedge work(w) \wedge laptop(l) \wedge agens(w,p) \wedge instrument(w,l), \quad (2.2)$$

which shows that there are *three* semantic entities (*p,l,w*), that they are of a certain type (*person, laptop, work*), and that the entities fulfill different roles (*p* is the agent in the event *w*, and *l* the instrument). Note that ‘unary’ predicates such as *work(w)* are equivalent to using a binary predicate and including \emptyset in *ENT*: *work(w, \emptyset)* or using declarative variable names such as *work \in ENT*, and *instance(w, work)* relations.

Importantly, with such a Neo-Davidson structure, we can attach further information about the event at our will, up to arbitrary complexity. For example, the time where the working happened, if the ventilator of the laptop is loud and running, or not, etc, with the variables referred to in different predicates.

Moreover, viewing each part in such a logical conjunction as a labeled **graph edge** ($pred(x,y) \equiv \langle x, :pred, y \rangle$), a **graph** such as in Figure 2.1 can be immediately created.

Predicate frames boost the descriptiveness of AMR. Such frames enable us to explicitly describe events and the roles of event participants. The frames are created by linguistic expert annotators and are freely available in PropBank (Palmer et al., 2005). In particular, PropBank contains several thousand English (sense-disambiguated) predicates and enumerates their core role arguments A_i - A_n , where *i,n* depend on the predicate (and its sense), but typically *i* = 0 and *n* ranges between one and five. These core roles are specified individually for every frame in PropBank, and each role has a specific description. E.g., in

S: A person works on their new laptop.

we would like a system to distinguish the correct sense for *work*. In this case, this would be `work-01`, which has a general sense of *work, being employed, acts, deeds*. Now we can inspect the following semantic role meaning structure:

0. A0(=A person) as the worker (the agent).
1. A1(\emptyset) as the job or project (the theme).
2. A2(\emptyset) as employer/benefactive.
3. A3(\emptyset) a co-worker.
4. A4(=their new laptop) as the instrument.

Core vs. non-core roles. Above, A0-A4 are frame-specific *core roles*. Additionally, we can attach to any predicate so called non-core roles, which are less-abstract roles of which we presume that they generalize over many different predicates, for instance, *location*, *time* or *instrument*. E.g., if *work* (in S) happens on a particular day, e.g., Tuesday, we can use a relation *time(work, tuesday)*. However, if non-core roles are empirically frequently realized in a predicate, they are often assigned a core-role (see above, *instrument* = A4). Even though there sometimes may be ambiguity in assigning roles to participants, for most cases we can resolve this by aiming at the most normative interpretation of the sentence. For instance, *Work on something* might also trigger the interpretation that *laptop* is a project (A1), as in *someone repairs, or builds a laptop*. However, in the absence of any further context, we consider *laptop* as the instrument, A4, since the interpretation that somebody uses a computer as an instrument to accomplish something seems more normative, especially in the absence of further context.

‘Format-wise’, AMR customary uses upper-case ‘ARGn’ edge labels to represent PropBank’s ‘An’ roles, lower-case for non-core roles, and sometimes also prepends a ‘:’ to edge labels, particularly when graphs are serialized as string. Such different notational choices can slightly vary among MRs and they do not imply a semantic difference. Therefore, in this thesis we will allow ourselves to generalize over particular edge label naming conventions, i.e., we let $An \equiv :ARGn \equiv ARGn \equiv argn \equiv arg_n$ (e.g., $A0 \equiv :ARG0 \equiv ARG0 \equiv arg0 \equiv arg_0$).

2.2.2 A deeper look into the AMR toolbox

Expressing a text with predicate frames such as they are contained in PropBank can already constitute a useful MR representation, and is targeted by a well-known NLP task called *Semantic Role Labeling*. However, through AMR we can model meaning more precisely and extensively. This is mainly because AMR provides us with tools to further decompose meaning. For instance, consider that in *S*, *their new laptop* is the instrument in the semantic role structure. AMR does not treat *their new laptop* as a single entity but instead would connect *new* with a modifier-edge to *laptop* (since the role of *new* is to modify the laptop, see also again Figure 2.1, top).⁵

Coreference modeling is also a powerful feature of AMR. Using variable nodes, we can refer to particular events, states, and persons multiple times, or distinguish among two or more instances of the same concept. For instance, decomposing *their new laptop* further, so that we know whom *their* actually refers to:

$$\exists p, l, w : person(p) \wedge work(w) \wedge laptop(l) \wedge arg0(w, p) \wedge arg4(w, l) \wedge \underline{poss}(p, l) \quad (2.3)$$

I.e., the difference to Eq. 2.2 is that we have added a part where p, l re-occurs in a *possession* relation. Similarly, in AMR, coreference is sometimes referred to as *re-entrancy*, referring to the re-entrant edges that occur in graphs as coreferences (see AMR graph in Figure 2.1, top: the *person* node has two incoming edges, among them one *possession* edge from *laptop*, explicating the meaning of *their*).

Negation modeling. AMR captures negation using *polarity* edge and reflects modality with *possible* edges. For instance, to alter the AMR of *S* to express the meaning (...) *cannot work* (...), we introduce an abstract *possible*(p), introduce a $arg1(p, r)$ and a $polarity(p, NEG)$. By contrast, when reflecting (...) *doesn't work* (...) we only insert $polarity(w,$

⁵Note that if such a modifier item is contained in PropBank as a one-place predicate with designated arguments, the AMR annotator should prefer selecting the pre-defined frame. In these cases, AMR trades some generality for more precise definition. Indeed, this is the case with *new*, which can be found as *new-01* (*be newly created; recently come into being*) in PropBank (c.f., <https://verbs.colorado.edu/propbank/framesets-english-aliases/new.html>). Therefore, in this case, we may select $\exists l, n : new_1(n) \wedge laptop(l) \wedge arg1(n, l)$ over $\exists l, n : new(n) \wedge laptop(l) \wedge mod(n, l)$.

NEG).⁶:

$$\exists p, l, w : person(p) \wedge work(w) \wedge laptop(l) \wedge arg0(w, p) \wedge arg4(w, l) \wedge \underline{polarity}(w, -). \quad (2.4)$$

Negation can be attached to any node, which is useful to model phrases such as *the laptop that is not new*:

$$\exists l, n : laptop(l) \wedge new(n) \wedge \underline{polarity}(n, -), \quad (2.5)$$

or meaningfully split composed *adjectives*, e.g., *unjust*: $\exists j just(j) \wedge polarity(h, -)$.

Focus modeling. AMRs are *rooted* graphs. The root concept shows the focus of the sentence. For instance, when expressed as AMRs, the sentences

S': The cat drinks milk.

S'': The milk is drunk by the cat.

differ only in their focus but otherwise express the same meaning: In AMR, we have $\langle ROOT, :root, cat \rangle$ in S' and $\langle ROOT, :root, milk \rangle$ in S'', all other triples being equal.

Some limitations of AMR. While AMR is a powerful tool for capturing the meaning of text, it is not (yet) fully complete. For instance, it lacks a complete first-order logic model because it currently cannot represent *scope* of phenomena such as quantification and negation, and it also lacks representation power for *tense and aspect*. However, there is active research on developing mitigating these concerns, by developing corresponding AMR modeling schemes (Donatelli et al., 2018; Pustejovsky et al., 2019) or translation mechanisms that can be applied to project AMR on first-order logics (Bos, 2016, 2019).

Second, AMRs are not created with compositional construction from the text, there are no explicit alignments between parts of a text and parts of its AMR structure. On one hand, this has the advantage of facilitating easier and more meaning-focused annotation. On the other hand, we cannot explicitly trace what parts in the sentence trigger particular meaning structures, which may be an issue in some applications.⁷

⁶In AMR, typically we use - as NEG

⁷A possible mitigation strategy is to build a 'post-hoc' alignment, where an automatic tool is used to build relations between the tokens in a sentence, and corresponding AMR structure.

2.2.3 From text to AMR and from AMR to text

Text2AMR. AMRs tend not to ‘appear out of nowhere’, and the human AMR building process is quite costly. Even after some targeted training, a human would need, on average, 10 minutes per sentence to invoke its AMR (Banarescu et al., 2013). Thus, AMRs often need to be inferred automatically based on a text.⁸ Therefore, researchers develop systems that generate AMR structures from text, called parsers. These parsers are typically trained and tested on large semantic graph banks (Knight et al., 2014, 2021), that cover a broad spectrum of text types and nowadays contain several thousands of human-crafted AMR graphs.

How to tell whether a parser is better than another? To automatically assess the quality of a parser, we usually preserve a test set with human-crafted AMRs (the *reference*) and then compare the parser’s outputted candidate graphs against the reference with a graph metric. While parsers tended to make lots of errors in times when they would be trained from scratch, recent AMR parsers are quite accurate and produce much fewer errors. This is mainly due to the paradigm of fine-tuning large pre-trained language models such as T5 or BART (Raffel et al., 2020; Lewis et al., 2020; Bevilacqua et al., 2021) that can boost model performance in many areas of NLP. To apply these powerful models to the task of creating AMR graphs, we need to express input (text) and output (graph) as text sequences. Parsing in such a sequence-to-sequence fashion is possible due to string linearization as described above, using DFS traversal from the root and brackets for one-line graph structure (see details in Section 2.1.1).

AMR2text. On the other hand, AMR2text generation systems aim at generating natural language text from an AMR. Therefore, these systems fall into the broad area of Natural Language Generation (NLG). Naturally, this process can be viewed as the inverse task to parsing/text2AMR, and thus usually the same data sets used for training and evaluating parsers are also used to train and evaluate AMR2text generation systems. Also very similar to text2AMR, against the backdrop of the emergence of pre-trained models, AMR2text systems that encode the graph as a linearized string have recently seen significant performance boosts (Bevilacqua et al., 2021). Different from text2AMR, but similar to other NLG systems, AMR2text systems are usually evaluated using automatic token-overlap matching metrics such as BLEU (Papineni et al., 2002) that compare the generated texts to reference texts from which the input meaning representations were constructed.

⁸We use ‘often’ here as to not preclude cases where AMRs may not need to be inferred. Such a case, for instance, could be a hypothetical communication of robots who communicate using AMR language.

2.3 Graph metrics and metric space

Our focus in this thesis lies on measuring similarities between MRs. After discussing the general concept of graph-based MRs above, we are now well prepared to touch on metrics between MRs. We start with a general *metric* between graph objects:

$$metric : G \times G \rightarrow \mathbb{R}. \quad (2.6)$$

A *metric space* arises then as a tuple $(metric, G)$ where the distance of any pair $g, g' \in G \times G$ is determined with our *metric*. Purely for convenience, we say that this metric measures **similarity** (the inverse of distance) by saying a **higher score means greater similarity**.

Note that a *canonical* metric space strictly requires satisfaction of certain axioms, such as non-negativity, triangle inequality, symmetry, or identity of indiscernibles. However, in practical cases, not all axioms can always be satisfied and furthermore their importance can be unclear and may vary among applications. Therefore, relaxed notions of metric spaces have been introduced, e.g., *pseudo-metrics*, *quasi-metrics*, or *meta-metrics* (Burago et al., 2022). In our work, we will allow ourselves to be a bit lenient in this matter of precise formal definition and only mention exact mathematical types of metrics where such a distinction is immediately helpful, e.g., for better understanding a metric’s goals or its behavior. So in general, similar to how the term ‘metric’ is broadly used in NLP, we say that a metric is a model/system/function that takes two objects as input and returns some number that can be understood as a distance, or similarity, of the input objects.

2.3.1 Graph similarity measures

General deliberations. As Shervashidze et al. (2011) put it, one can think of many different ways to calculate a similarity measure between graphs. Perhaps the most natural natural measure of graph similarity would be to check whether the graphs are structurally identical, i.e., isomorphic. This would provide us with a binary score that equals 1, if the graphs are isomorphic, and 0 otherwise. However, despite the popularity of the problem, no efficient algorithm is known, and it is presumed to be NP-complete (Garey and Johnson, 1979). When the two graphs are of different size, which is most regularly the case in applications, we can check for subgraph isomorphism. Subgraph isomorphism has actually been proven to be NP-complete (Garey and Johnson, 1979), suggesting that the isomorphism problem of same-size graphs is NP-complete, too.

Of course, for most purposes, we would like to move beyond a binary measure and obtain a more graded measure of similarity that puts the similarity of graphs on a spectrum. Again, one can think of many measures that are more fine-grained. A very simple measure that may come to mind is to create a simple statistics by counting matches of atomic graph parts such as node labels or edge labels, if available. However, while non-binary, efficient and perhaps in some cases effective, it clearly disrespects the graphs' structures. A less naïve approach would be to determine the size of the largest common subgraph in two graphs. But unfortunately, the problem of finding the largest common subgraph of two graphs is, again, NP-complete (Garey and Johnson, 1979). As we will see later, this kind of measurement has been proposed before for comparison of MR graphs (we will describe the measure in detail in the first part of our related work 3.1). Many other graph similarity measures have been proposed: graph edit distances calculate the effort of editing one graph such that it yields the other (Bunke and Allermann, 1983; Neuhaus et al., 2006; Gao et al., 2010) thus respecting the topology, as well as node and edge labels of graphs, but they are hard to parameterize and need to approximate NP-complete problems as intermediate steps. On the other hand, optimal assignment kernels (Kriege et al., 2016a,b), try to match substructures of graphs. Other approaches are based on creating invariant representations – but these tend to suffer from other problems, e.g., they restrict the type of graphs to unlabeled graphs, such as measures based on the skew-spectrum (Kondor and Borgwardt, 2008), or they may be difficult to adapt for labeled graphs, such as measures based on the graphlet spectrum (Kondor et al., 2009). The skew-spectrum projects the adjacency matrix of a graph onto a invariant matrix, while the graphlet spectrum captures statistics about nodes and node positions

2.3.2 Discussion

We'd like to have (MR) graph measures that fully assess *represented meaning*: they should be *graded*, *interpretable*, *efficient*, and *respect the graph topology*. So it seems probable, that at the end of the day, we may have to carefully weigh some trade-offs with regard to these objectives and won't be able to get our free lunch. Indeed, even if some efficient measures may be incapable of judging true (structural) graph-isomorphism, they may nevertheless provide us with an effective distance measurement that can meaningfully discriminate (to the best extent) our graphs, perhaps even better capturing true *semantic* MR/text isomorphism which could, in a sense, even stand in conflict with structural isomorphism (two structurally isomorphic MRs clearly have the same meaning, but

we will see that two structurally very different graphs can have the same meaning, too).

Among the most interesting techniques for finding a good trade-off, and measure graph distance efficiently and meaningfully, we find measures that have sprout from the famous *Weisfeiler-Leman* (WL) algorithm by Weisfeiler and Leman (1968). The iterative WL algorithm was originally intended for the efficient assessment of graph-isomorphism. In each iteration, a node collects its labels from its neighbors, and ii) compresses them into a new label. Finally, the two graphs can be projected onto two count vectors, where a particular index stands for a particular node-label and the vector of a graph contains the node label's count in a graph. Importantly, if the two vectors are different – we know that the two graphs are *not* (structurally) isomorphic. However, on the other hand, if the count vectors are the same, we cannot conclude that the graphs are isomorphic. But still, a very useful feature is that we can finally use fast and simple vector algebra on the two vectors (distilled from the graphs) to compute a fine graph similarity, e.g., using the dot-product. Inspired by this concept, recent WL-based graph measures (Shervashidze et al., 2011; Togninalli et al., 2019) are efficient and promise to respect and exploit graph topology, by restricting themselves to comparing graph substructures (up to an arbitrary size) in polynomial time.

2.4 Measuring MR similarity: can we use a graph measure off the shelf?

As the previous section suggests, there exists a myriad of ways to compare graphs. Let us think more about our application case: comparison of graphs that are MRs.

Let us first recall the general structural type of MR graphs: They possess a root, are directed, and carry node and edge labels. While the first two properties seem potentially neglectable and may not (much) restrict the set of meaningful graph similarity measures, it appears that the latter two properties (edge labels and node labels) may affect the nature of a suitable set of similarity measures much more, since these labels carry rich semantic information, for instance, they express semantic roles or events and states. So as MRs are actually designed to capture a text's *meaning*, it seems intuitive that the similarity of a pair of MRs should be expected to correspond to the similarity of a pair of texts, which they represent.

Therefore, the interaction between *structure* and *expressed meaning* seems more complex than for other types of graphs, and may sometimes even stand in conflict. In fact, we have to be aware of a possibly very crucial

Conjecture: A monotonic relation between topological graph structure distance and MR graph distance may *not* be what we wish for.

That means we anticipate cases of paired MRs where we would like to have

$$\exists a, b, c : \text{structSim}(a, b) > \text{structSim}(a, c) \wedge \text{realSim}(a, b) < \text{realSim}(a, c), \quad (2.7)$$

where a, b, c are some MR graphs, *structSim* a perfect measure of structural graph similarity and *realSim* a desired measure of MR meaning similarity. In other words, we conjecture that structural MR similarity may not be monotonically related to semantic MR similarity. Indeed, while for many types of graphs a very small change in structure may not lead to a very different overall meaning of the graph, it seems that an MR's overall meaning could quickly and drastically change, also due to how semantics are compositionally built.⁹ In particular, when speaking of MRs, phenomena of meaning compositionality might lead to the outcome that a small structural change (e.g., addition or removal of a single weakly connected node in a large MR graph), can drastically change the meaning that is captured in the MR (e.g., if we remove/add a *negation* node to an event node). On the other hand, broader structural differences may not necessarily imply a much different overall meaning. For example, if a pair of graphs represents texts that are (near-)paraphrases (e.g., *kitten–young cat*) the graphs can exhibit a significantly different structure (e.g., $\text{kitten}(k)$ vs. $\text{cat}(c) \wedge \text{young}(c) \wedge \text{mod}(c, y)$), which would lead to yield a topological/structural similarity that is too low, in the sense that it does not at all reflect the (near-)equivalency of the two MRs' meaning.

In sum, to make an attempt to answer the question in the title of this section: No, we probably cannot get a metric off-the-shelf in the hopes of measuring meaningful distances between MR graphs. Indeed, we will corroborate this insight in multiple places in this thesis.

⁹According to the *principle of compositionality*, which is conjectured by many scholars, the meaning of a text is *composed* from its parts (Boole, 1854; Montague, 1973; Pelletier, 1994).

2.5 Metric performance evaluation

Ultimately, a graph *metric* provides us with a scalar $s \in \mathbb{R}$ and thus performs a regression on pair-wise inputs. *Metric performance evaluation* wants to study the accuracy of the scores that a metric can provide us with.

Evaluation modes: intrinsic vs. extrinsic. Broadly speaking, there are two major modes of evaluation. **Intrinsic evaluation** would assess our regression model (metric) with an eye to whether it seems to do what we want it to do. For example, if we wanted to create an *efficient* graph metric, we could measure its efficiency, e.g., by theoretical complexity analysis, or empirical runtime analysis in average and non-average cases. **Extrinsic evaluation**, on the other hand, would assess the more general usefulness of our regression model for downstream tasks. For instance, we could visit a sentence classification problem (e.g., negative vs. positive sentiment): We could 1. MR-parse the sentences, and 2. construct a pair-wise similarity matrix using our *metric*. We could 3. feed this matrix into a kernel machine such as an SVM and evaluate the performance in this task using standard classification metrics such as Accuracy or F1 score. This way, we could get an insight of how well our regression model works for the downstream task of sentiment classification.

However, *extrinsic evaluation* and *intrinsic evaluation* are often not mutually exclusive. A middle-ground is found when inspecting the quality of our regression in immediate downstream tasks that are also indicative for the model's intrinsic behavior. For instance, we can perform sentence similarity rating through MRs and MR metrics. In this setup, we can directly compare our regression against a human reference regression with standard *regression metrics*, studying our model's intrinsic behavior and retrieving knowledge of how well our model performs in rating sentence similarity, an important task for semantic search.

Choosing regression evaluation metrics. Assume a data set $\{(x_i, y_i)\}_{i=1}^n$ of pairs. We generate scores $S = \{S_i\}_{i=1}^n = \{\text{metric}(x_i, y_i)\}_{i=1}^n$ using our regression and obtain reference scores $S^* = \{\text{metric}^*(x_i, y_i)\}_{i=1}^n$ with a reference regression *metric*^{*}.

The most simple measure of quality might be the mean absolute error of our scores against reference scores:

$$MAE(S, S^*) = \frac{1}{n} \sum_i \text{error}(i) = \frac{1}{n} \sum_i |\text{deviate}(i)| = \frac{1}{n} \sum_i |S_i - S_i^*|, \quad (2.8)$$

or its frequently found ‘cousins’ MSE (mean squared error) and RMSE (root of mean squared error). However, evaluation with such a metric is often problematic, since we’d ideally want to generalize over different scales and perhaps also distributions that our scores and reference scores may exhibit. We do not want to penalize equivalent metrics. Indeed, we can easily think of cases where equivalent metrics can exhibit an unwanted high MAE (MSE, RMSE). E.g., consider two same Gaussian distributions – the more we shift the mean of one, the higher the MAE.

Instead, we are more interested in the relation of predicted scores to reference scores, and an ideal relation that we would like to have is that they can predict each other. This property is assessed with *correlation measures*, such as the Pearson’s ρ :

$$\rho_{\text{Pearson}}(S, S^*) = \frac{\sum_{i=1}^n (S_i - \bar{S})(S_i^* - \bar{S}^*)}{\sqrt{\sum_{i=1}^n (S_i - \bar{S})^2} \sqrt{\sum_{i=1}^n (S_i^* - \bar{S}^*)^2}}. \quad (2.9)$$

Here, \bar{x} denotes the arithmetic mean of x . This formula precisely measures the linear relationship between two data sets (here: our scores and reference scores). Interestingly, from Pearson’s ρ , we can directly derive the Spearman’s ρ , since it arises by calculating Pearson’s ρ on the ranks of the data sets:

$$\rho_{\text{Spearman}}(S, S^*) = \rho_{\text{Pearson}}(\text{ranks}(S), \text{ranks}(S^*)). \quad (2.10)$$

Therefore, it tells us about how similar two measures are with respect to the *ranks* that they assign to data examples. In theory, the two ρ -measures can somewhat diverge, but practically, both ρ -correlation statistics often yield a similar outcome (Spearman’s ρ can be considered slightly more robust, while Pearson’s ρ has slightly more statistical power). Because either of these statistics is more meaningful for our purposes than absolute error statistics, in our thesis, as is also standard in the community (Reimers and Gurevych, 2019) we will focus on *correlation* measures for evaluation.

2.6 Unsupervised versus supervised metrics

It is interesting to assess metrics from a viewpoint that is concerned with the amount of supervision that is required by a metric to function well. For simplicity, we will do a binary categorization: i) general unsupervised or zero-shot metrics that are either constructed

using human intuition or are learnt without direct task-specific labels, and ii) supervised metrics that are learned for a specific target task.

In our thesis, we will first and foremost consider unsupervised or zero-shot metrics and control learning where possible. In fact, such metrics seem most appealing to us because they promise better generalization to different tasks. A supervised metric, by contrast, is learned using task-specific training data and thus may not transfer well to related tasks and different domains or applications, especially if it relies on a complex function for embedding our objects in the metric space, such as a neural network. Then, in many cases, without much deeper analysis, we cannot be assured that such a supervised metric has learnt anything generally useful, even though performance scores on a benchmark may seem high. Instead, the metric may have learned to exploit spurious cues and short cuts that indeed do predict labels in one type of data – but fail in other tasks where spurious knowledge is either not available or of a different nature. Such a ‘Clever Hans Effect’¹⁰ is a well known and long-standing problem in machine-learning known as *overfitting* (Dietterich, 1995). The issue becomes more severe with growing complexity of models (Niven and Kao, 2019). This is also evidenced by the fact that deeper, more complex text metrics that are trained with task-specific gold labels perform well for supervised tasks, but for unsupervised tasks such as semantic relatedness of sentences, simpler models perform better (Hill et al., 2016), or models that have not been trained using target-task specific gold information and are thus more robust (Reimers and Gurevych, 2019).

Nowadays, another related issue is the risk of *catastrophic forgetting*: when training a pre-trained model in a supervised manner for a particular task of interest, we can lose a lot of general information by teaching a model a specific task (Kemker et al., 2018). In sum, metrics that are not learnt but are created with human knowledge, or learnt with less supervision, or learnt in a very controlled way, provide us with a more general basis for meaningful distance measurements. Finally – if wished so – we can even build a supervised machine learning system on top of them, for downstream classification tasks, e.g., by employing a similarity-metric based kernel machine (Hofmann et al., 2008).

¹⁰Clever Hans was a horse that seemed capable of intellectual tasks, such as simple arithmetics. But in 1907, psychologist Oskar Pfungst demonstrated that the horse was not actually solving the mental tasks, but it was acting based on observed reactions of his trainer.

Chapter 3

Related work

3.1 MR metrics

We will see that metrics for other types of MRs are often taken over, are derived, or are inspired from AMR metrics. Therefore, we will dive straight into them.

3.1.1 AMR metrics

As of now, only few methods have been devised for measuring AMR similarity, and all of them are structural. They can be divided into two categories: i) approximating the solution to an NP-hard edge/triple-match maximizing node alignment (Cai and Knight, 2013), or ii) calculating the overlap of bags of graph parts that are extracted by traversing over the graphs (Song and Gildea, 2019; Anchiêta et al., 2019).

The ‘classical’ AMR metric: SMATCH The widely adopted SMATCH (Semantic *match*, SMATCH) metric (Cai and Knight, 2013) seeks to approximate an NP-hard graph alignment problem with a hill-climber, finally scoring matching triples. Formally, let $f_{map}(a, b)$ be the count of matching triples of two graphs a, b under any mapping function map that maps nodes from graph a to nodes from graph b (every node can have at maximum one correspondence in the other graph). For instance, given a triple from a : $\langle x, :relation, y \rangle$, and a triple from b : $\langle w, :relation, z \rangle$, then these triples match *iff* $map(x) = w \wedge map(y) = z$. So we would like to have a best

$$map^* = \operatorname{argmax}_{map} f_{map}(a, b), \quad (3.1)$$

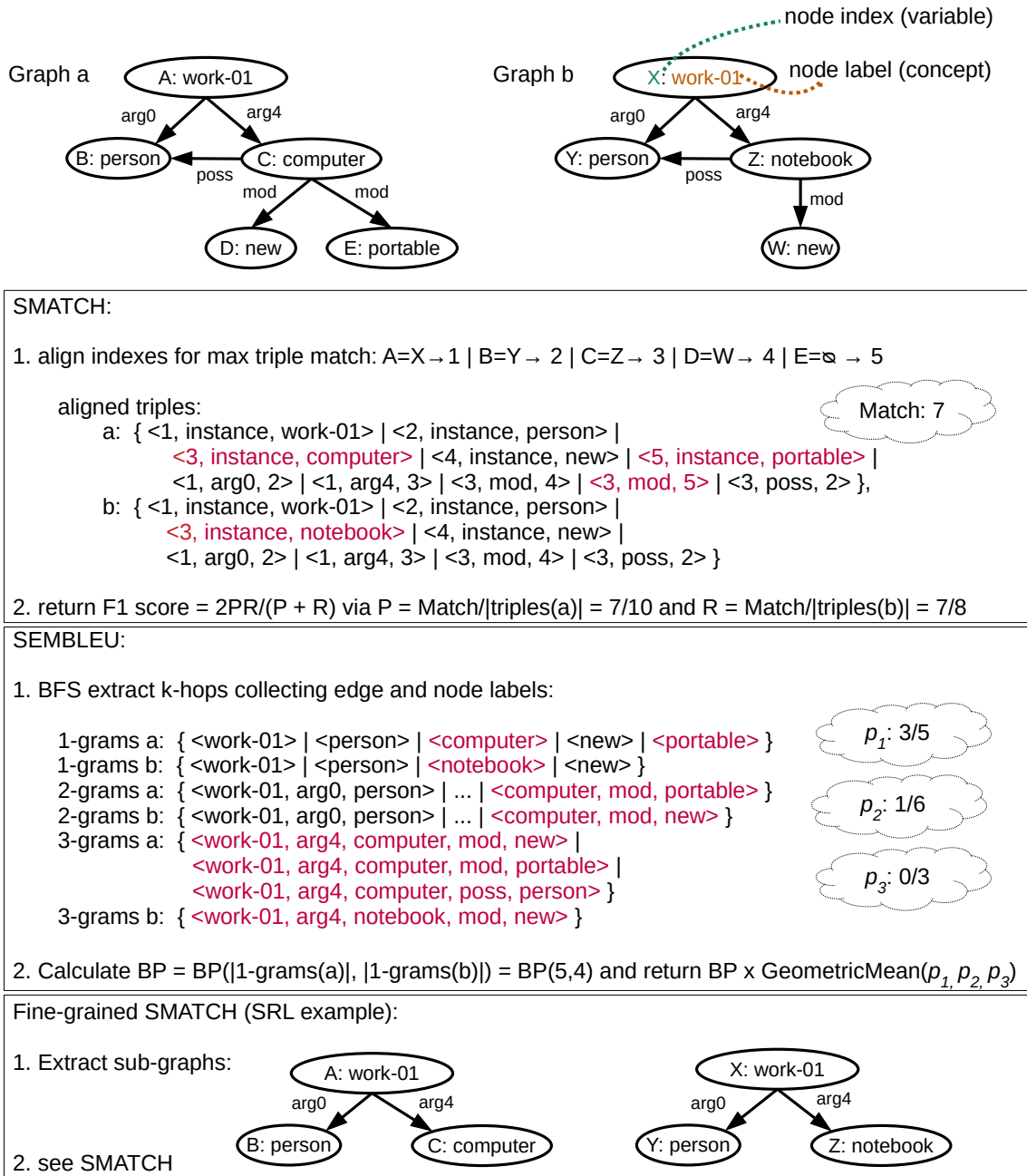


Figure 3.1: Inspection of three different AMR metric procedures. Clouds indicate the ratio of matching triples. Minor details are omitted for readability (special root node, smoothing in SEMBLEU, $k = 3$ in SEMBLEU and $w_k = \frac{1}{3} \forall k$, which results in an unweighted geometric mean).

that aligns the nodes such that the amount of matching triples is maximal. From there, we can compute a symmetric similarity score, e.g., based on F1:

$$F1 = \frac{2PR}{P+R}; \quad P = \frac{f_{map^*}(a,b)}{|triples(a)|}; \quad R = \frac{f_{map^*}(a,b)}{|triples(b)|}. \quad (3.2)$$

An example of SMATCH application is displayed in Figure 3.1: We 1. align the variables of two input graphs for a maximum triple overlap score, and then 2. return an $F1$ score based on the graphs’ matching triples.

SMATCH has been adapted by Cai and Lam (2019) to weight triple matches relative to their distance to the root, motivated by the hypothesis that “core semantics” tend to be located near a graph’s root and should be considered as more important.

BFS-based and alignment-free: SEMBLEU The problem in Eq. 3.1 is NP-complete and does not explicitly consider extended node neighborhoods. To mitigate these issues, SEMBLEU by Song and Gildea (2019) is aimed at efficiency and matching broader structures. To this aim, SEMBLEU extracts a bag of k -hop paths (per default: $k = 3$) from each AMR graph, starting from the root. During the traversal, AMR variables are replaced with their attached concepts, which alleviates the need for a costly alignment. Finally, inspired from practices in the popular NLP area of machine translation, SEMBLEU leverages the BLEU metric (Papineni et al., 2002) to compute a final score based on the extracted paths, calculating a (weighted) geometric mean over the k -gram precision scores. With weights adding up to 1, we have:

$$SEMBLEU = BP \cdot \exp \left(\sum_{k=1}^n w_k \log p_k \right) \quad (3.3)$$

$$BP = e^{\min \left\{ 1 - \frac{|b|}{|a|}, 0 \right\}}, \quad (3.4)$$

where p_k is BLEU’s *modified k-gram precision* that measures k -gram overlap of a candidate against a reference: $p_k = \frac{|kgram(a) \cap kgram(b)|}{|kgram(a)|}$. On the other hand, w_k is the (typically uniform) weight over chosen k -gram sizes. To counter cases where $p_k = 0$, SEMBLEU uses NIST geometric probability smoothing (Chen and Cherry, 2014).¹ The recall-focused ‘brevity penalty’ BP returns a value smaller than 1 when the candidate length a

¹NIST smoothing assigns a geometric sequence starting from $\frac{1}{2}$ to the k -grams with 0 matches: $p_k = \frac{1}{2^k}$ if $|kgram(a) \cap kgram(b)| = 0$.

is smaller than the reference length b (length is defined by SEMBLEU as the amount of nodes in a graph $|V|$).

An example for the SEMBLEU method is shown in Figure 3.1: We first crawl structures from the two graphs using a BFS traversal (tracing node and edge labels, to circumvent the alignment). Then 2. we calculate k -gram precision scores and return the BLEU score that is their geometric mean times the brevity penalty (which there is none in this case, since a does not contain fewer nodes than b , i.e., $BP = 1$).

At this point, we want to note that i) the omitted alignment greatly improves execution speed, but ii) we do not know whether we have to pay a price for this and how high a potential price might be. We will investigate these important questions in the next chapter of our thesis (Chapter 4). Also, we are unaware if better hyper-parameter configurations exist (e.g., regarding the k and the k -gram weights).

Fine-grained MR metrics MRs are *explicit* representations that capture *various different* types of semantic phenomena. Importantly, this means that we can measure similarity on subgraphs that capture the different semantic phenomena. With this in mind, Damonte et al. (2017) propose an AMR metric suite for such fine-grained aspectual assessments. In the end, it uses SMATCH for scoring the subgraph, but we could apply any other graph metric, too. An example for Semantic Role Label (SRL)-aspect assessment is shown in the bottom of Figure 3.1. We will leverage such metrics based on MR subgraphs in multiple places in our thesis and describe them in more detail where necessary.

Often inspired by AMR metrics: Metrics for Discourse Representation Structures and Metrics for Uniform Meaning Representation As discussed in our Background 2.2, other meaning representations such as DRS, Universal/Uniform MR, BabelNet MR, etc., bear great similarities to AMR. This is because these MRs are either derived from AMR (e.g., BabelNet, UMR) or can be expressed in an AMR-like graph format (e.g., DRS). Therefore, all metrics and applications that we will propose in this thesis allow straightforward extensions to other types of MRs. This is also reflected by the fact that, e.g., metrics for matching DRS, like the COUNTER metric (Abzianidze et al., 2019), or finding the largest common subgraph (Das et al., 2014), are, in principle, equivalent to SMATCH. Indicating more inspiration from AMR metrics, Liu et al. (2020) propose a faster metric for DRS evaluation that bears some similarities to SEMBLEU. Similarly, SMATCH has been adapted to compare Uniform Meaning Representations (Stengel-Eskin et al., 2020) and BabelNet Meaning Representation (Lorenzo et al., 2022).

3.1.2 Discussion

Clearly, previous strategies for computing MR similarity are rather coarse and thus likely do not meet our full demands for a finer similarity assessment (see also our discussion in Background section 2.4). In particular, this is because these metrics have been designed for a restricted use-case: mono-lingual evaluation of semantic parsers, where they determine a score for *structure overlap*. Determining structural overlap seems mostly legitimate in this scenario, since we are usually confronted with a reference MR and a parsed candidate MR from *the same sentence*. Therefore when evaluating a parser, we are less likely to encounter (near-)paraphrasal structures in the MRs or synonyms in MRs compared to the case where two MRs are grounded in *different sentences*. However, with extended use cases for MR metrics arising, there is increased awareness that structural matching of MRs is not sufficient for assessing the *meaning similarity* expressed by two MRs (Kapanipathi et al., 2021). This insufficiency has also been observed in cross-lingual AMR parsing evaluation (Biloshmi et al., 2020; Sheth et al., 2021; Uhrig et al., 2021), but is most prominent when attempting to compare the meaning of AMRs that represent different sentences.

That said, even in the case of mono-lingual parsing, proper MR evaluation may need to go beyond mere structural comparison, especially if abstract nodes can be projected, as is the case for a lot of MRs, including AMR.² Indeed, this issue can affect parser selection, where it gets increasingly harder to coarsely discriminate parsers. Against the backdrop of astonishing recent advances in AMR parsing, powered by the *language modeling and fine-tuning paradigm* (Bevilacqua et al., 2021), we find that parsers now achieve benchmark scores that surpass inter annotator agreement (IAA) estimates (according to structural measurement with SMATCH).³ One possible explanation for this is that structural metrics may not adequately assess (finer) differences in MRs anymore and thus fail to determine whether certain score differences are i) to be attributed to minor but valid differences in meaning interpretation or AMR structure, as they may also occur in human assessments, or whether the score differences are due to ii) significant meaning distorting errors. An example for i) are MR structures that would express paraphrastic sentences (in this case,

²Consider the sentence *I'd like to move to Berlin*. In AMR, *Berlin* triggers an abstract named entity structure. Whether *Berlin* triggers a *location*-node or a *city*-node should not influence the evaluation score (much), since both views are correct.

³Banarescu et al. (2013) find that an (optimistic) average annotator vs. consensus IAA (SMATCH) was 0.83 for newswire and 0.79 for web text. When newly trained annotators doubly annotated web text sentences, their annotator vs. annotator IAA was 0.71. Recent BART and T5 based models range between 0.82 and 0.84 SMATCH F1 scores.

purely structural assessment is bound to result in a similarity score that is too low), or different structures that represent valid interpretations of a sentence (e.g., *The man sees the woman with a telescope* – without further context, in an MR the telescope can be resolved to woman, or man). On the other hand, an example for ii) would be attaching a negation node to a predicate node in an MR (structural similarity between unaltered and altered MR may be high, but the MRs would semantically express a very different meaning).

MR metric evaluation: Benchmarking Metrics Metric evaluation is an active topic in NLP research and led to the emergence of benchmarks in various areas, most prominently, MT and NLG (Gardent et al., 2017; Zhu et al., 2018; Ma et al., 2019a). These benchmarks are useful since they help to assess and select metrics, and encourage their further development (Gehrmann et al., 2021). However, there is currently no established benchmark that defines a ground truth of *graded semantic similarity between pairs of MRs*, and how to measure similarity through these structural representations. Also, we do not have an established ground truth to assess *what* alternative AMR metrics such as SMATCH or SEMBLEU actually measure, and how their scores correlate with human judgments of the semantic similarity of sentences represented by AMRs. Later, in Chapter 5, we are going to address this lack of data and knowledge by building benchmark data sets for meaningful empirical evaluation of MR metrics and using them for investigation.

Metric evaluation for MT evaluation Metric evaluation for machine translation (MT) has received much attention over the recent years (Ma et al., 2019b; Mathur et al., 2020b; Freitag et al., 2021). When evaluating metrics for MT evaluation, it seems generally agreed upon that the main goal of a MT metric is high correlation to human ratings, mainly with respect to rating adequacy of a candidate against one (or a set of) gold reference(s). A recent shared task (Freitag et al., 2021) meta-evaluates popular metrics such as BLEU (Papineni et al., 2002) or BLEURT (Sellam et al., 2020), by comparing the metrics' scores to human scores for systems and individual segments. They find that the performance of each metric varies depending on the underlying domain (e.g., TED talks or news), and that most metrics struggle to penalize translations with errors in reversing negation or sentiment polarity, and show lower correlations for semantic phenomena including subordination, named entities and terminology. This indicates that there is potential for cross-pollination: clearly, AMR metric evaluation may profit from the vast amount of experience of metric evaluation for other tasks. On the other hand, MT evaluation may

profit from relating semantic representations, to better differentiate semantic errors with respect to their type and severity.

3.2 Meaning focused evaluation of natural language generation

Traditionally, the performance of NLG systems has been evaluated with word n-gram matching metrics such as the popular BLEU metric in MT (Papineni et al., 2002), or ROUGE (Lin, 2004) in document summarization. Yet, such metrics suffer from several well-known issues (Novikova et al., 2017; Nema and Khapra, 2018; Sai et al., 2020), e.g., due to their symbolic matching strategy they cannot account for paraphrases. Recently, to mitigate these issues, unsupervised (Zhang et al., 2020) or learned metrics (Sellam et al., 2020; Zhou and Xu, 2020) based on contextual language models have been proposed. For example, BERTscore (Zhang et al., 2020) uses BERT (Devlin et al., 2019) to encode tokens in candidate and reference, and then computes a score based on a cross-sentence word-similarity alignment. Compared with BLEU, these newer metrics are computationally more expensive but tend to show significantly higher agreement with human ratings. However, *all* of the aforementioned metrics (same as many others) tend to return scores that are hardly interpretable and therefore we often cannot tell what exactly they have measured.

These problems carry over to the evaluation of AMR2text generation, where systems aim at producing a sentence from an AMR structure: May and Priyadarshi (2017) find that BLEU does not well correspond to human ratings of generations from AMR, and Manning et al. (2020) show through human analysis that none of the existing automatic metrics can provide nuanced views on generation quality. Hence, we want to take a first step to address these issues by aiming at a clear separation of form and meaning, as called for by Bender and Koller (2020), through an MR-based metric.

First attempts of assessing semantic generation quality with MR related structures have been examined in MT using semantic role labeling (Lo, 2017) or word sense disambiguation (WSD) and natural language inference (NLI) (Carpuat, 2013; Poliak et al., 2018). Then there is SPICE that evaluates caption generation via inferred semantic propositions (Anderson et al., 2016). Just like the metric we are going to propose in Chapter 6, SPICE relies on automatic parses (a dependency parse of the caption and a scene graph predicted for the image) to evaluate content overlap of image and caption. Thus, SPICE is

a direct precursor of an NLG metric in V&L that relies on automatically produced structured representations. This thesis will also extend the previous work by showing ways of probing potentially harmful effects of incorporating automatic parsing components.

3.3 Metric extrapolation: Quality estimation of predicted structures

One of our research questions is motivated by the mere cost of creating human MRs, which is an arduous task. Thus, we pose the question: can we efficiently estimate the quality of a candidate MR (e.g., a parser’s output) by inferring a score that correctly *predicts* the similarity between the given candidate MR and a correct MR for the given sentence, where the latter remains implicit/latent? This way, we wouldn’t need to rely on a costly human reference, and could quickly assess how a parser generalizes to new data. While such *quality estimation* systems have not been built for MRs yet, we observe activity in related areas where complex linguistic structures are predicted and design of human gold annotations is costly.

Quality estimation for syntactic parsing and in MT. In syntactic parsing, the task of quality estimation has also been coined *accuracy prediction*. Same as in our presumed setup, the goal of this task is to predict parse accuracy metrics given only a sentence and its candidate parse. For instance, Ravi et al. (2008) propose a feature-based SVM regression model that predicts syntactic parser performance on different domains. An MR graph, however, differs in important ways from a syntactic tree. E.g., nodes in AMR do not explicitly correspond to words (as in dependency trees) or phrases (as in constituency trees). This makes any quality assessment harder, due to the absence of easy-gatherable textual evidence.

To estimate the quality of MT outputs, for example, Soricut and Narsale (2012) predict BLEU scores for machine-produced translations. Other researchers try to predict, i.a., the post-editing time or the missing words in an automatic translation (Joshi et al., 2016; Chatterjee et al., 2018; Kim et al., 2017; Specia et al., 2013). The fact that manually creating MR graphs is significantly more costly than a translation (due to requiring more time and trained linguists) provides another compelling argument for investigating automatic MR quality estimation techniques. To our knowledge, this thesis’ Chapter 7 is the first work to propose a quality estimation model for MR parsing.

3.4 Semantic textual similarity

As we’ve discussed before, measuring the similarity of two texts is a task that bridges many areas of NLP. To test whether we have created a metric that can adequately rate sentence similarity, it is common practice to leverage benchmark data sets that have been annotated by human annotators, using various scales or categories.

3.4.1 Data sets of human text similarity

In this section, we give an overview of some popular data sets that elicit textual semantic similarity ratings from humans.

5	<i>The two sentences are completely equivalent, as they mean the same thing.</i>
	The bird is bathing in the sink. Birdie is washing itself in the water basin.
4	<i>The two sentences are mostly equivalent, but some unimportant details differ.</i>
	Two boys on a couch are playing video games. Two boys are playing a video game.
3	<i>The two sentences are roughly equivalent, but some important information differs/missing.</i>
	John said he is considered a witness but not a suspect. “He is not a suspect anymore.” John said.
2	<i>The two sentences are not equivalent, but share some details.</i>
	They flew out of the nest in groups. They flew into the nest together.
1	<i>The two sentences are not equivalent, but are on the same topic.</i>
	The woman is playing the violin. The young lady enjoys listening to the guitar.
0	<i>The two sentences are completely dissimilar.</i>
	The black dog is running through the snow. A race car driver is driving his car through the mud.

Table 3.1: STS label explanation and examples, taken from Agirre et al. (2013).

Sentence similarity and relatedness. Popular benchmarks are STS (Agirre et al., 2013; Baudiš et al., 2016b; Cer et al., 2017) and SICK (Marelli et al., 2014). Both elicit human ratings of sentence similarity on a 5-point Likert scale. While STS aims at capturing

Task Instructions

Two snippets of text can mean the same thing even if they use very different words and phrases. Conversely, two texts that are superficially very similar in their word choice, phrasing and overall composition can have very different meanings.

For this task, you will compare two phrase or sentence length segments of text and select whether or not they have the same underlying meaning or message in terms of what they refer to, say or ask about the world.

For example, do both snippets refer to the exact same person, action, event, idea or thing? Or, are they similar but differ according to either large or small details?

Tips

- Assign labels as precisely as possible according to the underlying meaning of the two snippets rather than their superficial similarities or differences.
- Be careful of wording differences that have an important impact on what is being said or described.
- Ignore grammatical errors and awkward wordings as long as they do not obscure what is being conveyed.
- Avoid over labeling pairs with middle range scores such as "(3) Roughly Equivalent, ..." or "(2) Not equivalent, but share some details".
- Similarly, be careful of over reliance on extreme scores like "(5) Completely equivalent, ..." or "(0) On different topics."

Hot Keys

To navigate this HIT more quickly and without using a mouse, try making use of the following hot keys:

[tab] Next Question, [tab]+[shift] Previous Question, [up] / [down] (arrow keys) Navigate Pair Meaning Similarity Scale

Figure 3.2: STS annotator instructions from Agirre et al. (2016).

semantic similarity, SICK tests for *semantic* relatedness. These two aspects are highly related, but not the exact same (Budanitsky and Hirst, 2006; Kolb, 2009), but, still, the tasks often get subsumed under an umbrella term of *textual similarity*. In particular, the goal of the textual similarity datasets is to provide a standard setup for training, development and testing on different genres (news, captions, forums, etc.).

Let us inspect some concrete annotation instructions, to better understand the emergence of such data sets. The STS annotation instructions developed by Agirre et al. (2016) are shown in Figure 3.2. Finally, the collected annotations are post-processed, to increase data quality: To reduce label noise, annotator gold labels of amazon mechanical turk works are averaged over 5 different annotations, per sentence pair. To increase robustness, labels are averaged via median, and to increase and control for quality, annotators are filtered out who provide annotations that are found to deviate too much from the average annotator.⁴

The SICK creation bears strong similarities, but also strong differences. Similarly, to STS, every pair in SICK has been annotated with the degree to which two sentence meanings are related (on a 5-point scale). Human ratings were collected through a large crowd sourcing study with the *CrowdFlower* platform. The final gold relatedness labels were averaged over ten ratings from different annotators. The disagreement, measured in average standard deviation was 0.76. Much in contrast to STS creation, the annotators were not presented concrete instructions, rather, “to clarify the task to non-expert participants,

⁴A gold annotation from k-1 annotators is simulated. The remaining annotator’s agreement (measured in Pearson’s ρ) is calculated against the simulated gold annotation. If the correlation is below a certain threshold (here: $< 80\rho$), the annotator gets removed from the annotator pool.

while avoiding biasing their judgments with strict definitions, the instructions described the task only through examples of relatedness” (Marelli et al., 2014), e.g.:

- A and B are completely unrelated:
 - *Two girls are playing outdoors near a woman*
 - *The elephant is being ridden by the man*
- Very related sentences:
 - *A man is cooking pancakes*
 - *The man is cooking pancakes*

Discussion: Similarities and differences in relatedness and similarity annotation. A hidden difference in what is captured by either *relatedness* and *similarity* may be induced by data selection biases and biases through human annotator instructions, besides potential a-priori biases in different humans’ perception of similarity and relatedness. To elaborate on such a potential bias, we observe that in SICK, humans rated pairs based on examples that they were shown previously where two sentences are more or less related, while in STS the annotators received more precise instructions enhancing the clarity of the annotation task. Therefore, while probably being more precise, STS annotation instructions could bear the risk of biasing human judgments. Later in this thesis Section 9.9, we will examine if we can leverage MR metrics to shed more light on different such potential deviating similarity conceptualizations. Finally, there is a point where both data and similarity notions completely agree: the highest point on both STS and SICK scales means that two sentences are equivalent in meaning. Thus, if reduced to a binary paraphrase-classification task, the differences in relatedness and similarity may disappear, or, at least, reduce greatly. Some data sets have explicitly targeted such a binary classification setting, e.g., the Microsoft paraphrase corpus (Dolan and Brockett, 2005).

Similarity of natural language arguments. There also exist specific notions of sentence similarity that are of interest to larger NLP sub-communities. In particular, assessing the similarity of natural language arguments has received a lot of attention. It is a key task in argument mining (Reimers et al., 2019; Lenz et al., 2019) and a vital part of argument search (Maturana, 1988; Rissland et al., 1993; Wachsmuth et al., 2017; Ajjour et al., 2019; Chesnevar and Maguitman, 2004). Argument similarity ratings are also needed

for (case-based) argument retrieval (Rissland et al., 1993; Chesnevar and Maguitman, 2004), and even automated debaters (Slonim et al., 2021): to counter an opponent’s argument, one may retrieve an argument similar to theirs, but of opposite stance to the topic (Wachsmuth et al., 2018).⁵

Annotated datasets were introduced by Misra et al. (2016) who use a 6-point Likert scale (unrelated–equivalent) and Reimers et al. (2019) who introduce a newer data set with a larger number of different topics and improved annotator label averaging via MACE (Hovy et al., 2013). MACE calculates inter-rater agreement and uses this statistic to assign more trustworthy annotators higher weight (trustworthiness essentially means higher agreement to their average annotator colleague). Let us study an example for an argument pair that is judged highly similar, and an argument pair that is judged as not similar (both are from the topic *wind energy*).

- A and B are highly similar:
 - *Electricity is produced without burning fossil fuels and releasing harmful pollutants into the air.*
 - *Electricity generated by the wind, however, is clean-it does not emit either greenhouse gases like carbon dioxide or other harmful pollutants.*

- C and D are not similar:
 - *And every wind turbine slows the wind, thus reducing the wind energy available to any downwind turbines.*
 - *The wind turbine 100 may be installed on any terrain providing access to areas having desirable wind conditions.*

The similar arguments are indeed clearly similar, since they both highlight the positive environmental impact of wind energy, by reducing pollution.

Even though the dissimilar pair C, D shows some superficial similarity (induced by the shared topic), they highlight positive/negative aspects of wind turbines and thus have different stances to the topic, moreover, the second sentence is about a particular type of

⁵The stance can be determined using methods for argumentative relation classification (Kobbe et al., 2019; Opitz, 2019; Paul et al., 2020).

wind turbine, which may have further convinced the annotators in assigning a *dissimilarity* rating.⁶

Yet, a crucial issue is inherited from general sentence similarity: most methods are incapable of providing us with any deeper rationale for their predictions and it is unclear in which aspects two arguments are similar, or not, and why. It is thus also not clear whether and to what extent spurious clues or other artifacts may influence the similarity decision (Opitz and Frank, 2019c; Niven and Kao, 2019). Later, in Chapter 8, we aim at alleviating these issues by i) representing arguments with Abstract Meaning Representation (Banarescu et al., 2013) and conducting similarity assessment using well-defined graph metrics that provide explanatory AMR structure alignments; and ii) by investigating to what extent argument similarity can be projected from inferred AMR conclusions.

3.4.2 Automatic methods for rating text similarity

From bag-of words to sentence embeddings. Perhaps influenced by the fact that currently little is known about the different notions of humans about ‘relatedness’ and ‘similarity’ in general, and ‘argument similarity’, in particular, researchers address these tasks with similar methods. Up until today, a strong and generalizable baseline turns out to be treating a document as a ‘bag-of-words’ and measuring a simple overlap of two such ‘bags’.⁷ Sometimes such a simple metric is boosted with special term weighting strategies. Such term-weighting strategies can be traced back, at least, to Luhn (1957), who sets the weight of a term that occurs in a document proportional to the term frequency. Then there is the ‘dueling’ view of Jones (1972), who consider the *inverse* term frequency, which is based on the assumption that ‘less important’ terms occur more frequently in many texts (‘the’, ‘it’, ...), and thus should be down weighted since they presumably do not strongly mark content. In fact, this simple inverse term weighting strategy has turned out to be quite effective, and, as of 2015, is a component in more than 83% of text-based recommendation systems in digital libraries (Beel et al., 2016).

⁶It is interesting that in the dissimilar arguments, C can be seen as a suitable counter argument against D, defeating its premise that suggests that there is sufficient wind after building wind turbines, if there is sufficient wind before. Such an insight underpins the need for more fine-grained and explainable automatic similarity assessments.

⁷E.g., see (Opitz, 2023a).

Recent approaches employ pre-trained language models and infer distributed representations with language models such as BERT (Devlin et al., 2019) or InferSent (Conneau et al., 2017) and SBERT (Reimers et al., 2019) which can process sentences individually and thus alleviate the need for end-to-end similarity inference on each sentence pair. Instead, it infers the embedding of each sentence individually, and calculates similarity with simple vector algebra, which greatly reduces the number of costly model inferences for clustering and search (clustering: $O(n^2) \rightarrow O(n)$).

In sum, simple overlap statistics are efficient, somewhat effective, and the score calculation is transparent. Distributed vector representations can lead to enhanced effectiveness, but the similarity score calculation is intransparent. Both approaches lack high-level explainability that can sensibly explain their final rating. Therefore, we want to test whether we can measure the similarity in the space of MRs, to see how different meaning structures influence a final similarity score. We also want to test means of making the MR metrics more efficient – assessing text similarity through MRs with graph metrics can lead to limiting bottlenecks, since they tend to be slow, are often NP-hard (Cai and Knight, 2013) and the measurement relies on a parser for inferring MRs. Interestingly, concurrently to writing this thesis, we observe emerging interest in using AMR metrics for semantic search. For instance, Bonial et al. (2020) adapt SMATCH for a medically oriented search engine, and Müller and Kuwertz (2022) test SMATCH and SEMBLEU metrics for image retrieval via their captions. However, so-far, these approaches based on AMR metrics are still too ineffective and they appear barred from wide-spread adoption, mostly due to a substantial gap in accuracy and efficiency compared to state-of-the-art neural sentence embedding systems. In Chapter 9 of this thesis, we show that this lack of accuracy and efficiency can be mitigated by distilling MR metrics into a state-of-the-art search engine.

3.4.3 Explainability of decisions

Explainability of language models. While different linguistic indicators have been identified for or within BERT (Jawahar et al., 2019; Lepori and McCoy, 2020; Warstadt et al., 2019; Puccetti et al., 2021), this insight by itself does not provide us with any rationale for high (or low) sentence similarity in specific cases. And so, to achieve *local* explainability (Danilevsky et al., 2020), we would have to, at least, analyze attention weights (Clark et al., 2019; Wiegrefe and Pinter, 2019) or gradients (Selvaraju et al., 2017; Sanyal and Ren, 2021; Bastings and Filippova, 2020) of regions associated with

linguistic properties. But even then, it can be unclear how exactly to interpret the results (Jain and Wallace, 2019; Wiegrefe and Pinter, 2019; Wang et al., 2020a; Ferrando and Costa-jussà, 2021). In a different direction, Kaster et al. (2021) aim to explain BERTscore (Zhang et al., 2020) predictions with a regressor. Very recent metrics focus on explainability with prompts (Leiter et al., 2023). But unlike other explanation methods, these approaches are detached from the underlying models and may suffer from indirection effects. Token highlighting or saliency methods, on the other hand, do not provide us with a higher rationale of similarity. Instead, we would, e.g., like to have local and faithful higher-level self-explainability (Danilevsky et al., 2020) and structure a sentence embedding space into subspaces of meaning, informed by aspectual distances in MR meaning spaces. We will show that this can be achieved through MR metrics in Chapter 9.

Explanations in argumentation. Until recently, the quest for explanations in argumentation was mainly focused on theory development. The *Toulmin model* (Toulmin, 2003), for instance, offers a theory of what is needed to make an argument complete. *Argumentation schemes*, which develop taxonomies of argument types and argumentation fallacies (Walton, 2005; Walton et al., 2008) can be viewed as mechanisms for explaining functions, strengths and weaknesses of arguments. Other research aims at studying the computational and formal aspects of argumentation, e.g. abstract argumentation (Dung, 1995) and *Bayesian argumentation* (Zenker, 2013). Research in empirical *argument mining* led researchers to investigate practical methods for explanations (Lawrence, 2021; Becker et al., 2021; Gunning et al., 2019; Rago et al., 2021; Vassiliades et al., 2021). While most approaches focus on the analysis of linguistic aspects (Lauscher et al., 2021), e.g., by extracting selected features (Aker et al., 2017; Lugini and Litman, 2018) or leveraging discourse knowledge in language models (Opitz, 2019), others exploit large background knowledge graphs (Kobbe et al., 2019; Paul et al., 2020; Yuan et al., 2021) such as ConceptNet (Liu and Singh, 2004; Speer et al., 2017) or DBpedia (Mendes et al., 2012). An advantage of the approach that we will develop in Chapter 8 is the explicit graph alignment between two arguments' meaning graphs that better marks related structures, and that can help explain argument similarity judgments.

In sum, while much research has been devoted to improving the accuracy of similarity rating systems, little attention has been paid to i) leveraging MR metrics for deeper meaning-focused similarity assessments and also ii) uncovering the features that (in the

eyes of a human) make two sentences similar or dissimilar (Zeidler et al., 2022). To address both points, we provide novel generalizable MR metrics (c.f. e.g., Chapter 4) and find that MR-metrics that can potentially help uncover such features, while preserving strong rating accuracy (c.f. Chapters 8 and 9).

Part I

MR metric analysis and development

Chapter 4

MR metrics: assessment and development

4.1 Chapter outline

In this chapter, we first assess previously proposed MR metrics in a principled way, to better understand their nature and behavior, and detect perspectives for improvement. To address the detected improvement perspectives, we construct novel metrics. Overview:

1. We determine useful properties (‘principles’) for MR metrics in Section 4.2.
2. We employ our principles in Section 4.3 to analyze two previously proposed MR metrics that vastly differ in their approach (SMATCH, SEMBLEU). This way, we can categorize them and find out more about their strengths and drawbacks.
3. Based on our insights from 1. and 2., we develop novel and generalized MR metrics. Still within our analysis, we start in Section 4.3.2 with a straightforward extension of SMATCH to empower it for assessing graded lexical similarity of graph nodes (e.g., *cat* vs. *kitten*), which is crucial in generalized matching. Our metric analyses end with a summary (Section 4.4) and a reflection (Section 4.5). Finally, to further address the detected improvement perspectives, we construct novel MR metrics based on the Weisfeiler-Leman algorithm (Sections 4.6, 4.7) that allow contextualized matching of broader MR subgraphs (e.g., *kitten* vs. *cat*^{mod}→*young*).
4. We conclude the chapter with a discussion in Section 4.9.

Underlying work. The content of this chapter is mainly based on works by Opitz et al. (2020) and Opitz et al. (2021a).

4.2 Assessment of MR metrics through eight principles

In the last chapter, we introduced the SMATCH and SEMBLEU MR metrics (Section 3.1). Already there, we could notice that they differ a lot in their approach: SEMBLEU targets efficiency and extracts structures with a BFS traversal, while SMATCH computes a costly alignment and matches graph triples. However, so far, we cannot yet answer deeper questions on a metric’s properties and behavior, such as, e.g., *are there biases in a metric? Are there drawbacks that we incur when we ablate an alignment? Do both metrics have common drawbacks that we can improve upon?* If we could answer such questions, we could make more informed choices on metric selection, and become more aware of a metric’s drawbacks and how they could be alleviated. Therefore, to facilitate better understanding of MR metrics’ properties, and with an eye to perspectives for improvement, we establish *eight principles* for MR metric analysis.

Through the most neutral lens, we can view the principles as dimensions or properties which we can use to distinguish and classify MR metrics. However, we use the positively connotated term ‘principles’ because in the absence of further context we would tend to view these properties as desirable features of an MR metric. For instance, we would assume that a specific property could make a metric either more suitable for broader application, increase the meaningfulness of measurement, or offer some other advantage.

The first four principles are **mathematically motivated** and are mostly based on a mathematical notion of the term ‘metric’. These mathematically inspired principles give us some assurances about the behavior of a metric and ensure it is applicable to some particular applications such as clustering.

I ***Continuity, and upper-bound*** A similarity function should be continuous, with a natural edge case: a, b are equivalent (maximum similarity), often also presuming a lower-bound to indicate maximal distance. By choosing 1 as upper bound, and 0 as lower-bound, we obtain the following constraint on

$$metric : G \times G \rightarrow [0, 1]. \quad (4.1)$$

At some places in this thesis, due to convention, we project this score onto $[0, 100]$ and speak of *points*. Note that, speaking in terms of semantic similarity, the notion of ‘upper-bound’ is more natural than the ‘lower bound’. I.e., while the upper-bound would naturally lends itself to the interpretation that two inputs are the *same* or (*perhaps*) *paraphrases*, one could have adopt different notions of a lower-bound:

e.g., *unrelatedness, or contradiction*. Alternatively, we might not want to employ a lower-bound at all. Indeed, by setting the range of the *metric* to $(-\infty, 1]$, we can forgo a potentially arising need to build a conceptual definition of lower-bound.

II ***identity of indiscernibles*** This key principle is formalized by $metric(a,b) = 1 \Leftrightarrow a \equiv b$. It is violated if a metric assigns a value indicating equivalence to inputs that are not equivalent or if it considers equivalent inputs as different. Clearly, it increases the safety and interpretability of a metric.

III ***symmetry*** In many cases, we want a metric to be symmetric:

$$metric(a,b) = metric(b,a) \tag{4.2}$$

A metric would violate this principle if it assigns a different score after we reversed the argument order. Together with principles I and II, it extends the scope of the metric to usages beyond parser evaluation, as it also enables *sound IAA calculation, clustering and classification* of AMR graphs when we use the metric as a kernel in a classification system. In parser evaluation, one may dispense with any (strong) requirements for symmetry—however, the metric must then be applied in a standardized way, with a fixed order of arguments.

In cases where there is no defined reference, the asymmetry could be handled by aggregating $metric(a,b)$ and $metric(b,a)$, e.g., using the mean. However, it may be difficult to determine which aggregation technique we should pick, and how to interpret results. E.g. for $metric(a,b) = 0.1$ and $metric(b,a) = 0.9$, the arithmetic mean yields a score of 0.5 and the harmonic mean would yield a score much lower (0.18). In the example, the question on which score is more meaningful cannot be answered without further context. With a symmetric metric, we can forgo such and similar issues.

IV ***determinacy*** Repeated calculation over the same inputs should yield the same score. This principle is clearly desirable as it ensures reproducibility. A very small deviation may be tolerable, though it should be exactly quantifiable). Such Non-fully strictly deterministic metrics lead to rankings that are very unlikely to change under renewed metric calculation, while fully deterministic metrics *guarantee* that rankings do not change when we repeat calculation.

Now, we add a single principle to our original set of seven principles proposed in Opitz et al. (2020), with an eye to large-scale application of a metric:

- V *efficiency* focuses on practical usefulness, particularly for compute-intensive tasks such as MR-clustering or MR-search. Note that in contrast to most other principles, **efficiency** is not a binary feature and therefore it is more difficult to assess. Common ways of assessing **efficiency** may be empirical run-time tests or complexity analysis.

The next three principles we believe to be desirable specifically when comparing MR graphs. The first two of the following principles are **motivated by computer science and linguistics**, whereas the last one is **motivated from a linguistic and application-oriented perspective**.

- VI *no (unjustified or intransparent) bias*: Meaning representations consist of nodes and edges encoding specific information types. Unless explicitly justified and documented, a metric should not in unintended ways favor correctness or penalize divergence for specific types of substructures (e.g., leaf nodes). In case a metric favors or penalizes certain substructures more than others, in the interest of transparency, this should be made clear and explicit, and should be documented, verifiable and consistent. E.g., if we wish to give negation of the main predicate of a sentence a two times higher weight compared to negation in an embedded sentence, we want this to be made transparent. A concrete example for a *transparent* bias is found in Cai and Lam (2019). They analyze the impact of their top-down AMR parsing strategy by integrating a root-distance bias into SMATCH to focus on structures situated at the top of a graph. A *justified* bias can be transparent but doesn't have to be: a rather intransparent but nevertheless justified bias could be realized if it proves to be empirically indicative about human notion of similarity. This could be accomplished, e.g., by penalizing structural deviations more when they indicate an opposing meaning (in contrast to, e.g., if a structural deviation of the same degree represents a (near-)paraphrase in meaning).¹

¹After all, MRs represent *meaning of texts*, and text meaning similarity in the human mind clearly emerges in a non-linear fashion. E.g., we could be confronted with parsing errors of similar structural degree – but the structural deviations can express same, or even conflicting meanings. Emulating such a human similarity assessment in the MR space is promising for various reasons (e.g., explainability, deeper parsing evaluation), but seems non-trivial and thus it will be subject to repeated visits throughout some remaining parts of this thesis.

A metric does not have principles VI, if it has an intransparent and unjustified bias. In this case, application of the metric may potentially be hazardous since it could be unclear what is actually measured and to what degree the biases may affect the result.

VII *matching (graph-based) meaning representations – symbolic match* A goal of meaning representations is to show atomic conditions that determine the circumstances under which a sentence is true. Hence, our *metric* score should generally increase with increasing overlap of a and b , which we denote $f(a, b)$, the number of *matching* conditions. This overlap can be viewed from a **symbolic** or/and a **graded** perspective (cf., e.g., Schenker et al. (2005) who denote these perspectives as ‘syntactic’ vs. ‘semantic’). From the symbolic perspective, we can compare the nodes and edges of two graphs on a symbolic level, while from the graded perspective, we take into account the degree to which nodes and edges differ. Both types of matching involve a precondition: If a and b contain variables, we need a variable-mapping in order to match conditions from a and b .²

To test this principle, it may help to specify strict requirements and a measurement-objective of a metric. A straightforward and sensible choice could be achieved by letting the graph triples be the meaning conditions that we can match, and define the measurement-objective as the normalized size of the overlap of triples (aka Jaccard index J (Jaccard, 1912)). Let $t(x)$ be the set of triples of graph x , $t(y)$ be the set of triples of graph y , then

$$f(x, y) = |t(x) \cap t(y)|; \quad z(x, y) = |t(x) \cup t(y)|; \quad J(x, y) = f(x, y) / z(x, y), \quad (4.3)$$

which calculates the Jaccard index of two MR graphs. We can then say that a and b are considered more similar to each other than a and c iff a and b exhibit a greater relative agreement in their (symbolic) conditions:

$$metric(a, b) > metric(a, c) \Leftrightarrow \frac{f(a, b)}{z(a, b)} = J(a, b) > \frac{f(a, c)}{z(a, c)} = J(a, c). \quad (4.4)$$

²E.g., consider a graph a and its set of triples $t(a)$: $\{\langle x_1, instance, drink-1 \rangle \langle x_2, instance, cat \rangle, \langle x_1, arg0, x_2 \rangle, \langle x_1, arg1, x_3 \rangle, \langle x_3, instance, water \rangle\}$. When comparing a against an arbitrary graph b we need to judge whether a triple $\mathbf{t} \in t(a)$ is also contained in b : $\mathbf{t} \in t(b)$. For this, we need a mapping $map: vars(a) \rightarrow vars(b)$ where $vars(a) = \{x_1, \dots, x_n\}$, $vars(b) = \{y_1, \dots, y_m\}$ s.t. f is maximized.

With the above conditions, we have exactly and transparently specified a notion of similarity of MR graphs.

Discussion. While principle VII seems useful and enhances the transparency of a metric, we may have to forgo this principle in specific cases, e.g., if we want to take into account a graded semantic match of atomic graph elements or subgraphs, which is needed particularly when aiming to match MRs from different sentences. In such a case, we might have to, e.g., weight triples differently, or somehow learn to assess subgraph differences of Meaning Representations. This could potentially lead to a conflict in Eq. 4.4, trading in transparency for metric power. Indeed, a way to solve this conflict in favor of more metric power will be established in the next Principle VIII.

VIII **Graded similarity** To best get an intuition of the goal of this principle, consider two AMR graphs that match almost perfectly – except for two small divergent components. The extent of divergence can be measured by the degree of similarity of the two divergent components. In our case, we need linguistic knowledge to judge what degree of divergence we are dealing with and whether it is tolerable.

For example, consider that graph A contains a triple $\langle x, \text{instance}, \text{concept}A \rangle$ and graph B a triple $\langle y, \text{instance}, \text{concept}B \rangle$, while otherwise the graphs are equivalent, and the alignment has set $x=y$. Then, naturally we would like to have $f(A, B)$ higher when $\text{concept}A$ is similar to $\text{concept}B$ compared to the case where $\text{concept}A$ is dissimilar to $\text{concept}B$. In AMR, concepts are often abstract, so near-synonyms may even be fully admissible (*enemy-foe, location-place, etc.*).

While such (near-)synonyms are bound to occur frequently when we compare MR graphs of *different sentences* that may contain paraphrases, they can also occur in parser evaluation, where two different graphs represent the *same sentence*. For instance, whether *Berlin* in *I'd like to move to Berlin* gets projected on a location, or a city, can be a tolerable deviation.

Going beyond this, and to fully address this principle, we here also desire that *graded similarity* can extend from atomic concepts to subgraphs of arbitrary size, e.g., to reflect that $kitten(x)$ is very similar to $cat(x) \wedge mod(x, y) \wedge young(y)$.

Intermediate discussion. Principles I–III and VI–VII can be viewed as binary metric attributes/features that increase the transparency of an MR metric in the sense that when

we calculate a metric that exhibits a principle, we have exact guarantees about its behavior. Principle IV (determinacy), that we would specifically like to have for meaningful and reproducible parser evaluation: it can be strictly viewed as a binary feature, or it can be roughly projected on a binary feature, by saying that a small deviation in determinacy may perhaps be tolerable, if it is quantifiable. Principle V (efficiency), on the other hand, is focused on solving compute-intensive tasks that would require millions of metric calculations (e.g., clustering a large corpus containing n MRs would require $O(n^2)$ metric executions). Then, importantly, principle VIII is necessary not only for finer assessment of MR differences, but also to build applications where meaning representations of different sentences need to be studied.

4.3 Using our principles to assess AMR metrics

Recall that SEMBLEU differs significantly from SMATCH. A key difference is that SEMBLEU operates on reduced variable-free AMR graphs (which we here denote as g^{vf}) – instead of full-fledged AMR graphs. By eliminating variables, SEMBLEU can bypass an alignment search. This makes the calculation faster and alleviates a weakness of SMATCH: the hill-climbing search is slightly imprecise. However, SEMBLEU is not guided by aligned variables as anchors. Instead, SEMBLEU uses an n-gram statistic (BLEU) to compute an overlap score for graphs, based on k -hop paths extracted from g^{vf} , using the root node as the start for the extraction process. SMATCH, by contrast, acts directly on variable-bound graphs matching triples based on a selected alignment. Additionally, SEMBLEU can increase its k -parameter and SMATCH may match conjunctions of (inter-connected) triples. In the following analysis, however, we will adhere to their default configurations since this is how they are used in most applications.

Going over each principle, we will ask ourselves: *Why does a metric satisfy or violate a given principle?* and *What does this imply?* We start with principles from mathematics.

4.3.1 AMR metric principle analysis I–VII

We start with

I. Continuity, and upper-bound. This principle is fulfilled by both metrics as they are functions of the form $metric : G \times G \rightarrow [0, 1]$.

```

-----A-----Input-----B-----
( p / predicate-01          ( p / predicate-01
:arg0 ( x1 / man )          :arg0 ( x1 / man )
:arg1 ( x2 / man )          :arg1 x1
:arg2 x2 )                  :arg2 ( x2 / man ) )

-----Scores-----
SMATCH -> 0.667
SEMBLEU -> 1.0
-----

```

Figure 4.1: Two AMRs with semantic roles filled differently, SEMBLEU considers them as equivalent.

II. Identity of indiscernibles This principle is both intuitive and important: An AMR metric must return the maximum score if and only if the graphs are equivalent in meaning. SMATCH *conceptually satisfies* this principle, since it determines whether the two graphs are structurally isomorphic. It must be said, however, that this only holds in practice when using an optimal search strategy (Opitz, 2023c). When using a hill-climbing search, as is very common, Vanroy (2023) prove that SMATCH can get stuck in a local optimum, and return a score lower than one for exactly identical graphs.

SEMBLEU, on the other hand, *conceptually cannot comply* with this principle. Figure 4.1 shows an example of principle violation. Here, SEMBLEU yields a perfect score for two AMRs that differ in a single but crucial aspect: two of its ARG_x roles are filled with arguments that are meant to refer to distinct individuals that share the same concept. The graph on the left is an abstraction of, e.g. *The man₁ sees the other man₂ in the other man₂*, while the graph on the right is an abstraction of *The man₁ sees himself₁ in the other man₂*. SEMBLEU does not recognize the difference in meaning between a reflexive and a non-reflexive relation, assigning maximum similarity score, whereas SMATCH reflects such differences appropriately since it accounts for variables.

In sum, SEMBLEU does not satisfy principle II because it operates on a variable-free reduction of AMRs (g^{vf}). One could address this problem by reverting to canonical AMR graphs and adopting variable alignment in SEMBLEU. But this would adversely affect the advertised efficiency advantages over SMATCH. Re-integrating the alignment step would make SEMBLEU *less* efficient than SMATCH since it would add the complexity of breadth-first traversal, yielding a total complexity of $\mathcal{O}(\text{SMATCH})$ plus $\mathcal{O}(|V| + |E|)$.

III. Symmetry. This principle is fulfilled if $\forall a, b \in G : \text{metric}(a, b) = \text{metric}(b, a)$. Figure 4.2 shows an example where SEMBLEU does not comply with this principle, to a

-----A-----	-----Input-----	-----B-----
<pre>(a / and :op1 (h / heat-01 :arg1 (t / thing) :loc (b / between :op1 (w / we)) :degree (s / so)) :op2 (k / know-01 :polarity - :arg0 (i / i) :arg1 (t2 / thing :arg1-of (d / do-02))))</pre>	<pre>(k7 / know-01 :arg0 (i / i :arg0-of (d9 / do-02 :arg1 t8 :arg1 (t0 / thing :arg1-of (h2 / heat-01 :degree (s1 / so) :loc (b3 / between :op1 (w4 / we)))))) :arg1 (t8 / thing) :polarity -)</pre>	
-----Scores-----		
<pre>SEMBLEU (a,b) = 0.422</pre>	<pre><<</pre>	<pre>SEMBLEU (b,a) = 0.803</pre>
<pre>SMATCH (a,b) = 0.829</pre>	<pre>==</pre>	<pre>SMATCH (b,a) = 0.829</pre>

Figure 4.2: Large symmetry deviation of SEMBLEU for two parses of *Things are so heated between us, I don’t know what to do.*

significant extent: when comparing AMR graph a against b , it yields a score greater than 0.8, yet, when comparing b to a the score is smaller than 0.5.

We perform an experiment that quantifies this effect on a larger scale by assessing the frequency and the extent of such divergences. To this end, we parse 1368 development sentences from an AMR corpus (LDC2017T10) with an AMR parser (obtaining graph bank \mathcal{A}) and evaluate it against another graph bank \mathcal{B} (gold graphs or another parser-output). We quantify the symmetry violation by the *symmetry violation ratio* (Eq. 4.5) and the *mean symmetry violation* (Eq. 4.6) given some metric m :

$$svr = \frac{\sum_{i=1}^{|\mathcal{A}|} \mathbb{I}[m(\mathcal{A}_i, \mathcal{B}_i) \neq m(\mathcal{B}_i, \mathcal{A}_i)]}{|\mathcal{A}|} \quad (4.5)$$

$$msv = \frac{\sum_{i=1}^{|\mathcal{A}|} |m(\mathcal{A}_i, \mathcal{B}_i) - m(\mathcal{B}_i, \mathcal{A}_i)|}{|\mathcal{A}|} \quad (4.6)$$

We conduct the experiment with three AMR systems, CAMR (Wang et al., 2016), GPLA (Lyu and Titov, 2018) and JAMR (Flanigan et al., 2014), and the gold graphs. Moreover, to provide a baseline that allows us to better put the results into perspective, we also estimate the symmetry violation of BLEU (SEMBLEU’s MT ancestor) in an MT setting. Specifically, we fetch 16 system outputs of the WMT 2018 EN-DE metrics task (Ma et al., 2018) and calculate BLEU(a,b) and BLEU(b,a) of each sentence-pair (a,b) from the MT system’s output and the reference (using the same smoothing method as SEMBLEU). As *worst-case/avg.-case*, we use the outputs from the team where BLEU

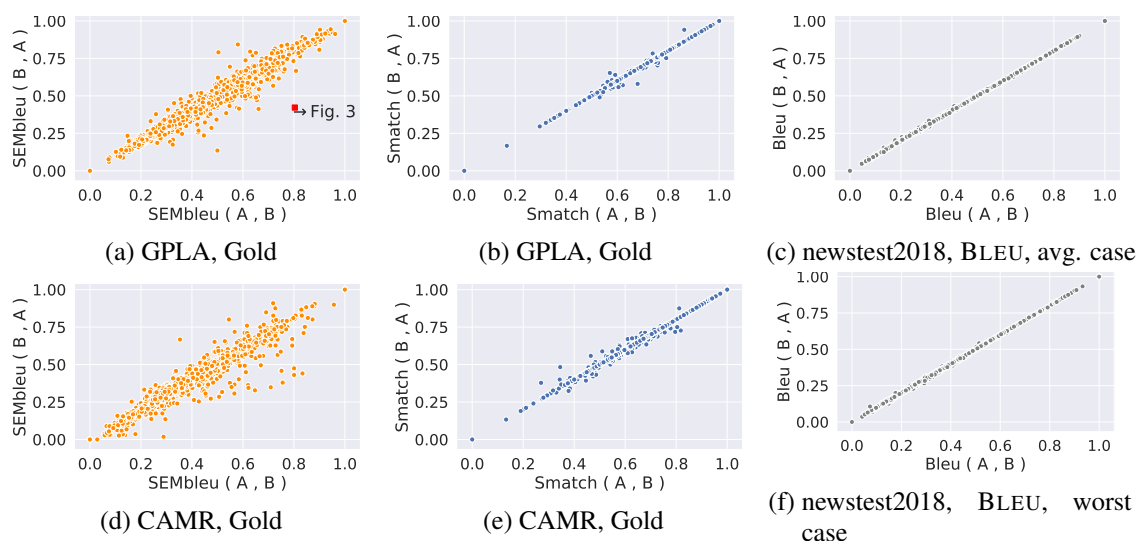


Figure 4.3: Symmetry evaluations of metrics. SEMBLEU (left column) and SMATCH (middle column) and BLEU as a ‘baseline’ in an MT task setting on newstest2018. SEMBLEU: large divergence, strong outliers. SMATCH: few divergences, few outliers; BLEU: many small divergences, zero outliers. (a) marks the case in Figure 4.2.

exhibits maximum/median msv .³

Table 4.1 shows that more than 80% of the evaluated AMR graph pairs lead to a symmetry violation with SEMBLEU (as opposed to less than 10% for SMATCH). The msv of SMATCH is considerably smaller compared to SEMBLEU: 0.1 vs. 3.2 points F1 score. Even though the BLEU metric is inherently asymmetric, most of the symmetry violations are negligible when applied in MT (high svr , low msv , Table 4.2). However, when applied to AMR graphs ‘via’ SEMBLEU the asymmetry is amplified by a factor of approximately 16 (0.2 vs. 3.2 points). Figure 4.3 visualizes the symmetry violations of SEMBLEU (left), SMATCH (middle) and BLEU (right). The SEMBLEU-plots show that the effect is widespread, some cases are extreme, many others are less extreme but still considerable. This stands in contrast to SMATCH but also to BLEU, which itself appears well calibrated and does not suffer from any major asymmetry.

In sum, symmetry violations with SMATCH are much fewer and less pronounced than those observed with SEMBLEU. In theory, SMATCH is fully symmetric, however, violations can occur due to alignment errors if we use the greedy variable-alignment search (future research can use the optimal solver from (Opitz, 2023c) to reduce the symmetry error to zero). By contrast, the symmetry violation of SEMBLEU is intrinsic to the

³worst: LMU uns.; avg.: LMU NMT (Huck et al., 2017).

Graph banks	symmetry violation			
	svr (% , $\Delta > 0.0001$)		msv (in points)	
	SMATCH	SEMBLEU	SMATCH	SEMBLEU
Gold \leftrightarrow GPLA	2.7	81.8	0.1	3.2
Gold \leftrightarrow CAMR	7.8	92.8	0.2	3.1
Gold \leftrightarrow JAMR	5.0	87.0	0.1	3.2
JAMR \leftrightarrow GPLA	4.2	86.0	0.1	3.0
CAMR \leftrightarrow GPLA	7.4	93.4	0.1	3.4
CAMR \leftrightarrow JAMR	7.9	91.6	0.2	3.3
avg.	5.8	88.8	0.1	3.2

Table 4.1: svr (Eq. 4.5), msv (Eq. 4.6) of AMR metrics.

data: newstest2018 \leftrightarrow (\cdot)	BLEU symmetry violation, MT	
	svr (% , $\Delta > 0.0001$)	msv (in points)
worst-case	81.3	0.2
avg-case	72.7	0.2

Table 4.2: svr (Eq. 4.5), msv (Eq. 4.6) of BLEU, MT setting.

	# restarts				
	1	2	3	5	7
corpus vs. corpus	$2.6e^{-4}$	$1.7e^{-4}$	$8.1e^{-5}$	$5.7e^{-5}$	$5.6e^{-5}$
graph vs. graph	$1.3e^{-3}$	$1.0e^{-3}$	$8.5e^{-4}$	$5.3e^{-4}$	$4.0e^{-4}$

Table 4.3: Expected determinacy error ϵ in SMATCH F1.

method since the underlying overlap measure BLEU is inherently asymmetric, however, this asymmetry is amplified in SEMBLEU compared to BLEU.⁴

IV. Determinacy. This principle states that repeated calculations of a metric should yield identical results, a feature that is particularly desirable in applications such as parser evaluation. Since there is no randomness in SEMBLEU, it fully complies with this principle. The most commonly used implementation of SMATCH, however, does not guarantee deterministic variable alignment results, since it aligns the variables by means of greedy hill-climbing. However, multiple random initializations together with the small set of

⁴As we show below (principle V), this is due to the way in which k-grams are extracted from variable-free AMR graphs.

AMR variables imply that the deviation will be $\leq \varepsilon$ (a small number close to 0). Additionally, $\varepsilon = 0$ is guaranteed when resorting to a (costly) ILP calculation. In a recent work (Opitz, 2023c) we revisit this issue and find that ILP optimal search is *feasible* in the standard evaluation case and offers us valuable upper-bounds and optimal solutions, resulting in slightly higher and fully deterministic SMATCH evaluation scores.

To see what happens if we use the common hill-climber, let us inspect Table 4.3 where we measure the expected ε : it displays the SMATCH F1 standard deviation with respect to 10 independent runs, on a corpus level and on a graph-pair level (arithmetic mean).⁵ We see that ε is small, even when only one random start is performed (corpus level: $\varepsilon=0.0003$, graph level: $\varepsilon=0.0013$).

We conclude that the hill-climbing seed in SMATCH is unlikely to have any significant effects on the final score, and multiple restarts provide an acceptable stability level. However, to secure transparency and true SMATCH scores in sensible applications such as parser evaluation and ranking, we may consider using optimal SMATCH (Opitz, 2023c). Indeed, if taking the feature of determinacy strictly, only SEMBLEU and SMATCH-ILP (Opitz, 2023c) are deterministic, if allowing a small tolerance, then SMATCH with hill-climber can be viewed to be deterministic, too. Recall that non-fully strictly deterministic metrics lead to parser rankings that are very unlikely to change under repeated metric calculation, while (fully) deterministic metrics *guarantee* that parser rankings do not change when we repeat calculation.

V. efficiency. While for most parsing evaluation applications, due to limited data size, it does not seem to matter much whether a metric is fast, or slow, the efficiency of a metric can have a great impact on – or even restrict access to – extended use-cases such as AMR clustering or AMR search on large-scale data, where we would need to calculate many comparisons of pairs. Clearly, SEMBLEU has the edge over SMATCH, since it does not calculate a costly alignment. Therefore, SEMBLEU needs only a few milliseconds to process 1,000 pairs, while SMATCH can require up to 60 seconds (Song and Gildea, 2019). On the other hand, at the cost of some of SMATCH’s power and risking the violation of other principles (e.g., identity of indiscernibles), it would be relatively straightforward to make it similarly fast as SEMBLEU. Indeed, we could replace all variables in triples with their concepts and match triples without an alignment. However, for simplicity, we summarize that SEMBLEU fulfills this principle, while SMATCH doesn’t.

⁵Data: dev set of LDC2017T10, parses by GPLA.

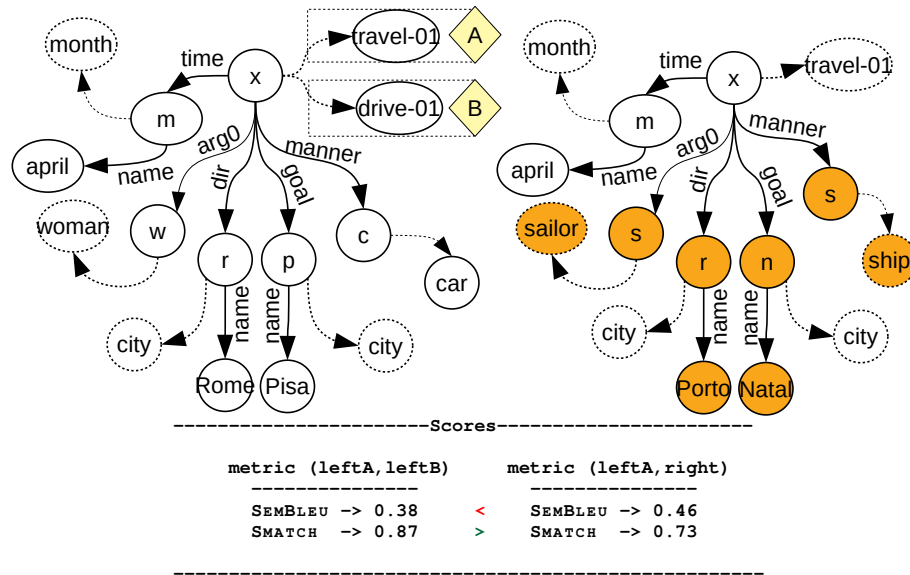


Figure 4.4: Left: *In April, a woman rides a car from Rome to Pisa.* root nodes A: *travel-01* vs. B: *drive-01*. Right: *In Apr., a sailor travels with a ship from P. to N.*

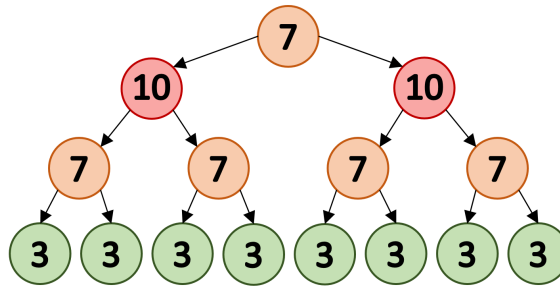


Figure 4.5: # of k -grams entered by a node in SEMBLEU.

VI. No unintentional bias. A similarity metric of MRs should not unjustifiably or unintentionally favor the correctness or penalize errors pertaining to any (sub-)structures of the graphs. However, we find that SEMBLEU is affected by a hidden bias that can affect certain types of structures differently, in particular structures that relate to high-degree nodes. The bias arises from two related factors: (i) when translating g to g^{vf} , SEMBLEU replaces variable nodes with concept nodes. Thus, nodes which were leaf nodes in g can be raised to highly connected nodes in g^{vf} . (ii) the graph traversal starts at the root node and is conducted in a breadth-first manner. These two factors have the effect that concept leaves – now occupying the position of (former) variable nodes with a high number of outgoing (and incoming) edges – will be visited and extracted more frequently than others.







			
SEMBLEU	$\mathcal{O}(3d)$	$\mathcal{O}(d^2 + d)$	$\mathcal{O}(d^2 + 2d)$
SMATCH	$\mathcal{O}(d)$	$\mathcal{O}(d)$	$\mathcal{O}(d)$

Table 4.4: Error impact depending on error location in a tree with node degree d .

The two factors in combination make SEMBLEU penalize a wrong concept node harshly when it is attached to a high-degree variable node (the leaf is raised to high-degree when transforming g to g^{vf}). Conversely, correctly or wrongly assigned concepts attached to nodes with low degree are only weakly considered. This may have severe consequences, e.g., for *negation*, since negation *always* occurs as a leaf in g and g^{vf} . Therefore, SEMBLEU is benevolent to polarity errors. As a full example, consider Figure 4.4. SEMBLEU considers two graphs that express quite distinct meanings (left and right) as more similar than graphs that are almost equivalent in meaning (left, variant a vs. b). This is because the leaf that is attached to the root is raised to a highly connected node in g^{vf} and thus is over-frequently contained in the extracted k-grams, whereas the other leaves will remain leaves in g^{vf} .

Analyzing and quantifying SEMBLEU’s bias. To better understand the bias, we study three limiting cases: (i) the root is wrong () (ii) d leaf nodes are wrong () and (iii) one branching node is wrong (). Depending on a specific node and its position in the graph, we would like to know onto how many k-grams (SEMBLEU) or triples (SMATCH) the errors are projected. For the sake of simplicity, we assume that the graph always comes in its simplified form g^{vf} , that it is a tree, and that every non-leaf node has the same out-degree d .

The result of our analysis is given in Table 4.4⁶ and exemplified in Figure 4.5. Both show that the number of times k-gram extraction visits a node heavily depends on its position and that with growing d , the bias gets amplified (Table 4.4).⁷ E.g., when $d=3$, 3 wrong leaves yield 9 wrong k-grams, and 1 wrong branching node can already yield 18 wrong k-grams. By contrast, in SMATCH the weight of d leaves always approximates the weight of 1 branching node of degree d .

⁶Proof sketch, SMATCH, d leaves: d triples, a root: d triples, a branching node: $d+1$ triples. SEMBLEU $_{k=3}^{w_k=1/3}$, d leaves: $3d$ k-grams (d tri, d bi, d uni). A root: d^2 tri, d bi, 1 uni. A branching node: d^2+d+1 tri, $d+1$ bi, 1 uni. \square

⁷Consider that in AMR, d can be quite high, e.g., a predicate with multiple arguments and additional modifiers.

In sum, in SMATCH the impact of a wrong node is constant for all node types and rises linearly with d . In SEMBLEU the impact of a node rises approximately quadratically with d and it also depends on the node type, since it raises some (but not all) leaves in g to connected nodes in g^{vf} .

Eliminating biases. A possible approach to reduce SEMBLEU’s biases could be to weigh the extracted k-gram matches according to the degree of the contained nodes. However, this would imply that we assume some k-grams (and thus also some nodes and edges) to be of greater importance than others – in other words, we would eliminate one bias by introducing another. Since the breadth-first traversal is the metric’s backbone, this issue may be hard to address well. When BLEU is used for MT evaluation, there is no such bias because the k-grams in a sentence appear linearly.

VII. Graph matching: symbolic perspective. This principle requires that a metric’s score grows with increasing overlap of the conditions that are simultaneously contained in a and b . SMATCH fulfills this principle since it matches two AMR graphs s.t. that the triple matches are maximized.⁸ Hence, SMATCH can be seen as a graph matching algorithm that works on any pair of graphs, including graphs with nodes that are variables. It fulfills the Jaccard-based overlap objective which symmetrically measures the amount of triples on which two graphs agree, normalized by their respective sizes (since SMATCH $F1 = 2J/(1 + J)$ is a monotonic relation).

Since SEMBLEU does not satisfy principles II and III (id. of indiscernibles and symmetry), it is a corollary that it cannot fulfill the overlap objective.⁹ Generally, SEMBLEU matches the results of a graph-to-bag-of-paths reduction function and the input may not be guaranteed to be recoverable from the output. Thus, matching the outputs of this function cannot be equated to matching the inputs on a graph-level.

⁸Again, same as in our analysis of Principle II (§4.3.1), if we are strict, this is only true conceptually for SMATCH if we use an optimal solver (Opitz, 2023c), while with hill-climbing we can find examples where this concept can be violated, e.g., c.f., (Vanroy, 2023).

⁹Proof by symmetry violation: $\exists a, b: \text{metric}(a, b) > \text{metric}(b, a) \Rightarrow f(a, b) > f(b, a) \rightarrow \text{✗}$, since $f(a, b) = |t(a) \cap t(b)| = |t(b) \cap t(a)| = f(b, a) \quad \square$. Another proof by identity of indiscernibles: $\exists a, b, c: \text{metric}(a, b) = \text{metric}(a, c) = 1 \wedge f(a, b)/z(a, b) = 1 > f(a, c)/z(a, c) \rightarrow \text{✗} \quad \square$

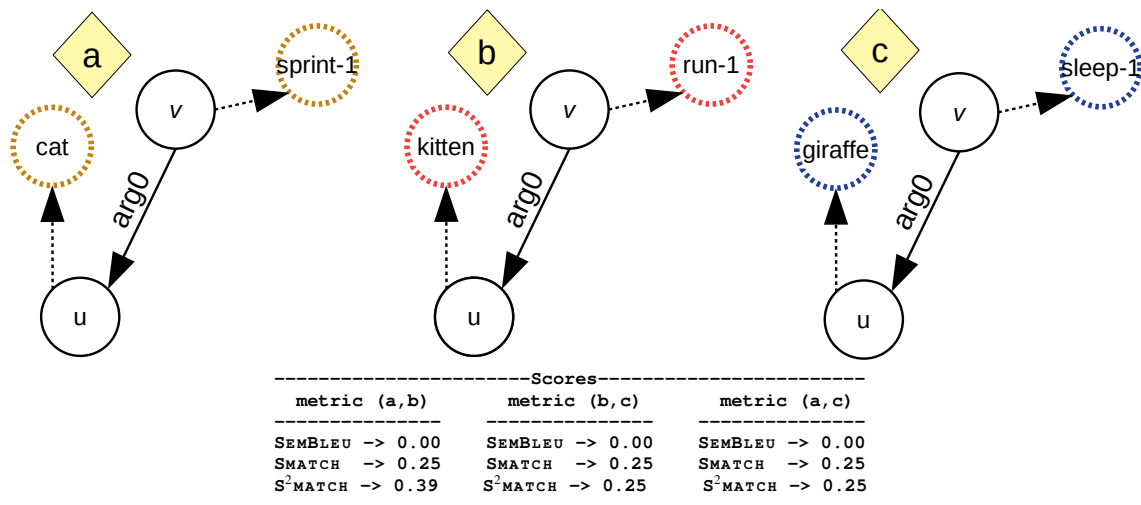


Figure 4.6: Three different MR graphs representing *The cat sprints*; *The kitten runs*; *The giraffe sleeps* and pairwise similarity scores from SEMBLEU, SMATCH and our new metric S²MATCH.

4.3.2 Towards enabling principle VIII with a novel metric: S²MATCH

This section focuses on principle VIII, semantically graded graph matching, a principle that none of the AMR metrics considered so-far satisfies. A fulfillment of this principle also increases the capacity of a metric to assess the semantic similarity of two AMR graphs from *different sentences*. E.g., when clustering AMR graphs or detecting paraphrases in AMR-parsed texts, the ability to abstract away from concrete lexicalizations is clearly desirable. Consider Figure 4.6 with three different graphs. Two of them (*a, b*) are similar in meaning and differ significantly from *c*. However, both SMATCH and SEMBLEU yield the same result in the sense that $metric(a,b) = metric(a,c)$. Put differently, neither metric takes into account that *giraffe* and *kitten* are two quite different concepts, while *cat* and *kitten* are more similar. However, we would like this to be reflected by our metric and obtain $metric(a,b) > metric(a,c)$ in such a case.

S²MATCH means (*Soft Semantic match*, pronounced: [estu:mætʃ]) and builds on SMATCH but differs from it in one important aspect: instead of maximizing the number of (hard) triple matches between two graphs during alignment search, we maximize the (soft) triple matches by taking into account the semantic similarity of concepts. Recall that an AMR graph contains two types of triples: instance and relation triples (e.g., Figure 4.6, left: $\langle u, instance, cat \rangle$ and $\langle v, arg0, u \rangle$). In SMATCH, two triples can only be matched if they are identical. In S²MATCH, we relax this constraint, which has also the potential to yield a

	avg. msv (Eq. 4.6)	determinacy error		
		1 restart	2 restarts	4 restarts
SMATCH	0.0011	$1.3e^{-3}$	$1.0e^{-3}$	$5.3e^{-4}$
S ² MATCH	0.0005	$9.0e^{-4}$	$6.1e^{-4}$	$2.1e^{-4}$
relative change	-54.6%	-30.7%	-39.0%	-60.3%

Table 4.5: S²MATCH improves upon SMATCH by reducing the extent of its non-determinacy.

different, and possibly, a better variable alignment. More precisely, in SMATCH we match two instance triples $\langle u, \text{instance}, x \rangle \in a$ and $\langle \text{map}(u), \text{instance}, y \rangle \in b$ as follows:

$$\text{hardMatch} = \mathbb{I}[x = y] \quad (4.7)$$

where $\mathbb{I}(c)$ equals 1 if c is true and 0 otherwise. S²MATCH relaxes this condition:

$$\text{softMatch} = 1 - d(x, y), \quad (4.8)$$

where d is an arbitrary distance function $d : X \times X \rightarrow [0, 1]$. E.g., in practice, if we represent the concepts as vectors $x, y \in \mathbb{R}^n$, we can use

$$d(x, y) = \min \left\{ 1, 1 - \frac{y^T x}{\|x\|_2 \|y\|_2} \right\}. \quad (4.9)$$

When plugged into Eq. 4.8, this results in the *cosine similarity* $\in [0, 1]$. It may be suitable to set a threshold τ (e.g., $\tau = 0.5$), to only consider the similarity between two concepts if it is above τ ($\text{softMatch} = 0$ if $1 - d(x, y) < \tau$).

To summarize, S²MATCH is designed to either yield the same score as SMATCH— or a slightly increased score when it aligns concepts that are symbolically distinct but semantically similar. In the following pilot experiments, we use cosine (Eq. 4.9) and $\tau = 0.5$ over 100 dimensional GloVe vectors (Pennington et al., 2014a).

An example, from parser evaluation, is shown in Figure 4.7. Here, S²MATCH increases the score to 63 F1 (+10 points) by detecting a more adequate alignment that accounts for the graded similarity of two related AMR concepts pairs. We believe that this is justified: The two graphs are very similar and an F1 of 53 is too low, doing the parser injustice.

On a technical note, the changes in alignments also have the outcome that S²MATCH mends some of SMATCH’s practical flaws, by reducing the hill-climbers imprecision: it better addresses principles III and IV, reducing the symmetry violation and determinacy error (Table 4.5).

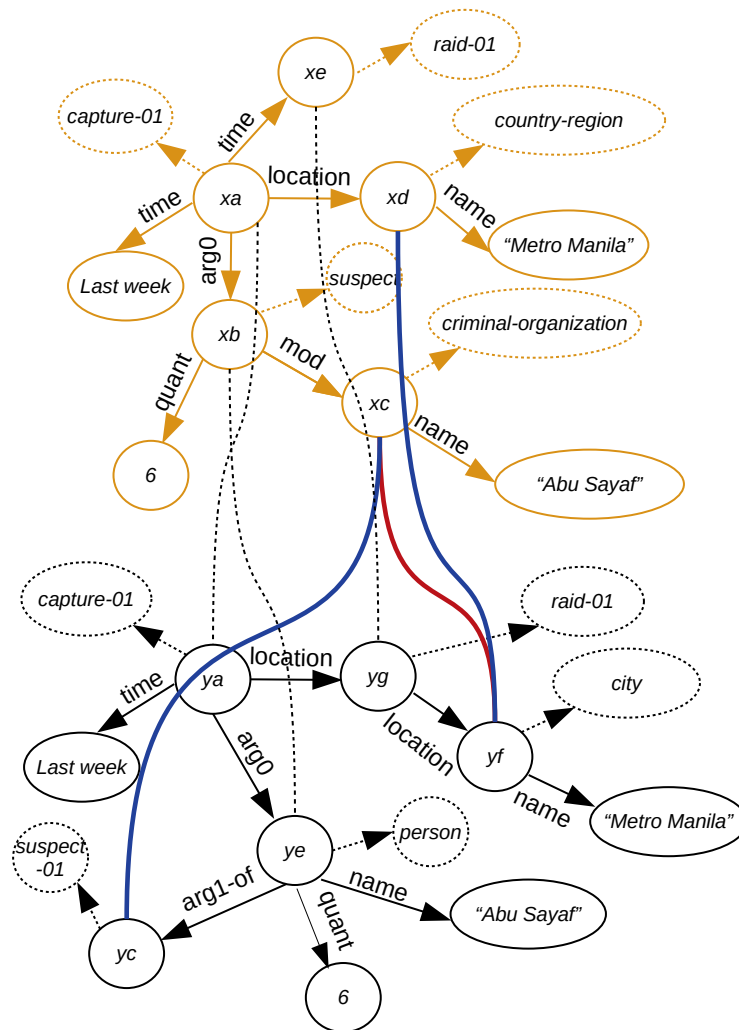


Figure 4.7: ‘6 Abu Sayyaf suspects were captured last week in a raid in Metro Manila.’ gold (top) vs. parsed AMR (bottom). SMATCH aligns *criminal-organization* to *city* (red); S²MATCH aligns *criminal-organization* to *suspect-01*, *city* to *country-region* (blue).

4.4 Summary of our metric analyses

Table 4.6 summarizes our analysis results. Principle I is fulfilled by all metrics as they exhibit *continuity*, and an *upper bound*. Principle II, however, is not satisfied by SEMBLEU since it can mistake two graphs of different meaning as equivalent. This is because it ablates a variable-alignment and thus cannot capture all coreferences. Yet, a positive outcome of this is that it is *fast to compute* (principle V), which is appealing for practical and large-scale applications where rapid metric computation is required. It also marks a point by fully satisfying principle IV, yielding fully deterministic results. SMATCH, by contrast, either needs to resort to a costly ILP solution or (in practice) often uses hill-climbing with

principle	SMATCH	+ILP (Opitz, 2023c)	SEMBLEU	S ² MATCH	+ILP (Opitz, 2023c)
I. Cont., upper-bound	✓	✓	✓	✓	✓
II. id. of indiscernibles	✓ _ε	✓	✗	✓ _{δ<ε}	✓
III. symmetry	✓ _ε	✓	✗	✓ _{δ<ε}	✓
IV. determinacy	✓ _ε	✓	✓	✓ _{δ<ε}	✓
V. efficiency	✗	✗	✓	✗	✗
VI. low bias	✓	✓	✗	✓	✓
VII. symb. graph matching	✓	✓	✗	✓	✓
VIII. graded graph matching	✗	✗	✗	✓ ^{LEX}	✓ ^{LEX}

Table 4.6: Evaluation of three AMR metrics using our eight principles. ✓_ε: fulfilled with a very small ε -error. +ILP column indicates a variant of the metric in the left neighbor column that uses optimal solution instead of hill-climbing.

multiple restarts to reduce divergence.¹⁰

A central insight brought out by our analysis is that SEMBLEU exhibits *biases* that are hard to control, conflicting with principle VI. This is caused by two (interacting) factors: (i) The extraction of k-grams is applied on the graph top to bottom and visits some nodes more frequently than others. (ii) It raises some (but not all) leaf nodes to connected nodes, and these nodes will be overly frequently contained in extracted k-grams. We have shown that these two factors in combination lead to large biases that researchers should be aware of when using SEMBLEU (Section 4.3.1). Its ‘ancestor’ BLEU does not suffer from such biases since it extracts k-grams linearly from a sentence.

Given that SEMBLEU is built on BLEU, it is inherently *asymmetric*. However, we have shown that the asymmetry (principle III) measured for BLEU in MT is amplified by SEMBLEU in AMR, mainly due to the biases it incurs (principle VI). While asymmetry can be tolerated in parser evaluation if outputs are compared against gold graphs in a standardized manner, it is difficult to apply an asymmetric metric to measure IAA or to compare parses for detecting paraphrases, or in tri-parsing, where no reference is available. If the asymmetry is amplified by a bias, it becomes harder to judge the scores. Finally, considering that SEMBLEU does not match AMR graphs on the graph-level but matches extracted bags-of-k-grams, it turns out that it cannot be categorized as a graph matching algorithm as defined in principle VI.

¹⁰Again, it is important to note that even in the case of multiple restarts, this can still be critical for some evaluation cases (Vanroy, 2023). In a very recent work (Opitz, 2023c), we re-examine the implementation of the standard evaluation protocol with SMATCH and show that using ILP is *feasible* in the average evaluation case and provides optimal scores that yield slightly higher evaluation scores. The ILP can be applied to even larger graphs by means of lossless compression of search space. We release all the code in the SMATCH++, a package for standardized AMR evaluation with SMATCH: <https://github.com/flipz357/smatchpp>.

Principle VII (symbolic graph matching) is clearly fulfilled by SMATCH. It searches for an optimal variable alignment and counts matching triples, therefore also being able to assess (structural) graph isomorphism. As a corollary, it fulfills principles I, II, III and VI.

Our principles also have helped us detect a weakness of all present AMR metrics: they operate on a discrete level and cannot assess graded meaning differences, an issue that is pointed out by our principle VIII. As a first step, we proposed S^2 MATCH: it preserves beneficial principles of SMATCH but is benevolent to slight lexical meaning deviations. Besides parser evaluation, this principle makes the metric also more suitable for other tasks, e.g., it can be used as a kernel in an SVM that classifies AMRs to determine whether two sentences are equivalent in meaning. In such a case, S^2 MATCH is bound to detect meaning-similarities that cannot be captured by SMATCH or SEMBLEU, e.g., due to paraphrases being projected into the parses.

4.5 Discussion: Limits of principle-based metric analysis and outlook

4.5.1 On dueling principles

On one hand, our analyses show that SMATCH is a solid metric for comparing MRs. Importantly, in contrast to more efficient alignment-free metrics, it can actually verify if two graphs are structurally isomorphic, or not.¹¹ But an MR graph is not just ‘some’ graph — it is a *meaning* representation. So at the end of the day, we may not be so much interested in designing a metric that merely assesses *symbolic structural* graph similarity, but instead we would like to have a graph metric that assesses *meaning* graph similarity. To achieve this goal, we attempted to make a first step with Principle VIII and S^2 MATCH. However, this step may not be enough: how meaning arises from subgraphs of the MR is highly non-trivial and it seems clear that different graph parts impact its overall meaning quite differently. Indeed, quite obviously, the overall meaning of an MR does not seem to monotonously depend on structural changes: A small structural change in meaning structure could change the expressed meaning to a larger degree than a larger structural change. As a simple example, imagine a meaning structure of a sentence in which we now artificially attach a negation label to the root vs. the meaning structure of an arbitrary

¹¹If used with an optimal solver (Opitz, 2023c).

paraphrase of the sentence. The former action (attaching a single negation node) will likely change the graph's structure only to a marginal degree (the relative degree depends on the overall size of the MR), but it probably has a large impact on the meaning that is conveyed by the graph (due to negating the main predicate). With structural similarity assessment, this might result in a similarity score that is misleadingly high. By contrast, the MR of the paraphrastic sentence could easily result in a much different graph structure, but the captured meaning will be almost the same, leading to a misleadingly low structural similarity score.

So, similar to how humans would construct an assessment of the similarity of two texts, we would like to have MR metrics that can more deeply assess the meaning similarity of two MRs. Indeed, to best accomplish that what is aimed at by Principle VIII, we want to work towards building a metric that can *approximate similarity as perceived by humans* through MR graphs. To achieve this, however, we might have to trade in some other principles, the result of which could affect potentially desirable features such as interpretability/transparency of measurement. E.g., we might have to increase some biases by differently weighting divergences of particular types of structures, to increase alignment of our metric with a human's view. Similarly, Principle VII that ensures a transparent and statistically clearly interpretable measurement (more equal triples \rightarrow higher score) may also be traded against Principle VIII. Other principles, however, could be expected to remain untouched. For instance, we see no obvious reason why we should generally forgo symmetry, since presumably a human similarity rating would also be (mostly) symmetric.

4.5.2 Parallels from machine translation evaluation research

There is an interesting parallel when viewing metrics for machine translation: There is the popular BLEU metric (Papineni et al., 2002) that has been recently criticized a lot for failing to measure more fine-grained meaning differences, leading to the development of other metrics such as BLEURT (Sellam et al., 2020) or BERTscore (Zhang et al., 2020) that achieve higher correlation with human raters. However, metrics like BERTscore or BLEURT are very complex and based on large black-box language models, so we cannot be sure how/what exactly they measure. On the other hand, we know *exactly* how/what BLEU measures, since it is expressed by a formula that humans (with a bit of prior math knowledge) can understand. In that sense, BLEU is a principled measurement. (Modulo brevity penalty,) BLEU calculates a geometric mean over k-gram precision scores. A

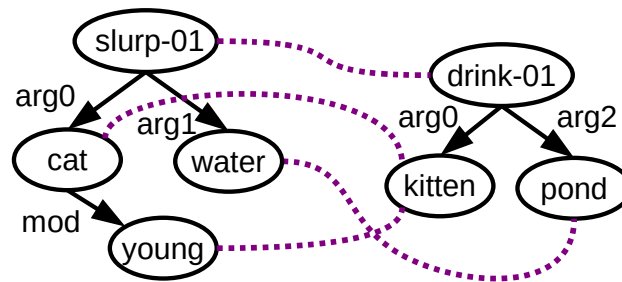


Figure 4.8: Similar MRs, with sketched alignments.

k-gram precision score tells us the probability that a predicted k-gram is also in the reference. A geometric mean performs a normalized multiplication. Therefore, BLEU tells us about: Given we predict any k-gram, how likely is it that it is found in the reference? The geometric mean then returns a normalized joint probability: Given we predict one 1gram,..., and one kgram, how likely is it that we can find them all in the reference? But, as is evident from recent developments, the MT community seems ready to forgo such transparency and probabilistic meaningful measure in favor of more semantically powerful metrics that assess the *semantic* adequacy of a candidate against a reference.

In sum, this i) underlines that often principles can be dueling, with some that are assigned higher importance winning over others ii) and it shows the importance to understand deeper meaning similarities also in the MR space, for MR-based tasks and evaluation. Thus we can say that there is a clear need to develop more powerful MR metrics that better capture subgraph similarities to better reflect the human view.

4.5.3 We need better MR metrics: Making the case with an example

To work more in the direction of MR metrics that can assess graded meaning similarity, we proposed S²MATCH. But S²MATCH is only a *first step* towards assessing graded MR differences. While it *can* consider graded meaning differences on *atomic parts of a meaning structure*, by construction, it *cannot* compare *subgraphs of different sizes*.

As an example, consider Figure 4.8, which shows two MRs that convey very similar meanings. All aforementioned metrics assign this pair a low similarity score, and – if alignment-based, as is SMATCH– find only subpar alignments. In particular, both SMATCH and S²MATCH align *drink-01* to *slurp-01* and *kitten* to *cat*, which results in a single matching triple $\langle x, \text{arg0}, y \rangle$ and a very low similarity score (0.14) for SMATCH, and three additional partially matching triples for S²MATCH (*kitten*–*cat*, *slurp*–*drink*, *pond*–*water*) with an increased similarity score (0.52) for S²MATCH. But importantly,

the example also shows two meaning *sub*-structures of different size, that express a near-equivalence in meaning (*young cat-kitten*). Therefore, we would like to have a metric that can account for such differences by providing us with a higher similarity score and, ideally, an explanatory graph alignment, similarly to what we have sketched in the Figure. Such an alignment must go beyond a structural 1:1 node mapping and allow us to map subgraphs of arbitrary size, i.e., sets of nodes, including their relations.

4.6 Building novel MR metrics from Weisfeiler-Leman

Previous MR metrics have complementary strengths and weaknesses. As discussed above, their shared weakness is that they cannot capture graded similarity of sub-structures. Other weaknesses are exhibited by some particular metrics (e.g., SEMBLEU’s unclear biases, or SMATCH’s slow execution efficiency). Therefore, we aim to propose new AMR metrics that are able to mitigate the weaknesses of different MR metrics, while unifying their strengths, aiming at the best of all worlds. Ideally, we would want want:

- i) an **interpretable alignment** and many principles (SMATCH);
- ii) a **fast metric** (SEMBLEU);
- iii) **matching larger substructures** (SEMBLEU)
- iv) and **assessment of graded similarity of AMR subgraphs** (extending S^2_{MATCH}).

This section proposes to make use of the *Weisfeiler-Leman graph kernel (WLK)* (Weisfeiler and Leman, 1968; Shervashidze et al., 2011) to assess AMR similarity. The idea is that WLK provides us with SEMBLEU-like matches of larger sub-structures, while bypassing potential biases induced by the BFS-traversal (see above, Section 4.3.1). We then describe the *Wasserstein Weisfeiler Leman kernel (WWLK)* (Togninalli et al., 2019) that is similar to WLK but provides i) an alignment of atomic and non-atomic substructures (going beyond SMATCH) and ii) a graded match of substructures (going beyond S^2_{MATCH}). Finally, we further adapt WWLK to $WWLK_{\Theta}$, a variant that we tailor to learn semantic edge parameters to achieve more control over MR graph similarity and build a more human-aligned MR similarity rating strategy.

4.6.1 Basic Weisfeiler-Leman Kernel (WLK)

The Weisfeiler-Leman kernel (WLK) method (Shervashidze et al., 2011) derives subgraph features from two input graphs. WLK has shown its power in many tasks, ranging

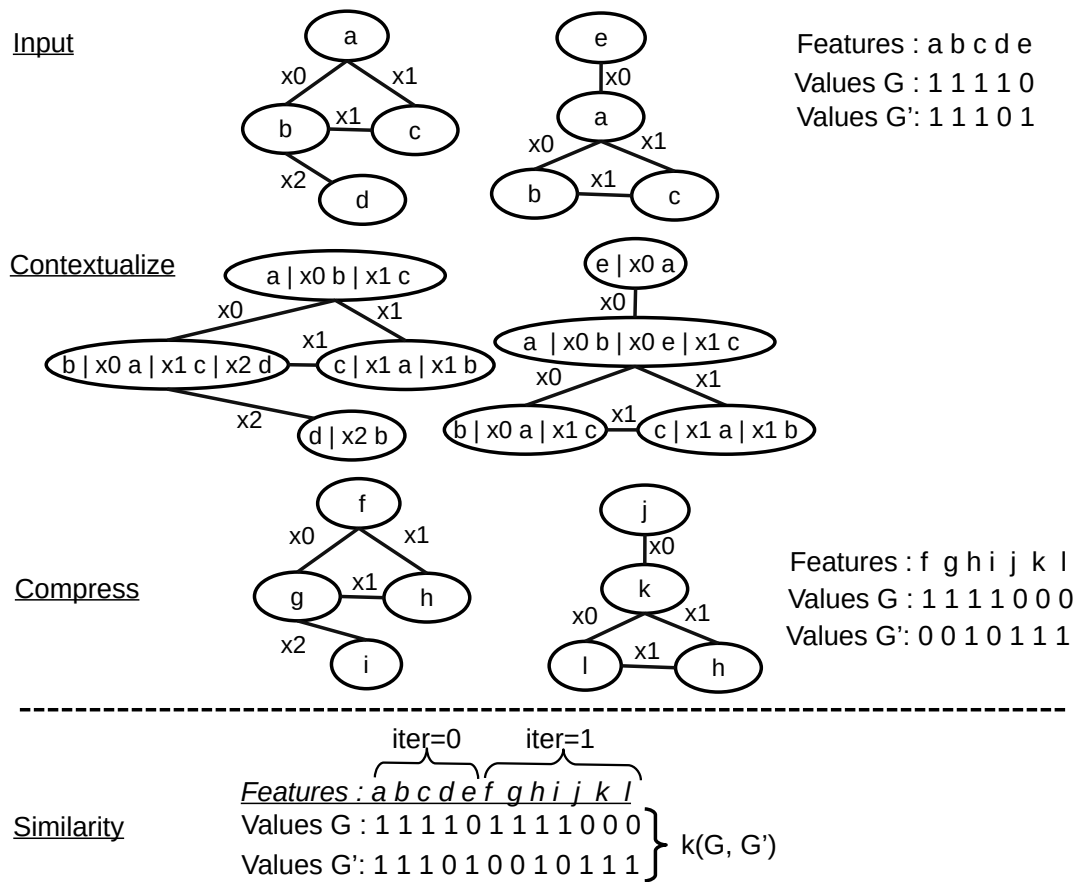


Figure 4.9: WLK example based on one iteration.

from protein classification to movie recommendation (Yanardag and Vishwanathan, 2015; Togninalli et al., 2019). However, so far, it has not been applied to (A)MR graphs. In the following, we will describe the WLK method.

Generally, a kernel can be viewed as a similarity measurement between two objects (Hofmann et al., 2008), in our case, two AMR graphs a, b . It is stated as $\mathbf{k}(a, b) = \langle \Phi(a), \Phi(b) \rangle$ where $\langle \cdot, \cdot \rangle : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}_+$ is an inner product and Φ maps an input to a feature vector that is built incrementally over K iterations. For our AMR graphs, one such iteration k works as follows: a) every node receives the labels of its neighbors and the labels of the edges connecting it to their neighbors, and stores them in a list (cf. Contextualize in Figure 4.9). b) The lists are alphabetically sorted and the string elements of the lists are concatenated to form new aggregate labels (cf. Compress in Figure 4.9). c) Two count vectors x_a^k and x_b^k are created where each dimension corresponds to a node label that is found in any of the two graphs and contains its count (cf. Features in Figure 4.9). Since every iteration yields two vectors (one for each input), we can concatenate the vectors over iterations and calculate the kernel (cf. Similarity in Figure 4.9):

$$\begin{aligned} \mathbf{k}(\cdot, \cdot) &= \langle \Phi_{WL}(a), \Phi_{WL}(b) \rangle \\ &= \langle \text{concat}(x_a^0, \dots, x_a^K), \text{concat}(x_b^0, \dots, x_b^K) \rangle \end{aligned} \quad (4.10)$$

Specifically, we use the cosine similarity kernel and two iterations ($K=2$), which implies that every node receives information from its neighbors and their immediate neighbors. For simplicity we will first treat edges as undirected, but later will experiment with various directionality parameterizations.

4.6.2 Wasserstein Weisfeiler-Leman (WWLK)

S^2_{MATCH} differs from all other AMR metrics in that it accepts close concept synonyms for alignment (up to a similarity threshold). But it comes with a restriction and a downside: i) it cannot assess graded similarity of (non-atomic) AMR subgraphs, which is crucial for assessing partial meaning agreement between AMRs (as illustrated in Figure 4.8), and ii) the alignment is costly to compute.

We hence propose to adopt a variant of WLK: the Wasserstein-Weisfeiler Leman kernel (WWLK) (Togninalli et al., 2019) for the following two reasons: i) WWLK can assess non-atomic subgraphs on a finer level, and ii) it provides graph alignments that are faster to compute than can be achieved by S_{MATCH} .

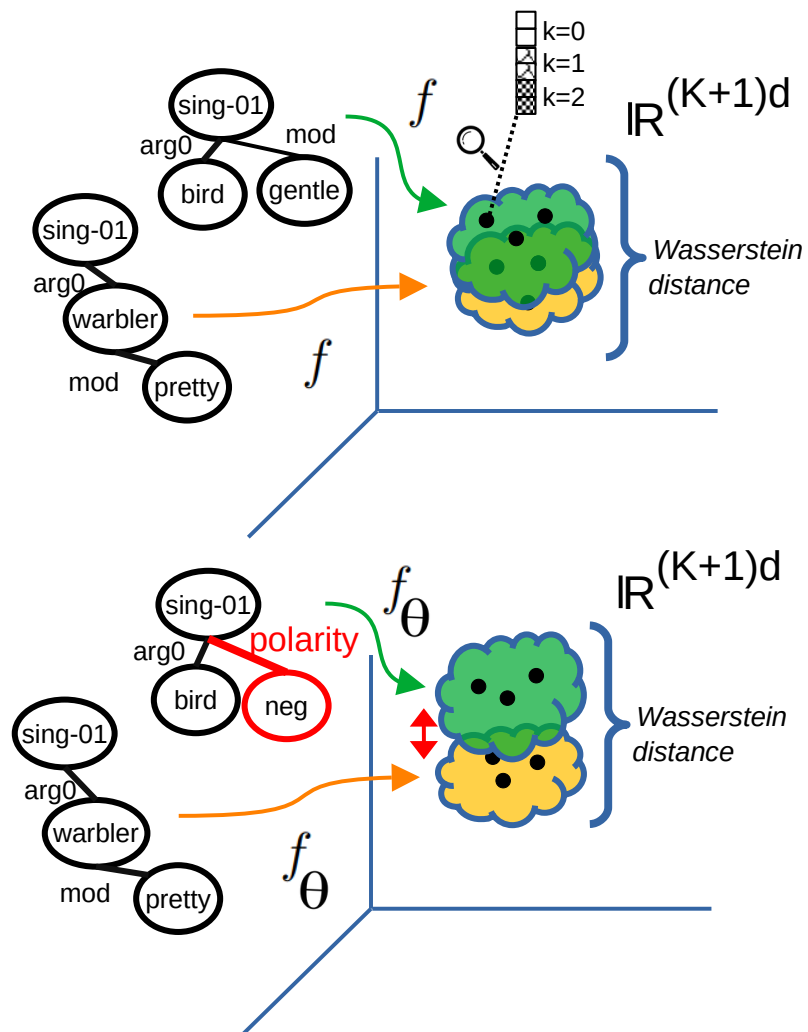


Figure 4.10: Wasserstein WLK example w/o learned edge parameters (top, c.f. Section 4.6.2) and w/ learnt edge parameters (bottom, c.f. Section 4.7). Learning these parameters allows us to adjust the embedded graphs such that they better take the (impact of) MR edges into account. Red: the distance increases because of a negation contrast between the two MRs that otherwise convey similar meaning.

WWLK works in **two steps**: 1. Given its initial node embeddings, we use WL to project the graph into a *latent space*, in which the final node embeddings describe *varying degrees of contextualization*. 2. Given a pair of such (WL) embedded graphs, a transportation plan is found that describes the minimum cost of translating one graph into the other. In the top graph of Figure 4.10, f indicates the first step, while *Wasserstein distance* indicates the second. Now, we describe the steps in closer detail.

Step 1: WL graph projection into latent space. Let $v = 1 \dots n$ be the nodes of AMR g . This graph is projected onto a matrix $\mathbb{R}^n \times \mathbb{R}^{(K+1)d}$ with

$$f(G) = hStack(X_g^0, \dots, X_g^K), \text{ where} \quad (4.11)$$

$$X_g^k = [x^k(1), \dots, x^k(n)]^T \in \mathbb{R}^n \times \mathbb{R}^d. \quad (4.12)$$

The function *hstack* concatenates matrices s.t. $(\begin{bmatrix} a & b \\ c & d \end{bmatrix}, \begin{bmatrix} x & y \\ w & z \end{bmatrix}) \rightarrow \begin{bmatrix} a & b & x & y \\ c & d & w & z \end{bmatrix}$. This means that, in the output space, every node is associated with a vector that is itself a concatenation of $K + 1$ vectors with d dimensions each, where k indicates the degree of contextualization (\mathcal{Q} in Figure 4.10). The embedding $x(v)^k \in \mathbb{R}^d$ for a node v in a certain iteration k is computed as follows:

$$x(v)^{k+1} = \frac{1}{2} \left(x(v)^k + \frac{1}{d(v)} \sum_{u \in \mathcal{N}_v} w(u, v) \cdot x(u)^k \right). \quad (4.13)$$

$d(v)$ is the degree of a node, \mathcal{N} returns the neighbors for a node, $w(u, v)$ can assign a weight to a node pair. The initial node embeddings, i.e., $x(\cdot)^0$, can be set up by looking up the node labels in a set of pre-trained word embeddings, or using random initialization. To distinguish between the discrete edge labels, we sample random weights.

Step 2: Computing the Wasserstein distance between two WL-embedded graphs.

The Wasserstein distance describes the minimum amount of work that is necessary to translate the (contextualized) nodes of one graph into the (contextualized) nodes of the other. It is computed based on pairwise euclidean distances from $f(a)$ with n nodes, and $f(b)$ with m nodes:

$$\text{distance} = \sum_{i=1}^n \sum_{j=1}^m \mathbf{T}_{i,j} D_{i,j} \quad (4.14)$$

Here, the ‘cost matrix’ $D \in \mathbb{R}^{n \times m}$ contains the euclidean distances between the n WL-embedded nodes from a and m WL-embedded nodes from b . I.e., $D_{i,j} = \|f(a)_i - f(b)_j\|_2$.

The *flow matrix* \mathbf{T} describes a transportation plan between the two graphs, i.e., $\mathbf{T}_{i,j} \geq 0$ states how much of node i from a flows to node j from b , the corresponding ‘local work’ can be stated as $flow(i, j) \cdot cost(i, j) := \mathbf{T}_{i,j} \cdot D_{i,j}$. To find the best \mathbf{T} , i.e., the transportation plan that minimizes the cumulative work needed (Eq. 4.14), we solve a constraint linear problem:¹²

$$\min \sum_{i=1}^n \sum_{j=1}^m \mathbf{T}_{i,j} D_{i,j} \quad (4.15)$$

$$s.t. : \mathbf{T}_{i,j} \geq 0, 1 \leq i \leq n, 1 \leq j \leq m \quad (4.16)$$

$$\sum_{j=1}^m \mathbf{T}_{i,j} = \frac{1}{m}, 1 \leq i \leq n \quad (4.17)$$

$$\sum_{i=1}^n \mathbf{T}_{i,j} = \frac{1}{n}, 1 \leq j \leq m \quad (4.18)$$

Note that i) the transportation plan \mathbf{T} describes an $n:m$ alignment between the nodes of the two graphs, and that ii) solving Eq. 4.15 has polynomial time complexity, while the (W)S⁽²⁾MATCH problem is NP-complete.

4.7 WWLK_θ with 0th-order optimization

Motivation: MR edge labels carry complex meaning. The embedding method of WWLK (Eq. 4.13) associates a weight $w(u, v) \in \mathbb{R}$ with each edge. For unlabeled graphs, $w(u, v)$ is simply set to one. To distinguish between the discrete AMR edge labels, in WWLK we have used random weights. However, AMR edge labels encode complex relations between nodes, and simply choosing random weights may not be enough. In fact, we hypothesize that different edge labels may impact the meaning similarity of AMR graphs in different ways. Whereas a modifier relation in an AMR graph configuration may or may not have a significant influence on the overall AMR graph similarity, an edge representing negation is bound to have a significant influence on the similarity of different AMR graphs. Consider the example in Figure 4.10: in the top figure, we embed AMRs for *The pretty warbler sings* and *The bird sings gently*, which have similar meanings. In the bottom figure, the second AMR has been changed to express the meaning of *The bird doesn't sing*, which clearly reduces the meaning similarity of the two AMRs. Hence, we hypothesize that learning edge parameters for different AMR relation types may help to

¹²We use <https://pypi.org/project/pyemd>

better adjust the graph embeddings, such that the Wasserstein distance may increase or decrease, depending on the specific meaning of AMR relation labels, and thus to better capture global meaning differences between AMRs (as outlined in Figure 4.10: f_θ).

Formally, to make the Wasserstein Weisfeiler-Leman kernel better account for *edge-labeled* AMR graphs, we learn a parameter set Θ that consists of parameters $\theta^{edgeLabel}$, where *edgeLabel* indicates the semantic relation, that is

$$edgeLabel \in L = \{\text{arg0}, \text{arg1}, \dots, \text{polarity}, \dots\}$$

Hence, in Eq. 4.13, we can set $w(u, v) = \theta^{label(u, v)}$ and apply multiplication $\theta^{label(u, v)} \cdot x(u)^k$. To facilitate the multiplication, we either may learn a matrix $\Theta \in \mathbb{R}^{|L| \times d}$ or a parameter vector $\Theta \in \mathbb{R}^{|L|}$. In our thesis, we constrain ourselves to the latter setting, i.e., our goal is to learn a parameter vector $\Theta \in \mathbb{R}^{|L|}$.

Learning edge labels with direct human feedback. To find suitable edge parameters Θ , we propose a zeroth order (gradient-free (Conn et al., 2009)) optimization setup, which has the advantage that we can *explicitly* teach our metric to better correlate with human ratings, optimizing the desired correlation objective without detours. In our case, we apply a simultaneous perturbation stochastic approximation (SPSA) procedure to estimate gradients (Spall, 1987, 1998; Wang, 2020).¹³

Let $sim(B, \Theta) = -WWLK_\Theta(B)$ be the similarity scores obtained from a (mini-)batch of graph pairs ($B = [(G_j, G'_j), \dots]$) as provided by (parametrized) $WWLK$. Now, let Y be the human reference scores. Then we design the loss function as $J(Y, \Theta) := 1 - correlation(sim(B, \Theta), Y)$. Further, let μ be coefficients that are sampled from a Bernoulli distribution. Then the gradient is estimated as follows:

$$\widehat{\nabla}_\Theta = \frac{J(Y, \Theta + c\mu) - J(Y, \Theta - c\mu)}{2c\mu}. \quad (4.19)$$

Finally, we can apply the common SGD learning rule: $\Theta^{t+1} = \Theta^t - \gamma \widehat{\nabla}_\Theta$. The learning rate γ and c decrease proportionally to t .

¹³It improves upon a classic Kiefer-Wolfowitz approximation (Kiefer, Wolfowitz, et al., 1952) by requiring, per gradient estimate, only 2 objective function evaluations instead of $2n$.

PRINCIPLE		SMATCH	SEMBLEU	S ² MATCH	WLK	WWLK	WWLK _θ
	Principle						
	I. Con., n-neg. bound	✓	✓	✓	✓	✓	✓
	II. id. of indisc.	✓	✗	✓	✗✓	✗✓	✗✓
	III. symmetry	✓	✗	✓	✓	✓	✓
	IV. determinacy	✓	✓	✓	✓	✗	✗
	V. efficiency	✗	★★★★	✗	★★★★	★★	★★
	VI. low bias	✓	✗	✓	✓	✓	✓
	VII. symb. matching	✓	✗	✓	✓	✓	✓
	VIII. graded matching	✗	✗	✓ ^{LEX}	✗	✓	✓

Table 4.7: Principle analysis update. On *efficiency*, in contrast to Table 4.6, we adopt a graded view, to indicate that WWLK is more efficient than SMATCH, but less efficient than SEMBLEU, which is in turn equally fast as WLK.

4.8 Taking a step back: principle analysis of WLK and WWLK

With WLK and WWLK we aimed at the combination of their strengths (e.g., broader match and efficiency as in SEMBLEU, alignment as in SMATCH, lexical matching as in S²MATCH), while mitigating their joint weaknesses. On one hand, we have learnt that theoretical principles are not everything we’d want in an MR metric, mainly because they are too focused on structural/symbolic matching (Section 4.5). On the other hand, nevertheless, it might be interesting to try updating the respective analysis result table (Table 4.6) with our novel metrics.

In particular, here we will focus on *id. of indiscernibles*, *determinacy*, *bias*, and *graded matching*.

Identity of indiscernible. We put ✗✓ for WWLK in the category *identity of indiscernibles* (two graphs have maximum similarity score, if, and only if they are equivalent). This is because, on one hand, we know from the Weisfeiler-Leman test that we can only view graph isomorphism from one angle: if a similarity score doesn’t equal one, then we know two graphs are not isomorphic. So there could be cases where two MRs are not equivalent but WWLK returns a score equal to one. Thus, when viewed harshly, it does not comply with the principle. However, while finding a counterexample for SEMBLEU was rather easy, we did not find such a case for (directional) WWLK.

Determinacy seems reduced in WWLK variants, since we initialize unknown nodes with random vectors. However, there are ways to mitigate this, without losing discriminatory power by naïvely using zero embeddings for unknowns. E.g.: i) we can *solve* the problem

by using a node embedding model that returns deterministic embeddings for any given node or edge label (e.g., using a character or sub-word based embedding model such as BERT (Devlin et al., 2019)); ii) and we can *mitigate* the problem by computing an expected distance over different initializations. The second method we will use to achieve stable results in the application case of parsing evaluation that will follow later in the next chapter (see Section 5.4).

Unintentional bias. Most metrics exhibit low bias. However, we are confronted with a new type of bias in $WWLK_{\theta}$: bias to human similarity. This bias is motivated and desirable (i.e., intentional), which is why we add a ✓.

Graded matching. To propose a first metric that addresses this principle partially, we proposed S^2_{MATCH} , that calculates $SMATCH$ by considering *lexical* graded similarity of concept nodes. Moving beyond the lexical level, we now have $WWLK$ that is the only MR metric that can calculate graded MR subgraph similarity from broader subgraph structures, and provides explanatory many-to-many alignment.

4.9 Discussion

Based on our insights from our studies on previous MR metrics, we proposed novel MR metrics that target i) the unification of strengths of previous metrics and ii) the incorporation of new features that aim at modeling more graded similarity between MR graphs, to generalize MR metric distances to new use-cases that go beyond mono-lingual parsing evaluation, which is a strictly limited scenario, because in that setup graphs are based on the same sentence. However, yet we don't know much about the empirical behavior of MR metrics. In particular we would like to know how well they might align with the human view on similarity. Therefore, next, we will develop and explore a meaningful evaluation measure that might help us answer this question.

Chapter 5

Extended empirical studies on MR metrics

5.1 Chapter outline

As of now, we know too little about the empirical behavior of MR metrics in different tasks, and how their performance could be measured. We are going to address these questions in this chapter. In particular:

1. We will construct the first benchmark to evaluate metrics of meaning representations using transparent objectives such as human sentence similarity and robustness stress tests (Section 5.2).
2. Through this benchmark, we evaluate MR metrics, outlining their strengths on different tasks and potential for further improvement (Section 5.3).
3. In Section 5.4 we will investigate MR metrics in an interesting practical evaluation setting: monolingual evaluation of high-performance MR parsers. We find that application of MR metrics provides us with new insights not only about the metrics, but also about high-performance MR parsing systems in general, which we summarize in Section 5.5.
4. We conclude this chapter with a discussion in Section 5.6.

Underlying work. The content of this chapter is mainly based on publications by Opitz et al. (2021a) and Opitz and Frank (2022a).

5.2 BAMBOO_W: A first benchmark for MR metrics

We will now construct BAMBOO_W, the first empirical benchmark for meaning representation metrics. BAMBOO_W is an acronym for a Benchmark for AMR Metrics based on Overt Objectives: it is aimed at maximizing the interpretability of results by defining multiple **overt objectives** that range from *sentence similarity objectives* to *stress tests* that probe a metric’s robustness against meaning-altering and meaning-preserving graph transformations. Later we will show the benefits of BAMBOO_W by using it to investigate MR metrics empirically, complementing our theoretical assessments from the chapter before.

Grounding MR similarity metrics in human ratings of semantic sentence similarity.

A natural and intuitive wish is that we would like to have MR metrics that reflect *human similarity* between the texts that the MRs represent. Therefore, as the main criterion for assessing MR similarity metrics, we use human judgments of the meaning similarity of sentences underlying pairs of MRs. Our primary assumption is: a metric of pairs of MR graphs a and b that represent sentences s and s' should reflect human judgments of semantic sentence similarity and relatedness:

$$mrMetric(a,b) \propto humanScore(s,s') \quad (5.1)$$

where \propto means *proportional to*¹.

5.2.1 Human similarity objectives

Similarity objectives. We select three notions of sentence similarity as evaluation targets for MR metrics. The three notions have been elicited by three human-rated evaluation datasets: i) the semantic textual similarity (STS) objective from Baudiš et al. (2016a,b); ii) the sentence relatedness objective (SICK) from Marelli et al. (2014); iii) the paraphrase detection objective (PARA) by Dolan and Brockett (2005).²

Each of these three evaluation data sets can be seen as a set of pairs of sentences (s_i, s'_i) with an associated score $humanScore(\cdot)$ that provides the human sentence relation assessment score reflecting *semantic similarity* (STS), *semantic relatedness* (SICK) and *whether sentences are paraprastic* (PARA). Hence, each of these data sets can be described as

¹In Opitz et al. (2021a) we used a *approximately* sign \approx instead of \propto , which would technically describe the most ideal situation. However, in the end our main goal would be to have strong correlation, which does not necessarily require a low *absolute* deviation from the human score, so \propto perhaps seems more apt.

²For more information about the construction process of the data sets, see our Related Work 3.4.

		data instances		(s. length)		graph statistics			
						# nodes		density	
source	train/dev/test	avg.	50 th	avg.	50 th	avg.	50 th		
STS	5749/1500/1379	9.9	8	14.1	12	0.10	0.08		
SICK	4500/500/4927	9.6	9	10.7	10	0.11	0.1		
PARA	3576/500/1275	18.9	19	30.6	30	0.04	0.04		

Table 5.1: BAMBOO_🌱 data set statistics of the **Main** partition. Sentence length (s. length, displayed for reference only) and graph statistics (average and median) are calculated on the training sets.

$\{(s_i, s'_i, humanScore(s_i, s'_i) = y_i)\}_{i=1}^n$. Both STS and SICK offer scores on Likert scales, ranging from *equivalence* (max) to *unrelated* (min), while PARA scores are binary, judging sentence pairs as being paraphrases (1), or not (0). We min-max normalize the Likert scale scores to the range $[0, 1]$ to facilitate standardized evaluation.

For BAMBOO_🌱, we replace each pair (s_i, s'_i) with their AMR parses: $(p_i = parse(s_i), p'_i = parse(s'_i))$, transforming the data into $\{(p_i, p'_i, y_i)\}_{i=1}^n$. This provides the main partition of the benchmarking data for BAMBOO_🌱, henceforth denoted as **Main**³. Statistics of **Main** are shown in Table 5.1). The sentences in PARA are longer compared to STS and SICK. The corresponding AMR graphs are, on average, much larger in number of nodes, but less complex with respect to the average density.⁴

AMR construction. We choose a strong parser that achieves high scores in the range of human-human inter-annotator agreement estimates in AMR banking: The parser yields 0.80-0.83 Smatch F1 on AMR2 and AMR3. The parser, henceforth denoted as T5S2S, is based on an AMR fine-tuned T5 language model (Raffel et al., 2020) and produces AMRs in a sequence-to-sequence fashion.⁵ It is on par with the current state-of-the-art that similarly relies on seq-to-seq (Xu et al., 2020), but the T5 backbone alleviates the need for massive MT pre-training.

Manual data quality assessment: three-way graph quality ratings. To obtain a better picture of the graph quality in BAMBOO_🌱 we perform manual quality inspections. From each data set (SICK, STS, PARA) we randomly select 100 sentences and create

³The other partitions, which are largely based on this data, will be introduced in Section 5.2.2.

⁴The lower average density could be caused, e.g., by the fact that the PARA data is sampled from news sources, which means that the AMRs contain more named entity structures that usually have more terminal nodes.

⁵<https://github.com/bjascob/amrlib>

	Parser	%gold \uparrow	%silver	%flawed \downarrow
STS	GPLA	43[33,53]	37[28,46]	20[12,27]
	T5S2S	54[44,64] $\dagger\ddagger$	41[31,50]	5[0,9] $\dagger\ddagger$
SICK	GPLA	38[28,47]	49[39,59]	13[6,19]
	T5S2S	48[38,58] \dagger	47[37,57]	5[0,9] $\dagger\ddagger$
PARA	GPLA	9[3,14]	52[43,62]	39[29,48]
	T5S2S	21[13,29] $\dagger\ddagger$	63[54, 73] $\dagger\ddagger$	16[8,23] $\dagger\ddagger$
ALL	GPLA	30[25,35]	46[40,52]	24[19,29]
	T5S2S	41[35,46] $\dagger\ddagger$	50[45,56]	9[5,12] $\dagger\ddagger$

Table 5.2: Three-way graph assessment. [x,y]: 95-confidence intervals estimated with bootstrap. \dagger (\ddagger) significant improvement of T5S2S over GPLA with $p < 0.05$ ($p < 0.005$).

their parses with T5S2S. Additionally, to establish a baseline, we also parse the same sentences with the GPLA parser of Lyu and Titov (2018), a neural graph prediction system that uses latent alignments (with 74.4 Smatch score on AMR2). This results in 300 GPLA parses and 300 T5S2S parses. A human annotator⁶ inspects the (shuffled) sample and assigns three-way labels: *flawed* – an AMR contains critical errors that distort the meaning significantly; *silver* – an AMR contains small errors that can potentially be neglected; *gold* – an AMR is acceptable.

Results in Table 5.2 show that the quality of T5S2S parses is substantially better than the baseline in all three data sets. The percentage of excellent parses increases considerably (STS: +11pp, SICK: +10pp, PARA: +11pp) while the percentage of flawed parses drops notably (STS: -15pp, SICK: -8pp, PARA: -23pp). The increases in gold parses and decreases in flawed parses are significant in all data sets ($p < 0.05$, 10,000 bootstrap samples of the sample means).⁷

5.2.2 Robustness challenges

Besides benchmarking MR metric scores against human ratings, we are also interested in assessing a metric’s robustness under **meaning-preserving** and **-altering** graph transformations. Assume we are given any pair of AMRs from paraphrases. A small change

⁶The human annotator is a proficient English speaker and has worked several years with AMR.

⁷ $\mathcal{H}_0(\text{gold})$: amount of gold graphs T5S2S \leq amount of gold graphs GPLA; $\mathcal{H}_0(\text{silver})$: amount of silver graphs T5S2S \leq amount of gold graphs GPLA; $\mathcal{H}_0(\text{flawed})$: amount of gold graphs T5S2S \geq amount of gold graphs GPLA.

in structure or node content can lead to two outcomes: the graphs still represent paraphrases, or they do not. We consider a metric to be robust if its ratings correctly reflect such changes.

Specifically, we apply three graph transformation strategies. i) Reification (**Reify**), which changes the graph’s surface structure, but not its meaning; ii) Concept synonym replacement (**Syno**), which also preserves meaning and may or may not change the graph surface structure; iii) Role confusion (**Arg**), which applies small changes to the graph structure that do not preserve its meaning.

Meaning-preserving translations

Generally, given a meaning-preserving function f of a graph, i.e.,

$$g \equiv f(g), \quad (5.2)$$

it is natural to expect that a semantic similarity function over the pair of transformed AMRs nevertheless stays stable, and thus satisfies:

$$metric(a, b) \approx metric(f(a), f(b)). \quad (5.3)$$

Reification translation (Reify), which we already visited in our Background Figure on graph translations (Figure 2.1), is an established way to rephrase AMRs. Formally, a reification is induced by a rule

$$edge(x, y) \xrightarrow{\text{reify}} instance(z, h(edge)_0) \quad (5.4)$$

$$\wedge h(edge)_1(z, x) \quad (5.5)$$

$$\wedge h(edge)_2(z, y), \quad (5.6)$$

where h returns, for a given edge, a new concept and corresponding edges from a dictionary, where the edges are either $:arg_i$ or $:op_i$. An example is displayed in Figure 5.1 (top, left, see also overview Figure 2.1 in the Background of this thesis.). Besides reification for *location*, other known types are *polarity-*, *modifier-*, or *time-reification*.⁸ Processing statistics of the applied reification operations are shown in Table 5.3.

⁸A complete list of reifications are given in the official AMR guidelines: <https://github.com/amrisi/amr-guidelines/blob/master/amr.md>

	STS		SICK		PARA	
	mean	th	mean	th	mean	th
Reify -OPS	2.74	[1, 2, 4]	1.17	[0, 1, 2]	5.14	[3, 5, 7]
Syno -OPS	0.80	[0, 1, 2]	1.31	[0, 1, 2]	1.30	[0, 1, 2]
Arg -OPS	1.33	[1, 1, 2]	1.11	[1, 1, 1]	1.80	[1, 2, 2]

Table 5.3: Statistics about the amount of transform operations that were conducted, on average, on one graph. [x,y,z]: 25th, 50th (median) and 75th percentile of the amount of operations.

Synonym concept node transform (Syno). Here, we iterate over MR node labels. For any label that shows a predicate from PropBank, we consult a manually created database of (near-)synonyms that are also contained in PropBank, and sample one for replacement. E.g., some sense of *fall* is near-equivalent to a sense of *decrease* (*car prices fell/decreased*). For concepts that are not predicates we run an ensemble of four WSD solvers⁹ (based on the concept and the sentence underlying the AMR) to identify its WordNet synset. From this synset we sample an alternative lemma.¹⁰ If an alternative lemma consists of multiple tokens where modifiers precede the noun, we replace the node with a graph-substructure. So, if the concept is *man* and we sample *adult_male*, we expand '*instance(x,man)*' with '*mod(x,y) ∧ instance(y,adult) ∧ instance(x,male)*'. Data processing statistics are shown in Table 5.3.

Meaning-altering graph transforms

Role confusion (Arg). A naïve MR metric could be one that treats an MR as a bag-of-nodes, omitting structural information, such as edges and edge-labels. Such metrics could exhibit misleadingly high correlation scores with human ratings, solely due to a high overlap in concept content.

Hence, we design adversarial instances that can probe an MR metric when confronted with cases of opposing factuality (e.g., polarity, modality or relation inverses), while concept overlap is largely preserved. We design a function

$$g \neq h(g), \tag{5.7}$$

⁹'Adapted lesk', 'Simple Lesk', 'Cosine Lesk', 'max sim' (Banerjee and Pedersen, 2002; Lesk, 1986; Pedersen, 2007): <https://github.com/alvations/pywsd>.

¹⁰To increase precision, we only perform this step if all solvers agree on the predicted synset.

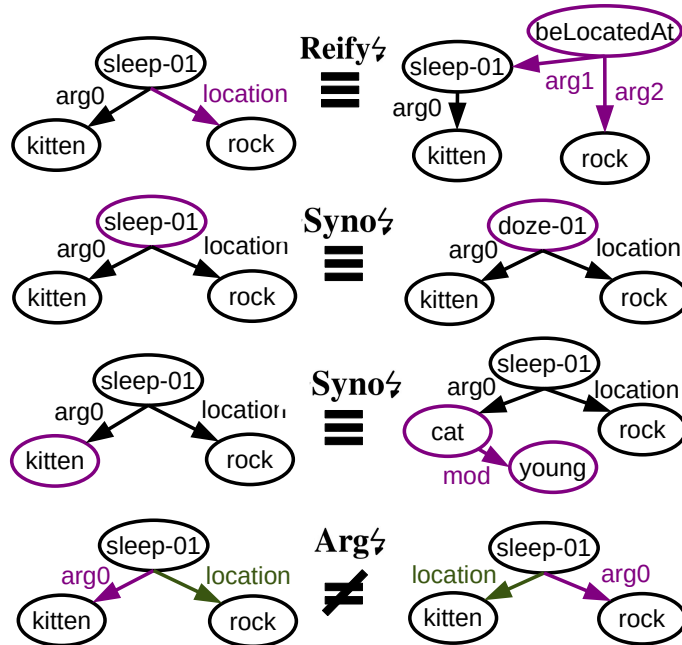


Figure 5.1: Examples for f and h graph transforms.

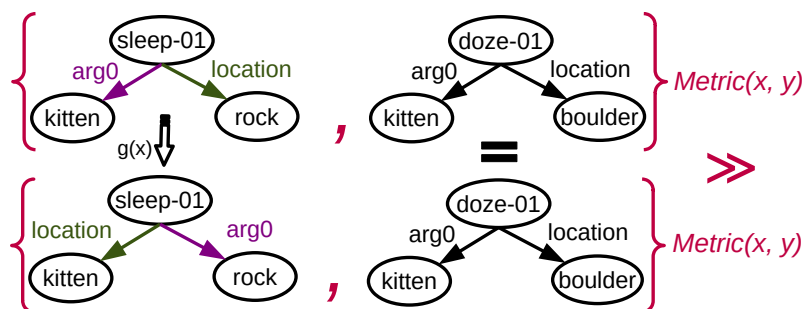


Figure 5.2: Metric objective example for Arg .

that confuses role labels (see **Arg**₇ in Figure 5.1). We make use of this function to turn two paraphrastic AMRs (a, b) into non-paraphrastic AMRs, by applying h to either a , or b , but not both.

In some cases h may create a meaning that still makes sense (*The tiger bites the snake.* \rightarrow *The snake bites the tiger.*), while in others, h may induce a nonsensical meaning (*The tiger jumps on the rock.* \rightarrow *The rock jumps on the tiger.*). However, this is not our primary concern, since in all cases, applying h achieves our main goal: it returns a **different meaning** that turns a paraphrase-relation between two AMRs into a non-paraphrastic one that is structurally not much different.

To implement **Arg**₇, for each data set (**PARA**, **STS**, **SICK**) we create one new data subset. First, i) we collect all paraphrases from the initial data (in **SICK** and **STS** these are pairs with maximum human score).¹¹ ii) We iterate over the AMR pairs (a, b) and randomly select the first or second AMR from the tuple. We then collect all n nodes with more than one outgoing edge. If $n = 0$, we skip this AMR pair (the pair will not be contained in the data). If $n > 0$, we apply the meaning altering function h and randomly flip edge labels. Finally, we add the original (a, b) to our data with the label *paraphrase*, and the altered pair ($a, g(b)$) with the label *non-paraphrase* (cf. Figure 5.2). Per graph, we allow a maximum of 3 role confusion operations (see Table 5.3 for processing statistics).

Discussion

Safety of robustness objectives. We have proposed three challenging robustness objectives, based on meaning-preserving and meaning altering graph transformations. **Reify**₇ changes the graph structure, but preserves the meaning. **Arg**₇ keeps the graph structure (modulo edge labels) while changing the meaning. **Syno**₇ changes node labels and possibly the graph structure and aims at preserving the meaning.

Reify₇ and **Arg**₇ are fully safe: they are well defined and are guaranteed to fulfill our goal (Eq. 5.2 and 5.7): meaning-preserving or -altering graph transforms. **Syno**₇ is more experimental and has (at least) three failure modes. In the first mode, depending on context, human similarity judgments could change when near-synonyms are chosen (sleep \rightarrow doze, a young cat \rightarrow kitten, etc.). The second mode occurs when WSD commits an error (e.g., minister (political sense) \rightarrow priest). A third mode are societal biases found in WordNet (e.g., the node *girl* may be mapped onto its ‘synonym’ *missy*). The third mode

¹¹This shrinks the train/dev/test size of STS (now: 474/106/158) and SICK (now: 246/50/238).

may not really be a failure, since it may not change the human rating, but, nevertheless, it may be undesirable.

In conclusion, **Reify** and **Arg** confusion constitute safe robustness challenges, while results on **Syno** may have to be taken with a grain of salt.

Status of the challenges in BAMBOO and outlook. We believe that a key benefit of the robustness challenges lies in their potential to provide complementary performance indicators, in addition to evaluation on the **Main** partition of BAMBOO (cf. Section 5.2). In particular, the challenges may serve to assess metrics more deeply, uncover potential weak spots, and help select among metrics, e.g., when performance differences on **Main** are small. In this work, however, the complementary nature of **Reify**, **Syno** or **Arg** versus **Main** is only reflected in the name of the partitions, and in our experiments, we consider all partitions equally. Future work may deviate from this setup.

Our proposed robustness challenges are also by no means exhaustive, and we believe that there is ample room for developing more challenges (*extending* BAMBOO) or experimenting with different setups of our challenges (*varying* BAMBOO¹²). For these reasons, it is possible that future work may implement alternative or enhanced setups, extensions and variations of BAMBOO.

5.3 Experimental insights from BAMBOO

Questions posed to BAMBOO. BAMBOO allows us to address several open questions: The first set of questions aims to gain more knowledge about previously released metrics. For example, we would like to know: *What semantic aspects of AMR does a metric measure? If a metric has hyper-parameters (e.g., SEMBLEU), which hyper-parameters are suitable (for a specific objective)? Does the costly alignment of SMATCH pay off, by yielding better predictions, or do the faster alignment-free metrics offer a ‘free-lunch’?* A second set of questions aims to evaluate our proposed novel AMR similarity metrics, and to assess their potential advantages.

Experimental setup. We evaluate all metrics on the test set of BAMBOO. The two hyper-parameters of S²MATCH, that determine when concepts are similar, are set with a small search on the development set (by contrast, S²MATCH_{default} denotes the default

¹²E.g., we may rectify only selected relations, or create more data, setting Eq. 5.3 to $metric(a,b) \approx metric(a,f(b))$, only applying f to one graph.

	speed	align	Main			Reify ₂			Syno ₂			Arg ₂			amean	hmean
			STS	SICK	PARA	STS	SICK	PARA	STS	SICK	PARA	STS	SICK	PARA		
SMATCH	-	✓	58.45	59.72	41.25	57.98	61.81	39.66	56.14	57.39	39.58	48.05	70.53	24.75	51.28	47.50
WSMATCH	-	✓	53.06	59.24	38.64	53.39	61.17	37.49	51.41	57.56	37.85	42.47	66.79	22.68	48.48	44.58
S ² MATCH _{de fault}	-	✓	56.38	58.15	42.16	55.65	60.04	40.41	56.05	57.17	40.92	46.51	70.90	26.58	50.91	47.80
S ² MATCH	-	✓	58.82	60.42	42.55	58.08	62.25	40.60	56.70	57.92	41.22	48.79	71.41	27.83	52.22	49.07
SEMA	++	✗	55.90	53.32	33.43	55.51	56.16	32.33	50.16	48.87	29.11	49.73	68.18	22.79	46.29	41.85
SEMBLEU _{k=1}	++	✗	66.03	62.88	39.72	61.76	62.10	38.17	61.83	58.83	37.10	1.99	1.47	1.40	41.11	5.78
SEMBLEU _{k=2}	++	✗	60.62	59.86	36.88	57.68	59.64	36.24	57.34	56.18	33.26	44.54	67.54	16.60	48.87	42.13
SEMBLEU _{k=3}	++	✗	56.49	57.76	32.47	54.84	57.70	33.25	52.82	53.47	28.44	49.06	69.49	24.27	47.50	42.82
SEMBLEU _{k=4}	++	✗	53.19	56.69	29.61	52.28	56.12	30.11	49.31	52.11	25.56	49.75	69.58	29.44	46.15	41.75
WLK (ours)	++	✗	64.86	61.52	37.35	62.69	62.55	36.49	59.41	56.60	33.71	45.89	64.70	19.47	50.44	44.35
WLK (ours)	+	✓	63.15	65.58	37.55	59.78	65.53	35.81	59.40	59.98	32.86	13.98	42.79	7.16	45.30	28.83
WLK [⊙] (ours)	+	✓	66.94	67.64	37.91	64.34	65.49	39.23	60.11	62.29	35.15	55.03	75.06	29.64	54.90	50.26

Table 5.4: BAMBOO benchmark result of AMR metrics. All numbers are Pearson’s $\rho \times 100$. ++: linear time complexity; +: polynomial time complexity; -: NP complete.

setup). $WWLK_{\theta}$ is trained with batch size 16 on the training data. S^2MATCH , $WWLK$ and $WWLK_{\theta}$ all make use of GloVe embeddings (Pennington et al., 2014a).

Our main evaluation metric is Pearson’s ρ between a metric’s output and the human ratings. Additionally, we consider two global performance measures to better rank AMR metrics: the arithmetic mean (*amean*) and the harmonic mean (*hmean*) over a metric’s results achieved in all tasks. *Hmean* is always \leq *amean* and is driven by low outliers. Hence, a large difference between *amean* and *hmean* serves as a warning light for a metric that is extremely vulnerable in a specific task.

5.3.1 BAMBOO studies previous metrics

Table 5.4 shows AMR metric results on BAMBOO across all three human similarity rating types (STS, SICK, PARA) and our four challenges: **Main** represents the standard setup (cf. Section 5.2.1), whereas **Reify**, **Syno** and **Arg** test the metric robustness (cf. Section 5.2.2). So far not introduced, we also include the SEMA metric (Anchiêta et al., 2019) in these experiments. SEMA is similar to Smatch (matching triplets) but does not compute an alignment.

SMATCH and S^2MATCH rank 1st and 2nd of previous metrics. SMATCH, our baseline metric, provides strong results across all tasks (Table 5.4, *amean*: 51.28). With default parameters, $S^2MATCH_{default}$ performs slightly worse on the main data for STS and SICK, but improves upon SMATCH on PARA, achieving a slight overall improvement with respect to *hmean* (+0.30), but not *amean* (-0.37). S^2MATCH is more robust against **Syno** (e.g., +4.6 on **Syno** STS vs. SMATCH), and when confronted with reified graphs (**Reify** STS +3.3 vs. SMATCH).

S^2MATCH , after setting its two hyper-parameters with a small search on the development data¹³, consistently improves upon SMATCH over all tasks (*amean*: +0.94, *hmean*: +1.57).

WSMATCH: Are nodes near the root more important? The hypothesis underlying WSMATCH is that concepts that are located near the top of an AMR have a higher impact on AMR similarity ratings. Interestingly, WSMATCH mostly falls short of SMATCH, offering substantially lower performance on all main tasks and all robustness checks, resulting in reduced overall *amean* and *hmean* scores (e.g., main STS: -5.39 vs. SMATCH, *amean*:

¹³STS/SICK: $\tau=0.90$, $\tau'=0.95$; PARA: $\tau=0.0$, $\tau'=0.95$

-2.8 vs. SMATCH, hmean: -2.9 vs. SMATCH). This contradicts the ‘core-semantics’ hypothesis and provides novel evidence that semantic concepts that influence human similarity ratings are not necessarily located close to AMR roots.¹⁴

BFS-based metrics I: SEMA increases speed but pays a price. Next, we find that SEMA achieves lower scores in almost all categories, when compared with SMATCH (amean: -4.99, hmean -5.65), ending up at rank 7 (according to hmean and amean) among prior metrics. It is similar to SMATCH in that it extracts triples from graphs, but differs by not providing an alignment. Therefore, it can only loosely model some phenomena, and we conclude that the increase in speed comes at the cost of a substantial drop in modeling capacity.

BFS-based metrics II: SEMBLEU is fast, but is sensitive to k . Results for SEMBLEU show that it is very sensible to parameterizations of k . Notably, $k=1$, which means that the method only extracts bags of nodes, achieves strong results on SICK and STS. On PARA, however, SEMBLEU is outperformed by S^2 MATCH, for all settings of k (best k ($k=2$): -2.8 amean, -4.7 hmean). Moreover, all variants of SEMBLEU are vulnerable to robustness checks. E.g., $k=2$, and, naturally, $k=1$ are easily fooled by **Argz**, where performance drops massively. $k=4$, on the other hand, is most robust against **Argz**, but overall it falls behind $k=2$.

Since SEMBLEU is asymmetric, we also re-compute the metric in a ‘symmetric’ way by averaging the metric result over different argument orders. We find that this can slightly increase its performance ($[k, \text{amean}, \text{hmean}]$: [1, +0.8, +0.6]; [2, +0.5, +0.4]; [3, +0.2, +0.2]; [4, +0.1, +0.0]).

In sum, our conclusions concerning SEMBLEU are: i) $\text{SEMBLEU}_{k=1}$ (but not $\text{SEMBLEU}_{k=3}$) performs well when measuring similarity and relatedness. However, $\text{SEMBLEU}_{k=1}$ is naïve and easily fooled (**Argz**). ii) Hence, we recommend $k=2$ as a good tradeoff between robustness and performance, with overall rank 4 (amean) and 6 (hmean).¹⁵

¹⁴Manual inspection of examples shows that low similarity can frequently be explained with differences in concrete concepts that tend to be distant to the root. E.g., the low similarity (0.16) of *Morsi supporters clash with riot police in Cairo* vs. *Protesters clash with riot police in Kiev* arises mostly from *Kiev* and *Cairo* and *Morsi*, however, these names (as are names in general in AMR) are distant to the root region, which is similar in both graphs (*clash, riot, protesters, supporters*).

¹⁵Setting $k=2$ stands in contrast to the original paper that recommended $k=3$, the common setting in MT. However, lower k in SEMBLEU reduces biases (as we have found out before via empirical analysis on BAMBOO₃, c.f., Section 5.3), which may explain the better result on BAMBOO₃.

5.3.2 BAMBOO assesses MR metrics

We now discuss results of our proposed metrics based on the Weisfeiler-Leman Kernel that we constructed in the previous Chapter (4).

Standard Weisfeiler-Leman (WLK) is fast and a strong baseline for AMR similarity.

First, we visit the symbolic Weisfeiler-Leman kernel (WLK). Like SEMBLEU and SEMA, it is alignment-free, and therefore very fast. However, it outperforms SEMBLEU and SEMA in almost all tasks (score difference against second best alignment-free metric: ([a|h]mean: +1.6, +1.5) but falls behind alignment-based SMATCH ([a|h]mean: -0.8, -3.2). Specifically, WLK proves robust against **Reify** but appears more vulnerable against **Syno** (-5 points on STS and SICK) and **Arg** (notably PARA, with -10 points).¹⁶

The better performance, compared to SEMBLEU and SEMA, may be due to the fact that WLK (unlike SEMBLEU and SEMA) does not perform BFS traversal from the root, which may reduce biases.

WWLK and WWLK_θ obtain first ranks. Basic WWLK exhibits strong performance on SICK (ranking second on main and first on **Reify**). However, it has large vulnerabilities, as exposed by **Arg**, where only SEMBLEU_{k=1} ranks lower. This can be explained by the fact that WWLK (7.2 Pearson’s ρ on PARA **Arg**) only weakly considers the semantic relations (whereas SEMBLEU_{k=1} does not consider semantic relations at all).

WWLK_θ, our proposed algorithm for edge label learning, mitigates this vulnerability (29.6 Pearson’s ρ on PARA **Arg**, 1st rank). Learning edge labels also helps assessing similarity (STS) and relatedness (SICK), with substantial improvements over standard WWLK and SMATCH (STS: 66.94, +3.9 vs. WWLK and +10.6 vs. SMATCH; SICK +2.1 vs. WWLK and +8.4 vs. SMATCH).

In sum, **WWLK_θ occupies rank 1 of all considered metrics** (amean and hmean), outperforming all non-alignment based metrics by large margins (amean +4.5 vs. WLK and +6.0 vs. SEMBLEU_{k=2}; hmean +5.9 vs. WLK and +8.1 vs. SEMBLEU_{k=2}), but also the alignment-based ones, albeit by lower margins (amean +2.7 vs. S²MATCH; hmean +1.2 vs. S²MATCH).

	K (#WL iters)							
	basic (K=2)		K=1		K=3		K=4	
	amean	hmean	amean	hmean	amean	hmean	amean	hmean
WLK	50.4	44.4	49.8	44.2	47.6	42.4	46.4	41.5
WWLK	45.3	28.8	43.4	15.3	45.7	31.4	42.3	24.0
WWLK _θ	54.9	50.3	52.2	35.4	55.2	51.1	50.8	47.3

Table 5.5: WLK variants with different K.

	undirected		TOP-DOWN		BOTTOM-UP		2WAYS	
	amean	hmean	amean	hmean	amean	hmean	amean	hmean
WLK	50.4	44.4	50.3	44.3	50.2	43.8	49.5	41.8
WWLK	45.3	28.8	43.7	22.0	41.6	9.9	44.8	24.1
WWLK _θ	54.9	50.3	53.8	46.1	50.2	18.7	55.3	51.0

Table 5.6: (W)WLK: message passing directions.

5.3.3 Analyzing hyper-parameters of our novel metrics WLK and WWLK

Setting K in (W)WLK. How does setting the number of iterations in Weisfeiler-Leman affect predictions? Table 5.5 shows K=2 is a good choice for all WLK variants. K=3 slightly increases performance in the latent variants (WWLK: +0.4 amean; WWLK_θ: +0.3 amean), but lowers performance for the fast symbolic matching WLK (-2.8 amean). This drop is somewhat expected: K>2 introduces much sparsity in the symbolic WLK feature space.

WL message passing direction. Even though AMR defines directional edges, for optimal similarity ratings, it was not a-priori clear in which directions the node contextualization should be restricted when attempting to model human similarity. Therefore, so far, our WLK variants have treated AMR graphs as undirected graphs (\leftrightarrow). In this experiment, we study three alternate scenarios: ‘TOP-DOWN’ (forward, \rightarrow), where information is only passed in the direction that AMR edges point at and ‘BOTTOM-UP’ (backwards, \leftarrow), where information is exclusively passed in the opposite direction, and 2WAY (\Leftrightarrow), where information is passed forwards, but for every edge $edge(x,y)$ we insert an $edge^{-1}(y,x)$. 2WAY facilitates more node interactions than either TOP-DOWN or BOTTOM-UP, while preserving directional information.

¹⁶Similar to SEMBLEU, we can mitigate this performance drop on **Argz** PARA by increasing the amount of passes K in WLK, however, this decreases overall amean and hmean.

	STS		SICK		PARA		AVERAGE	
	MHA	IMA	MHA	IMA	MHA	IMA	MHA	IMA
SM	[71, 73]	97.9	[66, 66]	99.9	[44, 44]	97.9	[60, 61]	98.6
WSM	[64, 65]	99.2	[67, 67]	99.8	[47, 49]	98.7	[59, 60]	99.2
S2M _{def}	[69, 70]	97.7	[62, 63]	99.3	[44, 47]	97.7	[58, 60]	98.2
S2M	[71, 73]	97.8	[69, 70]	98.6	[41, 46]	98.0	[60, 63]	98.1
SE	[66, 66]	97.7	[55, 55]	100	[42, 46]	99.0	[55, 56]	98.9
SB ₂	[68, 68]	97.2	[62, 62]	99.8	[41, 42]	98.8	[57, 58]	98.6
SB ₃	[66, 66]	98.4	[63, 63]	99.7	[33, 34]	99.3	[54, 54]	99.1
WLK	[72, 72]	98.2	[65, 65]	99.8	[43, 46]	97.9	[60, 61]	98.6
WWLK	[77, 78]	97.8	[65, 67]	98.1	[42, 46]	97.8	[61, 63]	97.9
WWLK _θ	[78, 78]	96.8	[67, 68]	98.1	[48, 48]	96.7	[64, 65]	97.2

Table 5.7: Retrospective sub-sample quality analysis of BAMBOO graph quality and sensitivity of metrics. All values are Pearson’s $\rho \times 100$. Metric Human Agreement (MHA): $[x, y]$, where x is the correlation (to human ratings) when the metric is executed on the uncorrected sample and y is the same assessment on the manually post-processed sample.

Our findings in Table 5.6 show a clear trend: treating AMR graphs as graphs with undirected edges offers better results than TOP-DOWN (e.g., WWLK-1.6 amean; -6.6 hmean) and considerably better results when compared to WLK in BOTTOM-UP mode (e.g., WWLK-3.7 amean; -18.9 hmean). Overall, 2WAY behaves similarly to the standard setup, with a slight improvement for WWLK_θ. Notably, the symbolic WLK variant, that does not use word embeddings, appears more robust in this experiment and differences between the three directional setups are small.

5.3.4 Revisiting the data quality in BAMBOO.

Initial quality analyses (Section 5.2.1) suggested that the quality of BAMBOO is high, with a large proportion of AMR graphs that are of gold or silver quality. In this experiment, we study how metric rankings and predictions could change when confronted with AMRs corrected by humans. From every data set, we randomly sample 50 AMR graph pairs (300 AMRs in total). In each AMR, the human annotator searched for mistakes, and corrected them.¹⁷

We study two settings. i) Intra metric agreement (IMA): For every metric, we calculate the correlation of its predictions for the initial graph pairs versus the predictions for the

¹⁷Overall, few corrections were necessary, as reflected in a high SMATCH between corrected and uncorrected graphs: 95.1 (STS), 96.8 (SICK), 97.9 (PARA).

graph pairs that are ensured to be correct. Note that, on one hand, a *high IMA for all metrics* would further corroborate the trustworthiness of BAMBOO_W results. However, on the other hand, a *high IMA for a single metric* cannot be interpreted as a marker for this metric’s quality. I.e., a maximum IMA (1.0) could also indicate that a metric is completely insensitive to human corrections. Furthermore, we study ii) Metric human agreement (MHA): Here, we correlate the metric scores against human ratings, once when fed the fully gold-ensured graph pairs and once when fed the standard graph pairs. Both measures, IMA, and IAA, can provide us with an indicator of how much metric ratings would change if BAMBOO_W would be fully human corrected.

Results are shown in Table 5.7. All metrics exhibit high IMA, suggesting that potential changes in their ratings, when fed gold-ensured graphs, are quite small. Furthermore, on average, all metrics tend to exhibit slightly better correlation with the human when computed on the gold-ensured graph pairs. However, supporting the assessment of IMA, the increments in MHA appear small, ranging from a minimum increment of +0.3 (SEMBLEU) to a maximum increment of +2.8 (S²MATCH), whereas WWLK yields an increment of +1.8. Generally, while this assessment has to be taken with a grain of salt due to the small sample size, it overall supports the validity of BAMBOO_W results, and indicates that all metrics may profit from accurate parsing.

5.3.5 Alignment discussion

Align or not align? We can group metrics for graph-based meaning representations into whether they compute an **alignment** between AMRs or not (Liu et al., 2020). A computed alignment, as in SMATCH, has the advantage that it lets us assess finer-grained AMR graph similarities and divergences, by creating and exploiting a mapping that shows which specific substructures of two graphs are more or less similar to each other. On the other hand, it was still an open question whether such an alignment is worth its computational cost and enhances similarity judgments.

Experiments on BAMBOO_W provide novel evidence on this matter: **alignment-based metrics may be preferred for better accuracy. Non-alignment based metrics may be preferred if speed matters most.** The latter situation may occur, e.g., when AMR metrics must be executed over a large cross-product of parses (for instance, to semantically cluster sentences from a corpus). For a balanced approach, WWLK_Θ offers a good trade-off: polynomial-time alignment and high accuracy.

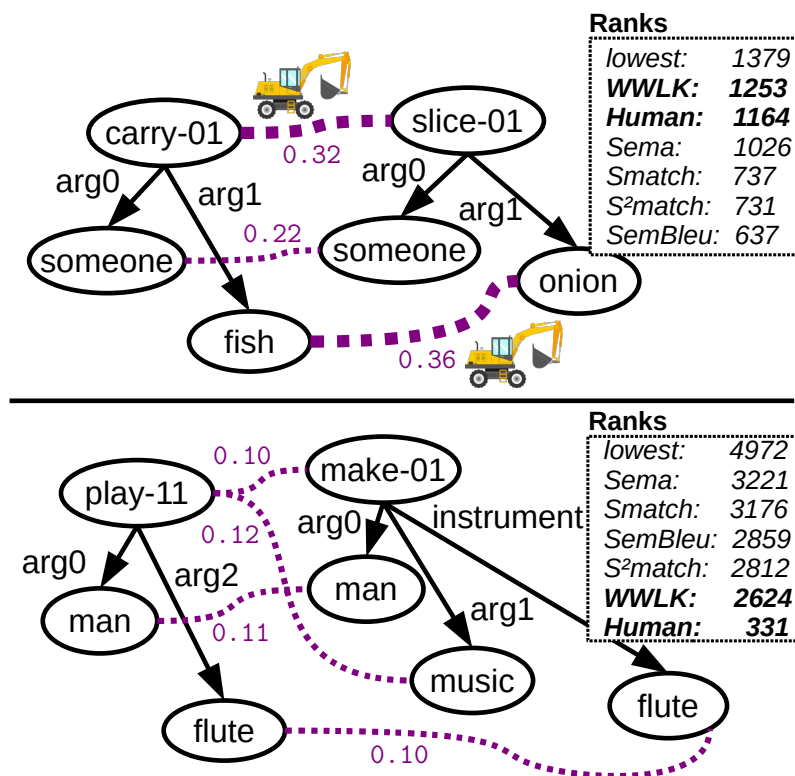


Figure 5.3: WWLK alignments and metric scores for dissimilar (top, STS) and similar (bottom, SICK) AMRs. Excavators indicate heavy Wasserstein work $flow \cdot cost$.

Example discussion I: Wasserstein transportation analysis explains disagreement.

Figure 5.3 (top) shows an example where the human-assigned similarity score is relatively low (rank 1164 of 1379). Due to the graphs having the same structure ($x \text{ arg0 } y; x \text{ arg1 } z$), the previous metrics (except SEMA) tend to assign similarities that are relatively too high. In particular, S²MATCH finds the exact same alignments in this case, but cannot assess the concept-relations more deeply. WWLK yields more informative alignments since they explain its decision to assign a more appropriate lower rank (1253 of 1379): substantial work is needed to transport, e.g., *carry-01* to *slice-01*.

Example discussion II: the value of n:m alignments.

Figure 5.3 (bottom) shows that WWLK produces valuable n:m alignments (*play-11* vs. *make-01* and *music*), which are needed to properly reflect similarity (note that SMATCH, WSMATCH and S²MATCH only provide 1-1 alignments). Yet, the example also shows that there is still a way to go. While humans assess this near-equivalence easily, providing a relatively high score (rank 331 of 4972), all metrics considered in this section, including ours, assign relative ranks that are too low (WWLK: 2624). Future work may incorporate external PropBank (Palmer

et al., 2005) knowledge into AMR metrics. In PropBank, sense 11 of *play* is defined as equivalent to *making music*.

5.3.6 Conclusions from BAMBOO_{MR} results

BAMBOO_{MR} is the first benchmark that allows researchers to assess AMR metrics empirically, setting the stage for future work on graph-based meaning representation metrics. We showcase the utility of BAMBOO_{MR}, by applying it to profile MR metrics, uncovering hitherto unknown strengths or weaknesses. We also saw that through BAMBOO_{MR} we are able to gain novel insight regarding suitable hyperparameters of different metric types, and to gain novel perspectives on how to further improve AMR similarity metrics to achieve better correlation with the degree of meaning similarity of paired sentences, as perceived by humans.

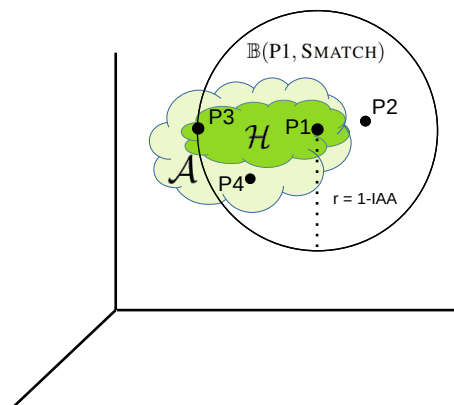


Figure 5.4: Sketch of MR IAA ball. The center ($P1$) is a reference MR, while $P2$, $P3$, $P4$ are candidates. Any MR x from the ball has high structural SMATCH agreement with $P1$, i.e., $\text{SMATCH}(x, P1) \geq$ estimated human IAA. However, they may fall in different categories: \mathcal{H} (green cloud) contains correct MR alternatives. Its superset \mathcal{A} (light cloud) contains acceptable MRs that may misrepresent the sentence meaning up to a minor degree. Other parses from the ball, e.g., $P2$, mis-represent the sentence’s meaning – despite possibly having higher SMATCH agreement with the reference than all other candidates.

5.4 Evaluating strong parsers with automatic and human AMR metrics

Now we’d like to test all the metrics that we have visited in a very classical application: evaluation of a mono-lingual parser against a gold reference corpus. Nowadays, this is particularly interesting, since thanks to astonishing recent advances in AMR parsing (mainly powered by the *language modeling and fine-tuning paradigm* (Bevilacqua et al., 2021)) we have parsers that now achieve benchmark scores that surpass IAA estimates, according to SMATCH.¹⁸ Therefore, we lack clarity on whether (fine) differences in SMATCH scores i) can be attributed to minor but valid divergences in interpretation or MR structure, as they may also occur in human assessments, or ii) if they constitute significant meaning distorting errors.

This fundamental issue is outlined in Figure 5.4. Four parses are located in the ball $\mathbb{B}(P1, \text{SMATCH})$ of estimated IAA, (gold) parse $P1$ being the center. However, the true set of possible human candidates \mathcal{H} is very likely much smaller than the ball and its shape

¹⁸Banarescu et al. (2013) find that an (optimistic) average annotator vs. consensus IAA, as measured in SMATCH, was 0.83 for newswire and 0.79 for web text. When newly trained annotators doubly annotated web text sentences, their annotator vs. annotator IAA was 0.71. Recent BART and T5 based models range between 0.82 and 0.84 SMATCH F1 scores.

is unknown.¹⁹ Besides, a superset of \mathcal{H} is a set of *acceptable* parses \mathcal{A} , i.e., parses that may have a small flaw which does not significantly distort the sentence meaning. Now, it can indeed happen that parse P2, as opposed to P3, has a lower distance to reference P1, i.e., to the center of $\mathbb{B}(\text{SMATCH})$ – but is not found in $\mathcal{A} \supseteq \mathcal{H}$, which marks it as an inaccurate candidate. On the other hand, P4 is contained in \mathcal{A} , but not in \mathcal{H} , which would make it acceptable, but less preferable than P3.

Research questions. Triggered by these considerations, we now tackle these key questions: *Do high-performance MR parsers indeed deliver accurate semantic graphs, as suggested by high benchmark scores that surpass human IAA estimates? Does a higher SMATCH against a single reference necessarily indicate better overall parse quality? And what steps can we take to mitigate potential issues when assessing the true performance of high-performance parsers?*

5.4.1 Study Setup: Data creation and MR metric setup

In this Section, we apply two popular high-performance parsers for creating candidate AMRs. Then we describe the human quality annotation, and give an overview of automatic AMR metrics that we consider in our subsequent studies.

Parsers and corpora. We choose the AMR3 benchmark²⁰ and the literary texts from the freely available Little Prince corpus.²¹ As parsers we choose T5- and BART-based systems, both on par with human IAA estimates, where BART achieves higher (SMATCH) scores on AMR3.²² We proceed as follows: we 1. parse the corpora with T5 and BART parsers and use SMATCH to select structurally diverging parse candidate pairs, and 2. sample 200 of those pairs, both for AMR3, and for Little Prince (i.e., 800 AMR candidates in total).

Annotation dimensions

Annotation dimension I: pairwise ranking. A human annotator is presented the sentence and two candidate graphs, assigning one of three labels and a free-text rationale.

¹⁹Under the unrealistic assumptions of an omniscient annotator and AMR being the ideal way of meaning representation, one might require that \mathcal{H} always has exactly one element.

²⁰LDC corpus [LDC2020T02](https://www.ldc.upenn.edu/LDC2020T02)

²¹From <https://amr.isi.edu/download.html>

²²See <https://github.com/bjascob/amr-lib-models> for more benchmarking statistics.


```

-----Reference AMR and Sentence-----
(1 / look-over-06          ``Looking over to the flag''
 :arg1 (f / flag))
-----Candidate parses-----
(1 / look-01              (z0 / look-01
 :direction (o / over)    :arg2 (z1 / flag)
 :destination (f / flag)) :direction (z2 / over))
-----Eval-----
Smatch (ref, cand): both score 0.2 (indicates low quality)
Human (sent, cand): both are acceptable
Human (cand, cand): no preference
-----

```

Figure 5.5: Data example: acceptable, low SMATCH. That is, $P \in \mathcal{H}$ but $P \notin \mathbb{B}(\text{SMATCH}, \text{ref})$.

```

-----Reference AMR (excerpt)-----
(i2 / imagine-01
 :arg0 (y / you)
 :arg1 (a / amaze-01
 :arg1 (i / i))
 :time-of (w / wake-01
 :arg0 (v / voice
 :mod (o / odd)
 :mod (l / little))
 :arg1 i))))
-----Candidate parse (excerpt)-----
(ii / imagine-01
 :arg0 (y / you)
 :arg1 (a / amaze-01
 :arg0 (v / voice
 :mod (l / little)
 :mod (o / odd))
 :arg1 (ii2 / i)))
Means: (..) imagine my amazement (..) by an odd little voice
Should mean: (..) imagine my amazement (..) when I was
awakened by an odd little voice
-----Eval-----
Smatch (ref, cand): scores 0.88 (indicates high quality)
Human (sent, cand): not acceptable
-----

```

Figure 5.6: Data example excerpt that shows an unacceptable parse with high SMATCH. That is, $P \notin \mathcal{A} \supseteq \mathcal{H}$ but $P \in \mathbb{B}(\text{SMATCH}, \text{ref})$

The labels are either +1 (prefer first graph), −1 (prefer second graph), or 0 (both are of same or very similar quality).

Annotation dimension II: parse acceptability. In addition, each graph is independently assigned a single label, considering only the sentence that it is supposed to represent. Here, the annotator makes a binary decision: +1, if the parse is acceptable, or 0, if the graph is not acceptable. A graph that is acceptable is fully valid, or may allow a very minor meaning deviation from the sentence, or a slightly weird but allowed interpretation that may differ from a normative interpretation. All other graphs are deemed not acceptable (0).

Example: Acceptable candidates, low SMATCH. Figure 5.5 shows an example of two graphs that have very low structural overlap with the reference (SMATCH = 0.2), but are acceptable. Here, the candidate graphs both differ from the reference because they tend to a more conservative interpretation, using the more general *look-01* predicate instead of

the *look-over-06* predicate in the human reference. In fact, the meaning of the reference can be considered, albeit valid, slightly weird, since *look-over-06* is defined in PropBank as *examining something idly*, which is a more ‘specific’ interpretation of the sentence in question. On the other hand, the candidate graphs differ from each other in the semantic role assigned to *flag*. In the first, *flag* is the destination of the *looking* action (which can be accepted), while in the second, we find a more questionable but still acceptable interpretation that *flag* is an *attribute of the thing that is looked at*.

Example: Candidate not acceptable, high SMATCH. An example that is inverse (high SMATCH, unacceptable) is shown in Figure 5.6, where the parse omits *awaken*. Albeit the factuality of the sentence is not (much) changed, and the structural deviation may legitimately imply that the odd voice is the cause of amazement, it misses a relevant piece of meaning and is therefore rated unacceptable.

Label statistics will be discussed in Section 5.4.2, where the human annotations are also contrasted against parser rankings of automatic metrics.

Metric selection

We distinguish metrics that aim specifically at *monolingual AMR parsing evaluation* from *multi-purpose MR metrics* that aim at generalized use-cases. Recall that MR metrics that are more targeted to evaluation of monolingual parsers typically have two features in common. First, they compare a candidate against a reference parse that both (try to) represent the *same sentence*. Second, they measure the amount of successfully reconstructed reference structure.²³

We also consider our novel multi-purpose MR metrics that aim to extend to use cases where MRs represent *different sentences*, such as evaluation of cross-lingual MR parsing, natural language generation (NLG) or rating semantic sentence similarity.

Monolingual AMR parsing metrics. We consider SMATCH, SEMA and SEMBLEU, as previously introduced. Recall that, per default, SEMBLEU uses $k=3$. But we additionally use $k=2$, following our insights from evaluation on BAMBOO₂ (cf. Section 5.2), where we found that $k=2$ better relates to human notions of sentence similarity.

²³The notion of *success* is mostly focused on structural matches, and can vary among metrics, usually depending on theoretical arguments of the developers of the metric.

Multi-purpose metrics: S²MATCH and WLK/WWLK. Targeting AMR metric application cases beyond monolingual parsing evaluation, such as measuring AMR similarity of different sentences, we introduced three metrics: i) **S²MATCH** that computes graded concept similarity (reflecting that, e.g., *cat* is more similar to *kitten* than to *plant*). ii) **WLK** applies the Weisfeiler-Leman kernel (Shervashidze et al., 2011) to compute a similarity score over feature vectors that describe graph statistics in different iterations of node contextualization. iii) **WWLK** (Wasserstein WLK, Togninalli et al. (2019)) projects the nodes of the graphs to a latent space partitioned into different degrees of node contextualization. Wasserstein distance is then used to match the graphs, based on a pair-wise node distance matrix.

Setup of multi-purpose metrics. For S²MATCH, WLK and WWLK we use the default setup, which consists of GloVe (Pennington et al., 2014a) embeddings and $k=2$ in WLK and WWLK, where k indicates the depth of node contextualizations.

Default WWLK initializes parameters randomly, if tokens are out of vocabulary (a random embedding for each OOV token type). To achieve deterministic results, without fixing a random seed, we could initialize the OOV parameters to 0. However, with this we’d lose valuable discriminative information on graph similarity. We therefore adopt a slight adaptation for WWLK and calculate the *expected* distance matrix before Wasserstein metric calculation, making results more reproducible while keeping discriminative power.

We also introduce **WWLK-k3e2n**, a WWLK variant with *edge2node* (*e2n*) transforms, more tailored to monolingual AMR parsing evaluation, which is the focus of this section. It increases the score impact of edge labels, motivated by the insight that edge labels are of particular importance in AMR parsing evaluation. It transforms an edge-labeled graph into an *equivalent* graph without edge-labels.²⁴ This is also known as ‘Levi transform’ (Levi, 1942), and has been previously advocated for AMR representation by Beck et al. (2018) and Ribeiro et al. (2019). Since due to the transform the distances in the graph will grow, we increase k by one ($k=3$). With this, we can set all edge weights to 1.

Simple baseline

To put the results into perspective, we introduce a very SIMPLE baseline: SIMPLE extracts bag-of-words (relation and concept labels) from two AMR graphs and computes the size of their intersection vs. the size of their union (aka *Jaccard Coefficient*).

²⁴E.g., $(x, \text{arg0}, z) \rightarrow (x, y) \wedge (y, z) \wedge (y, \text{arg0})$.

5.4.2 Study I: System-level scoring

Research questions. We focus on two questions:

1. How are the two parsers rated by humans?
2. How do metrics score our two parsers?

With 1. we aim to assess whether there is still room for AMR parser improvement, even though their SMATCH scores pass estimated human IAA. And for 2. we aim to know whether the metric rankings (still) appropriately reflect parser quality.

System scoring

Aggregation strategies: Micro vs. Macro. We have defined a metric between two AMRs. For ranking systems, we need to aggregate the individual pair-wise assessments into a single score. At this point, it is important to note that most papers use (only) micro SMATCH for ranking parsers, i.e., counting triple matches of aligned AMR pairs over all AMR pairs (before a final F1 score calculation).

Naturally, such micro corpus statistics are *unbiased* w.r.t. to whatever is defined as a single evaluation instance (in SMATCH: triples), but the trade-off is that they are biased towards instance type frequency and sentence length, since longer sentences tend to yield substantially more triples. Hence, the influence of a longer sentence may marginalize the influence of a shorter sentence. This issue may be further aggravated by the fact that longer sentences tend to contain more named entity phrases, and entity phrases typically trigger large simple structures, that are mostly easy to project.²⁵ Therefore, micro corpus statistics alone *could* potentially yield an incomplete assessment of parser performance. To shed more light on this issue, we provide additional evaluation via macro aggregation.

Statistics for micro and macro system scoring. We calculate two statistics. The first statistic shows the (micro/macro)-aggregated corpus score for a metric m , parsed corpus X and gold corpus G :

$$\begin{aligned} \mathbb{S}(m, X, G) \\ = \text{AGGR}(\{m(X_1, G_1), \dots, m(X_n, G_n)\}), \end{aligned}$$

²⁵As a small example, consider *The bird sings* vs. *Jon Bon Jovi sings*. The first sentence yields 3 triples, while the second sentence yields 8 triples, where the *John Bon Jovi* named entity structure has added 6 triples, outweighing the key semantic event *x sings*. Micro score would assign 2.6 times more importance to the second sentence/AMR.

		Little Prince						AMR3					
		P			S			P			S		
		BART	T5	Δ	BART	T5	Δ	BART	T5	Δ	BART	T5	Δ
	HUM	87	113	-26	0.58	0.69	-0.11	100	100	0.0	0.62	0.62	0.00
	SIMPLE	87	113	-26	0.69	0.7	-0.01	82	118	-36	0.75	0.75	0.00
Macro	SEMA	84	116	-32	0.6	0.63	-0.03	89	111	-22	0.68	0.68	0.00
	SEMBLEU-k2	90	110	-20	0.61	0.63	-0.02	98	102	-4	0.70	0.69	0.01
	SEMBLEU-k3	90	110	-20	0.51	0.53	-0.02	103	97	6	0.58	0.58	0.00
	SMATCH	94	106	-12	0.73	0.74	-0.01	95	105	-10	0.77	0.77	0.00
	S ² MATCH	93	107	-14	0.75	0.76	-0.01	95	105	-10	0.79	0.79	0.00
	WLK-k2	92	108	-16	0.63	0.65	-0.02	96	104	-8	0.69	0.69	0.00
	WWLK-k2	91	109	-18	0.79	0.8	-0.01	102	98	4	0.84	0.84	0.00
	WWLK-k3e2n	97	103	-6	0.72	0.73	-0.01	94	106	-12	0.78	0.78	0.00
Micro	SEMA	-	-	-	0.62	0.64	-0.02	-	-	-	0.69	0.68	0.01
	SEMBLEU	-	-	-	0.53	0.54	-0.01	-	-	-	0.60	0.57	0.03
	SMATCH	-	-	-	0.74	0.74	-0.01	-	-	-	0.77	0.75	0.02
	S ² MATCH	-	-	-	0.76	0.76	0.00	-	-	-	0.80	0.77	0.03

Table 5.8: Corpus level scoring results. Negative Δ shows preference for T5, positive Δ shows preference for BART.

For macro metrics, *AGGR* is the mean of pair-wise scores over all instances in a corpus X . In the case of the human metric, this is the ratio of acceptable parses in X . For micro metrics, *AGGR* computes overall matching triple F1 (SMATCH, SEMA) or overall k-gram BLEU (SEMBLEU). For WLK and WWLK, a micro variant is not implemented, hence we only show their macro scores.

The second statistic shows how often m prefers the parses in a parse corpus X over the these in Y :

$$\mathbb{P}(m, X, Y, G) = \sum_{i=1}^n \mathbb{I}[m(X_i, G_i) > m(Y_i, G_i)].$$

Here, $\mathbb{I}[c]$ denotes a function that returns 1 if the condition c is true, and zero in all other cases. For better comparability of numbers, we distribute cases where $m(X_i, G_i) = m(Y_i, G_i)$, which occur in the human annotation, evenly over $\mathbb{P}(m, X, Y, G)$ and $\mathbb{P}(m, Y, X, G)$.

Results

Results are shown in Table 5.8. In view of our research questions, we make interesting observations.

AMR parsing is far from solved. Considering the ratio of parses that were rated acceptable by the human (HUM, \mathbb{S}), they are surprisingly low, at only 0.58 (BART, Little Prince, Table 5.8); 0.69 (T5, Little Prince). Other parses have errors that substantially distort sentence meaning, even though major parts of the AMRs may structurally overlap.

Better SMATCH on AMR benchmark may not (always) imply a better parser. On AMR3, when inspecting corpus-SMATCH (micro SMATCH, Table 5.8), BART is considered the better parser, in comparison to T5 (+2 points). However, when consulting macro statistics, a different picture emerges. Here, BART and T5 obtain the same scores: AMR3, 0.62 vs. 0.62, Table 5.8. On the literary texts (Little Prince), where the domain is different and sentences tend to be shorter, T5 significantly (binomial test, $p < 0.05$) outperforms BART, both in the ratio of acceptable sentences (BART: 0.58, T5: 0.69), and in number of preferred candidates (BART: 87, T5: 113). Note that this insight is independent from our human annotations.

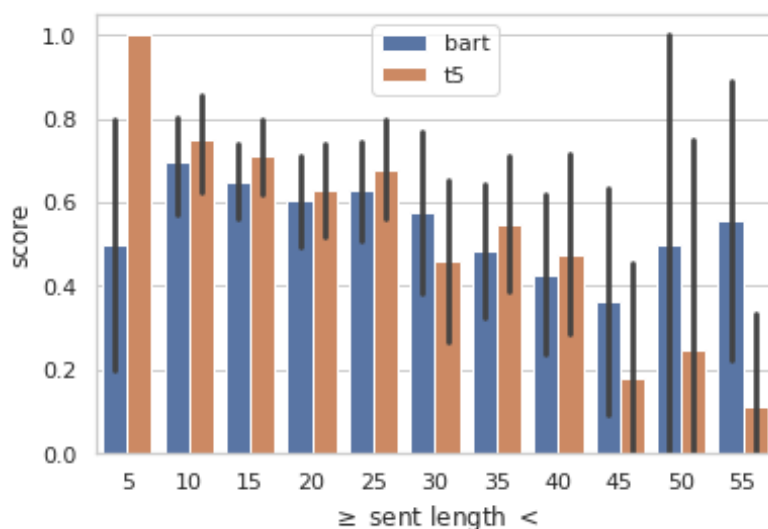


Figure 5.7: **Sentence length vs. human acceptability** on all annotated data. 55 includes all sentences longer than 55 tokens. See Figure 5.9 for occurrences of different sentence lengths.

All in all, this may suggest that BART tends to provide better performance for longer sentences, while T5 tends to provide better performance especially for shorter and medium-length sentences. Further analysis provides more evidence for this: In Figure 5.7 and Figure 5.8) we see that the longer the sentences the better seems the prediction by BART.

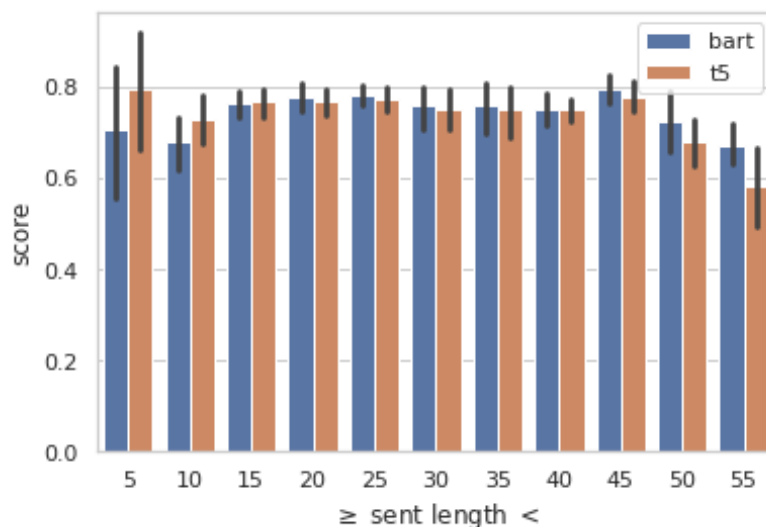


Figure 5.8: **Sentence length vs. Smatch** on all annotated data. 55 includes all sentences longer than 55 tokens. See Figure 5.9 for occurrences of different sentence lengths. Other metrics look similar.

While, the inverse picture emerges for T5. This could point at simple improvement potential by T5, when the longer sentences can be reasonably split before parsing and well fused together after parsing. For total sentence length distribution see Figure 5.9.

Metrics for system ranking. Regarding our tested metrics, especially the macro metrics, a clear pattern is that they mostly agree with the human ranking. However, our current results for the different metrics do not tell much, yet, about their suitability for AMR assessment and ranking. Even if a metric ranks a parser more similarly to the human, this may be for the wrong reasons, since this statistic filters out pair-wise correspondences to the human. This is also indicated by results of the simplistic bag-of-structure metric SIMPLE, which achieves the same results as human (HUM) on Little Prince, with respect to the number of preferred parses (\mathbb{P} , Little Prince, Table 5.8, HUM vs. SIMPLE). In that respect, it is more important to assess the pair-wise metric accuracy and metric specificity, which we will visit next in Sections 5.4.3 and 5.4.4.

5.4.3 Study II: Metric accuracy on parse level

Research questions. Now, we are interested in the metric accuracy, that is, agreement of AMR metrics with the human ratings. In particular, we would like to know, regarding:

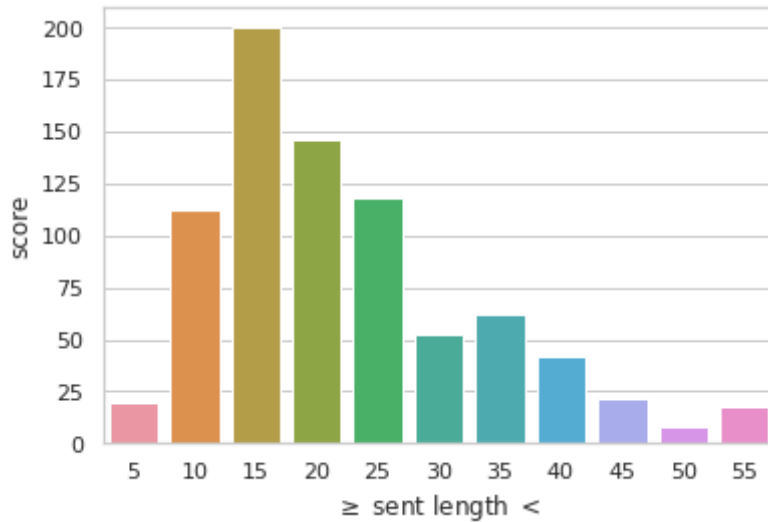


Figure 5.9: Sentence length occurrences. 55 includes all sentences longer than 55 tokens.

- Pair-wise parse accuracy: How do metrics agree with human preferences when ranking two candidates?
- Individual parse accuracy: Can metrics tell apart acceptable from unacceptable parses?

Note that these are hard tasks for metrics, since both T5 and BART show performance levels on par or above estimated measurements for human IAA. Therefore, smaller structural divergences from the reference can potentially have a bigger impact on parse acceptability (or preference) than larger structural deviations, that could express different (but valid) interpretations or (near-)paraphrases.

Evaluation metrics

Pairwise accuracy. Recall that the human assigned one of three ratings: 1, if AMR x is better, -1 , if AMR y is better, and 0 if there is no considerable quality difference between two candidate graphs x and y . A metric assigns two real values, $m(x, g)$ and $m(y, g)$, where g is the reference graph. Mapping the score to -1 or 1 is simple and intuitive, prompting us to introduce pair-wise accuracy. Consider a data set SD that contains all graph triplets (x, y, g) with a human preference sign (label -1 or $+1$). Further, let $\delta^m(x, y, g) = m(x, g) - m(y, g)$ the (signed) quality difference between x and y when using

m . Analogously $\delta^h(x, y)$ is the human preference. Then, the pairwise accuracy is

$$PA = \frac{1}{|D|} \sum_{(x,y,g) \in D} \mathbb{I}[\delta^m(x, y, g) \cdot \delta^h(x, y) > 0]. \quad (5.8)$$

It measures the ratio of candidate pairs where the metric has made the same signed decision as the human, in preferring one parse over the other.

Acceptability score. When rating acceptability, the human rates a single parse (given its sentence), assigning 1 (acceptable) or 0 (no acceptable). The metrics make use of the reference graph to compute a score. Aiming at an evaluation metric that makes as few assumptions as possible, we formulate the following expectation for an AMR graph metric to fulfill: the average rank of the scores for parses that have been labeled acceptable by the human should surpass the average rank of the scores for parses labeled as being not acceptable. Let \mathcal{J}^+ (\mathcal{J}^-) be the set of indices for which the human has assigned a label that indicates (un-)acceptability. Let $S = \{m(X_1, G_1) \dots m(X_n, G_n)\}$ be the metric m 's scores over all (x, g) parse/reference pairs, and R be the ranks of D . Let R^+ (and R^-) be the set of ranks indexed by \mathcal{J}^+ (and \mathcal{J}^-). Then

$$\mathbb{A}\Delta = \text{avg}(R^+) - \text{avg}(R^-) \quad (5.9)$$

To increase robustness, we use $\text{avg} := \text{median}$.

Results

The results are shown in Table 5.9. We conclude:

All metrics are suitable for pairwise-ranking of parses from high-performance parsers.

All metrics significantly outperform the random baseline with regard to the pair-wise ranking accuracy (PA). For Little Prince, SMATCH and S²MATCH yield the best performance, while for AMR3, WWLK-k3e2n has the best performance (closely followed by SEMBLEU-k2). Among different metrics, however, the differences are not large enough to confidently recommend one metric over the other.

Parse acceptability rating is hard. When tasked to rate parse acceptability ($\mathbb{A}\Delta$), all metrics show issues. For Little Prince, only SMATCH and WWLK-k3e2n significantly outperform the chance baseline, while for AMR3 all metrics are significantly above chance

	Little Prince		AMR3	
	PA	$\mathbb{A}\Delta$	PA	$\mathbb{A}\Delta$
HUM	1.0	233	1.0	234
RAND	0.5	0.0	0.5	0.0
SIMPLE	0.66 [†]	11.0	0.68 [†]	39.5 [†]
SEMA	0.66 [†]	24.3	0.7 [†]	35.3 [†]
SEMBLEU-k2	0.67 [†]	25.0	0.74 [†]	28.0
SEMBLEU-k3	0.63 [†]	32.0	0.68 [†]	29.0
SMATCH	0.72[†]	42.0 [†]	0.7 [†]	35.0 [†]
S ² MATCH	0.72[†]	35.3	0.7 [†]	42.3 [†]
WLK	0.66 [†]	28.0	0.68 [†]	41.5 [†]
WWLK-k2	0.63 [†]	20.5	0.73 [†]	51.0 [†]
WWLK-k3e2n	0.66 [†]	48.0[†]	0.76[†]	57.0[†]

Table 5.9: Metric agreement with human. †: random baseline (RAND) not contained in 95% confidence interval.

level, except SEMBLEU. Overall, however, the differences are not large enough to confidently recommend one metric over the other. On both corpora, best results are achieved with WWLK-k3e2n (Little Prince: 48.0, AMR3: 57.0).

Control experiment of metrics. We additionally parse a subset of 50 sentences with an older parser (Flanigan et al., 2014) that scores more than 20 points lower SMATCH, when compared with IAA as estimated in Banarescu et al. (2013). All metrics (with the exception of SIMPLE for one pair) correctly figure out all rankings and acceptability (according to the human, BART and T5 are preferred in all cases, except two cases where all three systems deliver equally valid graphs). This indicates that metrics indeed can accurately tell apart quality differences, *if* they are large enough and do not lie beyond human IAA.

5.4.4 Metric specificity

We found little evidence that could help us give recommendations on which metrics to prefer over others for monolingual parser evaluation in the high-performance regime. On the contrary, we found evidence that no metric can sufficiently assess parse acceptability. Therefore, it is interesting to see whether the metrics can provide *specific* views on parse quality and behaves differently from other metrics.

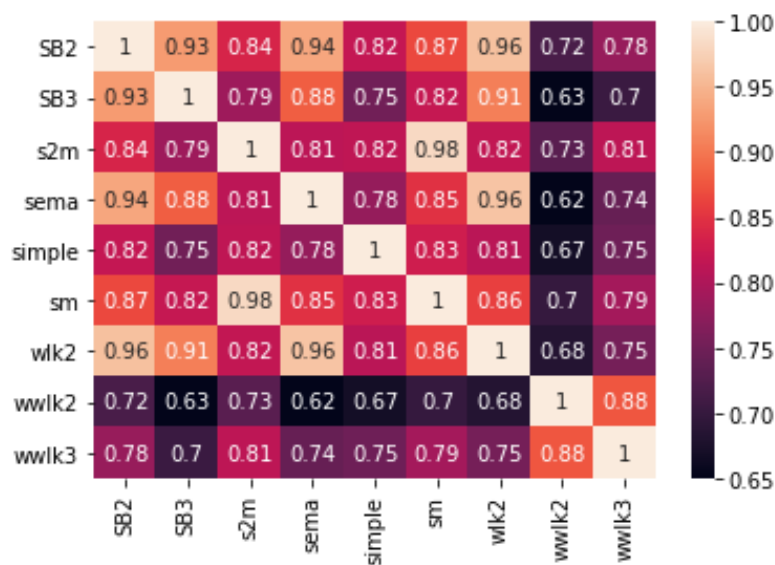


Figure 5.10: Inter-metric correlation on Little Prince.

Statistics. We compute Spearman’s ρ over metric pairs. Spearman’s ρ calculates Pearson’s ρ on the ranked predictions, which increases robustness.

Results are plotted in Figures 5.10 and 5.11. For both datasets, we see that the Wasserstein metrics provide rankings that differ more from the rankings assigned by other metrics, suggesting that they have unique features. On the other hand, the SEMBLEU metrics tend to agree the most with the rankings of the other metrics, suggesting that they share more features with other metrics. On a pair-wise level, the most similar metrics are SMATCH and S²MATCH, which is intuitive, since S²MATCH is an adaptation of SMATCH that also targets the comparison of AMRs from different sentences. Indeed, synonyms and similar concepts are unlikely to often occur in monolingual parsing, where parses contain exactly matching concepts. Further, WLK very much agrees with SEMBLEU, which seems intuitive, since both aim at comparing larger AMR subgraphs. Lowest agreement is exhibited between SEMA and WWLK, perhaps because these metrics are of different complexity and share different goals: simple and fast match of structures vs. graded assessment for general AMR similarity.

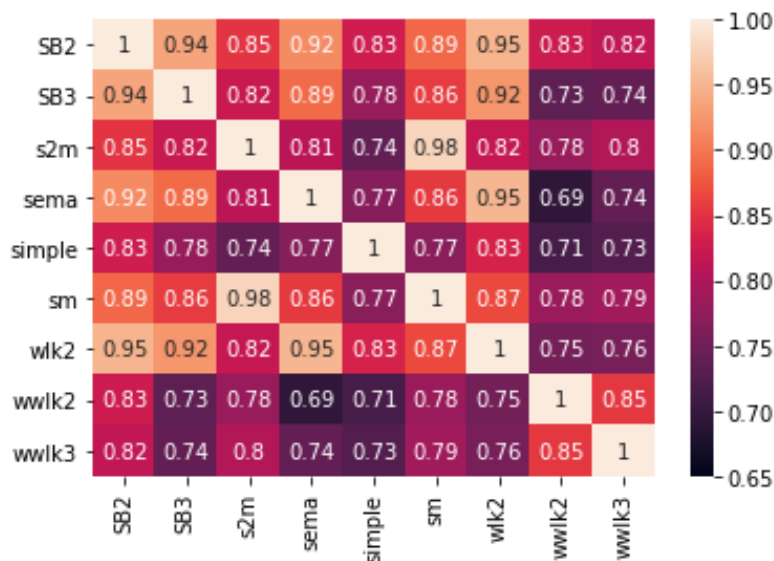


Figure 5.11: Inter-metric correlation on AMR3.

5.5 Discussion of limitations and recommendations for evaluating strong AMR parsers

5.5.1 Limitations

There are limitations of our study:

Limitation I: *Single vs. Double annotation.* While our quality annotations stem from an experienced human annotator, we would have liked to obtain annotations from a second annotator to measure IAA for AMR quality rating. This was partly precluded by the high costs of AMR annotation, which requires much time and experience. This is also reflected in the AMR benchmark corpora: the majority of graphs were created by a single annotator. Note, however, that some findings are independent of annotation (e.g., macro vs. micro metric corpus scoring, metric specificity).

Limitation II: *Assessing individual suitability of metrics for rating high-performance parsers.* Our study reports relevant findings on (monolingual) AMR parsing evaluation in high-performance regimes, and on upper bounds of AMR parsing. But an important question we had to leave open is the individual suitability of the metrics for comparing high-performance parsers.

Limitation III: Single-reference parses and ambiguity. Elaborating on *Limitation II* and recalling that AMR benchmarks have only single references, another caveat is that potentially correct metric behavior may be misinterpreted in our study. E.g., if a sentence allows two different interpretations, a metric might (correctly) yield a low score for the reference (different meaning), while the (reference-less) human rating may find the parse acceptable. This issue may also be mitigated by providing (costly) double annotation of AMR benchmark sentences.

To facilitate follow-up research, we release the annotated data. Our Little Prince annotations can be freely released, while AMR3 annotations require proof of LDC license.

5.5.2 Recommendations for parser selection and metric improvement perspectives

Main recommendations based on our study:

Recommendation I Besides micro aggregate scores we recommend using a **macro aggregate score** for parse evaluation (e.g., macro SMATCH, computed as an average over sentence scores): Commonly, only micro corpus statistics are used to compare and rank parsers. Yet, we found that macro (sentence-average) metrics can provide a valuable **complementary assessment** that can highlight important *additional* strengths of high-performance parsers.

Recommendation II We recommend conducting **more human evaluation of AMR parses**. With the available high-performance AMR parsers, it becomes more important to conduct manual analyses of parse quality. Our study provides evidence that AMR parsing still has large room for improvement, due to small but significant errors. Since this may not be noticeable for (current) metrics when given a single human reference, **future work on parsing may profit from careful human acceptability assessments**.

T5 vs. BART: which parser to prefer? Next to AMR parser developers, this question mainly concerns potential users of AMR parsers. Fine-tuned T5 and BART are both powerful AMR parsers. We observe a slight tendency that researchers prefer BART, possibly since it achieves slightly better SMATCH scores than T5 on the AMR3 benchmark. But

our work shows that differences between the systems are often finer than what can be assessed with structural overlap metrics (SMATCH), and both systems are generally strong but struggle with small but significant meaning errors.

In our study we found that when choosing between T5 and BART based AMR systems, **the choice might depend on the target domain**. Indeed, our results on Little Prince and AMR3 (mainly news) could indicate that **T5 may have an edge over BART when parsing literary texts**, and shorter sentences in general, while **BART has an edge over T5 when parsing longer sentences, and sentences from news sources**, especially if they are longer. However, it must be clearly noted that we do not know (yet) whether this insight carries over to other types of literary texts.

Perhaps, if we presume that performance is carried over to other types of literary texts, a possible explanation can be found in the data these two large models were trained on. BART uses the same training data as RoBERTa (Liu et al., 2019a), e.g., Wikipedia, book corpora and news. T5 leverages the colossal common crawl corpus (C4), that contains all kinds of texts scraped from the web. This *could* make T5 more robust to AMR domain changes, but less suitable for analysing longer sentences, since these may occur more frequently in BART’s corpora that seem more normative.

Which AMR metric to use? Our findings do not provide conclusive evidence on this question, partly due to insufficient data size, partly due to the general difficulty of the task. WWLK-k3e2n seems slightly more useful for detecting parse acceptability and pairwise ranking on news, while SMATCH yields best ranking on Little Prince.

However, our work shows that **it can be useful to calculate more than one metric to compare parsers**. In particular, we saw that predictions of structural matching metrics differ considerably from graded semantic similarity-based metrics, such as the WWLK metric variants. This suggests that these two types can provide complementary perspectives on parsing accuracy. Metric selection may, of course, also be driven by users’ specific desiderata, such as speed (SEMA, SEMBLEU, WLK), 1-1 alignment (SMATCH), n:m alignment (WWLK), or graded matching (SMATCH, WWLK).

5.6 Discussion


In this chapter, we empirically investigated MR metrics. Through our BAMBOO benchmark, we can investigate MR metrics with regard to two main objectives: sentence similarity and robustness to meaning-preserving and meaning-changing graph translation (for


BAMBOO Meaning Graph Similarity Benchmark

[➔ Here's](#) updated results on BAMBOO.

Contribute your results:

1. evaluate your metric
2. open an issue or a pull request
 - pull request: update both tables below
 - issue: report evaluation results, link to your metric, commit number (optional: paper link)

Figure 5.12: Snippet from BAMBOO  versioning on GitHub: <https://github.com/flimpz357/bamboo-amr-benchmark>.

an overview of some translations, recall Figure 2.1.2). We find that our WWLK metrics provide a valuable balance of efficiency and accuracy across the different objectives. For MR parsing evaluation, we investigated the capacity of metrics to discriminate MR parsers – based on a single gold reference. We find that if the parsers are of very different quality, discrimination is possible using any of the examined metrics. However, if both parsers are strong, all metrics struggle to provide us with a meaningfully differentiating picture. As of now, it remains an open question whether this is mainly due to issues in metrics or the fact that a single gold reference does not offer multiple interpretations, which can easily arise if there is not sufficient semantic context, as is generally the case in many sentences. To investigate this we require the design of more tests (possibly integrated into BAMBOO ) and need to develop metrics that can ideally take into account different interpretations without over-focusing on a single-reference. As a seemingly simple but very costly alternative or complement, we can consider to establish double annotation of MR reference. Importantly, a conjecture of our study is that *AMR parsing is far from solved*.²⁶

5.7 Creating a live benchmark with metric versioning

Benchmark results that are displayed in a static table ‘on paper’ have disadvantages, mainly due to being hard to update and extend. Indeed, concrete metric implementations are *software*, and software is bound to undergo changes, e.g., due to fixing bugs or implementing incremental improvements. These modifications might not change much the nature of a metric, but the practical reality is that they could lead to slight changes

²⁶An insight that now also has been corroborated by Groschwitz et al. (2023) who propose to address the issue with challenge sets.

in results. Similarly, the BAMBOO²⁷ evaluation itself may also be subject to updates, for instance, evaluation metrics could be improved, or sensible tasks be added. To better trace such developments, it makes sense to treat a benchmark like a piece of software and subject it to versioning. Therefore, we keep a official BAMBOO²⁷ github repository that invites researchers not only to post their metric results and compare against other metrics, but also to improve or extend the benchmarking²⁷ and allow us to keep metrics and results updated, under full transparency by exactly showing which version (commit uri) of a metric is used (Figure 5.12). The webpage of for our live benchmark is: <https://github.com/flipz357/bamboo-amr-benchmark>.

²⁷A constraint is that all data should be under public license. We have also considered to add the AMR parser evaluation challenge to BAMBOO²⁷. However, then BAMBOO²⁷ could not be freely distributed anymore, due to parts of the parser evaluation challenge being under non-public license. Therefore, as long as the license situation doesn't change, we decide to abstain from including the parser evaluation challenge in BAMBOO²⁷.

Part II

MR metrics for novel evaluation applications

Chapter 6

MRs in NLG evaluation

6.1 Chapter outline

In this chapter, we investigate MR metrics for an important NLP use-case where we have to cope with the absence of (at least) one of two MRs: The evaluation of generated text. While at first glance, this scenario seems to prevent us from directly applying MR metrics (since at maximum only the input MR is at our hands), we show that we can *project* the MRs with a strong parser and then execute our MR metrics, a very general method that we call REMATCH (Reconstruction Match). Besides making feasible a meaning-focused measurement, an additional advantage of such a projection is that in the setup of MR2text generation, we can perform the evaluation without a costly human reference. Moreover, the MR projection allows us to extend MR metric evaluation to all kinds of other generation evaluation tasks, such as machine translation or summarization, opening ways to profit from the metrics' controllable and interpretable way of taking measurements.

The remainder of this chapter is structured as follows:

1. After discussing our motivation in more detail (Section 6.2),
2. we formalize the problem of MR2text evaluation, and build a new metric that views text quality as composed of *F*orm and *M*eaning quality (*MF* score, Section 6.3).
3. We conduct two pilot studies: In Section 6.4, we re-rank NLG systems using our novel metric, assessing its discriminatory power and potential for providing us with coarse and fine-grained explanations for system quality scores. In our second pilot study (Section 6.5), we probe drawbacks of our metric, such as its dependence on a reliable MR parser or language model.
4. We conclude the chapter with a discussion (Section 6.6).

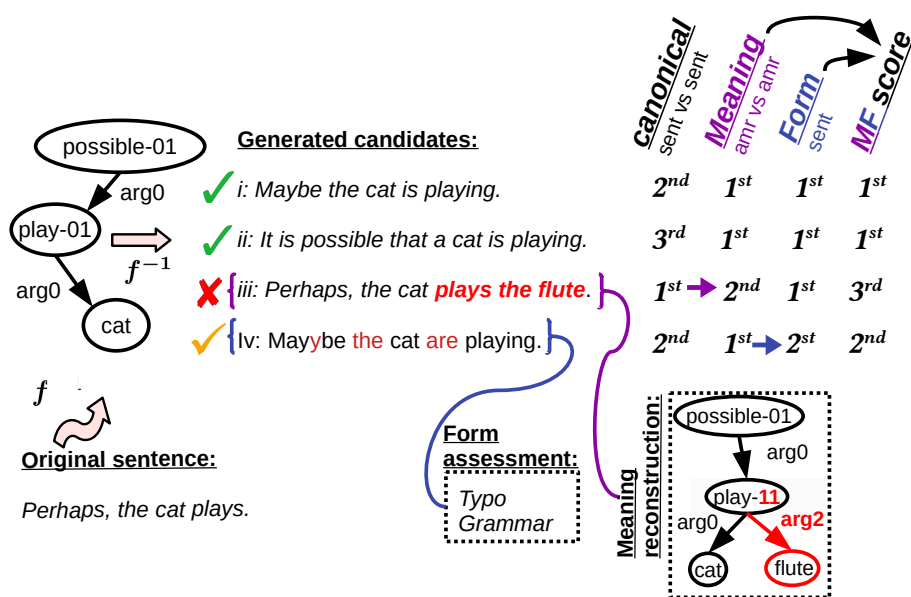


Figure 6.1: The *Canonical* evaluation matches n-grams from the sentences and assigns inappropriate ranks. Our metric \mathcal{MF}_β fuses *Meaning* and *Form* assessment and better reflects the ranking of the generations.

Underlying work. The content of this chapter is mainly based on work from Opitz and Frank (2021).

6.2 Motivation: Why standard metrics are insufficient for MR2text evaluation

Systems that generate texts from MR are evaluated by comparing the outputted text against a reference. However, the usually applied text matching metrics are known to suffer from issues such as high sensitivity to outliers (Mathur et al., 2020a), and lack of interpretability (Sai et al., 2020). In fact, some of these issues get compounded when evaluating MR2text. The core of the problem is that there are many ways to realize a sentence from a meaning representation. Figure 6.1 shows four candidate sentences (i-iv) for a given AMR (left).

One system generates (i): *Maybe the cat is playing.* while another generates (iii): *Perhaps, the cat plays the flute.* Clearly, (i) captures the meaning of the gold graph better than (iii), which contains ‘hallucinated’ content – a well-known issue in neural generation (Logan et al., 2019; Wang and Sennrich, 2020).

Yet, when using a canonical metric such as BLEU to evaluate sentences (i) and (iii)

against the reference, the system that produces hallucinations (**iii**) is greatly rewarded (54 BLEU points) to the disadvantage of systems that yield meaning preserving sentences (**i**) (18 points) and (**ii**) (5 points).

Therefore, we now want to aim at a (better) metric that **measures meaning preservation** of the generated output towards the MR given as input, by (re-)constructing an AMR from the generated sentence and comparing it to the input AMR. In Figure 6.1, Reconstruction is the result of parsing (**iii**). The reconstructed AMR exposes several meaning deviations (marked in red): it contains an alternate sense of *play* and contains an additional semantic role *arg2* with filler *flute*. By contrast, when converting sentences (**i**), (**ii**), or (**iv**) to AMRs, we obtain flawless reconstructions. We will measure preservation of Meaning using our MR graph matching metrics.

Figure 6.1 also illustrates that assessing meaning preservation is not sufficient to rate the quality of generations: (**iv**) captures the meaning of the AMR well – but its form is flawed: it suffers from wrong verb inflection, a common issue in low-resource text generation settings (Koponen et al., 2019).

In order to rate both meaning and form of a generated sentence, we combine the score for meaning reconstruction with a score called Form that **judges the sentence’s grammaticality and fluency**.

By these moves, we obtain a more suitable and explainable ranking with a combined MF score.¹ By clearly distinguishing between Meaning and Form, our MF score (henceforth denoted by \mathcal{MF}_β) also aligns well with recent calls to achieve a clearer separation of these aspects in NLU (Bender and Koller, 2020).

Generally, next we’ll proceed as follows:

(1) We propose two linguistically motivated principles that aim at a sound evaluation of MR2text systems, but may also extend to other generation tasks: the **principle of meaning preservation** and the **principle of (grammatical) form**.

(2) From these principles we derive and implement a (novel) \mathcal{MF}_β **score for language generation evaluation**² which is composed of individual metrics for transparently measuring and distinguishing meaning and form aspects. With a single parameter (β), \mathcal{MF}_β allows users to modulate these two views on generation quality to vary their impact on the final metric score.

¹See Figure 6.1: 1st/2nd rank: **i**; 3rd rank: **iv**; 4th rank: **iii**.

²We make code available at <https://github.com/Heidelberg-NLP/MFscore>.

(3) We conduct two major pilot studies involving (English) text generations from a range of competitive MR2text systems and human annotations. First we study the potential practical benefits of $\mathcal{M}\mathcal{F}_\beta$ when evaluating systems, such as its prospects to offer interpretability of scores and finer-grained system analyses. The second study probes potential weak spots of $\mathcal{M}\mathcal{F}_\beta$, e.g., its dependence on a strong MR parser.

6.3 Casting meaning and form into a metric: $\mathcal{M}\mathcal{F}_\beta$

While current NLG metrics lack *interpretability* and mainly focus on the form of generated text (Sai et al., 2020), in this work we emphasize the *meaning aspect* in NLG evaluation, which is most clearly dissociated from form when generating text from structured inputs such as MRs. At the same time, form and wording of the generated text cannot be ignored, as we want such systems to produce *natural and well-formed sentences*. Equipped with this two-fold objective, we start building our $\mathcal{M}\mathcal{F}_\beta$ score which aims at a *balanced combination* of both quality aspects: **meaning and form**.

6.3.1 From principles to $\mathcal{M}\mathcal{F}_\beta$

In a first step we introduce our

Principle of meaning \mathcal{M} . *Generated sentences should allow loss-less MR reconstruction.*

This principle expresses a key expectation for a system that generates NL sentences from abstract meaning representations. Namely, the generated sentence should reflect the meaning of the MR. So, in order to assess whether a generated sentence $s' = f^{-1}(m)$ is a valid generation for the input MR m , rather than matching s' against a reference sentence s , we perform this assessment **in the abstract MR domain**, by applying an inverse system f that *parses* the generated text back to an MR $m' = f(s') = f(f^{-1}(m))$. I.e., we desire a metric: $\mathcal{D} \times \mathcal{D} \rightarrow [0, 1]$ that satisfies: $s \equiv s' \iff \text{metric}(s, s') = 1 \iff m = m'$. Two texts are equivalent iff their meaning abstractions denote the same meaning.

In case $f(s')$ yields an MR $m' \neq m$, we can still determine the degree to which s' preserves the meaning of MR m by measuring the distance between m and m' with MR metrics, e.g., $MRmetric(m, m')$, such as SMATCH, S²MATCH or WWLK.

Note that computing $MRmetric(m, m')$ does not depend on a reference sentence, because the comparison is conducted purely in the abstract domain. This is mathematically

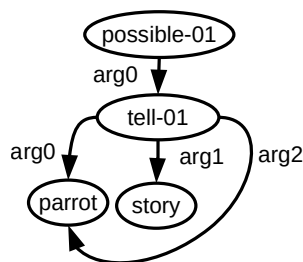


Figure 6.2: “Perhaps, the parrot is telling itself a story”.

more appealing for the evaluation of MR2text, since it solves the problem that one abstract representation may result in various (valid) surface realizations (cf. Appendix A.1). Finally, we also do not necessarily need to rely on a gold graph m , but can instead set $m = f(s)$, i.e., the parse of the reference sentence. This means that future application of \mathcal{M} to other kinds of text generation tasks is straightforward.

However, the principle \mathcal{M} alone is not sufficient: we also expect the system to generate grammatically well-formed and fluent text. For example, s' : *Possibly, it(self) tells parrot a story.* contains relevant content expressed in the AMR of Figure 6.2, but it is neither grammatically well-formed, nor a natural and fluent sentence. This leads us to our

Principle of form \mathcal{F} . *Generated sentences should be syntactically well-formed, natural and fluent.*

In the style of the well-established F_β score (Rijsbergen, 1979), we fuse these two principles into the $\mathcal{M}\mathcal{F}_\beta$ score:

$$\mathcal{M}\mathcal{F}_\beta = (1 + \beta^2) \frac{\text{Meaning} \times \text{Form}}{(\beta^2 \times \text{Meaning}) + \text{Form}} \quad (6.1)$$

Here, *Form* and *Meaning* are expressed as ratios that will be more closely described in the following section. β allows users to gauge the evaluation towards *Form* or *Meaning*, depending on specific application scenarios. Users may prefer the harmonic mean ($\beta = 1$) or may give *Meaning* double weight compared to *Form* (e.g., $\beta = .5$).³

In our experiments we consider extreme decompositions into *Meaning-only* ($\beta \rightarrow 0$) or *Form-only* ($\beta \rightarrow \infty$).

³Generally, *Form* receives β times as much importance compared with *Meaning*.

6.3.2 REMATCH: Measuring meaning with MR and MR metrics

We measure \mathcal{M} or *Meaning* (Meaning Preservation) with a score range in $[0, 1]$ by reconstructing/projecting the MR with an accurate AMR parser and computing an MR graph metric. We call this REMATCH (Reconstruction Match).

Given a generated sentence s' and source AMR m , we match $parse(s')$ against m by setting $REMATCH(s', m) := MRmetric(parse(s'), m)$. This means that we have to decide upon $parse$ and $MRmetric$. We propose two potential settings.

MR reconstruction. To reconstruct the MR with $parse$, we employ the parser of Cai and Lam (2020a) that achieves high SMATCH scores around 80 points on AMR benchmarks. We henceforth call it GSII. In our experiments we will also compare this parser against less recent parsers, indicating that selecting a strong parser has positive effects on the evaluation quality with $\mathcal{M}\mathcal{F}_\beta$ score, and so the accuracy of $\mathcal{M}\mathcal{F}_\beta$ may scale well with further future parsing improvements.⁴

Assessing \mathcal{M} with MR metrics. Besides having to select a parser for MR reconstruction, we have to select an MR metric. Now, having already studied and proposed MR metrics, we can deliberately pick a suitable metric: Since the scenario of MR-based text evaluation is still restricted in the sense that any two inputted structures strongly relate to the same/similar sentence/grounding, a structural metric like SMATCH seems sufficient and also provides very interpretable results, due to conforming to many of our principles, particularly PVII that ensures that the score is proportional to the amount of shared symbolic MR triples. However, due to potential noise in system outputs and different options to project similar abstract concepts (e.g., *location, place, ...*), we would also like to have a graded concept match, to help compensate for noise of minor lexical deviations from the original sentence all while keeping the score calculation most transparent. This parameterization can be achieved with S²MATCH (see Section 4.3.2), which we henceforth set as the default for our experiments on NLG evaluation with MR metrics. However, we will later also inspect how/if system rankings can be affected by usage of other MR metrics.

⁴Recall, however, that as we found out in Section 5.4, benchmarking of strong parsers is an open problem. But since the scores of the default parser in $\mathcal{M}\mathcal{F}_\beta$ are on par with human measured IAA (in SMATCH), we can confidently assume that a large proportion of projected meaning triples will be correct. Still, future work should strongly consider using the best parser available and determine this best parser through improved evaluation.

Discussion. Comparing to references by matching their meaning graphs has the prospect of offering interpretability and explanations, by detecting redundant or missing meaning components in the generations. In our studies, we will see that this assessment can be conducted by computing a *single graph overlap score* (e.g., $S^2\text{MATCH F1}$), or along *multiple dimensions of meaning*, such as SRL, coreference or WSD). Generally, $\mathcal{M}\mathcal{F}_\beta$ gives researchers the flexibility of choosing a *parser* or *mrMetric* to their liking. We chose a strong *parser* that achieves high IAA with humans, and an interpretable MR metric. Later, we will assess different parameterizations of $\mathcal{M}\mathcal{F}_\beta$ to assess potential weakspots of parsers and robustness to variation of MR metrics. Due to the explorative nature of this evaluation approach, we abstain from further parameter search for measuring \mathcal{M} while admitting that future work can probably find and determine more suitable parameterizations (e.g., also through development of stronger parsers or tailored MR similarity measurements).

6.3.3 Parameterizing form with LMs

Assessing sentence grammaticality and fluency is not an easy task (Heilman et al., 2014; Katinskaia and Ivanova, 2019). Recently, Lau et al. (2020) and Zhu and Bhat (2020) show that probability estimates based on language models can be used as an indicator for measuring complex notions of form and for measuring acceptability in context. For our $\mathcal{M}\mathcal{F}_\beta$ score we desire an interpretable ratio as input, which we base on LM predictions as follows.

Binary form assessment. Given a specific candidate generation s' , we use a binary variable to assess whether s' is of satisfactory form. For this, we first calculate the mean token probability:⁵

$$mtp(\cdot) = \frac{1}{n} \sum_{j=1}^n P(tok_j | ctx_j), \quad (6.2)$$

where ctx_j is different for uni-directional LMs ($ctx_j = tok_{1\dots j-1}$) and bi-directional LMs ($ctx_j = tok_{1\dots j-1, j+1\dots n}$). We compute mtp for the generated sentence s' and the reference s and calculate a preference score $prefScore = \frac{mtp(s')}{mtp(s') + mtp(s)}$. The decision of whether the *Form* of a generated sentence s' is acceptable is then calculated as

⁵We use the mean (instead of the product) because Bryant and Briscoe (2018) find that basing decisions on the mean works well in practice when assessing possible corrections of grammatical errors.

$$accept = \begin{cases} 1, & \text{if } prefScore \geq 0.5 - tol \\ 0, & \text{otherwise,} \end{cases}$$

where *tol* is a tolerance parameter. Less formally, a sentence is considered to have an acceptable surface form in relation to its reference if its form is estimated to be at least as good as the reference minus a tolerance, which we fix at 0.05. I.e., the corpus-level *Form* score reflects the ratio of generated sentences that are of acceptable form.⁶

Predictor selection. We consider GPT-2 (Radford et al., 2019), distil GPT-2 (Sanh et al., 2019), BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019b) as a basis for assessing *Form*. We conduct experiments on WebNLG (Gardent et al., 2017; Shimorina et al., 2017), which contains human fluency and grammaticality judgements for machine-generated sentences. We find that GPT-2 performs best: it discriminates sentences of poor and perfect fluency and grammaticality with an F1 score of approximately 0.8, and shows marginally better performance compared to the other LMs (see Appendix A.2 for the experiment details). We thus select GPT-2 as our LM for *Form* assessment.

Discussion. While the reconstruction of meaning does not depend on the reference sentence, we do make use of it, in *prefScore*, for better assessment of *Form*. One reason is that when assessing the form of a sentence s' that contains rare words, the ‘raw’ $mtp(s')$ may be too pessimistic and may not relate well to the quality of the form. Generally, the mtp (or any LM probability) itself is not well interpretable and hardly allows comparison to the mtp of other sentences (e.g., if they are about a different topic).

However, by relating the mtp of the generated sentence to the mtp of a (same-topic) reference, we gain three advantages: first, we do not, a-priori, penalize generations that contain rare words. Second, we obtain an interpretable corpus-level ratio (rate of sentences that are of acceptable form). This is important, since sound \mathcal{MF}_β calculation ideally requires two interpretable ratios as input. Third, by avoiding any string matching, we still keep form and meaning aspects clearly distinct.

⁶I.e., the *Form* score for a single sentence with $accept \geq 0.5 - tol$ equals 1.0. If a precise assessment for a single sentence is needed, we can fall back on *prefScore* (+/- *tol*).

6.3.4 Goals of our pilot studies

Our main aim is to establish, with the proposed $\mathcal{M}\mathcal{F}_\beta$ score for text generation, **i) a balanced and interpretable assessment** of generated text according to **Meaning and Form**. Yet, as detailed in Section 6.3.2 and Section 6.3.3, both components depend on a number of **ii) hyperparameters**, such as the parser applied for *Meaning* reconstruction, or the LM used for *Form* assessment. These parameters may also be subject to change over time. It is thus important to assess the effects of such factors on metric scores and system rankings. We investigate both aspects of $\mathcal{M}\mathcal{F}_\beta$ in **two pilot studies**.

In **the first study**, in Section 6.4, we aim to assess the prospects of $\mathcal{M}\mathcal{F}_\beta$ when ranking SOTA systems. We will see that $\mathcal{M}\mathcal{F}_\beta$ can explain system performance differences by disentangling *Form* and *Meaning*, an asset that no other metric can offer.

The second study, in Section 6.5, investigates the impact of $\mathcal{M}\mathcal{F}_\beta$'s dependence on a parser and a LM. We i) investigate the effects of using different parsers, ii) assess the potential suitability of $\mathcal{M}\mathcal{F}_\beta$ for other text generation tasks, by ablating the human gold graph from the evaluation and using $\mathcal{M}\mathcal{F}_\beta$ to evaluate generated text vs. reference text, and iii) validate the LM's binary predictions for *Form* in a manual annotation study.

6.4 Study I: Assessing potential for enhanced evaluation interpretability

Setup: data & metrics for system ranking. We obtain test predictions of several state-of-the-art *AMR2text generation systems* on LDC2017T10, the main benchmark for this task: (i) densely connected graph convolutional networks (Guo et al., 2019); (ii) Ribeiro et al. (2019)'s system that uses a dual graph representation; two concurrently published models (iii) based on graph transformers (Cai and Lam, 2020b; Wang et al., 2020b) and (iv) a model based on graph transformers that uses reconstruction information (Wang et al., 2020c) in a multi-task loss; finally, we obtain predictions of two system variants of Mager et al. (2020a) that fine-tune LMs and encode linearized graphs using (v) a large and (vi) a medium-sized LM. We true-case all sentences and parse them with GSII.

To put the results of $\mathcal{M}\mathcal{F}_\beta$ into perspective, we display the scores of several metrics that have been previously used for AMR2text: BLEU, METEOR, CHRF++. We also calculate BERTscore (Zhang et al., 2020) with RoBERTa-large (Liu et al., 2019b).⁷ Results

⁷BERTscore computes an F1-score over a *cosim*-based alignment of the contextual embeddings of paired sentences.

	popular NLG metrics				Meaning (MR metrics)					Form		$\mathcal{M}_{\mathcal{F}_1}$ Eq. 6.1	$\mathcal{M}_{\mathcal{F}_{0.5}}$ Eq. 6.1	
	BLEU	METEOR	chrF++	BERTsc.	SMATCH	WLK	WWLK	P	R	F1	%acc.			
<i>appUB</i>	na	na	na	na	79.9	73.8	85.8	83.1	80.1	81.5	100	89.8	84.6	
Ribeiro et al. (2019)	R'19	27.9 ⁽⁵⁾	33.2 ⁽⁷⁾	58.7 ⁽⁶⁾	92.7 ⁽⁴⁾	70.1 ⁽⁶⁾	66.2 ⁽⁶⁾	77.7 ⁽⁶⁾	76.5	67.7	71.9 ⁽⁶⁾	51.6 ⁽⁵⁾	60.1 ⁽⁵⁾	66.6 ⁽⁵⁾
Guo et al. (2019)	G'19	27.6 ⁽⁶⁾	33.7 ⁽⁶⁾	57.3 ⁽⁷⁾	92.4 ⁽⁷⁾	72.3 ⁽³⁾	66.5 ⁽⁵⁾	78.2 ⁽⁵⁾	78.2	70.0	73.9 ⁽³⁾	47.1 ⁽⁷⁾	57.5 ⁽⁷⁾	66.3 ⁽⁶⁾
Wang et al. (2020b)	Wb'20	27.3 ⁽⁷⁾	34.1 ⁽⁵⁾	59.3 ⁽⁵⁾	92.6 ⁽⁶⁾	70.0 ⁽⁷⁾	65.5 ⁽⁷⁾	74.8 ⁽⁷⁾	79.6	65.0	71.5 ⁽⁷⁾	49.5 ⁽⁶⁾	58.5 ⁽⁶⁾	65.7 ⁽⁷⁾
Cai and Lam (2020b)	C'20	29.8 ⁽⁴⁾	35.1 ⁽⁴⁾	59.4 ⁽⁴⁾	92.7 ⁽⁴⁾	71.7 ⁽⁵⁾	67.8 ⁽⁴⁾	79.4 ⁽⁴⁾	78.1	69.2	73.4 ⁽⁵⁾	51.9 ⁽⁴⁾	60.3 ⁽⁴⁾	67.0 ⁽⁴⁾
Mager et al. (2020a)-M	Mb'20	33.0 ⁽²⁾	37.3 ⁽²⁾	63.1 ⁽³⁾	93.9 ⁽²⁾	72.1 ⁽⁴⁾	68.3 ⁽³⁾	79.6 ⁽³⁾	79.5	68.7	73.7 ⁽⁴⁾	74.0 ⁽¹⁾	73.9 ⁽¹⁾	73.8 ⁽¹⁾
Mager et al. (2020a)-L	M'20	33.0 ⁽²⁾	37.7 ⁽¹⁾	63.9 ⁽²⁾	94.0 ⁽¹⁾	73.0 ⁽²⁾	69.1 ⁽²⁾	80.0 ⁽²⁾	80.8	69.2	74.5 ⁽²⁾	69.8 ⁽²⁾	72.1 ⁽²⁾	73.5 ⁽²⁾
Wang et al. (2020c)	W'20	33.9 ⁽¹⁾	37.1 ⁽³⁾	65.8 ⁽¹⁾	93.7 ⁽³⁾	73.8 ⁽¹⁾	70.0 ⁽¹⁾	81.0 ⁽¹⁾	80.3	70.9	75.3 ⁽¹⁾	55.7 ⁽³⁾	64.0 ⁽³⁾	70.3 ⁽³⁾

Table 6.1: Main metric results. *na* as upper-bound means that the upper-bound is not known and cannot be estimated. $\mathcal{M}_{\mathcal{F}_\beta}$ is calculated from *Form* and $S^2_{\text{MATCH}} F1$.

are displayed in Table 6.1, col. 3-6. $\mathcal{M}\mathcal{F}_\beta$ scores (col. 7-12) are divided into *Meaning* (REMATCH using GSII) and *Form* scores (based on GPT-2), and composite $\mathcal{M}\mathcal{F}_\beta$ scores with $\beta = 1$ (harmonic) and $\beta = 0.5$ (double weight on \mathcal{M}).

As an upper-bound approximation for REMATCH we propose parsing a gold sentence s and comparing the result against the gold MR m : $apprUB = \text{metric}(\text{parse}(s), m)$.⁸

6.4.1 Interpretability of system rankings

Surface matching metrics lack differentiation and interpretability. Table 6.1 shows that the baseline metrics tend to agree with each other on the ranking of systems, but there are also differences, for example, BERTscore and METEOR select M’20 as the best performing system while BLEU and CHRFF++ select W’20. While certain differences may be due to individual metric properties, e.g., METEOR allowing inexact word matching of synonyms, the underlying factors are difficult to assess, since the score differences between systems with switched ranks are small, and none of these metrics can provide us with a meaningful interpretation of their score that would extend beyond shallow surface statistics. Hence, these metrics cannot give us much intuition about why and when one system may be preferable over another.

Meaning vs. Form: How $\mathcal{M}\mathcal{F}_\beta$ explains system performance. We have seen that current metrics cannot provide us with convincing explanations as to why, e.g., W’20 should be preferred over M’20 (BLEU), or M’20 over W’20 (BERTscore). REMATCH metrics and $\mathcal{M}\mathcal{F}_\beta$ score, however, tell a story about how these systems differ, highlighting their complementary strengths by disentangling *Meaning* and *Form* (Bender and Koller, 2020): W’20 displays the highest REMATCH score, i.e., MRs constructed from its generations recover a maximum of the meaning contained in the input MR. M’20, by contrast, outperforms all systems in *Form* score. Looking at $\mathcal{M}\mathcal{F}_1$, the harmonic mean of both, both systems still occupy leading ranks, but W’20 falls back to 3rd rank, due to its weaker *Form* score.

Hence, given our metric principles, a user who cares about faithfulness to meaning, but less about fluency, should select W’20 (with consistently higher REMATCH over all

⁸This is the score of canonical parser evaluation. I.e., we would not expect the reconstruction m' of s' to score higher than had we applied *parse* to the original sentence: $\text{metric}(m', m) \leq \text{metric}(\text{parse}(s), m) = \text{apprUB}$. This is an idealization, as we can imagine cases where the original sentence s is more complex and thus more difficult to parse to an MR than a simpler generated paraphrase s' . Since we are interested in a very rough upper bound estimation, we abstract from such cases in our present work.

our MR metrics compared to M'20 by about $\Delta=1$ point) – a user who desires a system that preserves meaning well but also produces sentences of decent form, should select M(b)'20 (with $\mathcal{M}\mathcal{F}_{0.5}$ and $\mathcal{M}\mathcal{F}_1$ score differences against W'20 of $\Delta=3.5$ points and $\Delta=8$ points). Overall, $\mathcal{M}\mathcal{F}_\beta$ mostly agrees with BERTscore in the rankings of the teams, probably due to the additional regulation through *Form*, suggesting that BERTscore is quite *Form*-oriented. Indeed, $\mathcal{M}\mathcal{F}_\beta$'s larger score differences between the systems, due to *Form*, are striking, prompting us to investigate the *Form* predictions in closer detail (Section 6.5.2). We will see that using a different *Form* predictor as well as a manual native speaker annotation support our assessment of *Form*.

6.4.2 MR distance via REMATCH explains (re-)rankings

-----original sent-----		
Costa added that insurgents have been holding significant amounts of opium .		
-----original AMR-----		
<pre> (a / add-01 :arg0 (p / person :name (n / name :opl "Costa")) :arg1 (h / hold-01 :arg0 (i / insurgent) :arg1 (o / opium :quant (a2 / amount :arg1-of (s / significant-02)))) </pre>		
-----Candidate A-----Candidate B-----		
Costa added the insurgents to hold a significant amounts of opium .	>>	Costa added that the insurgents have held a significant amount of opium .
-----BLEU score-----		
37.7	>>	22.6
-----Reconstructions-----		
<pre> (c0 / add-02 (c0 / add-01 :arg0 (c2 / person :arg0 (c2 / person :name (c4 / name :name (c5 / name :opl "Costa") :opl "Costa")) :arg1 (c1 / insurgent) :arg1 (c1 / hold-01 :arg2 (c3 / hold-01 :arg0 (c4 / insurgent) :arg0 c1 :arg1 (c3 / opium :arg1 (c5 / opium :quant (c6 / amount :quant (c6 / amount :arg1-of (s / sign.-02 :arg1-of (s / sign.-02 :arg1-of (s / sign.-02)))))))) </pre>		
-----REMATCH scores-----		
S2match:	82.9 <<	100.00
Smatch:	81.1 <<	100.00
WLK:	77.2 <<	100.00
WWLK:	97.9 <<	100.00

Figure 6.3: Explainable *Meaning* score (re-)ranking.

An example for how REMATCH explains a re-ranking (different from BLEU) is shown in Figure 6.3. Here, the gold reference (both sentence and AMR) indicates that a person named Costa *adds* (as a communicative act⁹ that some insurgents have been holding large amounts of opium. However, system generation A (which is higher ranked by BLEU) chooses a different sense of *add*, *add-02*, which represents the action as an operation¹⁰, which results in an incoherent or nonsensical meaning representation where the person Costa *adds* (in the operational sense) the insurgent (as thing being added) to a circumstance to the effect that the insurgents hold a significant amount of opium. By contrast, system generation B preserves more of the gold MR’s meaning and clearly expresses that Costa performs an *act of communication* when he *adds* something. REMATCH ($\mathcal{M} \mathcal{F}_{\beta \rightarrow 0}$) is able to detect the meaning differences and assigns candidate B a significantly higher score than A, in fact, an S²MATCH score of 1.00 (since the graphs are structurally identical, the same score is achieved for other MR metrics).

6.4.3 MR distance via REMATCH explains negation error

```

-----original sent-----
      Since there is responsibility, we are not afraid.
-----original AMR-----
      (c / cause-01
       :arg0 (r / responsible-02)
       :arg1 (f / fear-01
              :polarity -
              :arg0 (w / we)))
-----Candidate 1-----Candidate 2-----
We are not responsible           We are not afraid
because we fear .                for responsibility .
-----pA=f(A)-----Reconstructions-----pB=f(B)-----
(c1 / cause-01                   (c1 / fear-01
 :arg0 (c5 / fear-01)             :arg0 (c5 / we)
 :arg1 (c4 / responsible-01)      :arg1 (c4 / responsible-03
 :arg0 (c10 / we)                 :arg0 c5)
 :polarity - )                    :polarity - )
-----ReMatch Negation Subgraph Scores-----
S2match:      0.00      <<      100.00
Smatch:       0.00      <<      100.00
WLK:          26.9      <<      100.00
WWLK:         42.0      <<      100.00

```

Figure 6.4: Explained negation confusion.

In Figure 6.4, both systems struggle to fully capture the meaning of the original MR $f(s)$. However, the system based on GPT medium (Mb’20) erroneously assesses that *we are not responsible* and *we fear*. However, quite the opposite is true: the gold graph and gold sentence states that *there is responsibility* and *there is no fear*. This important facet

⁹Sense *add-01* w/ roles: Arg0: *Speaker*; Arg1: *Utterance*.

¹⁰Sense *add-02* w/role set: Arg0: *adder*; Arg1: *thing being added*; Arg2: *thing being added to*; Arg3: *resulting sum*.

of meaning is better captured by C'20. The reconstruction shows that it reflects the gold negated concepts much better and does not distort facts that are core to the meaning. In consequence, the REMATCH scores are low for the left sentence with the distorted facts and maximum (for all MR metrics) for the sentence that sticks true to the facts.

6.4.4 MR distance via REMATCH explains SRL error

Figure 6.5 shows an example where REMATCH ranks two generated candidate sentences differently compared to BLEU. In this case, the gold sentence and the gold AMR both express that there is some soldier who tried to defuse a bomb and got injured in the process. Clearly, candidate generation A captures the meaning better, in fact, it captures it almost perfectly. However, since the surface text deviates from the gold sentence, BLEU overly penalizes this generation and assigns a very low score of 10.6 points. In contrast, candidate B matches the surface slightly better (12.2 points), but distorts the meaning: it does not contain any information about the soldier and states that *Disarming was injured*, which is grammatically correct, but semantically wrong, or even non-sense.

We see that the surface matching metric cannot explain its scores (beyond superficial statistics) and delivers a ranking that does not appropriately reflect the performance of the generation systems. However, REMATCH shows that the gold parse and the parse of candidate A agree with each other in the central *ARG1*-role of the main predicate *injure-01*: *it is the soldier who got injured*. On the other hand, in the reconstruction of the AMR of candidate B, the *ARG1* argument is filled differently: *it is the disarmament that gets injured*.

This assessment allows REMATCH to increment the score for generation A by a large margin, from 10.6 (BLEU) to 93.3 points (REMATCH, S²MATCH), expressing substantial agreement in meaning with the gold. The score for the candidate generation B also gets incremented – but it gets incremented much less, only to 70.2 points, expressing good to mediocre agreement. Interestingly, the difference is further dilated by WWLK (98 vs. 50 points), underlining the semantic problems in the right sentence/MR. Thus, by detecting the SRL confusion, REMATCH re-ranks the candidate generation such that the resulting ranking is more appropriate.

6.4.5 Assessing aspectual text quality using fine-grained MR distances

We apply SMATCH on aspectual subgraphs, as outlined in the bottom of Figure 3.1 in Chapter 3 by running aspectual subgraph-based evaluation with the goal to gain deeper

-----original sent-----	
Soldier injured during bomb defusion in Kathmandu after state of emergency expires .	
-----original AMR-----	
(i / injure-01 :arg0 (d / defuse-01 :arg1 (b / bomb) :location "Kathmandu") :arg1 (s / soldier) :time (a / after :op1 (e / expire-01 :arg1 (s2 / state :mod (e2 / emergency))))	
Candidate 1	Candidate 2
The Soldier was injured in the defuse of the bomb in Kathmandu after the emergency state expired .	Disarming the bomb in Kathmandu was injured in Kathmandu after state of emergency expires .
-----BLEU score-----	
score (A, s) = 10.6	score (B, s) = 12.2
-----Reconstructions-----	
(c0 / injure-01 :arg1 (c1 / soldier) :arg2 (c2 / defuse-01 :arg1 (c4 / bomb) :location "Kathmandu") :time (c3 / after :op1 (c6 / expire-01 :arg1 (c8 / state :mod (c9 / emergency))))	(c0 / injure-01 :arg1 (c1 / disarm-01 :arg1 (c4 / bomb) :location "Kathmandu" :time (c2 / after) :op1 (c5 / decline-02 :arg1 (c7 / state-01 :location c3 :mod (c8 / emergency))))
-----ReMatch SRL Subgraph Scores-----	
S2match:	93.3 >> 70.2
Smatch:	91.7 >> 66.7
WLK:	84.4 >> 34.4
WWLK:	98.9 >> 50.5

Figure 6.5: Explained SRL confusion.

insight into **how well system generations reflect or violate specific meaning aspects**. With this, we can investigate a system’s capacity to properly reflect negation (NEG); to generate correct surface forms for NEs (NER); assess how well a system captures coreference between entities (Coref); and whether or not the predicate-argument structures (SRL) of generated sentences appropriately reflect the source meaning. The results are shown in Table 6.2.

In sum, the system of W’20 appears to be the clear winner in most aspects of meaning. This is intuitive, since the system has been trained with an auxiliary signal that provides information on how well an AMR can be reconstructed from the generated sentence.

Furthermore, we observe, e.g., that R’19, which ranks last in the overall ranking, improves upon the best overall system by 3.4 points in NER recall and 1.9 points in F1. The analysis also corroborates that W’20 excels among competitors with best scores for coreference, SRL and negation, i.e., the more global aspects of sentence meaning. Such information can be valuable for researchers for deeper system analysis and for practitioners aiming for specific use cases.

	Reentrancies			SRL			negation			NER		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
<i>apprUB</i>	72.1	60.7	65.9	77.7	73.5	75.5	88.6	70.5	78.5	82.2	80.1	81.1
R'19	63.7	50.3	56.2	71.1	62.4	66.4	72.1	50.6	59.5	82.2	70.7	76.0
G'19	66.9	52.9	59.1	73.7	64.9	69.0	75.0	51.5	61.1	78.6	68.9	73.5
Wb'20	67.6	51.5	58.4	75.1	63.6	68.9	74.3	49.7	59.6	86.5	60.3	71.0
C'20	66.1	52.4	58.4	73.4	64.8	68.8	78.3	54.2	64.1	80.8	67.2	73.4
Mb'20	65.9	53.2	58.9	74.3	65.7	69.8	70.6	45.5	55.3	82.6	69.4	75.4
M'20	67.9	53.3	59.7	76.4	66.5	71.1	73.7	53.9	62.3	82.8	68.3	74.9
W'20	68.8	55.7	61.6	76.1	68.1	71.9	79.2	55.1	65.0	82.4	67.3	74.1

Table 6.2: Fine-grained corpus results using $\mathcal{M}\mathcal{F}_0$ (i.e., $S^2\text{MATCH}$) parameterized based on aspectual subgraphs.

6.5 Study II: Probing vulnerabilities of our approach

$\mathcal{M}\mathcal{F}_\beta$ has two apparent vulnerabilities: first, it depends on a parser for reconstruction. We have used a SOTA parser that is on par with human IAA. Yet, we cannot exclude the possibility that it introduces unwanted errors in computing $\mathcal{M}\mathcal{F}_\beta$ scores.

Second, the *Form* component is based on a LM and we have seen that it can change system rankings, even when it is discounted.¹¹ On the one hand, our LM was carefully selected, and other metrics such as BERTscore also heavily depend on LMs. On the other hand, we cannot exclude the possibility that the changed rankings are unjustified.

Our next studies investigate these weak spots more closely. First, in Section 6.5.1, we assess the outcome of $\mathcal{M}\mathcal{F}_\beta$ when using another parser and assess its potential portability to other text generation tasks by ablating the human gold graph and evaluate generated text against reference *text*. In Section 6.5.2 we conduct a human annotation study to assess whether the provided *Form* rankings are justified.

6.5.1 The parser: Achilles' heel of $\mathcal{M}\mathcal{F}_\beta$?

Using another parser. In this experiment we assess the robustness of REMATCH against using different parsers. This is important, since the metric and rankings could change with the parser. Here, we would hope that the difference of using one competitive parser over another will not be too extreme, especially with regard to system rankings. To investigate this issue, we apply two alternative parsers: i) GPLA (Lyu and Titov, 2018), a neural

¹¹In Table 6.1, both $\mathcal{M}\mathcal{F}_\beta$ with $\beta = 0.5$ and $\beta = 1.0$ slightly disagree with the ranks assigned by *Meaning* only.

	REMATCH F1				ranks REMATCH				ranks $\mathcal{M}\mathcal{F}_{0.5}$			
	TTSA	GPLA	GSII	GSII \blacklozenge	TTSA	GPLA	GSII	GSII \blacklozenge	TTSA	GPLA	GSII	GSII \blacklozenge
<i>apprUB</i>	73.7	76.2	81.5	86.4	0	0	0	0	0	0	0	0
R'19	66.9	70.1	71.9	72.3	7	7	6	6	5	5	5	5
G'19	69.7	72.2	73.9	73.7	3	3	3	4	6	6	6	6
Wb'20	67.3	70.2	71.5	71.6	6	6	7	7	7	7	7	7
C'20	69.1	70.4	72.2	73.4	4	5	5	5	4	4	4	4
Mb'20	68.9	70.5	73.7	74.2	5	4	4	3	1	2	1	1
M'20	69.8	72.5	74.5	75.1	2	2	2	2	2	1	2	2
W'20	70.5	73.1	75.3	75.4	1	1	1	1	3	3	3	3

Table 6.3: Analysis of our metric using different parsers (GPLA, TTSA GSII) or ablating the gold parse by comparing the parsed generation against the parse (distant) source sentence (GSII \blacklozenge).

graph-prediction system that jointly predicts latent alignments, concepts and relations, and ii) TTSA (Groschwitz et al., 2018), a neural transition-based parser that converts dependency trees to AMR graphs using a typed semantic algebra. We select GPLA and TTSA since they constitute technically quite distinct approaches compared to GSII.

The results are shown in Table 6.3 (columns labelled GPLA, TTSA and GSII). All variants tend to agree in the majority of their rankings¹² (e.g., $\text{REMATCH}^{\text{GPLA}}$ vs. $\text{REMATCH}^{\text{GSII}}$ F1: Spearman’s $\rho = 0.95$, Pearson’s $\rho = 0.96$, $p < 0.001$). When considering $\mathcal{M}\mathcal{F}_{\beta=0.5}$, the agreement further increases (e.g., $\mathcal{M}\mathcal{F}_{0.5}^{\text{GPLA}}$ vs. $\mathcal{M}\mathcal{F}_{0.5}^{\text{GSII}}$: Spearman’s $\rho = 0.95$, Pearson’s $\rho = 0.99$, $p < 0.001$).

However, while using TTSA or GPLA instead of GSII has little effect on the ranks, the absolute scores can differ (e.g., W'20 70.5 F1 w/ TTSA, 73.1 F1 w/ GPLA and 75.3 F1 w/ GSII). Yet, we find that none of the generation systems are unfairly treated by our main parser GSII since we observe (mostly uniform) increments from TTSA to GPLA and from GPLA to GSII. An unfair treatment could arise, e.g., if GSII generates bad AMR reconstructions for specific NLG systems but not so for others. However, we do not observe such tendencies. Hence we can conclude that GSII’s score increments stem from the fact that GSII yields better reconstructions for all systems.

¹²We observe one switch of ranks for TTSA-GPLA and GPLA-GSII and 2 rank switches for TTSA-GSII in REMATCH, and no rank switch for TTSA-GSII and one switch for TTSA-GPLA and GPLA-GSII, for $\mathcal{M}\mathcal{F}_{0.5}$.

	MR metric result				ranks $\mathcal{M}\mathcal{F}_1$				ranks $\mathcal{M}\mathcal{F}_{0.5}$			
	SM	WLK	WWLK	S ² M	SM	WLK	WWLK	S ² M	SM	WLK	WWLK	S ² M
<i>apprUB</i>	79.9	73.8	85.8	81.5	0	0	0	0	0	0	0	0
R'19	70.1 ₍₆₎	66.2 ₍₆₎	77.7 ₍₆₎	71.9 ₍₆₎	5	5	5	5	5	5	5	5
G'19	72.3 ₍₃₎	66.5 ₍₅₎	78.2 ₍₅₎	73.9 ₍₃₎	7	7	7	7	6	7	6	6
Wb'20	70.0 ₍₇₎	65.5 ₍₇₎	74.8 ₍₇₎	71.5 ₍₇₎	6	6	6	6	7	6	7	7
C'20	71.7 ₍₅₎	67.8 ₍₄₎	79.4 ₍₄₎	73.4 ₍₅₎	4	4	4	4	4	4	4	4
Mb'20	72.1 ₍₄₎	68.3 ₍₃₎	79.6 ₍₃₎	73.7 ₍₄₎	1	1	1	1	1	1	1	1
M'20	73.0 ₍₂₎	69.1 ₍₂₎	80.0 ₍₂₎	74.5 ₍₂₎	2	2	2	2	2	2	2	2
W'20	73.8 ₍₁₎	70.0 ₍₁₎	81.0 ₍₁₎	75.3 ₍₁₎	3	3	3	3	3	3	3	3

Table 6.4: Studying $\mathcal{M}\mathcal{F}_\beta$ ranking under variation of MR metric (parser: GSII).

$\mathcal{M}\mathcal{F}_\beta$ rankings under different MR metrics are displayed in Table 6.4. We see that, with one exception¹³, using different MR metrics does not lead to different $\mathcal{M}\mathcal{F}_\beta$ rankings of the examined systems, both for $\beta = 0.5$ (meaning-focused) and $\beta = 1$ (meaning-form balance).

Ablating the gold graph? Yes, we can. In lack of a gold standard for the automatic reconstructions, we elicit some indirect answers and insight about the parser’s quality, by considering the following question: *What is the effect on system rankings when we replace the input gold graphs with automatic parses of the distant source sentence?* If this effect is large, this will give us reasons to worry, as it would indicate that the parser is less reliable than expected given its high IAA with humans. On the other hand, if we only see a minor effect, this may increase the trust in our parser and indicate that $\mathcal{M}\mathcal{F}_\beta$ could be confidently applied for explainable evaluation in other generation tasks (such as MT or summarization), where we do not have gold AMRs, and would have to parse both generated and reference sentences.

The results of this experiment are displayed in Table 6.3: our standard setup is displayed in columns labeled GSII and the results of the setup where we replace the gold input graph with an automatic parse is indicated by GSII[♦]. When considering REMATCH scores, we see only one switched rank between Mb'20 and G'20 (3–4). However, note that the absolute F1 score Δ between these two systems is overall very small (GSII: 0.2; GSII[♦]: 0.5). Overall, the scores do not tend to differ much when the gold graph is ablated, we observe rather small (mostly positive) changes in system scores (GSII \rightarrow GSII[♦]): 0.1 / 1.2 / 0.4 (min/max/avg). In sum, we conclude from this experiment that ablating the gold graph does not have a major effect on the scores and rankings. And when considering the $\mathcal{M}\mathcal{F}_{\beta=0.5}$ score, the ranking stays fully stable (the same holds true for $\mathcal{M}\mathcal{F}_{\beta=1}$).

¹³There is one case where ranks 7 and 6 are switched (WLK and $\beta = 0.5$).

Discussion. We have shown that metric rankings are fairly robust to using different parsers and that we do not necessarily depend on gold AMR graphs to compute the measure. This offers prospects for **using $\mathcal{M}\mathcal{F}_\beta$ for an explainable assessment of systems that perform other kinds of text generation.** In order to measure \mathcal{M} , a parser could be applied to both the generated and the reference text, to measure their agreement in the domain of abstract meaning representation. This would in turn offer means for conducting fine-grained meaning analysis of generation tasks where the reference is a natural language sentence (e.g., in MT).

Recall, however, that AMR, as of now, does not capture some facets of meaning that may be of interest in some generation tasks. For instance, it does not capture tense or aspect. However, what we have investigated as a *potential weakness* of $\mathcal{M}\mathcal{F}_\beta$, namely the necessity to select a meaning parser, can also be viewed as a *potential strength*. E.g., Donatelli et al. (2018) show how tense and aspect can be captured with AMR. This indicates that $\mathcal{M}\mathcal{F}_\beta$ can indeed be used for a tense and aspect analysis of generated text – if we parameterize it with a dedicated parser. Finally, if output and reference do not consist of single sentences, it may be apt to use a parser that constructs MRs for discourse (e.g., DRS (Kamp, 1981)).

In summary, we conclude that $\mathcal{M}\mathcal{F}_\beta$, our proposed metric that aims to assess text generation quality by decomposing it into *form* and *meaning* aspects, is broadly applicable. However, different parser parametrizations may have to be considered in light of the specific nature of a generation task.

6.5.2 The *Form* component of $\mathcal{M}\mathcal{F}_\beta$

In Section 6.4.1, we have seen that the *Form* aspect of $\mathcal{M}\mathcal{F}_\beta$ can change system ranks. Notably, it has promoted M’20 as the best generation system, outranking W’20 (in agreement with BERTscore), whereas W’20 is selected by BLEU or REMATCH. Now, we aim to investigate whether these impactful decisions of the *Form* component were justified.

Human annotation. We ask a native speaker of English to rate 50 paired generations of M’20 and W’20, considering only grammaticality and fluency.¹⁴

Annotator and annotation. The English native speaker (UK) annotated 50 paired sentences of M’20 and W’20. They were presented in shuffled order and the annotator was

¹⁴The annotator was explicitly instructed not to consider whether a sentence ‘makes sense’, by presenting the *Green ideas sleep furiously* example as free from structural error.

Sys (W'20): He also said that our athletes do n't very use
of competition under strong sunlight .
Corr (human): He also said that our athletes are not very used
to competition under strong sunlight .
----> not acceptable

Sys (W'20): Sheng Chen , the 6 th position of Hubei province , who
was totally scored 342.60 at 342.60 points this year ,
is a temporary position .
Corr (human): Sheng Chen , the 6 th position of Hubei province , who
has totally scored 342.60 points this year ,
is in a temporary position .
----> not acceptable

Sys (W'20): The Chinese competitors are Lan Wei and Sheng Chen ,
qualify semi - final .
Corr (human): The Chinese competitor Lan Wei and Sheng Chen qualify
for the semi - final .
----> acceptable

Sys (M'20): Fengzhu Xu won many championships in international
competition before .
Corr (human): Fengzhu Xu won many championships in international
competitions before .
----> acceptable

Figure 6.6: Sentences of flawed form. --> refers to the binary acceptability judgment (Eq. 6.3.3).

tasked with assigning a nominal number, starting from zero, that indicates the amount of grammatical or fluency issues as assessed by the native speaker. Additionally, the human was asked to provide a correction. Examples of sentences of flawed form are shown in Figure 6.6.

Results of human evaluation. The annotator agreed in 42 of 50 pairs with the preference predicted by GPT-2 (a significant result: binomial test $p < 0.000001$). We find that the M'20 and Mb'20 generations are considerably better on the surface level, compared to generations of all other systems. For instance, the best system according to *Meaning*, W'20, frequently produces inflection mishaps: *Their hopes for entering the heat is already in-sight*, while we find few such violations with M'20 (here: *Their hopes for entering the heat are already in sight*). We also find errors with adverbials, e.g., W'20 writes *They are the most indoor training at home*, while M'20 writes *They are most trained indoors at home*. Arguably both sentences are not perfect but the second is substantially more well-formed.

	R'19	G'20	Wb'20	C'20	Mb'20	M'20	W'20
GPT-2	51.6 ₍₄₎	47.1 ₍₆₎	49.5 ₍₅₎	51.9 ₍₄₎	74.0 ₍₁₎	69.8 ₍₂₎	55.7 ₍₃₎
BERT	43.4 ₍₆₎	40.6 ₍₇₎	50.4 ₍₄₎	44.7 ₍₅₎	71.4 ₍₁₎	71.0 ₍₂₎	55.9 ₍₃₎

Table 6.5: *Form* scores when using a different LM.

Using a different LM. The human study indicates that GPT-2 is accurate to 84% when favoring one sentence over the other, with respect to fluency and grammaticality. However, when considering that there is a trend to building systems based on fine-tuned LMs, we need to assess whether they may be favored (too) much if *Form* is parameterized with a same or a highly similar LM to the one used by the NLG model. We find such a case in M'20: while it was not fine-tuned with the same GPT-2 that we used for *Form* assessment, they fine-tuned their model with its siblings GPT-2-medium and GPT-large, which may share structural similarities. Therefore, we also use BERT for *Form* assessment. The results in Table 6.5 support the conclusion from the human annotation: by large margins, both M'20 and Mb'20 deliver generations that are of significantly improved form and both agree on the group of the three best systems. Note that this insight can be provided by \mathcal{MF}_∞ , but it cannot be carved out by conventional metrics, since these do not disentangle *Form* and *Meaning*.

6.6 Discussion

To showcase the usefulness and assess perspectives of MR metrics for NLG evaluation, we introduced and explored the \mathcal{MF}_β score, a new metric tailored to evaluation of text generation from MRs, but also extensible to other generation tasks. \mathcal{MF}_β measures two natural objectives of text generation: *Form* measures fluency of the produced sentences and *Meaning* assesses to what extent the meaning of the input MR is reflected in the produced sentence. We show that \mathcal{MF}_β has the potential to yield a fine-grained performance assessment that go beyond what conventional metrics can provide. Using its β -parameter, \mathcal{MF}_β can be decomposed into complementary views – *Meaning* and *Form* – paving the way for custom gauging and selection of NLG systems. We have seen that \mathcal{MF}_β corresponds well to BERTscore when rankings systems, but overcomes its opaqueness by disentangling *Meaning*- and *Form*-related quality aspects. In sharp contrast to BERTscore, the *Form* component of \mathcal{MF}_β dispenses with string matching against reference sentences, offering an assessment independent of lexical alignment.

An important hyperparameter of our metric is the required MR parsing component for meaning reconstruction. We investigate the impact of its choice by choosing alternative high-performing parsers. Our study shows that absolute metric scores tend to increment when using a better parser, while system rankings are quite stable. Furthermore, we outline the potential of $\mathcal{M}\mathcal{F}_\beta$ to extend to further text generation tasks, by ablating the human gold graph from the evaluation, such that the metric score can be computed from candidate and reference text alone. Since benchmarking of systems needs deeper exploration, we recommend $\mathcal{M}\mathcal{F}_\beta$ score to obtain better diagnostics and explainability of text generation systems, including, but not limited to (A)MR2text.

Another interesting aspect that we have not explored in our pilot studies is the agreement to human ratings of system quality. Interestingly, recent work (Manning and Schneider, 2021) suggests that – with accurate (gold) parses available – a simple MR metric based measurement can provide better agreement to human ratings than highly parameterized black-box models such as BERTscore. Here, $\mathcal{M}\mathcal{F}_\beta$ offers a chance to investigate this direction further by studying more parameterizations that optimize for human agreement in NLG evaluation: How to best weight *Form* and *Meaning*? What MR metric (if available also: its hyper-parameters) to choose for measuring *Meaning*? Are there better ways for scoring *Form*?

Chapter 7

AMR quality estimation

7.1 Chapter outline

In the chapter before, we observed: for NLG evaluation of MR-to-text systems, at least one MR is hidden and needs to be projected. In this chapter, we investigate MR metric projection from an inverse, but also complementary viewpoint: the *efficient* evaluation of automatically generated MRs from text (i.e., MR parsing). Here, we are given a natural language sentence and an MR candidate parse that has been constructed by a system. In such a scenario, we want to apply a *quality estimation system* that informs us how well the candidate parse fits to the input sentence, without using the costly reference parse that is available in standard parsing evaluation. With such a quality estimation system that predicts/extrapolates parse(r) quality, we could quickly assess the performance of MR parsers on new data, or efficiently filter among different candidate MRs provided by multiple sources.

For simplicity, and because the parsing evaluation scenario is restricted (same sentences are underlying the candidate and reference), and because now not the metric is in direct focus but its extrapolation through machine learning, we will constrain ourselves in this chapter to the extrapolation of the maximally transparent structural SMATCH metric. The remainder of this chapter is structured as follows:

1. We provide a more detailed motivation in Section 7.2 and provide formal definition of the problem in Section 7.3.
2. We propose two neural graph encoding strategies that can be trained for cheaply projecting an MR metric: A structure-enriched LSTM that processes the graph as a serialized string with alignment information (Section 7.4); And a CNN that

```

(p4 / possible-01
  :arg1 (d5 / destabilize-01
    :-arg0 [:arg1] (c3 / country
      :quant (w2 / whole)))
  :condition (e1 / economy
    [:poss c3]
    :arg0-of (f0 / function-01
      [:pol -] )))

```

Figure 7.1: Parse of *Without a functioning economy, the whole country may destabilize* with errors outlined.

is inspired by simplicity and human annotator view, exploiting the structured and concise multi-line ‘Penman’ graph string serialization (Section 7.5).

3. We show how both graph encoding strategies can be used to predict MR quality along several semantic axes (Section 7.6). For evaluation, we construct training and testing data (Section 7.7) and test our proposed systems against baselines in Section 7.8, ablating various system parts.
4. We conclude the chapter with a discussion (Section 7.9).

Underlying work. The content of this chapter is mainly based on works from Opitz and Frank (2019b) and Opitz (2020).

7.2 Motivation: rating MR quality in the absence of human reference

While automatically generated MRs are leveraged to enhance a variety of natural language understanding tasks, there is a critical issue with automatically generated MRs (parses): they are often deficient. These deficiencies can be quite severe, even when newer parsers are used. For example, in Figure 7.1, a neural parser (Lyu and Titov, 2018) makes several errors when parsing *Without a functioning economy the whole country may destabilize*. E.g., it misses a negative polarity and classifies a patient argument as the agent by failing to see that *destabilize* here functions as an ergative verb (parser: the country is the causer

of destabilize; correct: the country is the object that is destabilized). In sum, the parse has misrepresented the sentence’s meaning.¹

However, assessing such deficiencies via comparison against a gold reference (as in classical parser evaluation) is often infeasible in practice: it takes a trained annotator and appr. 10 minutes to manually create one AMR (Banarescu et al., 2013). To mitigate this issue, we would like to be able to automatically rate the quality of MRs without the costly gold graphs. This would allow us to signal downstream task systems the incorporated graphs’ trustworthiness or select among different candidate graphs from different parsing systems.

Generally, as outlined in the related work (Chapter 3), the problem of measuring the quality of structured predictions in the absence of a costly gold reference is not unknown to the NLP community. In fact, it is a highly relevant task in several sub-communities, including *machine translation*. So by exploring strategies to predict an MR metric measurement in the absence of a gold reference and only a candidate input MR available, we make first steps to assess and address the quality estimation problem for *semantic parsers*.

7.3 Task formalization

We aim at rating the quality of MR graphs (‘parses’) in the absence of gold graphs. This problem boils down to answering the following question: *how well does a candidate MR graph capture a given natural language sentence?* Therefore, the exact goal in this task is to infer a mapping

$$f : X = S \times G \rightarrow \mathbb{R}^d, \quad (7.1)$$

that maps a sentence $s \in S$ together with a candidate MR graph $g \in G$ onto d scores, which describe the MR with regard to d quality dimensions of interest. A successful mapping function should strongly correlate with the gold scores as they would emerge from evaluation against gold graphs.

7.4 Model I: LSTM on enriched linearized graphs

We propose a neural hierarchical multi-output regression model (**LG-LSTM**) for quality estimation of MR parses. Its architecture is outlined in Figure 7.2.

¹?With a functioning economy, the whole country may cause something to destabilize.

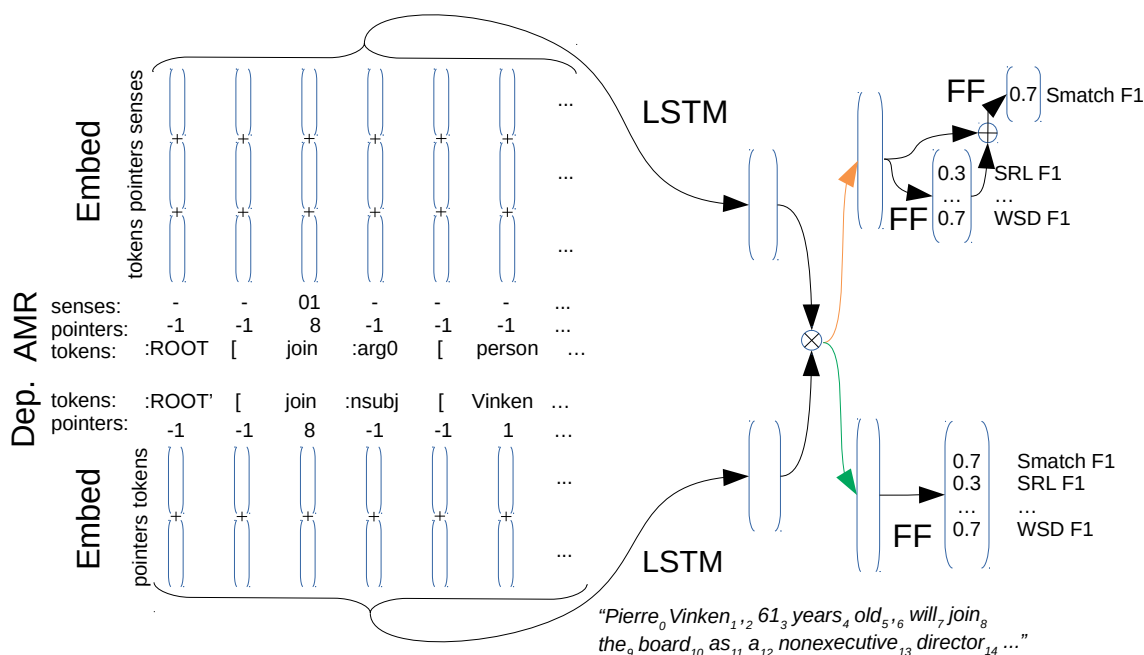


Figure 7.2: Our model I. green: Evaluation metrics computed in a non-hierarchical fashion. orange: Main evaluation metric is computed on top of secondary metrics. FF: basic feed-forward layer.

Inputs. Our model takes the following inputs: (i) a linearized MR and a linearized dependency graph (implementation details in Section 7.8). The motivation for feeding the dependency parse instead of the original sentence is due to the moderate similarity of dependency and MR structures. In addition, (ii) we produce alignments between sentence tokens and tokens in the sequential MR structure, as well as between sentence tokens and the linearized dependency structure, and feed these sequences of pointers to our quality estimation model. The intuition of using pointers is to provide the model with richer information via shallow alignment between MR, dependencies and the sequence of sentence tokens (see Section Section 7.8 for implementation details). Finally, (iii) we feed a sequence of PropBank sense indicators for MR predicates.

Joint encoding of MR and dependency parses for metric prediction. Embedding layers are shared between MR/dependency pointers and MR/dependency tokens. We embed the three sequences representing the MR graph (tokens, pointers and senses) in three matrices and sum them up element-wise (indicated with + in Figure 7.2). The same procedure is applied to the linearized dependency graph (tokens and pointers). The

resulting matrices are processed by two two-layered Bi-LSTMs to yield vectorized representations for (i) the MR graph and (ii) the dependency tree (i.e., the last states of forward and backward reads are concatenated). Thereafter, we apply element-wise multiplication, subtraction and addition to both vector representations and concatenate the resulting vectors (\otimes in Figure 7.2). The joint MR-dependency representation is further processed by a feed forward layer (FF) with sigmoid activation functions in order to predict, in total, 36 different metrics (green, Figure 7.2).

Hierarchical prediction of multiple metrics. The task naturally lends itself to be formulated in a hierarchical multi-task setup (orange, Figure 7.2). In this strand, we first compute aspectual subgraph metrics and on their basis we calculate the main scores (precision, recall, F1) as our primary metrics. In order to accomplish this, we collect the outputs from the subgraph metric prediction layer in a vector and concatenate it with the previous layer’s representation (\oplus in Figure 7.2). The resulting vector is fed through a last FF layer to predict the Precision/Recall/F1 SMATCH. Our intuition is that the estimated quality of the parse with respect to the aspectual metrics can help refine the prediction of the overall quality.

Discussion. Generally speaking, the LG-LSTM is a model that relies on graph linearization. Such type of model, despite its apparent simplicity, has proven to be an effective baseline or state-of-the-art method in various works about converting texts into graphs (Konstas et al., 2017; Noord and Bos, 2017b), or converting graphs into texts (Bastings et al., 2017; Beck et al., 2018; Song, 2019; Pourdamghani et al., 2016; Song et al., 2018; Vinyals et al., 2015; Mager et al., 2020b), or performing mathematically complex tasks modeled as graph-to-graph problems, such as symbolic integration (Lample and Charton, 2020).

7.5 Model II: Novel MR-as-image encoding with CNN

7.5.1 MR as image with latent channels

Now, we first motivate to treat MRs as images with latent channels in order to rate them efficiently. With this, we can invoke a lightweight CNN that evaluates MR quality in multiple dimensions of interest, and exploits *spatial Penman graph structure*. We call this model PCNN.

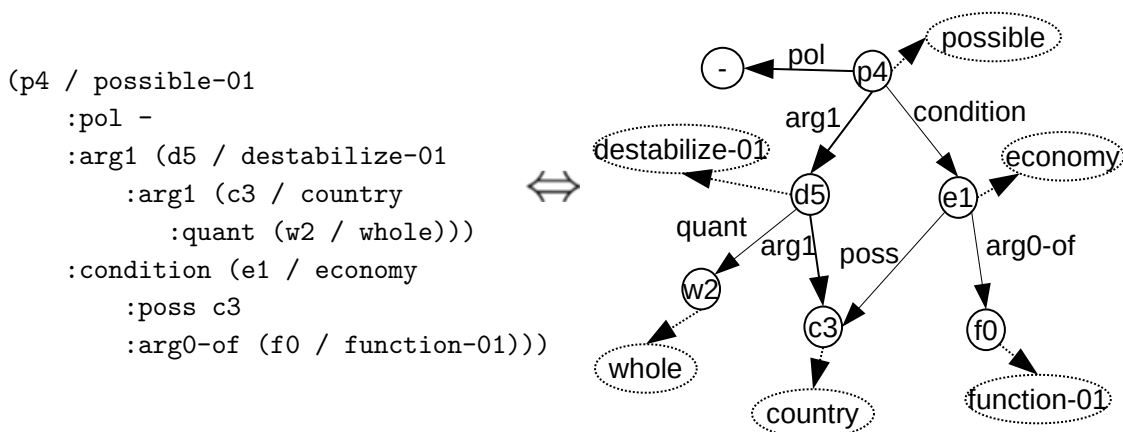


Figure 7.3: Recap of different displays for an MR structure of a sentence that has medium length (left: PENMAN notation, right: graphical visualization). See also the Figure in Background Section 2.1.

graph representation	computer processing	human understanding
triples/ G	✓ (e.g., GNN)	✗
graph visualization	✗	✓ (short sentences)
PENMAN, linearized string	✓ (e.g., LSTM)	✗
PENMAN, indents	✓ (this work)	✓

Table 7.1: Equivalent MR representations and their accessibility with respect to human or computer (✓: ‘okay’, ✗: ‘perhaps possible, but not well defined’).

The PENMAN notation and its (hidden) advantages. As we have already seen in our Background Chapter (Section 2.1.1), an interesting MR-as-string notation is called PENMAN-notation or *Sentence Plan Language* (Kasper, 1989; Mann, 1983): based on a depth-first traversal the graph a directed and rooted graph is serialized into a string. A clear advantage of this notation is that it allows for secure MR storage in text files. However, we argue that it has more advantages. For example, due to its clear structure, it allows humans a fairly quick understanding even of medium-sized to large MR structures (Figure 7.3, left). On the other hand, we argue that a graphical visualization of such medium-sized to large MRs (Figure 7.3, right) could hamper intuitive understanding, since the abundant visual signals (circles, arrows, etc.) may more easily overwhelm humans. Moreover, in every display, one would depend on an algorithm that needs to determine a suitable (and spacious) arrangement of the nodes, edges and edge labels. It may be for these reasons, that in the MR annotation tool², the graph that is under construction is always shown in PENMAN notation to the human user.

²<https://www.isi.edu/cgi-bin/div3/mt/amr-editor/login-gen-v1.7.cgi>

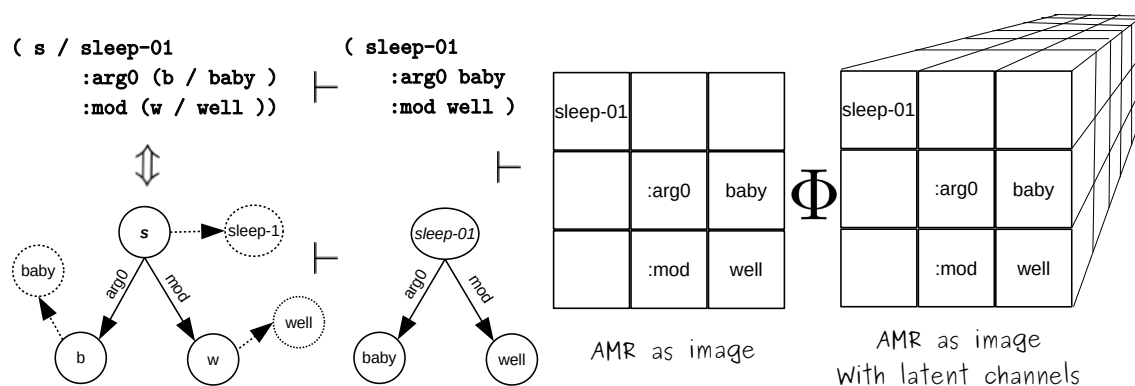


Figure 7.4: We transform the (simplified) PENMAN representation to an image and use Φ to add latent channels.

In sum, we find that the indented multi-line PENMAN form possesses three key advantages (Table 7.1): (i) it enables fairly easy human understanding, (ii), it is well-defined and (iii), which is what we will show next, it can be computationally exploited to better rate AMR quality.

AMR as image to preserve graph structure. Figure 7.4 describes our proposed sentence representation treatment.

After non-degenerate MR graph simplification (more details in *Preprocessing*, 7.8), we first project the PENMAN representation onto a small grid (‘image’). Each MR token (e.g., a node or an edge) is represented as a ‘categorical pixel’. Second, Φ adds latent ‘channels’ to the categorical pixels, which can be learned incrementally in an application. In other words, every MR token is represented by a fixed-sized vector of real numbers. These vectors are arranged such that the original graph structure is fully preserved.

7.5.2 A lightweight CNN to rate AMR quality

Again, we want to model f (Eq. 7.1) in order to estimate a suite of quality scores $y \in \mathbb{R}^d$ for any automatically generated MR graph, given only the graph and the sentence from whence it is derived. As with LG-LSTM, we will contrast the MR against the sentence’s dependency parse, exploiting observed structural similarities between these two types of information (Wang et al., 2015). The CNN architecture is outlined in Figure 7.5, it works in the following steps:

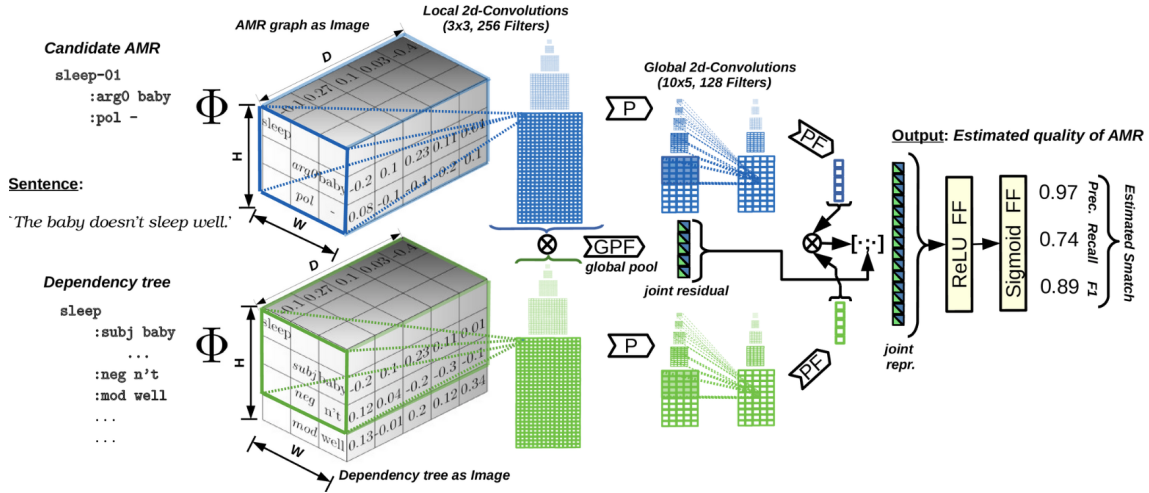


Figure 7.5: Our model II. Architecture for efficient AMR quality assessment.

Symbol embedding. The latent channels of AMR and dependency ‘pixels’ represent the embeddings of the ‘tokens’ or ‘symbols’ contained in the MR and dependency vocabulary. These symbols represent nodes or edges. We use two special tokens: the $\langle \text{tab} \rangle$ token, which represents the indentation level, and the $\langle \text{pad} \rangle$ token, which fills the remaining empty ‘pixels’. By embedding lookup, we obtain MR and dependency images with 128 latent channels and 45×15 ‘pixels’ (Φ in Figure 7.5; the amount of pixels is chosen such that more than 95% of training MRs can be fully captured).

Encoding local graph regions. Given MR and dependency images with 128 latent channels and 45×15 pixels, we apply to each of the two images 256 filters of size 3×3 , which is a standard type of kernel in CNNs. This converts both graphs to 256 feature maps each $\in \mathbb{R}^{45 \times 15}$ (same-padding), obtaining two three-dimensional tensors $L_{amr}^1, L_{dep}^1 \in \mathbb{R}^{45 \times 15 \times 256}$. From here, we construct our first joint representation, which matches local dependency regions with local MR regions:

$$j_{res} = GPF(L_{amr}^1 \otimes L_{dep}^1), \quad (7.2)$$

where $x \otimes y = [x \odot y; x \ominus y]$ denotes the concatenation of element-wise multiplication and element-wise subtraction. GPF is an operation that performs global pooling and vectorization (‘flattening’) of any input tensor. This means that $j_{res} \in \mathbb{R}^{512}$ is a joint representation of the locally matched dependency and AMR graph regions. This intermediate process is outlined in Figure 7.5 by \otimes (left) and GPF . Finally, we reduce the dimensions

of the two intermediate three-dimensional representations L_{amr}^1 and L_{dep}^1 with 3x3 max-pooling and obtain L_{amr}^2 and $L_{dep}^2 \in \mathbb{R}^{15 \times 5 \times 256}$

Encoding global graph regions. For a moment, we put the *joint residual* (j_{res}) aside and proceed by processing the locally convolved feature maps with larger filters. While the first convolutions allowed us to obtain abstract *local* graph regions L_{amr}^2 and L_{dep}^2 , we now aim at matching more *global* regions. More precisely, we use 128 2D filters of shape 10x5, followed by a 5x5 max-pooling operation on L_{amr}^2 and L_{dep}^2 . Thus, we have obtained vectorized abstract global graph representations $g_{amr}, g_{dep} \in \mathbb{R}^{384}$. Then, we construct a joint representation (right \otimes , Figure 7.5):

$$j_{glob} = g_{amr} \otimes g_{dep}. \quad (7.3)$$

At this point, together with the joint residual representation from the local region matching, we have arrived at two joint vector representations j_{glob} and j_{res} . We concatenate them ($[\cdot; \cdot]$ in Figure 7.5) to form one joint representation $\mathbf{j} \in \mathbb{R}^{1280}$:

$$\mathbf{j} = [j_{res}; j_{glob}] \quad (7.4)$$

Quality prediction. The shared representation \mathbf{j} is further processed by a feed-forward layer with ReLU activation functions (FF_{+ReLU} , Figure 7.5) and a consecutive feed-forward layer with sigmoid activation functions (FF_{+sigm} , Figure 7.5):

$$\mathbf{y} = \text{sigm}(\text{ReLU}(\mathbf{j}^T A) B), \quad (7.5)$$

where $A \in \mathbb{R}^{1280 \times h}$, $B \in \mathbb{R}^{h \times \dim(out)}$ are parameters of the model and

$$\text{sigm}(x) = \left(\frac{1}{1 + e^{-x_1}}, \dots, \frac{1}{1 + e^{-x_{\dim(out)}}} \right) \quad (7.6)$$

projects x onto $[0, 1]^{\dim(out)}$. When estimating the main MR metric scores we instantiate three output neurons ($\dim(out) = 3$) that represent main quality dimensions. In the case where we are interested in a more fine-grained assessment of AMR quality (e.g., knowledge-base linking quality), we can introduce more output neurons representing expected scores for various semantic aspects involved in MR parsing (as outlined next in Section 7.6).

To summarize, the residual joint representation should capture local similarities. On the other hand, the second joint representation aims to capture the more global and structural properties of the two graphs. Both types of information inform the final quality assessment of our model in the last layer.

7.6 Multi-quality dimensions

Main AMR quality dimensions. It is possible to learn to predict all MR metrics from the previous chapters. However, for simplicity, here we want to stick to the standard metrics that assess triple overlap. Hence, the main quality dimensions that we desire our model to predict are estimated **SMATCH F1/recall/precision**.

AMR sub-task quality dimensions. We also predict other quality dimensions to assess various MR aspects based on MR subgraphs as outlined in Figure 3.1 from our related work Section 3.1. A brief overview: (i) **Unlabeled**: SMATCH F1 when disregarding edge-labels. (ii) **No WSD**: SMATCH F1 when ignoring ProbBank senses. (iii) **Frames**: PropBank frame identification F1 (iii) **Wikification**: KB linking F1 score on *:wiki* relations. (iv) **Negations**: negation detection F1. (v) **NamedEnt**: NER F1. (vi) **NS frames**: F1 score for ProbBank frame identification when disregarding the sense. (vii) **Concepts** SMATCH F1 score for concept identification (viii) **SRL**: SMATCH F1 computed on arg-i roles only. (ix) **Reentrancy**: SMATCH F1 computed on re-entrant edges only. (x) **Ignore-Vars**: F1 when variable nodes are ignored. (xi) **Concepts**: F1 for concept detection.

7.7 Experimental data construction

Since our goal is to predict the accuracy of an automatic parse, we need a data set containing automatically produced AMR parses and their scores, as they would emerge from comparison to gold parses. Our data set, LDC2015E86, comprises 19,572 sentences and comes in a predefined training, development and test split. We parse this data set with three parsers, JAMR (Flanigan et al., 2014), CAMR (Wang et al., 2016) and GPLA (Lyu and Titov, 2018). Since the three parsers have been trained on the training data partition, we naturally obtain more accurate parses for the training partition than for development and test data.

parser	training		development	
	SMATCH (F1)	% def.	SMATCH (F1)	% def.
JAMR	0.79	86.7	0.69	91.8
CAMR	0.75	93.6	0.66	95.7
GPLA	0.86	83.4	0.76	90.0

Table 7.2: Parser output evaluation on training and development partitions of LDC2015E86. SMATCH F1: avg. over SMATCH F1 per sentence, % def.: percentage of deficient parses (i.e., parses with SMATCH F1 < 1).

Table 7.2, however, indicates that we still obtain a considerable amount of deficient parses for training. Based on the parser outputs we compute evaluations comparing the automatic parses with the gold parses by using SMATCH and fine-grained SMATCH sub-graph metrics. This allows us to create full-fledged training, development and test instances for our quality estimation task. Each instance consists of a sentence and an MR parse as input and a vector of metric scores as target.

Finally, our preliminary data set $\mathcal{D} = \{(S_i, G_i, y_i)\}_{i=1}^N$ contains 58,716 tuples (S_i, G_i, y_i) , where S_i is a natural language sentence, G_i is a ‘candidate’ AMR graph and $y_i \in \mathbb{R}^d$ is a 36-dimensional vector containing scores which represent the quality of the AMR graph in terms of precision, recall and F1 with respect to 12 different tasks captured by AMR (as outlined in Section 7.6).

Debiasing of the data. We observe three biases in the data. First, the graphs in the training section of our data are less deficient than in the development and testing data, because the parsers were trained on *(sentence, gold graph)* pairs from the training section. For our task, this means that the training section’s target scores are higher, on average, than the target scores in the other data partitions. To achieve more balance in this regard, we re-split the data randomly on the sentence-id level (such that a sentence does not appear in more than one partition with different parses).

Second, we observe that the data contains some superficial hidden clues that could give away the parse’s source. This bears the danger that a model does not learn to assess the *parse quality*, but to assess the *source of the parse*. And since some parsers are better or worse than others, the model could exploit this bias. For example, consider that one parser prefers to write *(r / run-01 :arg1 (c / cat) :polarity -)*, while the other parser prefers to write *(r / run-01 :polarity - :arg1 (c / cat))*. These two structures are semantically equivalent but differ on the surface. Hence, the arrangement of the output may provide

unwanted clues on the source of the parse. To alleviate this issue, we randomly re-arrange all parses on the surface, keeping their semantics.³⁴

A third bias stems from a design choice in the metric scripts used to calculate the target scores. More precisely, the extended SMATCH-metric script, per default, assigns a parse that does not contain a certain edge-type (e.g., `:argn`) the score 0 with respect to the specific quality dimension (in this case, SRL: 0.00 Precision/Recall/F1). However, if the gold parse also does not contain an edge of this type (i.e., `:argn`), then we believe that the correct default score should be 1, since the parse is, in the specific dimension, in perfect agreement with the gold (i.e., SRL: 1.00 Precision/Recall/F1). Therefore, we set all sub-task scores, where the predicted graph agrees with the gold graph in the absence of a feature, from 0 to 1.

7.8 Experiments on MR quality prediction

Preprocessing. Same as prior work, we dependency-parse and tokenize the sentences with spacy (Honnibal and Montani, 2017) and replace variables with corresponding concepts (e.g., `(j / jump-01 :arg0 (g / girl))` is translated to `(jump-01 :arg0 (girl))`). Re-entrancies are handled with pointers according to Noord and Bos (2017a), which ensures non-degenerate AMR simplification.⁵ Furthermore, we lower-case all tokens, remove quotation marks and join sub-structures that represent names.⁶ The vocabulary encompasses all tokens of frequency ≥ 5 , remaining ones are set to `<unk>`.

Training. All parameters are initialized randomly. We train for 5 epochs and select the parameters θ from the epoch where maximum development scores were achieved (with respect to average Pearson’s ρ over the quality dimensions). In training, we reduce the squared error with gradient descent (Adam rule (Kingma and Ba, 2019), learning rate =

³Technically, this is achieved by reformatting the parses such that in the depth-first writing-traversal at node n the out-going edges of n will be traversed in random order.

⁴Different variable names, e.g., `(r / run-01)` and `(x / run-01)` are not an issue in this work since the variables are handled via (Noord and Bos, 2017a). See also *Preprocessing*, Section 7.8

⁵For example, consider the sentence *The cat scratches itself* and its graph `(x / scratch-01 :arg0 (y / cat) :arg1 y)`. Replacing the variables with concepts would come at the cost of an information loss w.r.t. to coreference: `(scratch-01 :arg0 cat :arg1 cat)` — does the cat scratch itself or another cat? Hence, pointers are used to translate the graph into `(scratch-01 :arg0 *0* cat :arg1 *0*)`.

⁶E.g., `:name (name :op1 ‘Barack’ :op2 ‘Obama’)` is translated to `:name barack obama`.

	SMATCH	Ridge	GNN	LG-LSTM	PCNN	change %
P's ρ	F1	0.428	0.659	0.662 \pm 0.00	0.696 \pm 0.00	+5.14 †‡
	Precision	0.348	0.601	0.600 \pm 0.00	0.623 \pm 0.01	+3.83 †
	Recall	0.463	0.667	0.676 \pm 0.00	0.719 \pm 0.00	+6.36 †‡
RMSE	F1	0.155	0.132	0.130 \pm 0.00	0.128 \pm 0.00	-1.54
	Precision	0.146	0.127	0.126 \pm 0.00	0.126 \pm 0.00	+0.0
	Recall	0.169	0.141	0.142 \pm 0.00	0.136 \pm 0.00	-4.23

Table 7.3: Main results. Pearson’s corr. coefficient (row 1-3) is better if higher; root mean square error (RMSE, row 4-6) is better if lower. The quality dimensions are explained in Section 7.6. † (‡): $p < 0.05$ ($p < 0.005$), significant difference in the correlations with two-tailed test using Fisher ρ to z transformation (Fisher, 1915).

0.001, mini batch size = 64):

$$\theta^* = \arg \min_{\theta} \sum_{i=1}^{|\mathcal{D}|} \sum_{j=1}^{|M|} (y_{i,j} - f_{\theta}(s_i, g_i)_j)^2, \quad (7.7)$$

where M is the set of target metrics.

Baselines. To better put the main test results into perspective, we also display the results of two additional baselines: **GNN** (Song et al., 2018), where we encode the dependency tree and the AMR with a graph-recurrent encoder and perform regression on the joint averaged node embedding vectors. More precisely, $\left[\frac{1}{|V_A|} \sum_{v \in V_A} \text{emb}(v) \right] \otimes \left[\frac{1}{|V_D|} \sum_{v \in V_D} \text{emb}(v) \right]$. And **Ridge**, an l2-regularized linear regression that is based on shallow graph statistics. For the dependency graph (D) and the MR graph (A) we both compute $\phi(A|D) = [\text{density}, \text{avg. node degree}, \text{node count}, \text{edge count}, (\text{arg0}|\text{subj}) \text{ count}, (\text{arg1}|\text{obj}) \text{ count}]$, the final feature vector then is defined as $\Phi(x) = [\phi(A) - \phi(D); \phi(D); \phi(A); \frac{|\text{lemmas}(D) \cap \text{concepts}(A)|}{|\text{lemmas}(D) \cup \text{concepts}(A)|}]$

7.8.1 Results

Main AMR quality dimensions. The main quality of an MR graph is estimated in expected triple match ratios (SMATCH F1, Precision and Recall).

The results, averaged over 10 runs, are displayed in Table 7.3. With regard to estimated SMATCH F1, PCNN achieves a correlation with the gold scores of 0.695 Pearson’s ρ . This constitutes a significant improvement of appr. 5% over LG-LSTM. Similarly, recall and precision correlations improve by 6.36% and 3.83 % (from 0.676 to 0.719 and 0.600 to 0.623). While the improvement in predicted recall is significant at $p < 0.05$ and

	Quality Dim.	LG-LSTM	PCNN	change %
F1 Pearson's ρ	Concepts	0.508 \pm 0.01	0.545 \pm 0.01	+7.28 †
	Frames	0.420 \pm 0.01	0.488 \pm 0.01	+16.19 ††
	IgnoreVars	0.627 \pm 0.01	0.665 \pm 0.00	+6.06 ††
	NamedEnt.	0.429 \pm 0.02	0.460 \pm 0.01	+7.23 †
	Negations	0.685 \pm 0.02	0.746 \pm 0.01	+8.91 ††
	NoWSD	0.640 \pm 0.01	0.680 \pm 0.00	+6.25 ††
	NS-frames	0.419 \pm 0.02	0.505 \pm 0.01	+20.53 ††
	Reentrancies	0.508 \pm 0.01	0.602 \pm 0.00	+18.50 ††
	SRL	0.519 \pm 0.01	0.581 \pm 0.01	+11.95 ††
	Unlabeled	0.628 \pm 0.01	0.663 \pm 0.00	+5.57 ††
Wikification	0.901 \pm 0.00	0.904 \pm 0.00	+0.33	
F1 RMSE	Concepts	0.117 \pm 0.00	0.114 \pm 0.00	-2.56
	Frames	0.186 \pm 0.00	0.182 \pm 0.00	-2.15
	IgnoreVars	0.195 \pm 0.00	0.186 \pm 0.00	-4.62
	NamedEnt.	0.159 \pm 0.00	0.156 \pm 0.00	-1.89
	Negations	0.197 \pm 0.00	0.180 \pm 0.00	-8.63
	NoWSD	0.132 \pm 0.00	0.126 \pm 0.00	-4.55
	NS-frames	0.157 \pm 0.00	0.155 \pm 0.00	-1.27
	Reentrancies	0.285 \pm 0.00	0.265 \pm 0.00	-7.02
	SRL	0.189 \pm 0.00	0.181 \pm 0.00	-4.23
	Unlabeled	0.124 \pm 0.00	0.121 \pm 0.00	-2.42
Wikification	0.165 \pm 0.00	0.162 \pm 0.00	-1.82	

Table 7.4: Results for AMR quality rating w.r.t. various sub-tasks. † (‡): significance (c.f. caption Table 7.3).

$p < 0.005$, the improvement in predicted precision is significant at $p < 0.05$. When we consider the root mean square error (RMSE), we find that PCNN improves over the best baseline by -1.54% in estimated SMATCH F1 and -4.23% in estimated SMATCH recall. On the other hand, the RMS error in estimated precision remains unchanged.

AMR subtask quality. PCNN and LG-LSTM can also rate the quality of an AMR graph in a more fine-grained way.

The results are displayed in Table 7.4. Over almost every dimension we see considerable improvements by PCNN. For instance, a considerable improvement in Pearson's ρ is achieved for assessment of *frame prediction quality* ('NSFrames' in Table 7.4, +20.5% ρ) and *coreference quality* ('Reentrancies' in Table 7.4, +18.5%).

A substantial error reduction is achieved in *polarity* ('Negations', Table 7.4), where PCNN reduces the RMSE of the estimated F1 score by -8.6%. When rating the SRL-quality of an AMR parse, PCNN reduces the RMSE by appr. 4%. In general, improvements are obtained over almost all tested quality dimensions, both in RMSE reduction and increased correlation with the gold scores.

data	method	Pearson’s ρ			error
		P	R	F1	RMSE (F1)
$\frac{0}{2}$	LG-LSTM	0.72	0.78	0.77	0.138
	LG-LSTM _{+aux}	0.74	0.79	0.78	0.137
	PCNN	0.75	0.80	0.79	0.133
	PCNN _{+aux}	0.76	0.81	0.80	0.132
$\frac{1}{2}$	LG-LSTM	0.67	0.73	0.72	0.120
	PCNN	0.68	0.75	0.74	0.117
$\frac{2}{2}$	LG-LSTM	0.60	0.68	0.66	0.130
	PCNN	0.62	0.72	0.70	0.128

Table 7.5: Performance-effects of data debiasing steps. _{+aux} indicates a model variant that is trained using auxiliary losses that incorporate hierarchical information about the other AMR aspects in the training process.

7.8.2 Analysis

Effect of data debiasing. We want to study the effect of the data set cleaning steps by analyzing the performance of our method and the baseline on three different versions of the data, with respect to estimated SMATCH scores. The three versions are (i) $\frac{0}{2} = \text{AMRQUALITY}$, which is the original data; (ii) $\frac{1}{2}$, which is the data after the random re-split and score correction; (iii) $\frac{2}{2} = \text{AMRQUALITYCLEAN}$ which is our main data after the final debiasing step (shallow structure debiasing) has been applied.

The results are shown in Table 7.5. We can make three main observations: (i) from the first to the second debiasing step, both LG-LSTM and PCNN have in common that Pearson’s ρ and the error decrease. While we cannot exactly explain why ρ decreases, it is somewhat in line with recent research that observed performance drops when data was re-split (Gorman and Bedrick, 2019). On the other hand, the error decrease can be explained by the random re-split that balances the target scores. (ii) The second debiasing step leads to a decrease in ρ and an increase in error, also for both models. This indicates that we have successfully removed shallow biases from the data that can give away the parse’s source. (iii) On all considered versions of the data, the method performs better than the baseline.

AMRs: telling the good from the bad. In this experiment, we want to see how well the models can discriminate between good and bad graphs. To this aim, we create a five-way classification task: graphs are assigned the label ‘very bad’ (SMATCH F1 < 0.25), ‘bad’ ($0.25 \geq \text{SMATCH F1} < 0.5$), ‘good’ ($0.5 \geq \text{SMATCH F1} < 0.75$), ‘very good’ ($0.75 \geq$

	majority	random	LG-LSTM	PCNN
avg. F1	0.13	0.20	0.40	0.44 ^{†‡}
quadr. kappa	0.0	0.03	0.53	0.60 ^{†‡}

Table 7.6: Graph quality classification task. † (‡) significance with paired t-test at $p < 0.05$ ($p < 0.005$) over 10 random initializations.

	Quality Dim.	LG-LSTM	PCNN	PCNN (no dep.)
P's ρ	SMATCH F1	0.662 \pm 0.00	0.696 \pm 0.00	0.682 \pm 0.01
	SMATCH precision	0.600 \pm 0.00	0.623 \pm 0.01	0.614 \pm 0.01
	SMATCH recall	0.676 \pm 0.00	0.719 \pm 0.00	0.702 \pm 0.01
RMSE	SMATCH F1	0.130 \pm 0.00	0.128 \pm 0.00	0.128 \pm 0.00
	SMATCH precision	0.126 \pm 0.00	0.126 \pm 0.00	0.129 \pm 0.00
	SMATCH recall	0.142 \pm 0.00	0.136 \pm 0.00	0.139 \pm 0.00

Table 7.7: Right column: results of our system when we abstain from feeding the dependency tree, and only show the sentence together with the candidate MR.

SMATCH F1 < 0.95) and ‘excellent’ (SMATCH F1 ≥ 0.95), i.e., group of excellent graphs includes the gold graphs and (almost) flawless parses. Here, we do not retrain the models with a classification objective but convert the estimated SMATCH F1 to the corresponding label. Since the classes are situated on a nominal scale, and ordinary classification metrics would not fully reflect the performance, we also use quadratic weighted kappa (Cohen, 1968) for evaluation.

The results are shown in Table 7.6. All baselines, including LG-LSTM, are significantly outperformed PCNN, both in terms of macro F1⁷ (+4 points, 10% improvement) and quadratic kappa (+7 points, 13% improvement).

How important is the dependency information? To investigate this question, instead of feeding the dependency tree of the sentence, we only feed the sentence itself. To achieve this, we simply insert the tokens in the first row of the former dependency input image, and pad all remaining empty ‘pixels’. In this mode, the sentence encoding is similar to standard convolutional sentence encoders as they are typically used in many tasks (Kim, 2014).

The results are shown in the right column of Table 7.7. The performance drops are small but consistent across all analyzed dimensions, both in terms of error (0 to 2.2% increase) and Pearson’s ρ (1.4 to 2.4% decrease). This indicates that the dependency trees

⁷We use the arithmetic mean over the classes (Opitz and Burst, 2019).

GPU type	GTX Titan		GTX 1080	
method	LG-LSTM	PCNN	LG-LSTM	PCNN
avg. ep. time	722s	59s	1582s	64s
avg. W	105	166	45	128
kWh per epoch	0.021	0.003	0.020	0.002

Table 7.8: Efficiency analysis of two approaches.

contain information that can be exploited by our model to better judge the MR quality. We hypothesize that this is due to similarities between relations such as *subj/obj* (syntactic) or *arg0/arg1* (semantic), etc. Yet, we see that this simpler model, which does not see the dependency tree, still outperforms the baseline, except in estimated precision, where the error is increased by 2.4%.

Efficiency analysis. Recently, in many countries, there have been efforts to reduce energy consumption and carbon emission. Since deep learning typically requires intensive GPU computing, this aspect is of increasing importance to researchers and applicants (Strubell et al., 2019). To investigate energy consumption of our methods, we monitor their GPU usage during training, assessing the following quantities : (i) avg. time per epoch, (ii) avg. watts GPU usage, (iii) kilowatts per epoch (in kWh).

The results of this analysis are displayed in Table 7.8 and outlined in Figure 7.6. PCNN consumes approximately 6.6 times less total kWh on a GTX Titan (10 times less on a GTX 1080). Directly related, it also reduces the training time: prior work requires appr. 1500s training time per epoch (GTX 1080), while our method requires appr. 60s per epoch (GTX 1080). The main reason for this is that PCNN does not depend on recurrent operations and profits more from parallelism.

7.9 Discussion

In this chapter, we tested the feasibility of predicting distances between meaning structures when one meaning structure is hidden, and we only have its sentence; a problem that occurs when we want to rate parser output efficiently without constructing costly gold data. To this aim, we proposed two graph encoders that encode the candidate MR and the dependency tree of the source sentence, and learning to match them, extrapolating an MR metric. The best encoder is also the simplest: A novel and efficient CNN graph encoder that exploits the Penman multi-line spatial and interpretable serialized MR-string

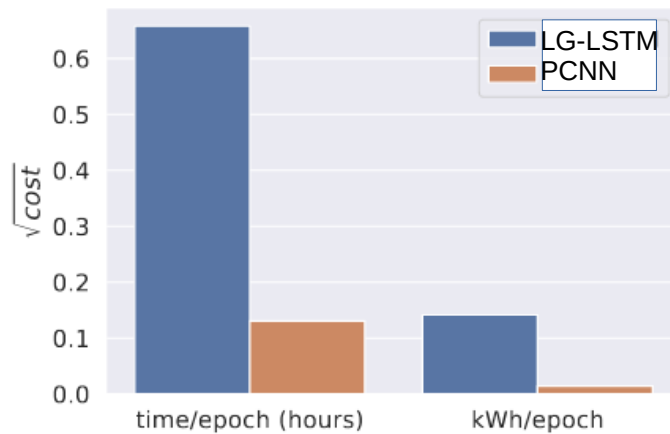


Figure 7.6: Training cost diagram of two approaches.

representation, that is also used in the human annotation process of MRs. A caveat is that we performed supervised training to learn to extrapolate the MR metrics. The training data set Incorporated parses from a selection of parsing systems, which could significantly limit the method's capability to generalize well to different/stronger parsers and data from different domains.

Part III

MR metrics for effective semantic similarity

Chapter 8

Exploring argumentation with MR metrics

8.1 Chapter outline

In the last two chapters, we want to explore extended and generalized use-cases for MR metrics, measuring semantic text similarity through the lens of MR. To make a first step, in this chapter we investigate a concrete application, that is measuring the similarity of natural language arguments. This is a task where the interest into explainable computational methods is specifically growing (Kobbe et al., 2019; Lawrence, 2021; Becker et al., 2020; Singh et al., 2021). In particular, we i) generally study the usefulness of MR representations and MR metrics for argumentation tasks, and ii) investigate the use of MR metrics for explaining relations between arguments.

The remainder of this chapter is structured as follows:

1. After discussing more background (Section 8.2) we introduce MR-argument similarity hypotheses (Section 8.3) and describe our concrete approach in Section 8.4.
2. We conduct experiments on argument similarity through the lens of MR and MR metrics (Section 8.5), obtaining state-of-the-art results.
3. We conduct extensive analysis in Section 8.6 and furthermore assess the usefulness of MR metrics for rating the quality of automatically generated argumentative conclusions.
4. We conclude the Chapter with a discussion (Section 8.7).

Underlying work. The content of this chapter is mainly based on Opitz et al. (2021b). The work received the best paper award at the Argument Mining Workshop 2021.

8.2 Research questions

When assessing the similarity of arguments, researchers typically use approaches that do not provide interpretable evidence or justifications for their ratings. Hence, the features that determine argument similarity remain elusive.

Indeed, previous methods (Reimers et al., 2019) for rating argument similarity suffer from a common flaw: beyond shallow statistics (word matches in bag-of-word models, or word similarities in distributional space), they do not provide any rationale for their predictions, and the prediction process is in general not transparent. Therefore, we know only little about the following question:

- *Which argument features correlate with human argument similarity decisions?*

In this work, we undertake a first attempt at answering this question, by testing two hypotheses:

- i) Representing arguments with Abstract Meaning Representations (AMRs) and using AMR graph metrics improves argument similarity rating and provides explanatory information.
- ii) Extending arguments with inferred conclusions can improve argument similarity rating.

8.3 Hypotheses

We base our models for explanatory argument similarity assessment on two hypotheses.

Hypothesis I: Abstract Meaning Representation of arguments supports explainable argument similarity assessment. There is growing interest in extracting graph structures from natural language arguments. Lenz et al. (2020), e.g., propose a pipeline for detecting and linking argumentative discourse units (ADUs). Al-Khatib et al. (2020) detect textual phrases and link them with *POS/NEG* relations, where *POS* indicates a positive influence and *NEG* a negative influence (inhibition), e.g., *sports NEG health issues*. However, such approaches lack finer semantic assessment: they do not distinguish word

senses, and the linked entities (phrases or ADUs) are taken as atoms, which hampers explainability: when linking *sports* and *health issues* with a *NEG* relation, we cannot differentiate *sports NEG issues* and *sports NEG health* (only the former is correct). We target a finer analysis of argumentative texts, by representing them with dense AMR graphs.

Recall that AMRs are directed, rooted and acyclic graphs that aim at capturing a sentence’s meaning (c.f. Background 2.2). Edges are labeled with semantic relation types (e.g., negation, cause, etc.) and vertices denote either variables or concepts (variables are instances of concepts and allow us to capture coreferences). Hence, AMR can capture various semantic phenomena that can play a role when assessing *argument* similarity. E.g., besides the obviously useful aspect of negation, AMR captures semantic roles and predicate senses (Kingsbury and Palmer, 2002). While it is clear that similar arguments tend to involve similar predicates and predicate senses, semantic structure and role assignment may also play a role. For instance, the claims: *consumption of alcohol leads to depression* vs. *depression leads to consumption of alcohol* are clearly distinct, while sharing the same concepts. Other AMR facets may also be useful. E.g., AMR captures coreferences and resolving them in different ways can induce significant meaning differences. Finally, AMR includes key semantic relations (location, cause, possession, etc.) that are often implicit or underspecified in language, hence their explicit representation in AMR provides a rich basis for assessing arguments.

Arguments represented with AMR can be compared with our AMR graph metrics, with the option to induce an explicit alignment between two argument graphs.

Hypothesis II: similar arguments lead to similar conclusions. We hypothesize that a key feature of similar arguments is that they invite for similar conclusions. Analogously, dissimilar arguments tend to lead to differing conclusions.

Consider the following two arguments:

- i) *Cannabis can have negative effects on brain development of teens.*
- ii) *Smoking cannabis is harmful for the lungs.*

The arguments are *dissimilar*, even though they share the same (negative) stance and argue from a similar perspective (health). This dissimilarity is also reflected in the conclusions that can be inferred from them: from i) we can infer that, i.a., *Cannabis consumption should be strictly controlled for age* or *Cannabis can have a negative impact on the brain*—while from ii) we could infer that *Cannabis, if consumed, should not be smoked* or *Cannabis smokers should get their lungs checked*.

As a complementary example, the *similarity* of two arguments may be reinforced by the *similarity* of their inferred conclusions, as shown below:

- i) *Fracking can contaminate water and water wells and suck towns dry.*
- ii) *As a water-poor state, fracking and its toxic wastewater presents a serious danger to our communities and ecosystems.*

Arguments i) and ii) are rated as similar, presumably because they point at detrimental ramifications of fracking related to water issues. This similarity is likely to be reflected in conclusions drawn from them, such as: i) *Fracking can lead to water issues* or ii) *Fracking poses dangers for water-poor states*.

8.4 Argument Similarity through MR Metrics

According **Hyp I**, we represent arguments with AMR graphs and rate their similarity with AMR metrics. To test **Hyp II** we infer conclusions from arguments with language models and compute similarity on arguments extended with their conclusion.

8.4.1 Models

We propose three model variants that aim at explaining argument similarity. Given two arguments a, a' and their *extrapolated* conclusions $c = \text{conclusion}(a)$, $c' = \text{conclusion}(a')$, we compute similarity in the space of abstract meaning representation using a similarity function f in three alternative ways: i) $f(a, a')$, between the two arguments, ii) $f(c, c')$ between their conclusions, iii) $f(a \oplus c, a' \oplus c')$, i.e., between the combinations of argument a and its derived conclusion c , where we use a simple decomposable weighting:

$$f(a \oplus c, a' \oplus c') = \lambda f(a, a') + (1 - \lambda) f(c, c') \quad (8.1)$$

If not specified otherwise, λ is set to 0.95.¹ The AMR metric f will be described in the following.

¹We choose a high value of λ since, clearly, the premises are bound to host the primary evidence for similarity, while a conclusion may serve as auxiliary information. In our experiments, we also consider extreme decompositions ($\lambda \in \{0, 1\}$).

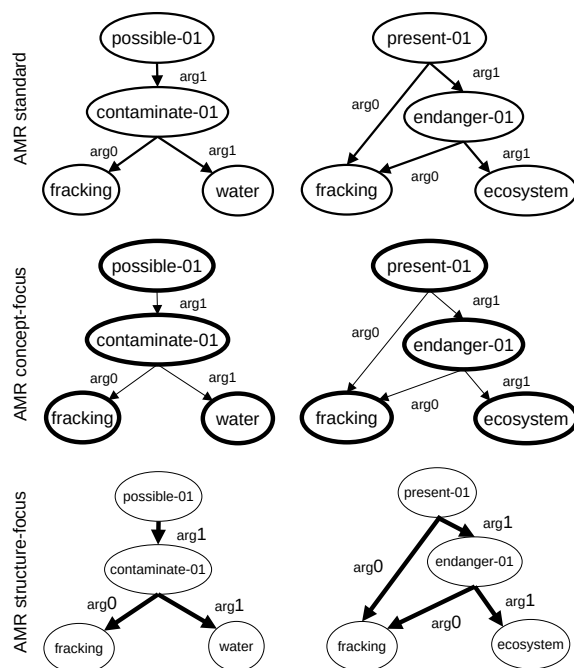


Figure 8.1: Standard, concept-focus and structure focus.

8.4.2 Implementation

AMR parser. Same as in BAMBOO², we parse all arguments from the data with the parser from `amrlib`², a strong fine-tuned T5 sequence-to-sequence model.

Conclusion generator. The task of conclusion generation has been recently investigated by Alshomary et al. (2020, 2021), and allows us to infer conclusions from given premises. Conclusion generation can be seen as the inverse of argument generation (Sato et al., 2015; Schiller et al., 2020). We generate conclusions from arguments using the T5 model (Raffel et al., 2020) pre-trained on summarization tasks. To encourage the model to generate informative conclusions (as opposed to summaries), we further fine-tune it on premise-conclusion samples from Stab and Gurevych (2017), which contain intelligible and rational conclusions of high linguistic quality.³

²<https://github.com/bjascob/amrlib>

³For further detail on this fine-tuning step, see Appendix A.3.

8.4.3 AMR metric variants for exploring argument similarity

As a basis here, we use S^2 MATCH and WWLK, since they admit measurement of more graded similarity. Since, so-far, little is known about the trade-off and interface between concrete and abstract semantics in human mental representations (Mkrtychian et al., 2019), we introduce two more variants that assess similarity from complementary perspectives: concept-focus, and structure-focus.

Two meta-variants: concept vs. structure. To better explore *argument similarity*, we devise two variants, the view the graph from complementary angles: 1) *concept-focus*, *C-focus*. The first metric variant focuses on conceptual matches (Figure 8.1, middle), i.e. the more concrete semantic aspects, by putting more weight on concept matches, and less on relational graph structure. 2) *structure-focus*, *S-focus* puts more weight on relational graph structure than on node labels (Figure 8.1, bottom). We speculate that C-focus will obtain a better score since human notion of argument similarity may be less influenced by an argument’s abstract semantic relational structure than the concepts that are involved in the arguments. However, both may provide valuable insights to better understand argument similarity.

Concept vs. Structure with S^2 MATCH. The first metric variant puts triple weight on concept matches (all triples that are *:instance* relations). The second variant puts triple weight on relation matches (all triples that are not *:instance* relations).

Concept vs. Structure with WWLK. Similarly, we can adapt WWLK in a straightforward manner to allow C-focus, and S-focus. For C-focus, we set $k = 0$, so that the Wasserstein distance is calculated on node embeddings only. For S-focus, we set $k = 4$, which increases the communication in the graph, and thus the impact of the graph’s structure.

8.5 Argument Similarity Prediction with MR Metrics: Experiments

8.5.1 Setup

Data set and evaluation metric. We use the UKP aspect corpus (Reimers et al., 2019), which contains 3,596 argument pairs on 28 topics that have been assigned a four-way

similarity rating: highly similar (HS), somewhat similar (SS), not similar (NS), different topic/‘can’t decide’ (DTORCD). Following Reimers et al. (2019), we frame the task as a binary prediction problem: *highly similar* (HS, SS) and *non-similar* (NS, DTORCD), and we conduct evaluation via cross validation with 4 folds. In every iteration, 7 topics serve as testing data, while the other 21 topics serve to tune a decision threshold of the metric score.⁴ As in Reimers et al. (2019), we evaluate the F1 score for each of the two labels and the arithmetic F1 mean (macro F1) (Opitz and Burst, 2019).

Baselines. We compare to previously established unsupervised baselines (Reimers et al., 2019): i) *Tfidf* calculates cosine similarity between Tfidf-weighted bag-of-word vectors; ii) *InferSent-(FastText|Glove)* leverages sentence embeddings produced by the InferSent model (Conneau et al., 2017) based on either FastText (Bojanowski et al., 2016) or GloVe (Pennington et al., 2014a) vectors, which are compared with cosine similarity; iii) *(GloVe|ELMo|BERT) Embedding* uses averaged GloVe embeddings or averaged contextualized embeddings from ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019) language models.

8.5.2 Results

Best system. Table 8.1 shows our main results. The AMR-based approach that is based on concept-focused WWLK scores, taking both the argument and its inferred conclusion into account, obtains rank 1 (69.22 macro F1), closely followed by S²MATCH also with concept focus that obtains rank 3 (68.70 macro F1) – both AMR approaches are outperforming all baselines, including the BERT baseline, by up to 4 points macro F1. The difference is significant (Student t-test). This system is closely followed by other AMR-based systems, e.g., using concept-focused S²MATCH that sees *only* the argument (68.17 macro F1), and standard S²MATCH taking both argument and conclusion into account (66.21 macro F1).

Does incorporating conclusions help? Interestingly, when making similarity judgments based on *only* conclusions, we still outperform the random baseline (rank 24). The low performance of this approach, in general, is expected, since clearly, argument similarity must be primarily determined based on the arguments, and hence, methods that rate

⁴Strictly speaking, this is not a fully unsupervised setup, however, we stick to this framing of the task to facilitate comparison to the previous work (Reimers et al., 2019).

	metric	model type	F1 score			rank
			macro	sim	not sim	
	human	?	78.34	74.74	81.94	0
Baselines	random	-	48.01	34.31	61.71	24
	Tf-Idf	$f(a, a')$	61.18	52.30	70.07	17
	InfSnt-fText	$f(a, a')$	66.21	58.66	73.76	7/8
	InfSnt-GloVe	$f(a, a')$	64.94	54.72	75.17	13
	GloVe Emb.	$f(a, a')$	64.68	56.32	73.04	14
	ELMo Emb.	$f(a, a')$	64.47	53.55	75.38	15
	<i>BERT Embe.</i>	$f(a, a')$	65.39	52.32	78.48	12
	WWLK variants	AMR	$f(a, a')$	$67.0^{\pm 0.8}$	$57.5^{\pm 1.2}$	$76.4^{\pm 0.5}$
AMR		$f(c, c')$	$61.5^{\pm 0.5}$	$51.2^{\pm 0.4}$	$71.8^{\pm 0.8}$	16
AMR		$f(a \oplus c, a' \oplus c')$	$67.4^{\pm 0.4}$	$58.2^{\pm 0.7}$	$76.5^{\pm 0.4}$	5 † ‡
AMR C-focus		$f(a, a')$	$68.72^{\pm 0.8}$	$60.44^{\pm 1.4}$	$76.99^{\pm 0.3}$	2 † ‡
AMR C-focus		$f(c, c')$	$62.03^{\pm 0.3}$	$52.08^{\pm 0.4}$	$71.97^{\pm 0.4}$	16
AMR C-focus		$f(a \oplus c, a' \oplus c')$	$69.22^{\pm 0.2}$	$60.55^{\pm 0.3}$	$77.89^{\pm 0.4}$	1 † ‡
AMR S-focus		$f(a, a')$	$66.2^{\pm 0.5}$	$56.98^{\pm 0.8}$	$75.41^{\pm 0.2}$	9
AMR S-focus		$f(c, c')$	$59.93^{\pm 0.5}$	$48.3^{\pm 0.7}$	$71.57^{\pm 0.5}$	21
AMR S-focus		$f(a \oplus c, a' \oplus c')$	$65.97^{\pm 0.3}$	$56.47^{\pm 0.4}$	$75.47^{\pm 0.2}$	10
S ² MATCH variants		AMR	$f(a, a')$	$65.44^{\pm 0.5}$	$55.23^{\pm 0.8}$	$75.66^{\pm 0.4}$
	AMR	$f(c, c')$	$57.31^{\pm 0.6}$	$45.73^{\pm 1.2}$	$68.89^{\pm 0.4}$	22
	AMR	$f(a \oplus c, a' \oplus c')$	$66.21^{\pm 0.3}$	$56.98^{\pm 0.6}$	$75.42^{\pm 0.1}$	7/8
	AMR C-focus	$f(a, a')$	$68.17^{\pm 0.3}$	$59.2^{\pm 0.6}$	$77.14^{\pm 0.2}$	4 † ‡
	AMR C-focus	$f(c, c')$	$60.29^{\pm 0.5}$	$49.33^{\pm 0.4}$	$71.26^{\pm 0.8}$	20
	AMR C-focus	$f(a \oplus c, a' \oplus c')$	$68.70^{\pm 0.5}$	$60.35^{\pm 1.0}$	$77.04^{\pm 0.1}$	3 †/‡
	AMR S-focus	$f(a, a')$	$60.74^{\pm 0.5}$	$49.94^{\pm 0.8}$	$71.55^{\pm 0.5}$	19
	AMR S-focus	$f(c, c')$	$56.48^{\pm 0.3}$	$44.96^{\pm 0.6}$	$67.99^{\pm 0.2}$	23
	AMR S-focus	$f(a \oplus c, a' \oplus c')$	$61.14^{\pm 0.3}$	$49.74^{\pm 0.5}$	$72.55^{\pm 0.5}$	18

Table 8.1: Main results for argument similarity. †/‡: significant improvement over all baselines with $p < 0.05/p < 0.005$ (Student t-test).

the similarity of arguments only seeing conclusion have an obvious disadvantage. Hence, the more interesting question is: Do inferred conclusions provide complementary information for the task? Our results show a tendency that this is the case. All AMR-based models that take both conclusion and argument into account (model type $f(a \oplus c, a' \oplus c')$) outperform models that only see the arguments. At this point, however, we cannot explain whether this is due to useful summaries or truly novel content that was generated, or a mix of both. We will investigate this question more deeply in Section 8.6.

Argument similarity: driven by abstract or concrete semantics? The strong performance of the concept-focused AMR metric shows that a large overlap in concepts tends to correlate with human ratings more than an overlap in abstract semantic structure. The structure-focused AMR methods (last block in Table 8.1), while significantly outperforming the random baseline, lag behind all other baselines. Note, however, that the standard AMR-based model, which weights concept and structure overlap equally, still provides strong performance compared to all baselines, with or without additional summaries.⁵

8.6 Analyses & Explainability

While these model ablations provide a global view of what matters in argument similarity rating, we now analyze the impact of finer semantic features.

8.6.1 Fine predictors of argument similarity

The previous experiment suggests that human argument similarity ratings can be modeled through a combination of different meaning facets, with a focus on concepts. We will now investigate how human argument similarity ratings correlate with specific meaning aspects represented in AMR graphs.

Setup. For this we leverage fine-grained AMR metrics (i.e., apply SMATCH on aspectual subgraphs, c.f. bottom of Figure 3.1 in Chapter 3), and compute similarity with respect to 6 meaning aspects i) named entities (NER); ii) negation; iii) lexical concepts; iv) predicate frames; v) coreference and vi) semantic roles (SRL). Instead of merging the

⁵Motivated by this result, we conduct two extreme ablations: concept-only and structure-only metrics. While the structure-only variant shows worse results than AMR S-focus (macro F1 $\Delta f(a, a')$: -2.7), concept-only variant and concept-focused are more or less on par (macro F1 $\Delta f(a, a')$: -0.2).

predictor	Pearson's ρ vs. human gold similarity		
	$f(a,a)$	$f(c,c)$	$f(ac,a'c')$
Concepts	0.492 [‡]	0.299 [‡]	0.492 [‡]
Sem. Role Labels (SRL)	0.400 [‡]	0.185 [‡]	0.402 [‡]
Predicate Frames	0.355 [‡]	0.232 [‡]	0.357 [‡]
Reentrancies (Coref.)	0.235 [‡]	0.085 [‡]	0.235 [‡]
Named Entity (NER)	0.076 [‡]	0.052 [‡]	0.077 [‡]
Negations	0.042 [†]	-0.011	0.042 [†]

Table 8.2: Semantic predictors of human argument similarity. †/‡: significant with $p < 0.05/p < 0.005$.

labels *somewhat similar* and *similar*, we keep them distinct and use a three-point Likert scale: 0 means *not similar* or *unrelated*, 0.5 means *somewhat similar*, and 1 means *highly similar*. To assess the correlation, we use **Pearson's correlation coefficient**.

Results of this univariate feature analysis are displayed in Table 8.2. As expected from the earlier experiment, shared concepts are strong predictors for argument similarity (Concepts, $\rho=0.49$). Also more abstract semantic features, such as similar semantic roles, have a solid signaling effect (SRL, $\rho=0.40$). Similarly, coreferences have predictive capacity, though at a lower range ($\rho=0.23$). On the other hand, negation or shared named entities do exhibit only small (yet still significant) predictive capacity (Negation, $\rho=0.04$ and NER, $\rho=0.08$). The low correlation of NE overlap with human similarity ratings can in part be explained by the fact that we do not find many arguments where this could potentially matter (in our data, only 1 to 2 out of 1,000 nodes represent person NEs). However, if humans were to rate argument similarity in a dataset that features many *arguments from expert opinion* (Godden and Walton, 2006; Wagemans, 2011), named entity overlap may have a significant predictive capacity. Also negation might be more important than what we see in this analysis, since it can be expressed in alternative ways (e.g., through antonyms) that are not encoded as such in AMR.

8.6.2 Example case with alignment

To illustrate the potential of using AMR for connecting and assessing arguments, we study an example case in Figure 8.2. It shows the graphs and graph alignments⁶ that were found, for the actual arguments and their automatically induced conclusions, for our running example on fracking.

⁶The alignments were computed with S2M^{Concept+Concl.}

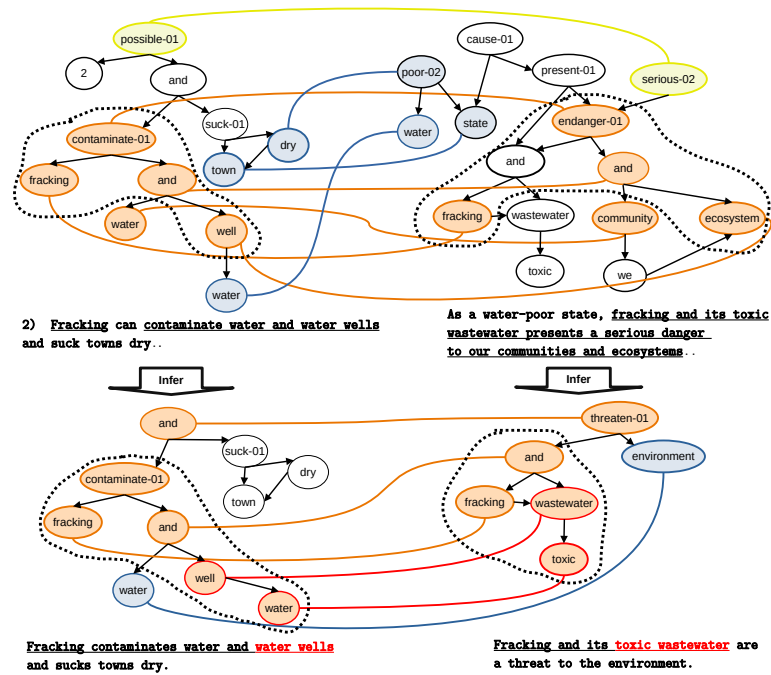


Figure 8.2: Full example (edge-labels omitted for simplified display) of explicit alignments between argument graphs (top) and automatically induced conclusions (bottom). Here, the conclusions help explaining argument similarity, since the alignment connects *fracking* in both graphs, as well as *water wells* and *toxic wastewater*, showing how *contaminating of the wells* (left graphs) actually happens: wells are polluted with toxic wastewater (right graphs).

Observations about argument alignment. The top figure shows the alignment of the two argument graphs, where important substructures have been linked. *Contamination of water and water wells* is linked to *endangering our communities and ecosystems* (orange nodes and alignment). It is also appropriate that *towns that are sucked dry* is linked to *water poor state* (blue). This link is very valuable since these statements stand in a semantic EXACERBATE-relation that may be important for the arguments' similarity (the water-poverty of states is exacerbated if towns are sucked dry). Ideally, we would like such alignments to be labeled with a corresponding semantic relation. In future work, we plan to achieve this by leveraging commonsense knowledge graphs like ConceptNet.

Observations about conclusion alignment. The bottom figure shows the alignment of the automatically deduced conclusions. For the left argument, the conclusion fails to produce an abstraction and more or less repeats the argument. For the argument on the right-hand side, however, the conclusion generator produced a more informative conclusion. From the input argument it concludes that *Fracking and its toxic wastewater are*

a threat to the environment—focusing on the negative environmental impact of fracking. This triggers a graph alignment which adds valuable new information (see clouds with dotted margins). The alignment makes explicit that *water wells* and *toxic wastewater* stand in a correspondence in the context of *fracking*. Specifically, we see how the *contamination of wells* (left graphs) happens: wells are polluted with toxic wastewater (right graphs). Additionally, the left graph helps explain parts of the meaning of the right graph: Fracking and toxic wastewater are a threat *because* fracking *contaminates* water and water wells.

8.6.3 Investigations of conclusion quality

An inferred conclusion can be more or less abstract or dissimilar from the input argument. This raises the question of the *quality* of an inferred conclusion. In fact, we can apply our AMR similarity metrics to quantify the similarity of an argument and its inferred conclusion—formally: $f(a, c)$ —which may be indicative of the *novelty* of a conclusion in relation to its premise. Hence, we investigate how AMR similarity metrics can be used to measure the *novelty* of a conclusion relative to its premise. Another aspect of conclusion quality is its *validity or justification*, i.e., to what extent it can be trusted. Clearly, a conclusion that is very similar to the premise has a high chance of being valid (as long as the premise is), whereas this is uncertain for parts of its meaning that do not match the premise.

In current research, not much is known about how to rate the quality of a conclusion drawn from an argument. We explore this question by performing a manual assessment of different quality aspects of conclusions, and investigate to what extent these can be assessed with our MR similarity metrics.

We randomly sample 100 argument-conclusion pairs per topic. The pairs are given to two annotators whom we ask to assign binary ratings regarding two questions: i) Is the conclusion *valid* based on the premise? With this we aim to assess whether the argument legitimizes the conclusion; and ii) Does the conclusion introduce some *novelty* relative to the argument? This should be denied if, e.g., the conclusion repeats the premise, and/or paraphrases it.

As shown in Figure 8.3, we measure moderate IAA, with slightly higher agreement for *novelty*. The results show that T5 often manages to produce either valid (*justification*, $\approx 65\text{-}75\%$ of cases) or novel content (*novelty*, $\approx 50\text{-}60\%$), but struggles to produce conclusions that fulfill both criteria (*justification & novelty*: $\approx 25\text{-}35\%$ of cases).

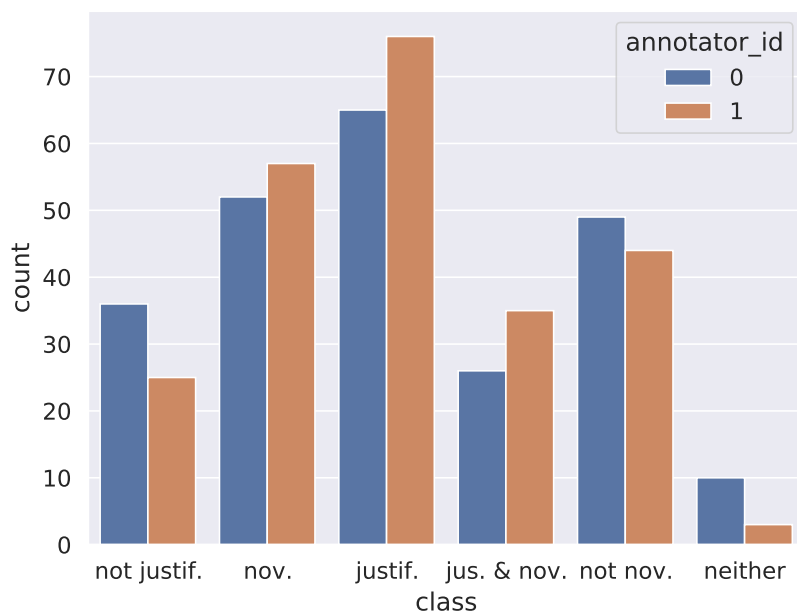


Figure 8.3: Annotation results of two quality aspects with IAA: $\mathcal{K}=0.49$ (*justification*) and $\mathcal{K}=0.57$ (*novelty*).

8.6.4 Can we predict conclusion quality?

We now extend the use of our metrics to assess conclusion quality by computing the similarity of argument and conclusion: $f(a, c)$. We calculate six graph similarity statistics of their AMRs to finally produce an aggregate score assessment: i) $|a \cap c|/|a|$ measures the relative amount of premise content that is contained in the conclusion (‘precision’); ii) $|a \cap c|/|c|$ measures the relative amount of conclusion content contained in the premise (‘recall’); iii) the harmonic mean of i) and ii) corresponds to main metric $f(a, c)$; and features iv-vi) apply a non-linear function to i)-iii), measuring the proximity to the feature means⁷, which expresses the idea that a conclusion that is both novel *and* justified may be situated at mean similarity of premise and conclusion, measured by $f(a, c)$.

We use a Linear SVM for predicting, in three binary classification tasks, either *justification*; *novelty* or *both*, using the feature set i)-vi).⁸

Results are seen in Table 8.3. Despite the small training data, performance is good for predicting *justified* (max. 68.6 F1) or *novel* (max. 70.0 F1). But predicting a conclusion to be *novel & justified* yields substantially lower performance (max. 58.3 F1), while still

⁷I.e., given the mean μ of a feature x , the new value x'_i of datum i is $x'_i = 1 - (\mu - x_i)^2$.

⁸We average all results over 25 runs of leave-one-out cross validations. When predicting either *justification*, or *novelty*, we average over the two annotators; when predicting *justification and novelty*, to increase the positive class labels slightly, the gold target are cases where one or two annotators annotated both *novel* and *justified*.

	justified	novel	both
random	0.5	0.5	0.5
i) $ a \cap c / a $	59.0 ++	58.7 --	53.4
ii) $ a \cap c / c $	68.6 +++	64.3 ---	52.4
iii) harm. mean i), ii)	61.3 +++	61.9 ---	52.2
iv) proximity to mean i	49.8	55.1 --	56.5 +
v) proximity to mean ii	35.9	52.0	58.3 +
vi) proximity to mean iii	30.5	54.4 -	58.1 +
i-vi combination	67.5	70.0	53.8

Table 8.3: Macro F1 scores for predicted conclusion quality using AMR-based models $f(a, c)$, assessing various aspects. For single features, + show positive correlation; - negative correlation (levels 0.05, 0.005, 0.0005).

above baseline. Feature correlations show that *novel* is negatively (-) associated with $f(a, c)$ (i-iii), while *justified* is positively (+) correlated with $f(a, c)$ (i-iii). We find much weaker correlation for *novel&justified*, tending to *mean* similarity (iv-vi).

Our analyses support **Hyp1** in that AMR metrics are able to rate *similarity* of arguments, of conclusions and of argument-conclusion pairs, and this also allows us to determine if a conclusion is *novel* or *justified*. While many justified conclusions are highly similar to the premise, **deciding their justification is difficult if they involve novelty**. We argue this is because *justification* cannot be determined from premises alone, but requires external knowledge. We leave this issue for future work.

8.6.5 Conclusion usefulness

Finally, we revisit our **Hyp2**, that by extending arguments with inferred conclusions, we can support assessment of argument similarity. This raises the issue of the *usefulness* of a conclusion, in terms of achieving good performance and interpretability of an argument similarity method. The aspect of the *usefulness of a conclusion* clearly differs from the question of its *quality*. For one, it is possible that a good conclusion is not useful for argument similarity rating, simply because the assessment of the paired argument premises already provides a confident and precise similarity judgment. On the other hand, a mediocre conclusion could provide complementary indications that can support the similarity judgment. In this final section we aim to assess factors that can determine this usefulness.

similarity (SIM) features					
feat. id	i	ii	iii	iv	v
	$f_{aa'}$	$f_{cc'}$	$f_{aa'} - f_{cc'}$	$(f_{ac} - f_{a'c'})/2$	hum
P. 's ρ	0.83	-22.81	26.6	-9.37	-14.72
p-value	> 0.05	$1.2e^{-43}$	$2.8e^{-59}$	$1.8e^{-8}$	$7.3e^{-19}$

Table 8.4: Predictors of conclusion usefulness.

Operationalizing conclusion usefulness. We define a score \mathcal{U} for the usefulness of a conclusion, based on a human rating y , the conclusion similarity $f(c, c')$ and argument similarity $f(a, a')$, as

$$\mathcal{U} = \frac{1}{1 + (y - f(c, c'))^2} + (y - f(a, a'))^2, \quad (8.2)$$

where \mathcal{U} is maximized *iff* the automatic similarity rating of the conclusions does not differ from the human rating, while the automatic similarity rating of the premises differs maximally from the human rating. It is in exactly these situations that a conclusion assessment will prove most useful.

Features for assessing conclusion usefulness \mathcal{U} . We assume the following features for modeling the usefulness of a conclusion, which we compute with our **similarity function** f : i) the similarity of the arguments $f(a, a')$; ii) the similarity of the conclusions $f(c, c')$; iii) the (signed) difference between the argument and the conclusion similarities $f(a, a') - f(c, c')$; iv) we compute the (signed) difference between the similarity of (a, c) and (a', c') : $\frac{f(a,c) - f(a',c')}{2}$; finally, v) y is the human rating.

Results. Table 8.4 shows that the highest predictive power for conclusion *usefulness* is feature iii): the similarity of the two arguments *minus* the similarity of the two conclusions. It exhibits a highly significant positive correlation with conclusion usefulness, and relates to the following scenario: If two arguments are considered to be similar, *but* the conclusions as dissimilar, this may signal that the arguments are rated dissimilar by the human, and the high initial rating may be reconsidered.

Table 8.5 shows a data sample where the conclusions help to correct an initial, over-optimistic similarity rating of the premises. The premises are rated *dissimilar* by the human, but since they contain similar concepts, such as *saving lives*, the AMR metric assigns

a)	<i>Because you may save up to eight lives through organ donation and enhance many others through tissue donation.</i>
c)	<i>organ donation is a great way to save up to eight lives.</i>
a')	<i>This medical research is important to understanding diseases in humans so that lives may be saved and improved.</i>
c')	<i>medical research is important to understand diseases in humans</i>

Table 8.5: AMR metrics detecting dissimilar arguments.

a high similarity rating (0.7) to the pair (a, a') . However, the automatically generated conclusions (c, c') are assigned low(er) similarity (0.2). The low rating can be explained by the fact that the conclusion generator has distilled different conclusions from the premises that reflect the different foci of the arguments: the first proposes that organ donations are good for saving lives, while the second argument proposes that generally more medical research should be conducted.

8.7 Discussion

Explanation dimensions. Our argument similarity rating approach may provide explanations in various dimensions. i) First and foremost, the explicit alignment and similarity computation based on MR and MR graph metrics, by relating similar concepts between arguments and their conclusions, provides insight into which components of two argument MRs relate to each other, with individual alignment scores, and to what extent they contribute to the overall score. Especially in light of recent observations showing supervised models to be prone to superficial cues in data sets (Opitz and Frank, 2019c; Niven and Kao, 2019; Heinzerling, 2020; Jo et al., 2021), this property is desirable. ii) We apply the fine-grained MR metrics according to the subgraph principle (c.f. 3.1, bottom) to assess semantic phenomena, such as negation or semantic roles. This can further **illuminate in which ways an argument pair is similar/dissimilar**. iii) By taking into account the similarity of automatically inferred conclusions, the similarity computed for premises may be re-adjusted in case the similarity of the inferred conclusions strongly differs.

The MR similarity statistics also enabled us to gain some first indications of what

could be considered a good conclusion (without a reference): e.g., our qualitative evaluations indicate that good conclusions tend to be neither very similar, nor very dissimilar to the premise. This seems plausible, since (too) high similarity may indicate a mere summary (reducing novelty), while (too) low similarity may indicate a lack of coherence (reducing validity).

Our approach also hinges on the quality of the inferred conclusions. The conclusions we obtained are often either *justified* or *novel*, but less often satisfy both conditions. In addition, we find that the degree of novelty is often rather small, perhaps reflecting that the T5 generator was pre-trained on summarization data and hence may tend to produce inferences that are *not* novel, since novelty is not a common characteristic of a summary. On the positive side, our approach can be fueled by an increasing amount of research on argument conclusion generation (Alshomary et al., 2020, 2021). In general, and particularly for our approach, it will be interesting to work with systems that produce not only a single, but multiple valid conclusions. Considering relations *across and within two conclusion sets* inferred from two premises using AMR metrics may provide key information on argument similarity.

Finally, by measuring the similarity of premises and their conclusions, our approach could shed light on another important question: *how to assess novelty and justification of a conclusion without reference?* This is an important question for research on argument conclusion generation since it lacks methods that can judge the quality of conclusions in the absence of (costly) references.⁹

Summary. We investigated two hypotheses: i) MR meaning representation and graph metrics help in assessing argument similarity, ii) automatically inferred conclusions can aid or reinforce the similarity assessment of arguments. We find solid evidence for the first hypothesis, especially when slightly adapting MR metrics to focus more on concrete concepts found in arguments. We find weak evidence that supports the second hypothesis, i.e., metrics improve consistently, but by small margins, when they are allowed to additionally consider the MRs of automatically inferred conclusions. We believe, however, that more substantial gains may be obtained in future work, by improving conclusion generation models such that they produce content that is both *valid and novel*. Finally, we

⁹Follow-up research on rating conclusion quality with regard to novelty and validity is performed by (Heinisch et al., 2022b) who propose robust training regimes and (Plenz et al., 2023) who infer commonsense knowledge graphs. More insights can be found in systems of the ArgMining 2022 shared task (Heinisch et al., 2022a) with the best performing systems using either knowledge (Saadat-Yazdi et al., 2022) or language model prompting (Meer et al., 2022).

have made first steps towards a *reference-less* metric for assessing novelty and justification of generated conclusions.

Chapter 9

Building efficient *and* effective similarity models from MR metrics

9.1 Chapter outline

Semantic similarity permeates many areas of NLP. Among others, it is a vital part of document search and information retrieval systems. Considering different axes of similarity, clearly we would like to say that two documents are similar if their meaning is similar. Thus, it seems very interesting to investigate MR metrics that perform the similarity assessment *directly* in the space of meaning. However, when using MR metrics simply “off the shelf”, we might run into *two major issues*. The first issue is a *efficiency bottleneck*: graph metric computation is typically costly, especially when conducted pair-wise over a larger set of documents, and parsing also tends to be quite slow and can require heavy additional machinery. Second, while we observed that MR metrics can outperform baselines such as bag-of-words and on top of this provide explanatory evidence with an alignment between meaning structures, MR metrics most probably lag behind in *accuracy* when one compares their performance against the performance of neural sentence embeddings that are derived from large pre-trained language models, since, without further improvements, they may not have learned to well take into account the importance of different meaning parts (e.g., consider that SMATCH assigns every meaning triple the same weight), a topic that we have already touched on in multiple places in this thesis, for instance, Section 2.4 in our Background 2.

To alleviate these anticipated problems of MR metrics (i.e, the *efficiency bottleneck*

and lack of *state-of-the-art accuracy*), in this last part of the thesis we first i) explore a strategy that allows us to emulate an NP-complete MR metric for extremely efficient application. Second, as the final part of this thesis, we show how to ii) use MR metrics as a signal to meaningfully structure neural embedding spaces into explainable features. This yields the best of two worlds: *strong* and *efficient* neural sentence embeddings that are also *explainable*. The remainder of this chapter is structured as follows:

1. In a pilot study (Section 9.2), we show that we can mitigate the NP-bottleneck of a graph metric through approximation strategies based on neural networks and data augmentation tricks.
2. We note that *accurate graph alignment* does not necessarily imply *a similarity rating that resembles that of a human*, the latter being crucial for semantic search (Section 9.3). We describe a solution by proposing semantically structured neural text embedding spaces (*S³BERT*): Under the guidance of MR metric signals, our method learns to decompose a document embedding into different semantic features, yielding an explainable, efficient and powerful vector space (Sections 9.4 and 9.5). The method is very general and can be adapted/customized for many different use-cases that require efficiency, accuracy, and explainability.
3. Starting in Section 9.6, we conduct extensive intrinsic and extrinsic evaluation of our semantic decomposition method. In Section 9.7, we assess explainability performance, and in Section 9.8 we examine performance on three diverse downstream similarity rating tasks. Lastly, we showcase our method in example studies, explaining similarity decisions for documents, and conducting a structural semantic data set difference analysis (Section 9.9)
4. As always, we conclude with a discussion (Section 9.10).

Underlying work. The content of this chapter is mainly based on works by Opitz et al. (2023a) and Opitz and Frank (2022b).

9.2 Pilot study: Fast similarity with learned SMATCH

A well-known bottle-neck of graph metrics that may limit more exploration, testing, and usage, is their time-complexity: Notably, computation of SMATCH can take more than a

minute to compare some 1,000 MR pairs (Song and Gildea, 2019). To understand that this can become problematic in many setups, consider a hypothetical user who desires exploring a (small) MR-parsed corpus with only $n = 1,000$ instances via clustering. The (symmetric) SMATCH needs to be executed over $(n^2 - n)/2 = 499,500$ pairs, resulting in a total time of more than 6 hours. The time complexity also negatively impacts MR evaluation time (Song and Gildea, 2019), as well as parsing efficiency of approaches involving re-inforcement learning (Naseem et al., 2019) or graph ensembling (Hoang et al., 2021), where MR metrics need to be run with high frequency. Furthermore, given recent interest into extended application settings, such as exploring MR for semantic search. Recently, there is a growing interest for using MR metrics in semantic search (Bonial et al., 2020; Müller and Kuwertz, 2022), we anticipate that this problem will become more pressing in the future.

As a first solution to mitigate these issues, we propose a method that learns to efficiently match MR graphs from a teacher SMATCH, thereby reducing MR clustering time from hours to seconds.

More precisely, we will:

1. Explore three different neural approaches that learn to synthesize SMATCH from scratch.
2. And show that we can approximate SMATCH up to a small error, by leveraging novel data augmentation tricks.

9.2.1 Learning NP-hard graph metric: problem definition and models

Recall that the SMATCH metric measures the structural overlap of two MR graphs (c.f., Section 3.1 in the Related Work 3). We i) compute an alignment between variable nodes of MRs and ii) assess triple matches based on the provided alignment. Formally, we start with two MR graphs a and b with variable nodes $X = (x_1, \dots, x_n)$ and $Y = (y_1, \dots, y_m)$. The goal is then to find an optimal *alignment*

$$\text{map}^* : X \rightarrow Y, \tag{9.1}$$

searching for a *map* that maximizes the number of *triple matches* for the two graphs. E.g., assume $\langle x, \text{arg0}, y \rangle \in a$ and $\langle v, \text{arg0}, u \rangle \in b$. Recall that if $x = v$ and $y = u$, we count *one*

triple match. Again, also for the general SMATCH formula, we refer the reader, e.g., to Eq. 3.1 in the Related Work 3.

Setup

Experimental data creation. We create the data for our experiments as follows: 1. We parse 59,255 sentences of the LDC2020T02 AMR dataset with a parser (Lyu and Titov, 2018) to obtain graphs that can be aligned to reference graphs; 2. For every parallel graph pair $pair = (a, b)$, we use SMATCH (ORACLE) to compute an F1 score s and the alignment map^* . We shuffle the data and split it into training, development and test set (56255-1500-1500).

Objective and approach. The task is to reproduce the teacher ORACLE as precisely as possible. We design and test three different approaches. The first is indirect, in that it predicts the alignment, from which we compute the score. The second directly predicts the scores. The third approach enhances the second, to make it even more efficient.

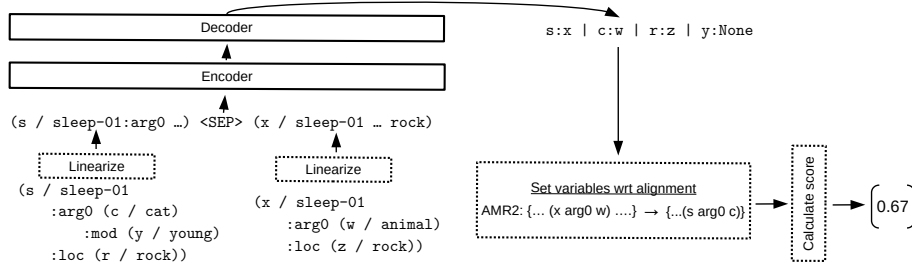


Figure 9.1: Seq2seq SMATCH alignment-learner.

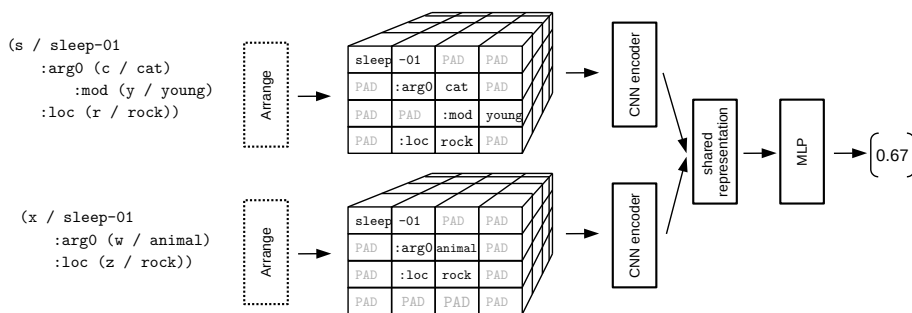


Figure 9.2: Implicit CNN-based SMATCH graph metric predictor.

Synthesis option I: Alignment learning

Here, we aim to learn the alignment itself (Eq. 9.1) with an NMT model, as illustrated in Figure 9.1. For the input, we linearize the two AMRs and concatenate the linearized token sequences with a special $\langle \text{SEP} \rangle$ token. The output consists of a sequence $x_j:y_k \dots x_i:y_m \dots$ where in every pair $u:v$, u is a variable node from the first AMR mapped to a node v from the second AMR. The SMATCH score is then calculated based on the predicted alignment.

To predict the node alignments/mapping of variables, we use a transformer based encoder-decoder NMT model. Details about the network structure and hyperparameters are stated in Appendix A.2.

Synthesis option II: SMATCH prediction

In this setup, we aim to predict SMATCH F1 scores for pairs of AMRs directly, in a single step. This means that we directly learn SMATCH F1 with a neural network and our target is the ORACLE F1 score.

To learn this mapping, we adapt the convolutional neural network from Chapter 7, as shown in Figure 9.2. The model was originally intended to assess AMR accuracy, i.e., measuring AMR parse quality without a reference. Taking inspiration from human annotators, who exploit a spatial ‘Penman’ arrangement of AMR graphs for better understanding, it models directed-acyclic and rooted graphs as 2d structures, employing a CNN for processing, which is highly efficient. To feed a pair of AMRs, we remove the dependency graph encoder of the model and replace it with the AMR graph encoder. Moreover, we increase the depth of the network by adding one more MLP layer after convolutional encoding. A basic mean squared error is employed as loss function. More details about hyperparameters are stated in Appendix A.3.

Synthesis option III: AMR Vector learning

Inspired by Reimers and Gurevych (2019), we aim to make the CNN even more efficient, by alleviating the need for pair-wise model inferences. Hence, instead of computing a shared representation of two CNN-encoded AMRs, we process each representation with an MLP (w/ shared parameters), to obtain two AMR vectors $NN(a)$ and $NN(b)$. These

vectors are then tuned with a distance loss \mathcal{L} against ORACLE score s :

$$\mathcal{L} = \sum_{(a,b,s)} \left([1 - |NN(a) - NN(b)|] - s \right)^2, \quad (9.2)$$

where $||$ is returns a vector distance $\in [0, 1]$. This approach enables extremely fast search and clustering. Indeed, the required (clustering-)model inferences are $O(n)$ instead of $O(n^2)$, since the model can infer a vector for each graph individually (allowing application of simple vector algebra for similarity).

Data compression and extension tricks

Vocabulary reduction trick. The SMATCH metric measures the structural overlap of two graphs. This means that we can greatly reduce our vocabulary, by assigning each graph pair a *local vocabulary* (see Figure 9.3, ‘anonymize’).

First, we gather all nodes from two graphs a and b , computing a joint vocabulary over the concept nodes. We then relabel the concepts with integers starting from 1. E.g., consider AMR a : ($r / run-01 :arg0 (d / duck)$), and AMR b : ($x / run-01 :arg0 (y / duck) :mod (z / fast)$). The gold alignment is $map^* = \{(r,x), (d,y), (\emptyset, z)\}$. Now, we set the shared concepts and relations to the same index $run=run=1$ and $duck=duck=2$ and $:arg0=:arg0=3$ and distribute the rest of the indices $r=4, d=5, x=6, y=7, z=8, fast=9, :mod=10$. This yields equivalent AMRs $a' = (4 / 1 :3 (5 / 2))$ and $b' = (6 / 1 :3 (7 / 2) :10 (8 / 9))$. The target alignment then equals $map^* = \{(4,6), (5,7), (\emptyset, 8)\}$. This strategy greatly reduces the vocabulary size, in our case from 40k tokens to less than 700.

Auxiliary data creation trick. We also find that we can cheaply create auxiliary gold data. We re-assign different indices to AMR tokens, and correspondingly modify the ORACLE alignment (Figure 9.3, ‘permute’). In our experiments, we permute the existing token-index vocabularies 10 times, resulting in a ten-fold increase of the training data. We expect that, with this strategy, the model will better learn properties of permutation invariance, which in turn will help it synthesize the algorithm.

9.2.2 Evaluation

Output post-processing. For the score synthesis (Option II) and vector synthesis (Option III), no further post-processing is required, since we directly obtain the estimated

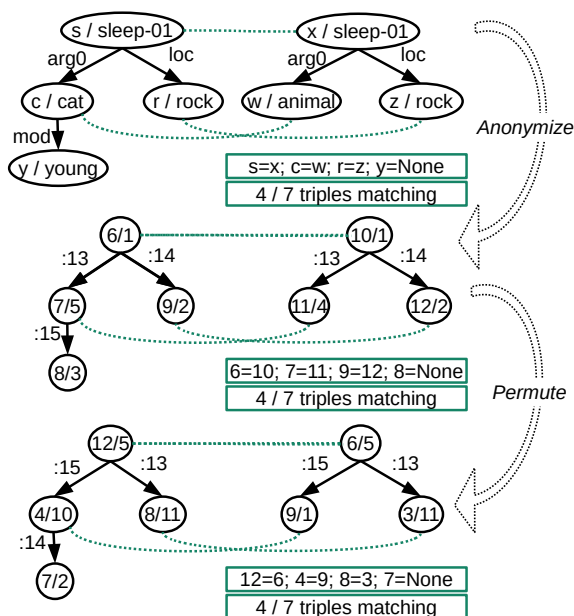


Figure 9.3: AMR graph anonymization and permutation.

SMATCH scores as output. In the explicitly synthesized alignment algorithm, however, we get map , which is the predicted alignment from the sequence-to-sequence model. In this case, we simply feed map as an argument into the SMATCH Eq. 3.1, to obtain the scores.

Evaluation. We compare the predicted scores \hat{y} against the gold scores y with Pearson’s ρ . However, for the model that predicts the explicit alignment (Option I), we can compute another interesting and meaningful metric. For this, we first calculate the average SMATCH score over AMR pairs given the gold alignment map^* , and then we calculate the average SMATCH score over AMR pairs given the predicted alignment \widehat{map} using the SMATCH Eq. 3.1. Note, that the SMATCH score based on the gold alignment constitutes an upper bound (max). Therefore, the SMATCH score based on the predicted alignment shows us how close we are to this upper bound. Our baseline consists of scores that are computed from a random alignment ($random$).

Results (Table 9.1). Our best model is the NMT approach using both data augmentation tricks. Obtaining 98.4 ρ , it very closely approximates the ORACLE, while being about 30 times faster than ORACLE and 76.2 points better than the random baseline. Perhaps the best tradeoff between speed and approximation performance is gained by the simple CNN score synthesis (96.8 ρ , 200x faster than ORACLE), also using both data tricks.

	data trick	Eq. 3.1	Pearson's ρ	time ^(secs)
ORACLE	na	77.5	100	28680
rand. baseline	na	13.5	22.2	0.4
align. synthesis		39.0	52.8	1089
align. synthesis	voc	64.5	80.0	1089
align. synthesis	voc+aug	76.4	98.4	1089
score synthesis		na	<u>87.5</u>	140
score synthesis	voc	na	82.0	140
score synthesis	voc+aug	na	96.8	140
vector synthesis		na	84.7	0.7
vector synthesis	voc	na	75.6	0.7
vector synthesis	voc+aug	na	94.2	0.7

Table 9.1: Results of experiments. time: Approximate time for computing a pair-wise distance matrix on 1k AMRs on a TI 1080 GPU.

The vector synthesis falls a bit shorter in performance (94.2 ρ), but it is extremely fast and achieves a 40,000x speed-up compared to ORACLE and about 1500x compared to the NMT approach.¹

Consistently, the data extension (*aug*) is very useful. However, the vocabulary reduction (*voc*) is only useful for the NMT model (+27.2 points), whereas the scores are lowered for the CNN-based models (−5.5 for score synthesis, −9.1, *vector synthesis*). We conjecture that the CNNs learn SMATCH more indirectly by exploiting token similarities in the global vocabulary, and therefore struggle more to build a generalizable algorithm, in contrast to the bigger NMT transformer that learns to assess tokens fully from their given graph context.

9.2.3 Discussion

We tested methods for learning to solve the hard graph matching problem, exploring different neural architectures, and data augmentation strategies that help models to generalize. Our best models increase SMATCH calculation speed by a large factor while incurring only small losses in accuracy that can be tolerated in many use cases.

A noteworthy limitation of all tested methods, however, remains the alignment of larger MR graphs with many variables. On one hand, when the alignment candidate

¹Note also that all models in Table 9.1 are significantly better ($p < 0.001$) than the random baseline (one-sided test w/ z-transform).

data type	data size	Δ vs. ORACLE		
		SMATCH F1	Pea's ρ	better
full	1500	-1.1	-1.6	-
< 5 vars	505	-0.6	-1.2	yes
< 10 vars	1041	-0.7	-1.2	yes
< 15 vars	1206	-0.9	-1.2	yes
< 20 vars	1353	-0.9	-1.2	yes
< 25 vars	1449	-0.9	-1.3	yes
> 5 vars	940	-1.5	-2.2	no
> 10 vars	476	-2.1	-3.6	no
> 15 vars	318	-2.3	-5.4	no
> 20 vars	183	-3.0	-10.1	no
> 25 vars	83	-4.7	-19.3	no
> 30 vars	37	-8.0	-25.5	no
> 35 vars	20	-12.5	-41.1	no
single snt AMRs	1421	-1.0	-1.5	yes
multi snt AMRs	79	-2.7	-9.6	no

Table 9.2: Experiments on different test subsets that represent different problem complexities predicted with our best model (*align. synthesis+voc+aug*). $\langle x \rangle$ vars means that one of two graphs contains $\langle x \rangle$ variables. *better*: is the drop in accuracy of the model vs. ORACLE smaller compared with the model tested on all data?

space increases, the runtime of SMATCH increases exponentially, while our considered approaches remain fast. However, in such a scenario, the neural models are bound to trade in some accuracy. Table 9.2 assesses the effect size for differently sized alignment candidate spaces: while the model overall copes with different search space sizes, the accuracy loss is more considerable for large problems. We conclude that the fast and accurate alignment of *larger* AMR graphs remains a challenging and unsolved problem. In this regard, we believe that our proposed data extension trick in combination with long-sequence transformers (Beltagy et al., 2020; Rae et al., 2020; Choromanski et al., 2021) may provide valuable means to address this limitation in future work.

9.3 *Efficient, explainable and effective similarity metrics*

It has become widely known that models based on large-pretrained language models, such as S(entence)BERT, provide effective and efficient sentence embeddings that show high correlation to human similarity ratings. However, they lack interpretability. On the other hand, in our thesis we've become aware that graph metrics for MRs are interpretable since they can make explicit the semantic aspects in which two sentences are similar. However, we also know that the MR metrics tend to be slow, rely on parsers, and most likely are too naïve to reach state-of-the-art performance when rating sentence similarity (as opposed to the former metrics from large language models that are learned from very large data).

Therefore, in the last part of this thesis, we aim at the best of both worlds, by learning to induce Semantically Structured Sentence sentence transformer embeddings (S³BERT). Our S³BERT embeddings are composed of explainable sub-embeddings that emphasize various semantic sentence features (e.g., semantic roles, negation, or quantification). We show how to i) learn a decomposition of the sentence embeddings into semantic features, through approximation of a suite of interpretable MR graph metrics, and how to ii) preserve the overall power of the neural embeddings by controlling the decomposition learning process with a second objective that enforces consistency with the similarity ratings of an SBERT teacher model. In our experimental studies, we show that our approach offers interpretability – while fully preserving the effectiveness and efficiency of the neural sentence embeddings.

9.4 Structuring embedding spaces with graph metric guidance

Preliminary I: Neural (e.g., SBERT) sentence embeddings and similarity. Let SB be a function that maps an input sentence s to a vector $e \in \mathbb{R}^d$. Given two sentence vectors $e = SB(s)$ and $e' = SB(s')$, we can compute, e.g., the cosine similarity of sentences:

$$\text{sim}(e, e') = \frac{e^T e'}{|e||e'|}. \quad (9.3)$$


Preliminary II: AMR and AMR metrics. Recall that an MR $a \in G$ represents the meaning of a sentence in a directed acyclic graph. The graph makes key aspects of meaning explicit, e.g., semantic roles or negation. Hence, given a pair of AMR graphs $\langle a, b \rangle \in A \times A$, an MR metric can measure *overall* graph similarity, or similarity with respect to *specific aspects*, if we execute it on dedicated subgraphs. We denote such a metric as

$$m^k : G \times G \rightarrow [0, 1], \quad (9.4)$$

where k indicates a particular semantic aspect, in view of which the graphs' similarity is assessed, e.g. negation. The AMR metrics we will apply in this study will be described in more detail in Section 9.5.

9.4.1 Structured embedding spaces: Formal problem definition and objective

Problem statement. We aim to shape neural sentence embeddings in such a way that different sub-embeddings represent specific meaning aspects. This process of *sentence embedding decomposition* is illustrated in Figure 9.4 (right): SBERT produces two embeddings e and e' that consist of sub-embeddings $F_1 \dots F_K, R$ and $F'_1 \dots F'_K, R'$. E.g., F_k may express negation features, while F_z expresses semantic role features of a sentence. The residual R offers space to model sentence features not covered by the pre-defined set of semantic features.

Having established such decompositions, we can compute, e.g., sentence similarity with respect to semantic roles ($k = SRL$) by choosing subspaces $F_{SRL} \subset e = SB(s)$ and $F'_{SRL} \subset e' = SB(s')$, and calculating $\text{sim}(F_{SRL}, F'_{SRL})$ on the subspaces. This is indicated as  in Figure 9.4.

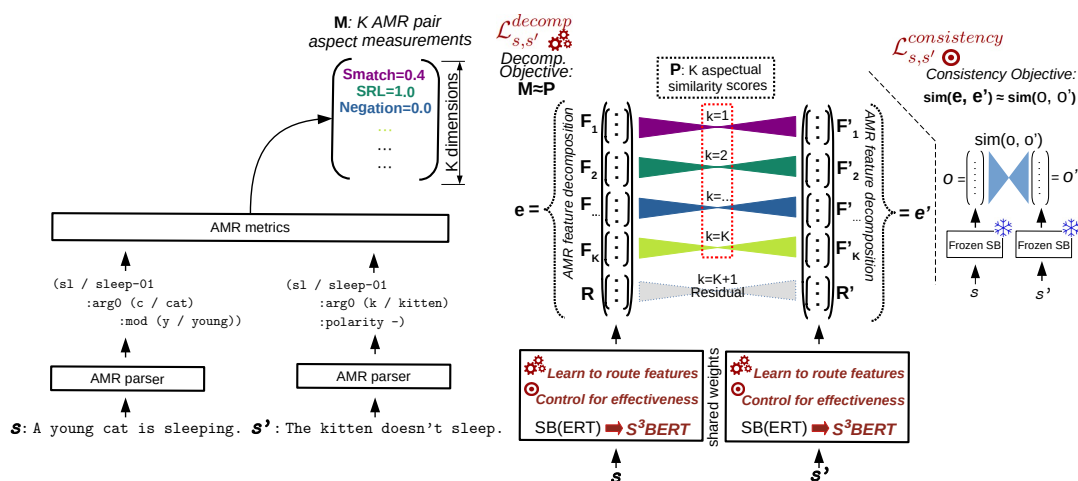


Figure 9.4: Overview of approach. \star The decomposition objective structures the sentence embedding space into AMR sentence features ($F_1 \dots F_K$): The process is guided by AMR metric approximation, through which S^3BERT learns to disentangle and route the features. \odot The consistency objective is aimed at preventing catastrophic forgetting: To preserve the overall effectiveness of the neural sentence embeddings, it controls the decomposition learning process and helps modeling the residual (R).

Assigning embedding dimensions to features. For convenience, let $i: \{1 \dots K\} \rightarrow [0, d] \times [0, d]$ denote an AMR aspect-embedding assignment function where d is the dimension of the (full) sentence embedding. This allows us to map any semantic category to a range of specific sentence embedding indices. E.g., a h -dimensional embedding for SRL sentence features for a sentence s can be accessed via $SB(s)_{i(SRL)}$, where $v_{(start, end)}$ yields all dimensions from $start$ to end of a vector v . Since we aim at a non-overlap decomposition, we ensure that $i(k) \cap i(k') \neq \emptyset$ iff $k = k'$. We call a model consisting of a tuple (SB, i) , a Semantically Structured Sentence transformer, in short S^3BERT .

9.4.2 Learning to partition the semantic space

We presume that a standard pre-trained neural sentence embedding model (e.g., SBERT) already contains some semantic features in some embedding dimensions. Hence, we want to achieve an arrangement of the embedding space according to our pre-defined partitioning, but also give it the chance to instill new knowledge about AMR semantics.

In addition, to preserve the neural model's high accuracy, we aim to control the decomposition process in a way that lets us route internal semantic knowledge *not* captured by AMR to the residual embedding. To this end, we propose a two-fold objective: *Score decomposition* and *Score consistency*.

Target scores from AMR metrics. We build an AMR metric target \mathbf{M} as shown in Figure 9.4 (left). Two AMRs, constructed from two sentences, are assessed with AMR metrics in K semantic aspects (Eq. 9.4) yielding $\mathbf{M} \in \mathcal{M} = \mathbb{R}^K$. Additionally, let \mathbf{P} be S³BERT’s AMR metric predictions, i.e., $\mathbf{P} = [\text{sim}(F_1, F'_1), \dots, \text{sim}(F_K, F'_K)]$.

For a training instance (s, s', \mathbf{M}) , we calculate the following decomposition loss:

$$\mathcal{L}_{s,s'}^{\text{decomp}} = \frac{1}{K} \sum_{k=1}^K \left[\mathbf{M}_k - \underbrace{\beta^k \text{sim}(SB(s)_{i(k)}, SB(s')_{i(k)})}_{\mathbf{P}_k} \right]^2, \quad (9.5)$$

with β^k a learnable scalar for easier projection onto a specific AMR metric’s scale. The objective is also outlined as $\mathbf{P} \approx \mathbf{M}$ in Figure 9.4.

Note that AMR graphs and metrics are only needed for training, not for inference.

9.4.3 Preventing catastrophic forgetting

When training S³BERT only with the *decomposition objective* (Eq. 9.5), there is a great risk it will unlearn important information, since it is unrealistic to expect that sentence similarity can be *fully* composed from the K aspects measured by MR metrics. We also know that MR metrics lag behind pre-trained neural embedding models in similarity rating accuracy, so forcing the MR metrics as a target could further risk the loss of useful neural prior information. Hence, we control the decomposition learning process to include a *residual* sub-embedding, to rescue important parts of semantic information not captured by MR and MR metrics. To this end, we propose a *consistency objective*.

Given a frozen SBERT (SB^*), and a training example (s, s') :

$$\mathcal{L}_{s,s'}^{\text{consistency}} = \left(\text{sim}(SB^*(s), SB^*(s')) - \text{sim}(SB(s), SB(s')) \right)^2.$$

Less formally, this means that the control is established by imposing that S³BERT’s overall similarity ratings be in accordance with a frozen SBERT’s original ratings, but otherwise leaving freedom for the choice of structure in S³BERT’s embedding space. Given independence of pairwise-targets, we can compute the loss efficiently on b^2 examples in batches of size b .

9.4.4 Global objective

We finally combine the *consistency objective* and the *decomposition objective*. The cumulative loss for a batch $B = \{(S_i, S'_i, \mathcal{M}_i)\}_{i=1}^b$ is

$$\mathbf{L} = \frac{\alpha}{b} \sum_{i=1}^b \mathcal{L}_{S_i, S'_i}^{decomp} + \frac{1}{b^2} \sum_{i=1}^b \sum_{j=1}^b \mathcal{L}_{S_i, S'_j}^{consistency}, \quad (9.6)$$

where α weighs the two parts (we use $\alpha = 1$).

9.5 AMR metrics and data construction

In Section 9.4, Eq. 9.4, we formally described an MR metric. Now we consider the concrete metric instances we will use for S³BERT decomposition. We here distinguish *general* metrics that assess global AMR graph similarity, and *aspectual* metrics that aim at assessing AMR similarity with respect to specific semantic categories, e.g., semantic roles.

9.5.1 Global AMR similarity

SMATCH for assessing the structural overlap of two semantic MR graphs (c.f., 3.1).

WLK and WWLK for more modulated similarity of MR graphs (c.f., 4.6).

9.5.2 Aspectual AMR similarity

FINESMATCH for calculating SMATCH on interpretable MR subgraphs. Here we use **Frames**: graph similarity with regard to predicates. **Named entity**: graph similarity based on named entity substructures (*person, city, ...*). **Negation**: graph similarity based on expressions of negation. **Concepts**: graph similarity based on node labels only. **Coreference**: graph similarity focused on co-referent structures. **SRL**: graph similarity considering predicate substructures. Finally, **Unlabeled**: not considering semantic edge labels.²

Additionally, we observe that AMR contains information about quantifiers and define **quantSim**, which measures the (normalized) overlap of quantifier structure of two AMRs.

²We follow our setup from Chapter 7 and set metric values to 1.00 (as opposed to 0.00) in cases where neither of the graphs contains structures of the given aspect (e.g., named entities are absent from both graphs), since the graphs can then be considered to (vacuously) agree in the given aspect.

Although AMR lacks modeling of quantifier scope (Bos, 2016), estimating the overlap of quantificational structure can give indications of semantic sentence similarity.

Graph statistics. In addition, we introduce graph metrics that target other aspects modeled by AMR: **MaxIndegreeSim**, **maxOutDegreeSim** and **maxDegreeSim**. From each graph in a pair of AMRs, we extract the node that is best connected (either outdegree, indegree, or indegree+outdegree). We compare these nodes with cosine similarity using GloVe embeddings (Pennington et al., 2014a). The motivation for this is that two Meaning Representations that share the same focus are more likely to be similar (Lambrecht, 1996). Similarly, **rootSim** compares the similarity of MR roots, motivated by Cai and Lam (2019), who speculate that more important concepts are closer to the root.

9.5.3 Data setup

For the decomposition objective we need training instances of paired sentences with AMR metric scores attached. We proceed as follows:

1) We collect 1,500,000 sentence pairs from data sets that contain sentences that can roughly be viewed as similar.³ 2) We parse these sentences with a good off-the-shelf AMR parser.⁴ 3) For each training sentence pair we create a positive (a, a^+) and a negative (a, a^-) datum, where the negative pair is formed by replacing AMR a^+ with an AMR sampled from a random pair. Thereby we show S³BERT both AMR metric outputs computed from similar AMRs, and unrelated AMRs (that may still share some abstract semantic features). 4) We execute our AMR metrics (c.f. Section 9.5.1 & §9.5.2) over all pairs from step 3). Step 4) took approx. 3 days, since AMR metrics tend to have high computational complexity.

For experimentation, we cut off a development and testing set with 2,500 positive pairs each.⁵

³AllNLI, CoCo, flickr captions, quora duplicate questions.

⁴<https://github.com/bjascob/amrlib> The parser is based on a fine-tuned T5 (Raffel et al., 2020) language model and reports more than 80 Smatch points on AMR3. On a GPU Ti 1080 the parsing took approx. 3 weeks.

⁵Using only similar sentence pairs for validation increases the AMR metric prediction difficulty and provides a useful lower bound for correlation.

9.6 Evaluation Study Setup

Our two objectives aim at creating S³BERT embeddings by partitioning a sentence transformer’s output space into features that capture different semantic AMR aspects, while controlling the decomposition process such that we prevent any forgetting of knowledge and preserve the power of the neural embeddings.

Hence, two key questions need to be addressed:

- 1.) Will S³BERT partition its sentence embedding space into interpretable semantic aspects?
- 2.) If so, what is the price? Does our consistency objective succeed in controlling the decomposition process such that it retains SBERT’s extraneous knowledge of sentence semantics?

Basic setup. We use a standard SBERT model⁶ with 11 layers and allow tuning of the last two layers. The sentence embedding dimension is $d = 384$, the sub-embedding dimension is set to $h = 16$ for all 15 aspects of AMR, which implies that the dimension of the residual is $384 - (15 \times 16) = 144$. More details on the model architecture and the training hyper-parameters can be found in Appendix A.5. In all result tables, † indicates statistically significant improvement over the runner-up (Student t-test, $p < 0.05$, five random runs)

9.7 Evaluation of S³BERT space partitioning

Our goal is to make SBERT embeddings more interpretable, by partitioning the sentence embedding space into multiple semantically meaningful sub-embeddings. We now aim to answer research question 1) whether these sub-embeddings relate to the AMR metric aspects they were trained to predict.

Data setup. We use the 2,500 testing sentence pairs we had split from our generated data. For each semantic aspect, we calculate cosine similarities of the corresponding sub-embeddings. We then calculate the Spearmanr correlation of these predictions vs. the ground truth AMR metric similarities.

⁶Pre-trained All-MiniLM-L12-v2 from the sentence transformers library.

Baseline setup. We consider three baselines. Same as S^3 BERT, all baselines are based on standard SBERT model.⁶

SB-full (no partitioning): We use the complete embedding, which means that we predict the same value for all AMR aspects. This baseline is bound to provide strong correlations with most metrics (in our experiments on BAMBOO₃, Section 5.3, we have shown that AMR metrics correlate with human sentence similarity, and so does SBERT), but obviously lacks the interpretability we are aiming for. We therefore instantiate two more baselines that can be directly compared, since they partition the space according to semantic aspects.

SB-rand (partitioning): We assign 16 embedding dimensions randomly to every semantic aspect.

SB-ILP (partitioning): We use an integer linear program to assign the semantic aspects to different SBERT dimensions. We create a bi-partite weighted graph with node sets (V_{SB}, V_{SEM}) with SBERT dimensions (V_{SB}) , and the targeted semantic aspects (V_{SEM}) . Then, we introduce weighted edges $(i, j) \in V_{SB} \times V_{SEM}$, where a weight $\omega(i, j)$ is the Spearmanr correlation of SBERT values in dimension i vs. the metric scores for aspect j across all (development) data instances. We solve

$$\max \sum_{(i,j) \in V_{SB} \times V_{SEM}} \omega(i, j) \cdot x_{ij} \quad (9.7)$$

$$s.t. \sum_j x_{ij} \leq 1 \quad \forall i \in V_{SB} \quad (9.8)$$

$$\sum_i x_{ij} \geq 1 \quad \forall j \in V_{SEM} \quad (9.9)$$

The binary decision variables $x_{ij} \in \{0, 1\}$ indicate whether an SBERT dimension is part of a specific sub-embedding. The first constraint decomposes SBERT embeddings into non-overlapping parts, one for each aspect. The second constraint ensures that each semantic aspect is modeled.

Results are displayed in Table 9.3. First, we see that the global AMR metrics WLK and WWLK are best modeled with the cosine distance computed on full SBERT embeddings (unpartitioned, Table 9.3) and we can't model them as well with a sub-embedding. This seems intuitive: the power of a low-dimensional sub-embedding is too low to express the complexity of the two Weisfeiler graph metrics that aim at capturing broader AMR

aspect	SB-full	partitioning models		
		SB-rand	SB-ILP	S ³ BERT
SMATCH	64.6	57.1	57.9	68.2 [†]
WLK	76.7 [†]	63.5	64.2	74.6
WWLK	75.1	62.0	63.8	74.4
Frames	46.0	40.8	45.2	66.4 [†]
Unlabeled	58.4	52.3	54.7	65.1 [†]
Named Ent.	-14.4	-1.1	-0.3	51.1 [†]
Negation	-2.00	-0.0	3.4	33.0 [†]
Concepts	76.7 [†]	64.5	72.3	74.0
Coreference	23.2	10.3	13.6	43.3 [†]
SRL	48.3	40.8	44.9	60.8 [†]
maxIndegreeSim	27.0	23.6	24.0	32.5 [†]
maxOutDegreeSim	22.3	17.5	19.4	42.5 [†]
maxDegreeSim	22.3	18.0	19.7	30.0 [†]
rootSim	25.5	21.7	25.1	43.1 [†]
quantSim	11.5	10.0	11.8	74.6 [†]

Table 9.3: Spearmanr x 100 of AMR aspects. *Italics*: overall best. **bold**: best partitioning approach. underlined: improvement by more than 20 Spearmanr points.

sub-structures. However, the structural SMATCH, which does not match structures beyond triples, can be better modeled in a sub-embedding (+3.8 vs. SB-full). Nonetheless, compared to the best partitioning baseline (SB-ILP), our approach provides substantial improvements (Spearmanr points, WLK +10.4, WWLK +10.6).

Therefore, it is more interesting to study the fine-grained semantic aspects measured by our aspectual AMR metrics. We find that there are three AMR features that are very poorly modeled with global SBERT embeddings: *named entities*, *negation*, *quantification*. They also cannot be extracted with the SB-ILP baseline. By contrast, S³BERT clearly improves over these baselines. E.g., *negation* modeling improves from a negative correlation to a significant positive correlation of 33.0 Spearmanr. *Quantifier similarity* increases from 11.8 Spearmanr to 74.6.

9.8 Correlation with human judgements

Relating to research question 2) on whether we can effectively prevent SBERT from forgetting prior knowledge when teaching it to predict AMR metrics, we test how well our

approach compares to human ratings of sentence similarity in the typical zero shot setting. As our main goal is to increase the interpretability of SBERT predictions, we consider S³BERT achieving SBERT’s original performance on this task a satisfying objective.

9.8.1 Sentence semantic similarity

Test data. We use sentence semantic similarity data with human ratings. The STS (STSb) benchmark (Baudiš et al., 2016b) assesses semantic similarity and SICK (Marelli et al., 2014) relatedness.⁷

Evaluation metric. We again use Spearmanr. To assess *efficiency*, we display the approximate time for a metric to process 1,000 pairs. We also want to assess the *explainability* of the methods, which can be complicated (Danilevsky et al., 2020). To keep it as simple as possible, we assign ★★ when a metric is fully transparent and the score can be traced in the meaning space via graph alignment (SMATCH, WWLK), and ★ if there is a dedicated mechanism of explanation (e.g., via a linguistically decomposable score, as in S³BERT).

Baselines. As baselines we use: 1. SBERT and 2. our S³BERT from which we ablate a) the decomposition objective (S³BERT^{dec}) or b) the consistency objective (S³BERT^{cons.}). Assessing S³BERT^{cons.} is key, since it shows the performance when we only focus on learning AMR features – a significantly reduced score would prove the importance of counter-balancing decomposition with our consistency objective. For reference, we also include results from a simplistic baseline (word overlap) and the AMR metrics computed from the AMR graphs of sentences.

Results are shown in Table 9.4. Interestingly, while one main goal was to prevent a performance drop, S³BERT tends to outperform all baselines, including SBERT (significant improvement for STSb).

It is important to note that catastrophic forgetting indeed occurs if learning is not controlled by the consistency objective. In this case, the performance drops by about 20-30 points (S³BERT^{cons.} in Table 9.4). We conclude that our consistency objective effectively prevented any loss of embedding power.

⁷We min-max normalize the Likert-scale ratings of both datasets to the range between 0 and 1.

system	speed (1k pairs)	xplain	STSb	SICK
bag-of-words	0s	-	43.2	53.3
bag-of-nodes	31m (p) + 0.0s (i)	-	60.4	61.6
SMATCH	31m (p) + 49s (i)	★★	57.2	59.1
WLK	31m (p) + 1s (i)	-	63.9	61.4
WWLK	31m (p) + 5s (i)	★★	62.5	64.7
SBERT	1s (i)	-	83.1	78.9
S ³ BERT	1s (i)	★	83.7[†]	79.1
S ³ BERT ^{dec}	1s (i)	-	83.0	78.9
S ³ BERT ^{cons.}	1s (i)	★	51.7	58.1

Table 9.4: Results on STSb and SICK using Spearmanr x 100; Speed measurements of parser (p) and metric inference (i), units are minutes (m) and seconds (s).

system	xplain	3-Likert Spea's r	binary classif. F1 scores		
			Macro	Sim	¬ Sim.
RE19	-	-	65.4	52.3	78.5
BH21	-	34.8	-	-	-
OP21	★★	-	68.6	60.4	77.0
SBERT	-	54.2	71.7	63.8	79.6
S ³ BERT	★	56.4[†]	72.9[†]	65.7[†]	80.1[†]
S ³ BERT ^{cons.}	★	28.2	55.6	53.7	57.4

Table 9.5: Results on argument similarity prediction.

9.8.2 Argument similarity

Testing data. Besides the STS and SICK benchmarks we use the challenging UKPA(spect) data (Reimers et al., 2019) with high-quality similarity ratings of natural language arguments from 28 controversial topics such as, e.g., *GMO* or *Fracking*.

Evaluation metric. Argument pairs in UKPA have one of four labels: *dissimilar*, *unrelated*, *somewhat similar* and *highly similar*. Originally, the task was evaluated as a binary classification task (Reimers et al., 2019), by mapping the *similar* and *highly similar* labels to 1, and the other two labels to zero. A similarity metric’s scores are then mapped to binary decisions via a simple threshold-search script. To conform with this work, we also evaluate using this setup. But to account for the fine-grained labels, we also use a second metric based on (Spearmanr) correlation, following Behrendt and Harmeling (2021) who propose a 3-Likert scale that maps *dissimilar* and *unrelated* to 0, *somewhat similar* to 0.5, *highly similar* to 1.0.

Baselines. Table 9.5 shows the results of the best systems reported for i) a BERT-based approach (Reimers et al., 2019) (RE19), ii) the AMR-based SMATCH-variant approach of our argumentation-focused work before (Chapter 8), and iii) Behrendt and Harmeling (2021) (BH21), who pre-train BERT on other argumentation datasets for 3-Likert style rating.

Results. S³BERT significantly outperforms all baselines, including SBERT, in the classification setting, and in the correlation evaluation setting. When assessing interpretability, OP21 offers ★★ because it is based on SMATCH and the score can be *fully* traced. However, it is less efficient, due to the cost of executing AMR metrics and parser, and lags behind in accuracy. Again, we can conclude that our approach offers a valuable balance between interpretability and performance. Finally, this experiment further corroborates that controlling the decomposition learning process is paramount: without consistency objective, the accuracy is almost halved (S³BERT^{cons.} in Table 9.5).

9.8.3 Ablation and parametrization experiments

Upper-bounds for MR metric approximation. While not the main objective of our work, the approximation of computationally expensive AMR metrics can be considered

an interesting task on its own. We hence explore two AMR metric approximation upper-bounds: i) $S^3BERT^{cons.}$: Naturally, the consistency objective is orthogonal to the AMR metric approximation objective and by ablating the consistency objective, we can obtain an upper-bound for the prediction of AMR metric scores. ii) $S^3BERT^{cons.}+parser$: At the cost of making our approach much less efficient, we train $S^3BERT^{cons.}$ directly on (linearized) AMR graph strings instead of their underlying sentences, which allows us to infer metric scores directly from AMR graphs.

The results of these setups are given in Table 9.6. We see that both modifications can yield, to some extent, better AMR metric approximation accuracy, across all tested aspects. However, considering our second key goal of preserving the overall power of sentence embeddings, it is important to note that these improvements come at great cost, because if we do not control the decomposition process with our consistency objective, the similarity rating effectivity of the neural embeddings deteriorates (see $S^3BERT^{cons.}$ in Table 9.4 for sentence similarity and Table 9.5 for argument similarity). On top of this, $S^3BERT^{cons.}+parser$ will also lose much *efficiency*.⁸

9.8.4 AMR metric approximation inspection

How well can we approximate the AMR metrics, in a setup where we only care about AMR metric prediction performance? The answer is shown in Table 9.6. Best AMR metric approximation is achieved when we train on AMR graphs instead of sentence pairs, and switching off the consistency objective that prevents the catastrophic forgetting. In other words, we could manage to increase AMR approximation accuracy by roughly up to 15 points by paying the price of additional runtime (for parsing) and forgetting of SBERT’s strong overall performance.

Effect of parser quality. For creating AMRs, we used a strong parser that yields high SMATCH scores on AMR benchmarks. To investigate the effect of using another parser, we re-ran our first experiment (decomposition) with metrics computed from parses of the older JAMR (Flanigan et al., 2014) parser, that achieves more than 20 points lower SMATCH on AMR benchmarks. We observe moderately (+1-3 correlation points) better results across all categories with the more recent parser. This implies that there is potential room for further improvement of our method by using an even more accurate parser, but judging from the marginally lower score of JAMR, the gain may be small.

⁸Due to slow AMR parsing (c.f. Table 9.4).

aspect	S ³ BERT	S ³ BERT ^{cons.}	S ³ BERT ^{cons.} +parser
SMATCH	68.2	77.0	80.3
WLK	74.6	79.3	78.9
WWLK	74.4	81.5	82.3
Frames	66.4	79.6	80.3
Unlabeled	65.1	75.5	78.0
Named Ent.	51.1	58.0	61.9
Negation	33.0	34.5	35.5
Concepts	74.0	78.5	76.4
Coreference	43.3	57.4	72.1
SRL	60.8	74.3	83.0
maxIndegreeSim	32.5	37.3	37.5
maxOutDegreeSim	42.5	59.9	65.4
maxDegreeSim	30.0	40.6	42.7
rootSim	43.1	57.4	81.2
quantSim	74.6	75.7	76.1

Table 9.6: AMR metric approximation upper-bounds. $S^3BERT^{cons.}$: S³BERT without consistency objective (trades sentence similarity rating performance for better AMR approximation). $S^3BERT^{cons.}+parser$: S³BERT without consistency objective and inference on linearized AMR graphs (trades sentence similarity rating performance *and* efficiency for better AMR approximation).

Size of training data. We observe that the AMR metric approximation accuracy profits from growing size of the training data (see Table 9.7).

9.9 Data analyses with S³BERT

9.9.1 Studying S³BERT predictions

We find many interesting cases where S³BERT is able to explain its similarity scores, some of which are listed in Table 9.8.

For example, both S³BERT and SBERT assign a high similarity score (0.70–0.73) to *two cats are looking at a window vs. a white cat looking out of a window*, while the human similarity rating is just above average (.52). Here, a low similarity rating of -0.15 in S³BERT’s **quantifier feature** provides a (possible) rationale for the much lower human score, due to a strong contrast in quantifier meaning (*two* vs. *a*).

When confronted with **negation**, both SBERT and S³BERT assign moderately high scores to *The man likes cheese vs. the man doesn’t like cheese*. But S³BERT can explain this: its high *concept* similarity score increases the overall rating, while a (very) low similarity score for *negation* (-0.30) regulates the rating downwards. We also see differences in how negation of a matrix verb affects the S³BERT negation feature – compared with negation applied to a coordinated sentence. *Three boys in karate costumes [aren’t | are]*

aspect	amount of training data			
	rand (0k)	50k	300k	1500k
SMATCH	57.1	59.4	60.2	68.2
WLK	63.5	64.1	70.2	74.6
WWLK	62.0	65.8	67.0	74.4
Frames	40.8	44.2	53.6	66.4
Unlabeled	52.3	53.6	54.1	65.1
Named Ent.	-1.1	11.4	31.8	51.1
Negation	-0.0	17.8	29.0	33.0
Concepts	76.7	69.6	71.2	74.0
Coreference	23.2	23.9	25.2	43.3
SRL	48.3	49.4	50.0	60.8
maxIndegreeSim	27.0	26.7	26.4	32.5
maxOutDegreeSim	22.3	22.4	23.1	42.5
maxDegreeSim	22.3	22.1	22.5	30.0
rootSim	25.5	26.4	28.9	43.1
quantSim	11.5	47.1	65.4	74.6

Table 9.7: AMR prediction performance w.r.t. different training data sizes.

id	sentence pairs	humSim	SBERT	S ³ BERT	notable feature similarities
1	two cats are looking at a window a white cat looking out of a window	0.52	0.70	0.72	concepts: 0.87↑↑; quant: -0.15↓↓
2	three men posing in a tent three men eating in a kitchen	0.24	0.39	0.42	quant:0.99↑↑; Frames: -0.02↓↓, Unlabeled: 0.6 ↑
3	rocky and apollo creed are running down the beach the men are jogging on the beach	0.6	0.33	0.32	maxDegSim: 0.4↑, NamedEnt: -0.72↓↓
4	a man is smoking a baby is sucking on a pacifier	0.0	0.06	0.06	rootSim↑↑: 0.4
5	a dog prepares to herd three sheep with horns a dog and sheep run together	0.44	0.63	0.65	SRL: 0.56↓; Frames: 0.45↓, Concepts: 0.85↑
6	The cat scratches itself The cat scratches another cat	na	0.81	0.78	Concepts: 0.9 ↓; Negation 0.56↓; Coref: 0.41↓↓
7	The man likes cheese The man doesn't like cheese	na	0.80	0.77	Concepts: 0.90 ↑; Negation: -0.3 ↓↓
8	Recruits are talking to an officer An officer is talking to the recruits	0.68	0.97	0.98	SRL: 0.96 ↓; Negation: 0.90 ↓; Unlabeled: 0.99 ↑
9	A dog is teasing a monkey at the zoo A monkey is teasing a dog at the zoo	0.63	0.99	0.99	SRL: 0.96 ↓; Negation: 0.97 ↓; maxDegr: 1.0 ↑
10	Three boys in karate costumes aren't fighting Three boys in karate costumes are fighting	0.58	0.86	0.86	Concepts: 0.92↑; Negation: -0.31↓↓
11	A child is walking down the street and a jeep is pulling up	0.63	0.95	0.92	Concepts: 0.95↑; Negation: -0.22↓↓
	A child is walking down the street and a jeep is not pulling up				

Table 9.8: Prediction Examples from STSb and SICK, or own construction (human rating: na).

fighting results in lower negation agreement (Negation feature similarity: -0.31) compared to negation applying to the predicate of a sub-ordinate sentence, as in *A child is walking down the street and a jeep [is not | is] pulling up* (Negation feature similarity: -0.22).

Coreference can also explain key differences in meaning: *The cat scratches a cat* and *The cat scratches itself* are highly rated in all aspects (0.78–0.8 overall similarity) – except for coreference, with similarity of only 0.41, signaling a key difference reflected

in coreference structures.

Comparing the **foci of sentences** can also provide explanatory information. E.g., the human score for *a man is smoking* and *a baby is sucking on a pacifier* is zero, indicating complete dissimilarity. But S³BERT and SBERT assign scores that indicate moderate similarity. S³BERT’s features may explain this, in that the sentences’ foci (root sim) are somewhat related (0.4, *smoking* vs. *sucking*).

9.9.2 Studying predictors of human scores

What features can predict *human similarity scores* and how may the assessment of argument similarity as opposed to sentence similarity differ from each other? In search for answers to these questions, we perform a quantitative analysis of S³BERT’s fine-grained features. We proceed as follows: Let *SIM* be S³BERT’s similarity ratings for a pairwise data set, and *HUM* be the corresponding human ratings. Now, let *FEASIM* be the fine-grained S³BERT feature similarities for a feature *FEA* (e.g., SRL aspect). Then we compute, for each *FEA*, *Spearmanr*(*FEASIM*, *SIM*) and *Spearmanr*(*FEASIM*, *HUM*), both on STS and argumentation benchmarks. In other words, we analyze predictive capacity of features for a) system vs. b) human similarity in c) different domains/tasks.

Analysis results are shown in Table 9.9. Interestingly, for *human argument similarity*, the residual has much lower predictive power (26.1), suggesting that human argument similarity notions differ significantly from sentence similarity. Indeed, another key difference can be found in the importance of quantification similarity, which is marginal (-4.2) for argumentation, but not for STS (51.6). We speculate that users judging argument similarity tend to generalize over quantifier differences, being more focused on general statements and concepts, as opposed to, e.g., numerical precision. Notably, human argument similarity is markedly well predicted by **Frames** – this feature alone achieves state-of-the-art results, indicating a marked importance of predicate frames for argument similarity.

Of course, although the analysis may give some interesting indications about similarity as perceived by humans (and SBERT), it has to be taken with a grain of salt, one reason being, e.g., that the shown statistics are influenced by AMR metric prediction accuracy, which varies across aspects (c.f. Table 9.3). Our study also indicates that neither sentence nor argument similarity can be fully explained by any feature. We hypothesize that we may need to go beyond what sentence transformers and (current) AMR metrics can measure, e.g., by incorporating background knowledge. Our method may offer a way to

inject such background knowledge into sentence embeddings, via distillation of dedicated metrics.

9.9.3 Evaluation with a CheckList

Mainly focusing on NLG evaluation metric inspection, Zeidler et al. (2022) adopt the CheckList paradigm (Ribeiro et al., 2020) and annotate a subset of similar sentences from the SICK corpus with an occurring semantic phenomenon. For instance, one part of the check list is labeled *hyponymy* – it comprises sentence pairs that can be related with a hypernymy relation (e.g., *A cat runs, A kitten runs*). Therefore, we can inspect the correlation of a metric in such diverse phenomena. Even though the AMR aspects might not exactly translate to their aspects, and some phenomena come with a very small data size, evaluation with such a CheckList could provide additional insights. Table A.4 of Appendix A.6 shows an extensive evaluation of various metrics, including S³BERT sub-embeddings on all categories of the CheckList.

In comparison to the competitors, including NLG metrics based on large language models such as BERTscore, 8 sub-embeddings obtain higher average correlation. At this point, however, it is important to underline that the performance of a sub-embedding with respect to a certain CheckList semantic category (even if sufficient examples were available) does not tell us much about the degree that the respective sub-embedding has managed to capture the semantic category. Instead, this type of CheckList asks a ‘posterior’ question: *Given sentence pairs of a certain semantic phenomenon, tell us how similar the sentences are in the eye of the human annotator*. This question is related to what we are interested in, but it is not the same, since sub-embeddings are more tailored to tell us *if* there is a certain semantic divergence/agreement happening between sentences (and in which aspects), *and* how overall similarity is changed by it.

9.10 Discussion

We propose a method for decomposing neural sentence embedding spaces into different sub-spaces, with the goal of obtaining sentence similarity ratings that are *accurate*, *efficient* and *explainable*. The sub-spaces express facets of meaning as captured by MR and MR metrics, such as *Negation* or *Semantic Roles*. The *decomposition objective* partitions the semantic space via targeted synthesis of MR metrics. The effectiveness of neural

sentence embeddings is preserved by a *consistency objective* that controls the decomposition process and routes global semantic information not expressed by MR into a *residual embedding*. The S³BERT embeddings are more explainable and are on par, or even outperform, the accuracy of a strong standard sentence transformer. The approach allows straightforward extension to customized metrics of meaning similarity.

Notably, in contrast to other explainability approaches that inspect, e.g., gradients or attention weights to assess salience of input tokens, our partition-approach targets explainability in the *decision-space*. Indeed, semantic partitioning of the ‘last neural layer’ lets us inspect how abstract features are weighted in order to arrive at a certain decision/output. In our future work (Section 10.3) we discuss cross-pollination of methods for ‘input-space’ explainability⁹ and our work targeting ‘decision-space’ explainability.

⁹E.g., see the work of Moeller et al. (2023) who show a way to apply the ‘gradient integration principle’ (Sanyal and Ren, 2021) to the Siamese text embedding encoders that we have used, understanding the similarity contributions of individual input tokens.

	aspectual semantic feature														global AMR feature			
	data	Conc.	Frame	NE	Neg.	Coref	SRL	IDgr	ODgr	Dgr	\sqrt{Sim}	quant	Sma.	Unlab.	WLK	W ² LK	Resid.	
FEASIM vs. HUM	STSB	73.8 ₍₁₎	68.7	60.4	53.6	65.6	70.8 ₍₂₎	66.8	64.8	69.9 ₍₃₎	67.2	51.6	72.7	68.1	75.1	72.8	83.3	
vs. SIM	STSB	88.3 ₍₁₎	81.5	75.6	61.9	80.0	84.4 ₍₂₎	81.2 ₍₃₎	78.7	81.2 ₍₃₎	77.5	60.1	86.1	83.4	88.9	86.4	99.3	
vs. HUM	UKP	51.3	61.3 ₍₁₎	26.9	52.1 ₍₃₎	42.9	43.7	33.6	57.1 ₍₂₎	42.0	45.4	-4.2	30.3	37.8	10.9	25.2	26.1	
vs. SIM	UKP	98.3 ₍₁₎	86.7	85.0	93.3 ₍₂₎	91.7	90.0	90.0	91.7 ₍₃₎	85.0	86.7	63.3	91.7	86.7	81.7	86.7	96.7	

Table 9.9: Similarity investigation with S³BERT feature analysis. **bold/(n)**: best from a feature group (rank 1–3).

Part IV

Conclusions and future work

Chapter 10

Conclusions and outlook

10.1 Conclusions

Proper analysis of *semantic similarity* is important for many machine learning tasks, including semantic search and evaluation of various system outputs. On the hunt for *more meaningful* semantic similarity measures, we studied metrics of meaning representations (MRs) that *capture semantics explicitly*. While we certainly have to leave some questions open¹, our answer to the important question of whether MR-based distances are interesting objects of study is generally positive: MR metrics can conduct explainable and meaning-grounded similarity measurements that support various applications such as (*inter alia*) NLG diagnostics, semantic clustering and search of graphs and texts, as well as extended parsing evaluation.²

In Part I of this thesis, we contributed **a theoretical and comparative study of previous MR metrics**. Based on insights of our analysis, we developed **a metric for graded similarity of atomic meaning graph constitutes** (e.g., matching a *kitten* vs. a *cat* node, using our S²MATCH metric, Chapter 4), and **a novel MR metric for graded similarity of subgraphs** (e.g., matching *kitten* against the structure $cat \xrightarrow{mod} young$, using our WWLK metric, Chapter 4). To assess MR metrics empirically, we introduced the **first benchmark for MR metrics** (BAMBOO³, Chapter 5) that tests metrics with respect to several objectives such as sentence similarity and metric robustness against meaning preserving/altering graph transformations. We showed that our novel metrics achieve good

¹We will touch on some of them in Section 10.3.

²Meanwhile, we also find that this thesis' ideas have started to get picked up, e.g., for parsing evaluation (Lorenzo et al., 2023; Vasylenko et al., 2023), cross-lingual analysis of MRs (Uhrig et al., 2021; Wein and Schneider, 2022; Leung et al., 2022; Wein and Schneider, 2023), MR2text evaluation (Hoyle et al., 2021; Manning and Schneider, 2021; Ribeiro et al., 2021; Li et al., 2022; Montella et al., 2023), or building advanced neural MR metrics (Shou and Lin, 2023). We study asymmetric versions of our MR metrics for interpretable NLI (Opitz et al., 2023b), and propose more standardized parsing evaluation (Opitz, 2023c,b).

results in most tasks. Finally, we tested metrics in a restricted but popular setup: the evaluation of high-performance MR parsers. We found that all metrics struggle with rating finer MR quality and that MR parsing is far from solved, pointing at the need for more future research on parsing evaluation.

In Part II, we showed that we can use MR metrics to view the evaluation of automatically generated texts through MR-based semantic distances. This way, we could detect the aspects where the generated hypothesis and reference are actually similar, or dissimilar (e.g., negation, semantic roles, etc.), and diagnose issues in automatic systems. We also showed that we can ablate costly textual human reference texts by comparing the MR of the generation against the MR from which the generation stems, resulting in the **first system for referenceless evaluation of MR2text generation**, Chapter 6).

On the other hand, we looked at the task of efficiently rating automatic MR constructions, by predicting metric scores when the costly reference graph is absent. Potential application cases are filtering candidates in ensemble parsing, estimating parser performance on unseen and new data sets, or possibly aiding human annotators in an active learning setup to improve parsing. To address this task, we proposed **the first systems for referenceless semantic parser quality estimation** (Section 7).

Finally, in Part III, we explored novel and generalized use-cases of MRs and MR metrics by aiming to build interpretable text similarity metrics for applications like text search or text clustering. In a first demonstration, **we showed that MR metrics can be used for rating the similarity of natural language arguments in a transparent, accurate and explainable manner**. We also showed that it can be valuable to automatically extrapolate conclusions of premises and view them through the lens of MR, all based on our hypothesis that similar arguments lead to similar conclusions. Additionally, we tested our MR metrics for rating the quality of arguments, which is a difficult but important task in argument mining. We find that good argumentative conclusions are semantically neither too close, nor too far, from their premise(s), an assessment that can be automatically supported with MR metrics.

However, we also observe **a critical issue of MR metrics**: They tend to be slow and therefore their application to large-scale use-cases is strictly limited. As a first step to alleviate this issue, we conducted a pilot experiment where we showed that it is possible to approximate costly graph matching via SMATCH, proposing a **novel approach to accurately approximate NP-complete MR alignment** (Chapter 9). To this aim, we proposed data augmentation with graph permutation and vocabulary reduction. With these data augmentations, we showed that we can train i) a machine translation model to predict

MR application / task	possible setup
1:1 node MR-alignment	SMATCH/S ² MATCH (max. efficiency: learned SMATCH/S ² MATCH)
n:m node MR-alignment	WWLK
n:m edge MR-alignment	WWLK + edge-to-node translation
MR-parser eval	ANY
MR-quality estimation	project ANY (e.g., using extrapolated SMATCH, or MR metric + oracle parser)
MR2text eval w/o ref	project ANY (e.g., via parser)
NLG eval text vs. text	S ³ BERT or ANY + parser
Semantic search: graph	S ³ BERT + parser or efficient MR metric (e.g., SEMBLEU, learned SMATCH)
Semantic search: text	S ³ BERT or efficient MR metric + parser
Explain-low-level	MR metrics with graph alignment (SMATCH, S ² MATCH, WWLK)
Explain-high-level	ANY on MR subgraphs or S ³ BERT feature decomposition

Table 10.1: High-level MR-metric method decision guide for a selection of general applications that we visited in this thesis. ANY means any MR metric that measures distance in explicit MR metric spaces (SMATCH, S²MATCH, SEMBLEU, SEMA, WLK, WWLK,...)

the alignment and score with great accuracy and ii) a very fast but slightly less accurate convolutional neural network that directly predicts the graph similarity score.

Finally, still not satisfied with the efficiency of measurement in meaning space (due to the dependency on a slow MR construction mechanism) and also aiming at an accuracy that resembles human similarity ratings, **we showed that we can inject interpretability into a state-of-the-art neural model via guidance from MR metrics** (Chapter 9). More precisely, we employed MR metrics for showing the neural network how to decompose its sentence embeddings space into various features that express different text semantics. To control this decomposition process, we enforced that the model’s overall similarities are consistent with the similarity ratings of a teacher model. We found that the resulting model preserves full efficiency, since it does neither need a parser nor MR metrics in inference, and keeps or even extends the state-of-the-art accuracy in different benchmark data sets and domains – while also offering valuable interpretability.

10.2 Decision guide for MR metric application

We briefly want to gather the visited applications/tasks and associated MR metrics in a ‘decision guide’ (Table 10.1).

An interesting multi-purpose application is the **alignment of MRs**. For instance, it can be used to explain scores, or merge graphs. A 1-1 node alignment is provided by SMATCH,

and improved by S^2 MATCH. As a fast alternative, we have learned-SMATCH (using alignment prediction with NMT) but would have to admit an accuracy loss³. WWLK (so-far) is the only metric that provides a many-to-many node alignment, which is useful for aligning MRs from different sentences, where arbitrarily-sized sub-structures have to be related (e.g., *kitten* as a node, and *cat :mod young* as a subgraph). Note that an edge-to-node graph translation ‘trick’ as outlined in Figure 2.1 (Background Chapter 2) can empower us to also directly align edges.

When performing **MR-parser evaluation** against a reference, we can simply take an MR metric off the shelf. By using multiple metrics, or an ensemble of metrics, where the selection is possibly informed by human accuracy from BAMBOO³, we can obtain a more comprehensive picture of parsing performance.

Interestingly, we saw that we do not need a costly reference in some evaluation cases, ranging from the quality assessment of MR candidate parses to MR2text candidate sentences. In particular, for efficient **quality estimation of predicted MR graphs**, we tested and proposed different neural graph encoders that learn to project an oracle metric in the absence of the MR reference. On the other hand, for **evaluation of MR2text** systems, we proposed to project a candidate MR by parsing the text output, and comparing this candidate MR against the input MR in the MR space. Such a strategy is unsupervised and thus comes with reduced biases, and also allows to ablate a costly reference. The latter ‘principle’ (parser + metric) also may extend to different NLG evaluation tasks, where reference text is available. The reference text then would also have to be parsed in order to apply MR metrics. Alternatively, of course, we could employ S^3 BERT to yield less transparent but more efficient and (probably) more accurate explainable NLG evaluation.

Then we studied the novel application of MR metrics for **semantic search and similarity through MR graphs**. Such an application may be of particular interest to, e.g., linguists that desire to cluster or search particular semantic patterns. Almost all MR metrics seem suitable for this, but their suitability may be strictly limited by efficiency bottlenecks such as NP-completeness. Thus, we can resort to WLK and SEMBLEU, or learned SMATCH, or even S^3 BERT trained directly on parses.

Finally, we investigated the **semantic search on texts** based on MR metrics. We created S^3 BERT that decomposes a neural sentence transformer embedding space into meaningful MR subspaces, guided by MR metric distillation. S^3 BERT yields vectors that are equally powerful but show how the similarity ratings are composed and explained from

³The accuracy loss can be strongly reduced by our training data augmentation strategies

different aspects as measured in the implicit MR space. Since a parser is not required, the efficiency of a neural sentence transformer is also fully retained.

Lastly, we consider two different levels of **explainability application: high-level, or low-level**. Here, we use *high-level explainability* for providing high-level explanations for decisions that are human-interpretable, and *low-level explainability* to highlight that it may be possible for a human to fully trace and understand score computation. SMATCH, S²MATCH, and WWLK are explainable on both levels: On the low level, they are explainable because we can exactly and quickly trace how the final score emerges. On a high-level, they offer explainability via application of targeted aspectual subgraphs (e.g., on SRL-focused subgraphs). Of course, using the same principle (measuring distances of aspectual subgraphs), we can also explain other graph metrics on a high level, such as WLK. However, a fine-grained assessment may not be directly accessible in *all* MR metrics. For instance, SEMBLEU is limited to processing connected graphs, and subgraph extraction risks loss of connectivity. The idea of higher-level explainability by examining aspectual subgraph distances is also reflected by S³BERT, where we use the principle to learn a meaningful embedding space decomposition.

10.3 Outlook and future work

Having arrived at the end of this thesis, and taking a step back, we notice ample room for future work. This includes questions that we had to leave open and new questions that have arisen during our explorations. We give a brief overview over some of them:

- **State-of-the-art similarity metrics in explicit MR space:** While we showed that MR metrics can support interpretable similarity measurements and even help to improve powerful neural text embeddings, we are still lacking graph metrics that more accurately reflect similarity. We hypothesize that we need to build more informed graph metrics that better know how to weight different types of meaning graph similarities/divergences. Consider SMATCH that assigns all graph triples the same weight: it clearly can't account for the many cases where we would like to attach different relevance to different types of meaning components of a text. While our WWLK metric makes a step forward by including broader matches and adjustable edge weights, its modeling scheme still seems too naïve to exhibit the power of a

human (or neural) assessment. Fortunately, the search for better MR graph metrics may be boosted by the continuous improvement of parsers.⁴

- **Hybrid metrics:** Directly connecting to the point above, it may be interesting to study hybrid metrics of text and MRs for performing a two-way measurement based on text/MR similarity. In a subsequent work to this thesis, using a simple combination of MR and text metrics, we find indications that the two domains can be complementary and together help improve accuracy (Opitz et al., 2023b).
- **Comparing large MR graphs:** An open question concerns the application of metrics between larger meaning representations. Indeed, the MRs that we considered do not often represent more than one sentence. When graphs grow much larger, running SMATCH to the optimal solution may be infeasible, and the solutions with heuristics such as hill-climbing SMATCH and our learned SMATCH will deteriorate. While the issue can be mitigated by our WLK and WWLK metrics, future work can explore more ways for comparing large MRs of multiple sentences, paragraphs, or full documents of unrestricted size.
- **Referenceless parser quality estimation:** The comparison of MR parsers is a challenging problem, particularly when a costly reference is absent. While we found that we can mitigate this issue via referenceless parsing evaluation metrics, our tested metrics were based on a neural network trained on candidate parses and are thus subject to biases from learning and data selection. Besides, our approach is necessarily tailored to a specific metric, and thus it will inherit any of its drawbacks. To alleviate these issues in a referenceless metric, a different paradigm could be useful. More specifically, different NLP systems could inform us about MR quality. For instance, an MR text generation system could provide us with a set of possible texts generated from a candidate MR. To our benefit, we could then exploit a large tool box of NLG evaluation metrics, including strong faithfulness measures (Steen et al., 2023) that may be particularly attractive to verify the factual coherence of semantic parses. Alternatively, or in conjunction, we could leverage NLP systems that solve MR-related tasks. E.g., we could compare the output of a good coreference resolution system to parts of our candidate MR to verify the MRs

⁴Recall that for BAMBOO₂, we found improvements when using human corrected parses for all tested metrics, and the findings of Manning and Schneider (2021) also suggest that better parsers lead to better MR metric performance. If gold parses are available, they find that MR metrics can be better than large neural metrics.

references. In the end, a sensible referenceless metric could even perform *better* than a reference-based metric, since the latter typically hinges on a single reference and therefore disfavors different but valid human interpretations.

- **From *End-explainability* to *End-to-End explainability*:** We believe that the explainability of our S³BERT embedding-partition technique can be fruitfully related to explainability methods that calculate salience of input text structures, e.g., through gradient integration (Sanyal and Ren, 2021), or Shapley values (Shapley, 1951). This would allow us to trace back the aspectual sub-embeddings to different parts of the input, and could elicit how aspectual similarities are affected by structures in the input. Combining our ‘decision explainability’ with the ‘input structure explainability’ of conventional saliency methods could provide us with a powerful tool for deep structured linguistic data analyses and exploration. An anticipated bottleneck, however, is a greatly increased inference cost due to the conventional explainability methods, losing efficiency of S³BERT explainability as is.
- **Can we inform Meaning Representation Design with MR metrics?** In most of our experiments, we used a particular MR framework (AMR). An interesting meta-feature that is shared by many MR frameworks is that their design is primarily driven by human intuition, linguistic education, experience, or even specific user desiderata (Pavlova et al., 2023) – i.e., all aspects that are not immediately *empirically grounded*. So an interesting question is: can we use MR metrics to elicit MR design improvements, or detect MR design principles that are good/flawed? This may be possible by carefully examining MRs of paraphrastic sentences, together with their MR metric scores: If the similarity scores are high, this is a signal that we did something right in the MR design. On the other hand, low scores may either point at some issue in the MR metric, *or/and* it might show us some aspect in which the two MR structures are too dissimilar, potentially revealing an MR design issue that could be alleviated by sensible design improvement.

Overall, the field of meaning representations and their metrics leaves plenty of room for interesting future research.

Appendix A

Appendix

A.1 On the soundness of comparing MR-generated sentences in the MR domain

First, we provide a simple example for our argument (it is safer to compare texts generated from AMR in the AMR domain) and then a simple proposition together with its proof. The example is displayed in Fig. A.1, where, similar to MR2text, we see a (surjective) function that generates concrete objects from abstract objects (e.g., *mammal* \rightarrow {*dog*, *mouse*, *cow*}). Now, imagine we are given *mammal* and are tasked with generating a single concrete instance. How can we assess whether our output is correct? We cannot safely assess this by testing whether the output (e.g., *cow*) is the same as another instance of *mammal* (e.g., *dog*). Instead, we can re-apply the abstraction f to *cow* and conduct the comparison safely in the abstract domain.

Proposition. a) *The canonical MR2text evaluation setup, that matches generated sentence s' to distant source sentence s , is not well defined.* b) *This issue can be alleviated by*

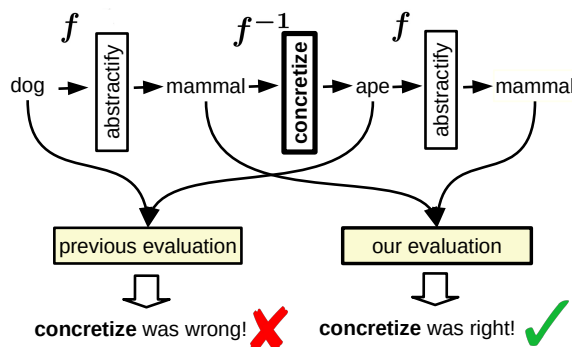


Figure A.1: A critical issue and its alleviation.

LM	F1 score			
	grammaticality		fluency	
	poor/perfect	all	poor/perfect	all
GPT2	0.80	0.74	0.80	0.71
GPT2-distill	0.79	0.73	0.76	0.70
BERT	0.80	0.72	0.80	0.72
RoBERTa	0.66	0.72	0.69	0.72

Table A.1: Results for assessing the *Form* score prediction (corpus-level) of different LMs for NLG-generated sentences against humans judgements (separated by grammaticality and fluency); all: all 12k generated sentences vs. ‘poor/perfect’: the 5k instances of best/worst generations in both grammaticality and fluency.

grounding the evaluation in the MR domain by re-applying parse, abstaining from direct use of s (thereby using MR2text generation as a right inverse function).

Proof. Let X be a set of concrete objects (e.g., sentences) and f a (surjective) function from X to Y (e.g., ‘sent-to-MR’), where Y contains abstract objects (e.g., MRs), s.t. $|Y| < |X|$. Then, using $f^{-1} : Y \rightarrow X$ (e.g., ‘MR-to-sent’) as right-inverse is well-defined: $f \circ f^{-1} = id_Y$ (Proposition b), but using it solely as left-inverse (as done in previous evaluation) does not guarantee a well-defined result: $f^{-1} \circ f \neq id_X$ (Proposition a). \square

A.2 Form predictor selection experiment

To estimate how well they are able to assess *Form*, we make use of human-assigned scores for data from the WebNLG task as provided by Gardent et al. (2017). It contains grammaticality and fluency judgments by humans for more than 2000 machine-generated sentences. We report the F1 score, both for grammaticality and fluency, by converting the human assessment scores to *accept* predictions, and using them as a gold standard to evaluate the LM-based *accept* predictions over (i) all 12k sentence pairs¹ and (ii) only the 5k sentence pairs where both grammaticality and fluency were either rated as ‘perfect’ (max. score) or ‘poor’ (min. score) by the human.²

The results are displayed in Table A.1 and show (i) that the LMs lie very close to each other with respect to their capacity to predict fluency and grammatically, and (ii) that both fluency and grammaticality can be predicted fairly well.

¹This includes all generated sentences from a given input, as provided by Gardent et al. (2017) and Shimorina et al. (2017)

²The ratings are based on a 3-point Likert scale.

A.3 Fine-tuning the conclusion generator

To fine-tune the sequence to sequence language model T5 for conclusion generation, we create training data from the the persuasive essays dataset of Stab and Gurevych (2017) as follows: From all premise-conclusion-pairs annotated in this dataset, we retrieved all claims with their annotated premises. In addition, we employ all annotated major claims with their supportive claims as premise-conclusion-pairs.³ We discarded samples for which we cannot retrieve any premise. Each resulting premises-conclusion-sample has 3.1 premises on average.

We split the data into 80% instances for training, and 10% for validation and testing, each. For each sample, we input the concatenated premises by encoding the string template `summarize:<premises>` and train with the conclusion as a target by applying a cross-entropy loss for each token. We guide the training process with an early stopping mechanism to ensure the best accuracy (ignoring padding tokens) on our validation dataset. In inference, we apply a 5-beam-search in combination with sampling over the 20 most probable tokens per inference step.

To assess the quality and relatedness of the generated conclusions, we manually compared the predicted conclusions with their premises in our test split. Since we observed promising and appropriate conclusion generations, we were encouraged to utilize the learned capabilities of the fine-tuned language model to generate conclusions for the argumentative sentences in the UKP aspect corpus.

A.4 Hyper-parameter setups

A.4.1 Sequence-to-sequence network parameters

Hyper-parameters for the NMT approach are displayed in Table A.2. The best model is determined on the development data by calculating BLEU against the reference alignments.

A.4.2 CNN network parameters

Hyper-parameters for the CNN approach are displayed in Table A.2. The best model is determined on the development data by calculating Pearson’s ρ correlation of predicted

³Whenever we encounter multiple premises or supportive claims of a single claim, we concatenate them in document order.

parameter	value
embedding size	512
encoder	4 transformer layers w/ 4 heads
decoder	4 transformer layers w/ 4 heads
feed forw. dim	2048
loss	cross-entropy
weight init	xavier
optimizer	adam
learning rate	0.0002
batch size	8192 (tokens)

Table A.2: Overview of NMT hyper-parameters.

parameter	value
emb. dimension	100
'pixels'	60x15
CNN encoder	concatenate(256 3x3 convs, 3x3 max pool 128 5x5 convs, 5x5 max pool)
MLP	relu layer followed by lin. regressor
weight init	xavier
optimizer	adam
learning rate	0.001
batch size	64

Table A.3: Overview of CNN hyper-parameters.

# example	antonymy 157	article 77	Co-hypo. 35	hypo. 11	negation 156	omission 155	part. syno. 26	passive 78	SRL 8	Sub. clause 69	AVG -
BERTScore	9.6	9.5	6.3	80.6	5.3	32.5	5.8	-10.4	-45.8	-3.3	9.0
BLEU	12.2	3.2	18.6	38.4	-5.4	11.8	-5.1	-8.0	9.3	-14.2	6.1
chrF++	15.0	4.2	10.1	6.5	-4.5	21.1	9.0	-12.2	1.2	-12.1	3.8
GraCo	7.1	-11.9	3.2	11.4	-9.4	12.8	22.9	-6.7	34.9	27.7	9.1
GraCo (G)	12.8	17.3	-1.2	34.3	-0.4	-0.8	-36.1	-12.0	-16.7	14.0	1.1
GraCo (r)	-1.6	-11.9	-5.2	43.5	-0.3	15.4	9.0	-3.8	21.7	33.0	10.0
Graco (g, R)	0.5	17.3	10.3	20.1	0.4	3.4	-5.2	-10.3	-16.7	11.6	3.0
Meteor	26.9	6.5	26.4	42.4	-4.5	6.3	32.2	-7.9	-24.2	-2.7	10.1
S ² MATCH	10.1	-4.6	4.7	48.5	-1.5	18.3	-5.7	2.7	-16.5	13.2	6.9
SMATCH	10.2	-4.6	7.7	48.5	-1.5	18.3	-5.7	2.7	-16.5	13.2	7.2
WLK	20.1	6.7	13.7	41.8	0.5	18.1	-17.0	2.7	-9.3	12.9	9.0
WWLK	10.9	-4.6	-9.9	77.3	-2.6	16.0	-4.0	2.7	-27.7	10.7	6.9
Concept	29.0	11.0	41.8	11.8	7.2	26.8	14.6	0.3	-15.7	11.9	13.9
Frame	-4.0	-0.8	46.8	21.1	1.7	19.8	3.8	-6.5	-41.0	11.9	5.3
NE	19.5	-19.1	45.5	21.1	-2.5	24.6	20.2	6.7	-48.2	6.0	7.4
Neg	12.2	-4.5	13.1	12.5	-6.1	10.1	26.1	22.6	-13.3	6.3	7.9
Coref	19.5	-1.4	28.9	13.6	5.3	17.8	26.2	10.8	-4.8	2.7	11.9
SRL	29.5	-2.9	8.1	2.5	0.0	16.2	21.9	15.6	-22.9	7.7	7.6
Idegr	24.8	2.1	51.4	16.5	1.6	18.9	17.9	8.0	-22.9	10.2	12.9
ODegr	9.1	0.7	57.8	18.7	11.9	15.7	28.8	17.6	-34.9	-11.6	11.4
Dgr	15.3	-4.0	-0.6	6.8	-5.3	20.4	4.8	13.6	-12.0	-4.3	3.5
root sim	4.3	0.7	36.1	14.8	1.0	21.3	10.3	18.6	9.6	-3.7	11.3
quant	12.0	7.5	37.0	13.5	-3.6	27.9	17.1	23.1	-59.0	23.7	9.9
Smatch	17.6	-1.3	23.7	16.7	6.6	16.3	11.0	3.7	-38.6	-4.8	5.1
Unl	21.7	-10.6	11.8	17.9	-4.0	9.0	11.7	21.3	-20.5	-4.6	5.4
WLK	30.8	-5.4	34.4	6.5	-3.8	22.3	19.4	9.8	32.5	20.2	16.7
WWLK	28.0	4.4	22.2	17.7	3.7	25.7	17.0	4.3	7.2	4.5	13.5
Residual	29.8	-1.1	52.7	30.5	4.4	30.8	36.1	14.8	-13.3	4.6	18.9

Table A.4: Evaluation on CheckList (Zeidler et al., 2022), Spearman correlation. Lower part of the Table show S³BERT sub-embeddings. ‘GraCo’ is a metric by Zeidler et al. (2022), we refer the reader to this paper.

scores and gold scores.

A.5 S3BERT Hyper-parameters and training

We use distilSBERT model as basis. Batch size is set to 64, the learning rate (after 100 warm-up steps) is set to 0.00001. We train for 8 epochs, evaluating every 1000 steps. Afterwards we select the model from the evaluation step where we achieve minimum development loss.

A.6 Results on semantic CheckList

For results on CheckList (Zeidler et al., 2022), see Table A.4.

List of Figures

2.1	Meaning preserving, quasi meaning preserving, and meaning breaking (A)MR translations.	16
3.1	Inspection of three different AMR metric procedures. Clouds indicate the ratio of matching triples. Minor details are omitted for readability (special root node, smoothing in SEMBLEU, $k = 3$ in SEMBLEU and $w_k = \frac{1}{3}\forall k$, which results in an unweighted geometric mean).	32
3.2	STS annotator instructions from Agirre et al. (2016).	40
4.1	Two AMRs with semantic roles filled differently, SEMBLEU considers them as equivalent.	56
4.2	Large symmetry deviation of SEMBLEU for two parses of <i>Things are so heated between us, I don't know what to do.</i>	57
4.3	Symmetry evaluations of metrics. SEMBLEU (left column) and SMATCH (middle column) and BLEU as a 'baseline' in an MT task setting on newstest2018. SEMBLEU: large divergence, strong outliers. SMATCH: few divergences, few outliers; BLEU: many small divergences, zero outliers. (a) marks the case in Figure 4.2.	58
4.4	Left: <i>In April, a woman rides a car from Rome to Pisa.</i> root nodes A: <i>travel-01</i> vs. B: <i>drive-01</i> . Right: <i>In Apr., a sailor travels with a ship from P. to N.</i>	61
4.5	# of k -grams entered by a node in SEMBLEU.	61
4.6	Three different MR graphs representing <i>The cat sprints; The kitten runs; The giraffe sleeps</i> and pairwise similarity scores from SEMBLEU, SMATCH and our new metric S^2MATCH	64

4.7	'6 Abu Sayyaf suspects were captured last week in a raid in Metro Manila.' gold (top) vs. parsed AMR (bottom). SMATCH aligns <i>criminal-organization</i> to <i>city</i> (red); S ² MATCH aligns <i>criminal-organization</i> to <i>suspect-01</i> , <i>city</i> to <i>country-region</i> (blue).	66
4.8	Similar MRs, with sketched alignments.	70
4.9	WLK example based on one iteration.	72
4.10	Wasserstein WLK example w/o learned edge parameters (top, c.f. Section 4.6.2) and w/ learnt edge parameters (bottom, c.f. Section 4.7). Learning these parameters allows us to adjust the embedded graphs such that they better take the (impact of) MR edges into account. Red: the distance increases because of a negation contrast between the two MRs that otherwise convey similar meaning.	74
5.1	Examples for f and h graph transforms.	87
5.2	Metric objective example for Argζ	87
5.3	WWLK alignments and metric scores for dissimilar (top, STS) and similar (bottom, SICK) AMRs. Excavators indicate heavy Wasserstein work <i>flow · cost</i>	97
5.4	Sketch of MR IAA ball. The center (P1) is a reference MR, while P2, P3, P4 are candidates. Any MR x from the ball has high structural SMATCH agreement with P1, i.e., $\text{SMATCH}(x, P1) \geq \text{estimated human IAA}$. However, they may fall in different categories: \mathcal{H} (green cloud) contains correct MR alternatives. Its superset \mathcal{A} (light cloud) contains acceptable MRs that may misrepresent the sentence meaning up to a minor degree. Other parses from the ball, e.g., P2, mis-represent the sentence's meaning – despite possibly having higher SMATCH agreement with the reference than all other candidates.	99
5.5	Data example: acceptable, low SMATCH. That is, $P \in \mathcal{H}$ but $P \notin \mathbb{B}(\text{SMATCH}, \text{ref})$. 101	
5.6	Data example excerpt that shows an unacceptable parse with high SMATCH. That is, $P \notin \mathcal{A} \supseteq \mathcal{H}$ but $P \in \mathbb{B}(\text{SMATCH}, \text{ref})$	101
5.7	Sentence length vs. human acceptability on all annotated data. 55 includes all sentences longer than 55 tokens. See Figure 5.9 for occurrences of different sentence lengths.	106

5.8	Sentence length vs. Smatch on all annotated data. 55 includes all sentences longer than 55 tokens. See Figure 5.9 for occurrences of different sentence lengths. Other metrics look similar.	107
5.9	Sentence length occurrences. 55 includes all sentences longer than 55 tokens.	108
5.10	Inter-metric correlation on Little Prince.	111
5.11	Inter-metric correlation on AMR3.	112
5.12	Snippet from BAMBOO versioning on GitHub: https://github.com/flipz357/bamboo-amr-benchmark	115
6.1	The <i>Canonical</i> evaluation matches n-grams from the sentences and assigns inappropriate ranks. Our metric $\mathcal{M}\mathcal{F}_\beta$ fuses <i>Meaning</i> and <i>Form</i> assessment and better reflects the ranking of the generations.	120
6.2	<i>“Perhaps, the parrot is telling itself a story”</i>	123
6.3	Explainable <i>Meaning</i> score (re-)ranking.	130
6.4	Explained negation confusion.	131
6.5	Explained SRL confusion.	133
6.6	Sentences of flawed form. --> refers to the binary acceptability judgment (Eq. 6.3.3).	138
7.1	Parse of <i>Without a functioning economy, the whole country may destabilize</i> with errors outlined.	142
7.2	Our model I. green: Evaluation metrics computed in a non-hierarchical fashion. orange: Main evaluation metric is computed on top of secondary metrics. FF: basic feed-forward layer.	144
7.3	Recap of different displays for an MR structure of a sentence that has medium length (left: PENMAN notation, right: graphical visualization). See also the Figure in Background Section 2.1.	146
7.4	We transform the (simplified) PENMAN representation to an image and use Φ to add latent channels.	147
7.5	Our model II. Architecture for efficient AMR quality assessment.	148
7.6	Training cost diagram of two approaches.	158
8.1	Standard, concept-focus and structure focus.	165

8.2	Full example (edge-labels omitted for simplified display) of explicit alignments between argument graphs (top) and automatically induced conclusions (bottom). Here, the conclusions help explaining argument similarity, since the alignment connects <i>fracking</i> in both graphs, as well as <i>water wells</i> and <i>toxic wastewater</i> , showing how <i>contaminating of the wells</i> (left graphs) actually happens: wells are polluted with toxic wastewater (right graphs).	171
8.3	Annotation results of two quality aspects with IAA: $\mathcal{K}=0.49$ (<i>justification</i>) and $\mathcal{K}=0.57$ (<i>novelty</i>).	173
9.1	Seq2seq SMATCH alignment-learner.	182
9.2	Implicit CNN-based SMATCH graph metric predictor.	182
9.3	AMR graph anonymization and permutation.	185
9.4	Overview of approach. ⚙️ The decomposition objective structures the sentence embedding space into AMR sentence features ($F_1 \dots F_K$): The process is guided by AMR metric approximation, through which S ³ BERT learns to disentangle and route the features. ☉ The consistency objective is aimed at preventing catastrophic forgetting: To preserve the overall effectiveness of the neural sentence embeddings, it controls the decomposition learning process and helps modeling the residual (R).	190
A.1	A critical issue and its alleviation.	217

List of Tables

3.1	STS label explanation and examples, taken from Agirre et al. (2013).	39
4.1	svr (Eq. 4.5), msv (Eq. 4.6) of AMR metrics.	59
4.2	svr (Eq. 4.5), msv (Eq. 4.6) of BLEU, MT setting.	59
4.3	Expected determinacy error ϵ in SMATCH F1.	59
4.4	Error impact depending on error location in a tree with node degree d .	62
4.5	S ² MATCH improves upon SMATCH by reducing the extent of its non-determinacy.	65
4.6	Evaluation of three AMR metrics using our eight principles. \checkmark_ϵ : fulfilled with a very small ϵ -error. +ILP column indicates a variant of the metric in the left neighbor column that uses optimal solution instead of hill-climbing.	67
4.7	Principle analysis update. On <i>efficiency</i> , in contrast to Table 4.6, we adopt a graded view, to indicate that WWLK is more efficient than SMATCH, but less efficient than SEMBLEU, which is in turn equally fast as WLK.	78
5.1	BAMBOO data set statistics of the Main partition. Sentence length (s. length, displayed for reference only) and graph statistics (average and median) are calculated on the training sets.	83
5.2	Three-way graph assessment. [x,y]: 95-confidence intervals estimated with bootstrap. † (‡) significant improvement of T5S2S over GPLA with $p < 0.05$ ($p < 0.005$).	84
5.3	Statistics about the amount of transform operations that were conducted, on average, on one graph. [x,y,z]: 25th, 50th (median) and 75th percentile of the amount of operations.	86
5.4	BAMBOO benchmark result of AMR metrics. All numbers are Pearson's $\rho \times 100$. ++: linear time complexity; +: polynomial time complexity; -: NP complete.	90

5.5	WLK variants with different K	94
5.6	(W)WLK: message passing directions.	94
5.7	Retrospective sub-sample quality analysis of BAMBOO graph quality and sensitivity of metrics. All values are Pearson’s $\rho \times 100$. Metric Human Agreement (MHA): $[x,y]$, where x is the correlation (to human ratings) when the metric is executed on the uncorrected sample and y is the same assessment on the manually post-processed sample.	95
5.8	Corpus level scoring results. Negative Δ shows preference for T5, positive Δ shows preference for BART.	105
5.9	Metric agreement with human. †: random baseline (RAND) not contained in 95% confidence interval.	110
6.1	Main metric results. <i>na</i> as upper-bound means that the upper-bound is not known and cannot be estimated. $\mathcal{M}\mathcal{F}_\beta$ is calculated from <i>Form</i> and $S^2\text{MATCH F1}$	128
6.2	Fine-grained corpus results using $\mathcal{M}\mathcal{F}_0$ (i.e., $S^2\text{MATCH}$) parameterized based on aspectual subgraphs.	134
6.3	Analysis of our metric using different parsers (GPLA, TTSA GSII) or ablating the gold parse by comparing the parsed generation against the parse (distant) source sentence (GSII \blacklozenge).	135
6.4	Studying $\mathcal{M}\mathcal{F}_\beta$ ranking under variation of MR metric (parser: GSII).	136
6.5	<i>Form</i> scores when using a different LM.	139
7.1	Equivalent MR representations and their accessibility with respect to human or computer (✓: ‘okay’, ✗: ‘perhaps possible, but not well defined’).	146
7.2	Parser output evaluation on training and development partitions of LDC2015E86. SMATCH F1: avg. over SMATCH F1 per sentence, % def.: percentage of deficient parses (i.e., parses with SMATCH F1 < 1).	151
7.3	Main results. Pearson’s corr. coefficient (row 1-3) is better if higher; root mean square error (RMSE, row 4-6) is better if lower. The quality dimensions are explained in Section 7.6. † (‡): $p < 0.05$ ($p < 0.005$), significant difference in the correlations with two-tailed test using Fisher ρ to z transformation (Fisher, 1915).	153
7.4	Results for AMR quality rating w.r.t. various sub-tasks. † (‡): significance (c.f. caption Table 7.3).	154

7.5	Performance-effects of data debiasing steps. <i>+aux</i> indicates a model variant that is trained using auxiliary losses that incorporate hierarchical information about the other AMR aspects in the training process.	155
7.6	Graph quality classification task. † (‡) significance with paired t-test at $p < 0.05$ ($p < 0.005$) over 10 random initializations.	156
7.7	Right column: results of our system when we abstain from feeding the dependency tree, and only show the sentence together with the candidate MR.	156
7.8	Efficiency analysis of two approaches.	157
8.1	Main results for argument similarity. †/‡: significant improvement over all baselines with $p < 0.05/p < 0.005$ (Student t-test).	168
8.2	Semantic predictors of human argument similarity. †/‡: significant with $p < 0.05/p < 0.005$	170
8.3	Macro F1 scores for predicted conclusion quality using AMR-based models $f(a, c)$, assessing various aspects. For single features, + show positive correlation; - negative correlation (levels 0.05, 0.005, 0.0005).	174
8.4	Predictors of conclusion usefulness.	175
8.5	AMR metrics detecting dissimilar arguments.	176
9.1	Results of experiments. time: Approximate time for computing a pairwise distance matrix on 1k AMRs on a TI 1080 GPU.	186
9.2	Experiments on different test subsets that represent different problem complexities predicted with our best model (<i>align. synthesis+voc+aug</i>). $\langle \rangle x$ vars means that one of two graphs contains $\langle \rangle x$ variables. <i>better</i> : is the drop in accuracy of the model vs. ORACLE smaller compared with the model tested on all data?	187
9.3	Spearmanr x 100 of AMR aspects. <i>Italics</i> : overall best. bold : best partitioning approach. <u>underlined</u> : improvement by more than 20 Spearmanr points.	196
9.4	Results on STSb and SICK using Spearmanr x 100; Speed measurements of parser (p) and metric inference (i), units are minutes (m) and seconds (s). 198	198
9.5	Results on argument similarity prediction.	198

9.6	AMR metric approximation upper-bounds. $S^3BERT^{cons.}$: S^3BERT without consistency objective (trades sentence similarity rating performance for better AMR approximation). $S^3BERT^{cons.}+parser$: S^3BERT without consistency objective and inference on linearized AMR graphs (trades sentence similarity rating performance <i>and</i> efficiency for better AMR approximation).	201
9.7	AMR prediction performance w.r.t. different training data sizes.	202
9.8	Prediction Examples from STSb and SICK, or own construction (human rating: na).	202
9.9	Similarity investigation with S^3BERT feature analysis. bold/(n) : best from a feature group (rank 1–3).	206
10.1	High-level MR-metric method decision guide for a selection of general applications that we visited in this thesis. ANY means any MR metric that measures distance in explicit MR metric spaces (SMATCH, S^2MATCH , SEMBLEU, SEMA, WLK, WWLK,...)	211
A.1	Results for assessing the <i>Form</i> score prediction (corpus-level) of different LMs for NLG-generated sentences against humans judgements (separated by grammaticality and fluency); all: all 12k generated sentences vs. 'poor/perfect': the 5k instances of best/worst generations in both grammaticality and fluency.	218
A.2	Overview of NMT hyper-parameters.	220
A.3	Overview of CNN hyper-parameters.	220
A.4	Evaluation on CheckList (Zeidler et al., 2022), Speamanr correlation. Lower part of the Table show S^3BERT sub-embeddings. 'GraCo' is a metric by Zeidler et al. (2022), we refer the reader to this paper.	221

Bibliography

- Abzianidze, Lasha, Rik van Noord, Hessel Haagsma, and Johan Bos (2019). “The First Shared Task on Discourse Representation Structure Parsing”. In: *Proceedings of the IWCS Shared Task on Semantic Parsing*. Gothenburg, Sweden: Association for Computational Linguistics. DOI: [10.18653/v1/W19-1201](https://doi.org/10.18653/v1/W19-1201). URL: <https://www.aclweb.org/anthology/W19-1201>.
- Agirre, Eneko, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe (2016). “SemEval-2016 Task 1: Semantic Textual Similarity, Monolingual and Cross-Lingual Evaluation”. In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. San Diego, California: Association for Computational Linguistics, pp. 497–511. DOI: [10.18653/v1/S16-1081](https://doi.org/10.18653/v1/S16-1081). URL: <https://aclanthology.org/S16-1081>.
- Agirre, Eneko, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo (2013). “*SEM 2013 shared task: Semantic Textual Similarity”. In: *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*. Atlanta, Georgia, USA: Association for Computational Linguistics, pp. 32–43. URL: <https://aclanthology.org/S13-1004>.
- Ajjour, Yamen, Henning Wachsmuth, Johannes Kiesel, Martin Potthast, Matthias Hagen, and Benno Stein (2019). “Data acquisition for argument search: The args. me corpus”. In: *Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz)*. Springer, pp. 48–59.
- Aker, Ahmet, Alfred Sliwa, Yuan Ma, Ruishen Lui, Niravkumar Borad, Seyedeh Ziyaei, and Mina Ghobadi (2017). “What works and what does not: Classifier and feature analysis for argument mining”. In: *Proceedings of the 4th Workshop on Argument Mining*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 91–96. DOI: [10.18653/v1/W17-5112](https://doi.org/10.18653/v1/W17-5112). URL: <https://aclanthology.org/W17-5112>.

- Al-Khatib, Khalid, Yufang Hou, Henning Wachsmuth, Charles Jochim, Francesca Bonin, and Benno Stein (2020). “End-to-end argumentation knowledge graph construction”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 05, pp. 7367–7374.
- Alshomary, Milad, Wei-Fan Chen, Timon Gurcke, and Henning Wachsmuth (2021). “Belief-based Generation of Argumentative Claims”. In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online: Association for Computational Linguistics, pp. 224–233. URL: <https://aclanthology.org/2021.eacl-main.17>.
- Alshomary, Milad, Shahbaz Syed, Martin Potthast, and Henning Wachsmuth (2020). “Target Inference in Argument Conclusion Generation”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 4334–4345. DOI: [10.18653/v1/2020.acl-main.399](https://doi.org/10.18653/v1/2020.acl-main.399). URL: <https://aclanthology.org/2020.acl-main.399>.
- Anchiêta, Rafael Torres, Marco Antonio Sobrevilla Cabezudo, and Thiago Alexandre Salgueiro Pardo (2019). “SEMA: an Extended Semantic Evaluation for AMR”. In: *(To appear) Proceedings of the 20th Computational Linguistics and Intelligent Text Processing*. Ed. by Alexander Gelbukh. Springer International Publishg.
- Anderson, Peter, Basura Fernando, Mark Johnson, and Stephen Gould (2016). “SPICE: Semantic Propositional Image Caption Evaluation”. In: *Computer Vision – ECCV 2016*. Ed. by Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling. Cham: Springer International Publishing, pp. 382–398. ISBN: 978-3-319-46454-1.
- Banarescu, Laura, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider (2013). “Abstract Meaning Representation for Sembanking”. In: *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*. Sofia, Bulgaria: Association for Computational Linguistics, pp. 178–186. URL: <https://www.aclweb.org/anthology/W13-2322>.
- Banerjee, Satanjeev and Ted Pedersen (2002). “An adapted Lesk algorithm for word sense disambiguation using WordNet”. In: *International conference on intelligent text processing and computational linguistics*. Springer, pp. 136–145.
- Bastings, Jasmijn and Katja Filippova (2020). “The elephant in the interpretability room: Why use attention as explanation when we have saliency methods?” In: *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*. Online: Association for Computational Linguistics, pp. 149–155. DOI: [10.18653/v1/2020.blackboxnlp.10](https://doi.org/10.18653/v1/2020.blackboxnlp.10).

- 18653/v1/2020.blackboxnlp-1.14. URL: <https://aclanthology.org/2020.blackboxnlp-1.14>.
- Bastings, Joost, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Sima'an (2017). "Graph Convolutional Encoders for Syntax-aware Neural Machine Translation". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 1957–1967. DOI: 10.18653/v1/D17-1209. URL: <https://www.aclweb.org/anthology/D17-1209>.
- Baudiš, Petr, Jan Pichl, Tomáš Vyskočil, and Jan Šedivý (2016a). "Sentence pair scoring: Towards unified framework for text comprehension". In: *arXiv preprint arXiv:1603.06127*.
- Baudiš, Petr, Silvestr Stanko, and Jan Šedivý (2016b). "Joint Learning of Sentence Embeddings for Relevance and Entailment". In: *Proceedings of the 1st Workshop on Representation Learning for NLP*. Berlin, Germany: Association for Computational Linguistics, pp. 8–17. DOI: 10.18653/v1/W16-1602. URL: <https://www.aclweb.org/anthology/W16-1602>.
- Beck, Daniel, Gholamreza Haffari, and Trevor Cohn (2018). "Graph-to-Sequence Learning using Gated Graph Neural Networks". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 273–283. DOI: 10.18653/v1/P18-1026. URL: <https://aclanthology.org/P18-1026>.
- Becker, Maria, Ioana Hulpuș, Juri Opitz, Debjit Paul, Jonathan Kobbe, Heiner Stuckenschmidt, and Anette Frank (2020). "Explaining arguments with background knowledge". In: *Datenbank-Spektrum* 20.2, pp. 131–141.
- Becker, Maria, Siting Liang, and Anette Frank (2021). "Reconstructing Implicit Knowledge with Language Models". In: *Proceedings of Deep Learning Inside Out (DeeLIO): The 2nd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*. Online: Association for Computational Linguistics, pp. 11–24. DOI: 10.18653/v1/2021.deelio-1.2. URL: <https://aclanthology.org/2021.deelio-1.2>.
- Beel, Joeran, Bela Gipp, Stefan Langer, and Corinna Breitinger (2016). "Paper recommender systems: a literature survey". In: *International Journal on Digital Libraries* 17.4, pp. 305–338.
- Behrendt, Maike and Stefan Harmeling (2021). "ArgueBERT: How To Improve BERT Embeddings for Measuring the Similarity of Arguments". In: *Proceedings of the 17th*

- Conference on Natural Language Processing (KONVENS 2021)*. Düsseldorf, Germany: KONVENS 2021 Organizers, pp. 28–36. URL: <https://aclanthology.org/2021.konvens-1.3>.
- Beltagy, Iz, Matthew E Peters, and Arman Cohan (2020). “Longformer: The long-document transformer”. In: *arXiv preprint arXiv:2004.05150*.
- Bender, Emily M. and Alexander Koller (2020). “Climbing towards NLU: On Meaning, Form, and Understanding in the Age of Data”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 5185–5198. DOI: [10.18653/v1/2020.acl-main.463](https://doi.org/10.18653/v1/2020.acl-main.463). URL: <https://www.aclweb.org/anthology/2020.acl-main.463>.
- Bevilacqua, Michele, Rexhina Blloshmi, and Roberto Navigli (2021). “One SPRING to rule them both: Symmetric AMR semantic parsing and generation without a complex pipeline”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 14, pp. 12564–12573.
- Blloshmi, Rexhina, Rocco Tripodi, and Roberto Navigli (2020). “XL-AMR: Enabling Cross-Lingual AMR Parsing with Transfer Learning Techniques”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pp. 2487–2500. DOI: [10.18653/v1/2020.emnlp-main.195](https://doi.org/10.18653/v1/2020.emnlp-main.195). URL: <https://www.aclweb.org/anthology/2020.emnlp-main.195>.
- Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov (2016). “Enriching Word Vectors with Subword Information”. In: *arXiv preprint arXiv:1607.04606*.
- Bonial, Claire, Stephanie M. Lukin, David Doughty, Steven Hill, and Clare Voss (2020). “InfoForager: Leveraging Semantic Search with AMR for COVID-19 Research”. In: *Proceedings of the Second International Workshop on Designing Meaning Representations*. Barcelona Spain (online): Association for Computational Linguistics, pp. 67–77. URL: <https://www.aclweb.org/anthology/2020.dmr-1.7>.
- Boole, George (1854). *An investigation of the laws of thought: on which are founded the mathematical theories of logic and probabilities*. Dover.
- Bos, Johan (2016). “Expressive power of abstract meaning representations”. In: *Computational Linguistics* 42.3, pp. 527–535.
- Bos, Johan (2019). “Separating Argument Structure from Logical Structure in AMR”. In: *arXiv preprint arXiv:1908.01355*.

- Bryant, Christopher and Ted Briscoe (2018). “Language Model Based Grammatical Error Correction without Annotated Training Data”. In: *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 247–253. DOI: [10.18653/v1/W18-0529](https://doi.org/10.18653/v1/W18-0529). URL: <https://www.aclweb.org/anthology/W18-0529>.
- Budanitsky, Alexander and Graeme Hirst (2006). “Evaluating wordnet-based measures of lexical semantic relatedness”. In: *Computational linguistics* 32.1, pp. 13–47.
- Bunke, Horst and Gudrun Allermann (1983). “Inexact graph matching for structural pattern recognition”. In: *Pattern Recognition Letters* 1.4, pp. 245–253.
- Burago, Dmitri, Yuri Burago, and Sergei Ivanov (2022). *A course in metric geometry*. Vol. 33. American Mathematical Society.
- Cai, Deng and Wai Lam (2019). “Core Semantic First: A Top-down Approach for AMR Parsing”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 3799–3809. DOI: [10.18653/v1/D19-1393](https://doi.org/10.18653/v1/D19-1393). URL: <https://www.aclweb.org/anthology/D19-1393>.
- Cai, Deng and Wai Lam (2020a). “AMR Parsing via Graph-Sequence Iterative Inference”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 1290–1301. DOI: [10.18653/v1/2020.acl-main.119](https://doi.org/10.18653/v1/2020.acl-main.119). URL: <https://www.aclweb.org/anthology/2020.acl-main.119>.
- Cai, Deng and Wai Lam (2020b). “Graph Transformer for Graph-to-Sequence Learning”. In: *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, pp. 7464–7471. URL: <https://aaai.org/ojs/index.php/AAAI/article/view/6243>.
- Cai, Shu and Kevin Knight (2013). “Smatch: an Evaluation Metric for Semantic Feature Structures”. In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Sofia, Bulgaria: Association for Computational Linguistics, pp. 748–752. URL: <https://www.aclweb.org/anthology/P13-2131>.

- Carpuat, Marine (2013). “A Semantic Evaluation of Machine Translation Lexical Choice”. In: *Proceedings of the Seventh Workshop on Syntax, Semantics and Structure in Statistical Translation*. Atlanta, Georgia: Association for Computational Linguistics, pp. 1–10. URL: <https://www.aclweb.org/anthology/W13-0801>.
- Cer, Daniel, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia (2017). “SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation”. In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Vancouver, Canada: Association for Computational Linguistics, pp. 1–14. DOI: [10.18653/v1/S17-2001](https://doi.org/10.18653/v1/S17-2001). URL: <https://www.aclweb.org/anthology/S17-2001>.
- Chatterjee, Rajen, Matteo Negri, Marco Turchi, Frédéric Blain, and Lucia Specia (2018). “Combining Quality Estimation and Automatic Post-editing to Enhance Machine Translation output”. In: *AMTA (1)*. Association for Machine Translation in the Americas, pp. 26–38.
- Chen, Boxing and Colin Cherry (2014). “A Systematic Comparison of Smoothing Techniques for Sentence-Level BLEU”. In: *Proceedings of the Ninth Workshop on Statistical Machine Translation*. Baltimore, Maryland, USA: Association for Computational Linguistics, pp. 362–367. DOI: [10.3115/v1/W14-3346](https://doi.org/10.3115/v1/W14-3346). URL: <https://www.aclweb.org/anthology/W14-3346>.
- Chesnevar, Carlos Iván and Ana G Maguitman (2004). “Arguenet: An argument-based recommender system for solving web search queries”. In: *2004 2nd International IEEE Conference on Intelligent Systems. Proceedings (IEEE Cat. No. 04EX791)*. Vol. 1. IEEE, pp. 282–287.
- Choromanski, Krzysztof Marcin et al. (2021). “Rethinking Attention with Performers”. In: *International Conference on Learning Representations*.
- Clark, Kevin, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning (2019). “What Does BERT Look at? An Analysis of BERT’s Attention”. In: *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Florence, Italy: Association for Computational Linguistics, pp. 276–286. DOI: [10.18653/v1/W19-4828](https://doi.org/10.18653/v1/W19-4828). URL: <https://aclanthology.org/W19-4828>.
- Cohen, Jacob (1968). “Weighted kappa: nominal scale agreement with provision for scaled disagreement or partial credit”. In: *Psychological bulletin*, pp. 213–220.
- Conn, Andrew R, Katya Scheinberg, and Luis N Vicente (2009). *Introduction to derivative-free optimization*. SIAM.

- Conneau, Alexis, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes (2017). “Supervised Learning of Universal Sentence Representations from Natural Language Inference Data”. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 670–680. DOI: [10.18653/v1/D17-1070](https://doi.org/10.18653/v1/D17-1070). URL: <https://aclanthology.org/D17-1070>.
- Damonte, Marco, Shay B. Cohen, and Giorgio Satta (2017). “An Incremental Parser for Abstract Meaning Representation”. In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Valencia, Spain: Association for Computational Linguistics, pp. 536–546. URL: <https://aclanthology.org/E17-1051>.
- Danilevsky, Marina, Kun Qian, Ranit Aharonov, Yannis Katsis, Ban Kawas, and Prithviraj Sen (2020). “A Survey of the State of Explainable AI for Natural Language Processing”. In: *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*. Suzhou, China: Association for Computational Linguistics, pp. 447–459. URL: <https://aclanthology.org/2020.aacl-main.46>.
- Das, Nibaran, Swarnendu Ghosh, Teresa Gonçalves, and Paulo Quaresma (2014). “Comparison of different graph distance metrics for semantic text based classification”. In: *Polibits* 49, pp. 51–58.
- Davidson, Donald and Nicholas Rescher (1967). “The logical form of action sentences”. In: *1967*, pp. 105–122.
- Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019). “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186. DOI: [10.18653/v1/N19-1423](https://doi.org/10.18653/v1/N19-1423). URL: <https://www.aclweb.org/anthology/N19-1423>.
- Dietterich, Tom (1995). “Overfitting and undercomputing in machine learning”. In: *ACM computing surveys (CSUR)* 27.3, pp. 326–327.

- Dolan, William B. and Chris Brockett (2005). “Automatically Constructing a Corpus of Sentential Paraphrases”. In: *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*. URL: <https://www.aclweb.org/anthology/I05-5002>.
- Donatelli, Lucia, Michael Regan, William Croft, and Nathan Schneider (2018). “Annotation of Tense and Aspect Semantics for Sentential AMR”. In: *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, pp. 96–108. URL: <https://www.aclweb.org/anthology/W18-4912>.
- Dowty, David (1991). “Thematic proto-roles and argument selection”. In: *language* 67.3, pp. 547–619.
- Dung, Phan Minh (1995). “On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games”. In: *Artificial intelligence* 77.2, pp. 321–357.
- Erk, Katrin and Sebastian Padó (2008). “A Structured Vector Space Model for Word Meaning in Context”. In: *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. Honolulu, Hawaii: Association for Computational Linguistics, pp. 897–906. URL: <https://aclanthology.org/D08-1094>.
- Ferrando, Javier and Marta R. Costa-jussà (2021). “Attention Weights in Transformer NMT Fail Aligning Words Between Sequences but Largely Explain Model Predictions”. In: *Findings of the Association for Computational Linguistics: EMNLP 2021*. Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 434–443. DOI: [10.18653/v1/2021.findings-emnlp.39](https://doi.org/10.18653/v1/2021.findings-emnlp.39). URL: <https://aclanthology.org/2021.findings-emnlp.39>.
- Fisher, Ronald A (1915). “Frequency distribution of the values of the correlation coefficient in samples from an indefinitely large population”. In: *Biometrika* 10.4, pp. 507–521.
- Flanigan, Jeffrey, Sam Thomson, Jaime Carbonell, Chris Dyer, and Noah A. Smith (2014). “A Discriminative Graph-Based Parser for the Abstract Meaning Representation”. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Baltimore, Maryland: Association for Computational Linguistics, pp. 1426–1436. DOI: [10.3115/v1/P14-1134](https://doi.org/10.3115/v1/P14-1134). URL: <https://aclanthology.org/P14-1134>.

- Freitag, Markus, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, George Foster, Alon Lavie, and Ondřej Bojar (2021). “Results of the WMT21 Metrics Shared Task: Evaluating Metrics with Expert-based Human Evaluations on TED and News Domain”. In: *Proceedings of the Sixth Conference on Machine Translation*. Online: Association for Computational Linguistics, pp. 733–774. URL: <https://aclanthology.org/2021.wmt-1.73>.
- Gao, Xinbo, Bing Xiao, Dacheng Tao, and Xuelong Li (2010). “A survey of graph edit distance”. In: *Pattern Analysis and applications* 13.1, pp. 113–129.
- Gardent, Claire, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini (2017). “The WebNLG Challenge: Generating Text from RDF Data”. In: *Proceedings of the 10th International Conference on Natural Language Generation*. Santiago de Compostela, Spain: Association for Computational Linguistics, pp. 124–133. DOI: [10.18653/v1/W17-3518](https://doi.org/10.18653/v1/W17-3518). URL: <https://www.aclweb.org/anthology/W17-3518>.
- Garey, M. R. and David S. Johnson (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman. ISBN: 0-7167-1044-7.
- Gehrmann, Sebastian, Tosin Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Aremu Anuoluwapo, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna Clinciu, Dipanjan Das, Kaustubh D Dhole, et al. (2021). “The GEM Benchmark: Natural Language Generation, its Evaluation and Metrics”. In: *arXiv preprint arXiv:2102.01672*.
- Godden, David M and Douglas Walton (2006). “Argument from expert opinion as legal evidence: Critical questions and admissibility criteria of expert testimony in the American legal system”. In: *Ratio Juris* 19.3, pp. 261–286.
- Gorman, Kyle and Steven Bedrick (2019). “We Need to Talk about Standard Splits”. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 2786–2791. DOI: [10.18653/v1/P19-1267](https://doi.org/10.18653/v1/P19-1267). URL: <https://www.aclweb.org/anthology/P19-1267>.
- Groschwitz, Jonas, Shay B Cohen, Lucia Donatelli, and Meaghan Fowlie (2023). “AMR Parsing is Far from Solved: GrAPES, the Granular AMR Parsing Evaluation Suite”. In: *arXiv preprint arXiv:2312.03480*.
- Groschwitz, Jonas, Matthias Lindemann, Meaghan Fowlie, Mark Johnson, and Alexander Koller (2018). “AMR dependency parsing with a typed semantic algebra”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational

- Linguistics, pp. 1831–1841. DOI: [10.18653/v1/P18-1170](https://doi.org/10.18653/v1/P18-1170). URL: <https://www.aclweb.org/anthology/P18-1170>.
- Gunning, David, Mark Stefik, Jaesik Choi, Timothy Miller, Simone Stumpf, and Guang-Zhong Yang (2019). “XAI—Explainable artificial intelligence”. In: *Science Robotics* 4.37.
- Guo, Zhijiang, Yan Zhang, Zhiyang Teng, and Wei Lu (2019). “Densely Connected Graph Convolutional Networks for Graph-to-Sequence Learning”. In: *Transactions of the Association for Computational Linguistics* 7, pp. 297–312. DOI: [10.1162/tacl_a_00269](https://doi.org/10.1162/tacl_a_00269). URL: <https://www.aclweb.org/anthology/Q19-1019>.
- Harary, Frank and Robert Z Norman (1960). “Some properties of line digraphs”. In: *Rendiconti del circolo matematico di palermo* 9.2, pp. 161–168.
- Heilman, Michael, Aoife Cahill, Nitin Madnani, Melissa Lopez, Matthew Mulholland, and Joel Tetreault (2014). “Predicting Grammaticality on an Ordinal Scale”. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Baltimore, Maryland: Association for Computational Linguistics, pp. 174–180. DOI: [10.3115/v1/P14-2029](https://doi.org/10.3115/v1/P14-2029). URL: <https://www.aclweb.org/anthology/P14-2029>.
- Heinisch, Philipp, Anette Frank, Juri Opitz, Moritz Plenz, and Philipp Cimiano (2022a). “Overview of the 2022 Validity and Novelty Prediction Shared Task”. In: *Proceedings of the 9th Workshop on Argument Mining*. Online and in Gyeongju, Republic of Korea: International Conference on Computational Linguistics, pp. 84–94. URL: <https://aclanthology.org/2022.argmining-1.7>.
- Heinisch, Philipp, Moritz Plenz, Juri Opitz, Anette Frank, and Philipp Cimiano (2022b). “Data Augmentation for Improving the Prediction of Validity and Novelty of Argumentative Conclusions”. In: *Proceedings of the 9th Workshop on Argument Mining*. Online and in Gyeongju, Republic of Korea: International Conference on Computational Linguistics, pp. 19–33. URL: <https://aclanthology.org/2022.argmining-1.2>.
- Heinzerling, Benjamin (2020). “NLP’s Clever Hans Moment has Arrived”. In: *Journal of Cognitive Science* 21.1, pp. 159–168.

- Hill, Felix, Kyunghyun Cho, and Anna Korhonen (2016). “Learning Distributed Representations of Sentences from Unlabelled Data”. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, pp. 1367–1377. DOI: [10.18653/v1/N16-1162](https://doi.org/10.18653/v1/N16-1162). URL: <https://aclanthology.org/N16-1162>.
- Hoang, Thanh Lam, Gabriele Picco, Yufang Hou, Young-Suk Lee, Lam Nguyen, Dzung Phan, Vanessa Lopez, and Ramon Fernandez Astudillo (2021). “Ensembling Graph Predictions for AMR Parsing”. In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan. Vol. 34. Curran Associates, Inc., pp. 8495–8505.
- Hochreiter, Sepp and Jürgen Schmidhuber (1997). “Long Short-Term Memory”. In: *Neural Comput.* 9.8, pp. 1735–1780. ISSN: 0899-7667. DOI: [10.1162/neco.1997.9.8.1735](https://doi.org/10.1162/neco.1997.9.8.1735). URL: <http://dx.doi.org/10.1162/neco.1997.9.8.1735>.
- Hofmann, Thomas, Bernhard Schölkopf, and Alexander J Smola (2008). “Kernel methods in machine learning”. In: *The annals of statistics*, pp. 1171–1220.
- Honnibal, Matthew and Ines Montani (2017). “spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing”. To appear.
- Hovy, Dirk, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard Hovy (2013). “Learning Whom to Trust with MACE”. In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Atlanta, Georgia: Association for Computational Linguistics, pp. 1120–1130. URL: <https://aclanthology.org/N13-1132>.
- Hoyle, Alexander Miserlis, Ana Marasović, and Noah A. Smith (2021). “Promoting Graph Awareness in Linearized Graph-to-Text Generation”. In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Online: Association for Computational Linguistics, pp. 944–956. DOI: [10.18653/v1/2021.findings-acl.82](https://doi.org/10.18653/v1/2021.findings-acl.82). URL: <https://aclanthology.org/2021.findings-acl.82>.
- Huck, Matthias, Fabienne Braune, and Alexander Fraser (2017). “LMU Munich’s Neural Machine Translation Systems for News Articles and Health Information Texts”. In: *Proceedings of the Second Conference on Machine Translation*. Copenhagen, Denmark, pp. 315–322. DOI: [10.18653/v1/W17-4730](https://doi.org/10.18653/v1/W17-4730). URL: <https://www.aclweb.org/anthology/W17-4730>.

- Jaccard, Paul (1912). “The distribution of the flora in the alpine zone”. In: *New phytologist* 11.2, pp. 37–50.
- Jain, Sarthak and Byron C. Wallace (2019). “Attention is not Explanation”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 3543–3556. DOI: [10.18653/v1/N19-1357](https://doi.org/10.18653/v1/N19-1357). URL: <https://aclanthology.org/N19-1357>.
- Jawahar, Ganesh, Benoît Sagot, and Djamé Seddah (2019). “What Does BERT Learn about the Structure of Language?” In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 3651–3657. DOI: [10.18653/v1/P19-1356](https://doi.org/10.18653/v1/P19-1356). URL: <https://aclanthology.org/P19-1356>.
- Jo, Yohan, Sejin Bang, Chris Reed, and Eduard H. Hovy (2021). “Classifying Argumentative Relations Using Logical Mechanisms and Argumentation Schemes”. In: *CoRR* abs/2105.07571. arXiv: [2105.07571](https://arxiv.org/abs/2105.07571). URL: <https://arxiv.org/abs/2105.07571>.
- Jones, Karen Sparck (1972). “A statistical interpretation of term specificity and its application in retrieval”. In: *Journal of documentation*.
- Joshi, Nisheeth, Iti Mathur, Hemant Darbari, and Ajai Kumar (2016). “Quality Estimation of English-Hindi Machine Translation Systems”. In: *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies*. ICTCS ’16. Udaipur, India: ACM, 53:1–53:5. ISBN: 978-1-4503-3962-9. DOI: [10.1145/2905055.2905259](https://doi.org/10.1145/2905055.2905259). URL: <http://doi.acm.org/10.1145/2905055.2905259>.
- Kamp, Hans (1981). “A theory of truth and semantic representation”. In: *Formal semantics—the essential readings*, pp. 189–222.
- Kapanipathi, Pavan, Ibrahim Abdelaziz, Srinivas Ravishankar, Salim Roukos, Alexander Gray, Ramon Astudillo, Maria Chang, Cristina Cornelio, Saswati Dana, Achille Fokoue, et al. (2021). “Leveraging Abstract Meaning Representation for Knowledge Base Question Answering”. In: *Findings of the Association for Computational Linguistics: ACL*.

- Kasper, Robert T. (1989). “A Flexible Interface for Linking Applications to Penman’s Sentence Generator”. In: *Proceedings of the Workshop on Speech and Natural Language*. HLT ’89. Philadelphia, Pennsylvania: Association for Computational Linguistics, pp. 153–158. DOI: [10.3115/100964.100979](https://doi.org/10.3115/100964.100979). URL: <https://doi.org/10.3115/100964.100979>.
- Kaster, Marvin, Wei Zhao, and Steffen Eger (2021). “Global Explainability of BERT-Based Evaluation Metrics by Disentangling along Linguistic Factors”. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 8912–8925. DOI: [10.18653/v1/2021.emnlp-main.701](https://aclanthology.org/2021.emnlp-main.701). URL: <https://aclanthology.org/2021.emnlp-main.701>.
- Katinskaia, Anisia and Sardana Ivanova (2019). “Multiple Admissibility: Judging Grammaticality using Unlabeled Data in Language Learning”. In: *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*. Florence, Italy: Association for Computational Linguistics, pp. 12–22. DOI: [10.18653/v1/W19-3702](https://www.aclweb.org/anthology/W19-3702). URL: <https://www.aclweb.org/anthology/W19-3702>.
- Kemker, Ronald, Marc McClure, Angelina Abitino, Tyler Hayes, and Christopher Kanan (2018). “Measuring Catastrophic Forgetting in Neural Networks”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 32.1. DOI: [10.1609/aaai.v32i1.11651](https://ojs.aaai.org/index.php/AAAI/article/view/11651). URL: <https://ojs.aaai.org/index.php/AAAI/article/view/11651>.
- Kiefer, Jack, Jacob Wolfowitz, et al. (1952). “Stochastic estimation of the maximum of a regression function”. In: *The Annals of Mathematical Statistics* 23.3, pp. 462–466.
- Kim, Hyun, Hun-Young Jung, Hongseok Kwon, Jong-Hyeok Lee, and Seung-Hoon Na (2017). “Predictor-Estimator: Neural Quality Estimation Based on Target Word Prediction for Machine Translation”. In: *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 17.1, 3:1–3:22. ISSN: 2375-4699. DOI: [10.1145/3109480](https://doi.acm.org/10.1145/3109480). URL: [http://doi.acm.org/10.1145/3109480](https://doi.acm.org/10.1145/3109480).
- Kim, Yoon (2014). “Convolutional Neural Networks for Sentence Classification”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pp. 1746–1751. DOI: [10.3115/v1/D14-1181](https://www.aclweb.org/anthology/D14-1181). URL: <https://www.aclweb.org/anthology/D14-1181>.
- Kingma, Diederik P and J Adam Ba (2019). “A method for stochastic optimization. arXiv 2014”. In: *arXiv preprint arXiv:1412.6980* 434.

- Kingsbury, Paul and Martha Palmer (2002). “From TreeBank to PropBank”. In: *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC’02)*.
- Knight, Kevin, Bianca Badarau, Laura Baranescu, Claire Bonial, Madalina Bardocz, Kira Griffitt, Ulf Hermjakob, Daniel Marcu, Martha Palmer, Tim O’Gorman, et al. (2021). “Abstract meaning representation (amr) annotation release 3.0”. In:
- Knight, Kevin, Laura Baranescu, Claire Bonial, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Daniel Marcu, Martha Palmer, and Nathan Schneider (2014). “Abstract meaning representation (AMR) annotation release 1.0 LDC2014T12”. In: *Web Download. Philadelphia: Linguistic Data Consortium*.
- Kobbe, Jonathan, Juri Opitz, Maria Becker, Ioana Hulpus, Heiner Stuckenschmidt, and Anette Frank (2019). “Exploiting Background Knowledge for Argumentative Relation Classification”. In: *2nd Conference on Language, Data and Knowledge (LDK 2019)*. Ed. by Maria Eskevich, Gerard de Melo, Christian Fäth, John P. McCrae, Paul Buitelaar, Christian Chiarcos, Bettina Klimek, and Milan Dojchinovski. Vol. 70. OpenAccess Series in Informatics (OASICs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 8:1–8:14. ISBN: 978-3-95977-105-4. DOI: [10 . 4230 / OASICs . LDK . 2019 . 8](https://doi.org/10.4230/OASICs.LDK.2019.8). URL: <http://drops.dagstuhl.de/opus/volltexte/2019/10372>.
- Kolb, Peter (2009). “Experiments on the difference between semantic similarity and relatedness”. In: *Proceedings of the 17th Nordic Conference of Computational Linguistics (NODALIDA 2009)*. Odense, Denmark: Northern European Association for Language Technology (NEALT), pp. 81–88. URL: <https://www.aclweb.org/anthology/W09-4613>.
- Kondor, Risi and Karsten M Borgwardt (2008). “The skew spectrum of graphs”. In: *Proceedings of the 25th international conference on Machine learning*, pp. 496–503.
- Kondor, Risi, Nino Shervashidze, and Karsten M Borgwardt (2009). “The graphlet spectrum”. In: *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 529–536.
- Konstas, Ioannis, Srinivasan Iyer, Mark Yatskar, Yejin Choi, and Luke Zettlemoyer (2017). “Neural AMR: Sequence-to-Sequence Models for Parsing and Generation”. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, pp. 146–157. DOI: [10 . 18653 / v1 / P17 - 1014](https://doi.org/10.18653/v1/P17-1014). URL: <https://www.aclweb.org/anthology/P17-1014>.

- Koponen, Maarit, Leena Salmi, and Markku Nikulin (2019). “A product and process analysis of post-editor corrections on neural, statistical and rule-based machine translation output”. In: *Machine Translation* 33.1-2, pp. 61–90.
- Krausz, József (1943). “Démonstration nouvelle d’une théoreme de Whitney sur les réseaux”. In: *Mat. Fiz. Lapok* 50.1, pp. 75–85.
- Kriege, Nils M, Pierre-Louis Giscard, and Richard Wilson (2016a). “On valid optimal assignment kernels and applications to graph classification”. In: *Advances in neural information processing systems* 29.
- Kriege, Nils M, Pierre-Louis Giscard, and Richard Wilson (2016b). “On valid optimal assignment kernels and applications to graph classification”. In: *Advances in neural information processing systems* 29.
- Kroch, Anthony S. (1978). “Toward a theory of social dialect variation”. In: *Language in Society* 7.1, 17–36. DOI: [10.1017/S0047404500005315](https://doi.org/10.1017/S0047404500005315).
- Lambrech, Knud (1996). *Information structure and sentence form: Topic, focus, and the mental representations of discourse referents*. Vol. 71. Cambridge university press.
- Lample, Guillaume and François Charton (2020). “Deep Learning For Symbolic Mathematics”. In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. URL: <https://openreview.net/forum?id=SleZYeHFDS>.
- Lasersohn, Peter (2016). *A semantics for groups and events*. Routledge.
- Lau, Jey Han, Carlos S Armendariz, Shalom Lappin, Matthew Purver, and Chang Shu (2020). “How Furiously Can Colourless Green Ideas Sleep? Sentence Acceptability in Context”. In: *arXiv preprint arXiv:2004.00881*.
- Lauscher, Anne, Henning Wachsmuth, Iryna Gurevych, and Goran Glavaš (2021). *Scientia Potentia Est – On the Role of Knowledge in Computational Argumentation*. arXiv: [2107.00281](https://arxiv.org/abs/2107.00281) [cs.CL].
- Lawrence, John (2021). “Explainable argument mining”. PhD thesis. University of Dundee.
- Leiter, Christoph, Juri Opitz, Daniel Deutsch, Yang Gao, Rotem Dror, and Steffen Eger (2023). “The Eval4NLP 2023 Shared Task on Prompting Large Language Models as Explainable Metrics”. In: *Proceedings of the 4th Workshop on Evaluation and Comparison for NLP systems*.
- Lenz, Mirko, Stefan Ollinger, Premtim Sahitaj, and Ralph Bergmann (2019). “Semantic textual similarity measures for case-based retrieval of argument graphs”. In: *International Conference on Case-Based Reasoning*. Springer, pp. 219–234.

- Lenz, Mirko, Premtim Sahitaj, Sean Kallenberg, Christopher Coors, Lorik Dumani, Ralf Schenkel, and Ralph Bergmann (2020). “Towards an argument mining pipeline transforming texts to argument graphs”. In: *Computational Models of Argument: Proceedings of COMMA 2020* 326, p. 263.
- Lepori, Michael and R. Thomas McCoy (2020). “Picking BERT’s Brain: Probing for Linguistic Dependencies in Contextualized Embeddings Using Representational Similarity Analysis”. In: *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, pp. 3637–3651. DOI: [10.18653/v1/2020.coling-main.325](https://doi.org/10.18653/v1/2020.coling-main.325). URL: <https://aclanthology.org/2020.coling-main.325>.
- Lesk, Michael (1986). “Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone”. In: *Proceedings of the 5th annual international conference on Systems documentation*, pp. 24–26.
- Leung, Wai Ching, Shira Wein, and Nathan Schneider (2022). “Semantic Similarity as a Window into Vector- and Graph-Based Metrics”. In: *Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*. Abu Dhabi, United Arab Emirates (Hybrid): Association for Computational Linguistics, pp. 106–115. URL: <https://aclanthology.org/2022.gem-1.8>.
- Levi, Friedrich Wilhelm (1942). *Finite geometrical systems: six public lectures delivered in February, 1940, at the University of Calcutta*. University of Calcutta.
- Lewis, Mike, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer (2020). “BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 7871–7880. DOI: [10.18653/v1/2020.acl-main.703](https://doi.org/10.18653/v1/2020.acl-main.703). URL: <https://aclanthology.org/2020.acl-main.703>.
- Li, Liang, Ruiying Geng, Bowen Li, Can Ma, Yinliang Yue, Binhua Li, and Yongbin Li (2022). “Graph-to-Text Generation with Dynamic Structure Pruning”. In: *arXiv preprint arXiv:2209.07258*.
- Lin, Chin-Yew (2004). “ROUGE: A Package for Automatic Evaluation of Summaries”. In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, pp. 74–81. URL: <https://www.aclweb.org/anthology/W04-1013>.

- Liu, Hugo and Push Singh (2004). “ConceptNet—a practical commonsense reasoning tool-kit”. In: *BT technology journal* 22.4, pp. 211–226.
- Liu, Jiangming, Shay B. Cohen, and Mirella Lapata (2020). “Dscorer: A Fast Evaluation Metric for Discourse Representation Structure Parsing”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 4547–4554. DOI: [10.18653/v1/2020.acl-main.416](https://doi.org/10.18653/v1/2020.acl-main.416). URL: <https://aclanthology.org/2020.acl-main.416>.
- Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov (2019a). “Roberta: A robustly optimized bert pretraining approach”. In: *arXiv preprint arXiv:1907.11692*.
- Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov (2019b). “RoBERTa: A Robustly Optimized BERT Pretraining Approach”. In: *arXiv preprint arXiv:1907.11692*.
- Lo, Chi-kiu (2017). “MEANT 2.0: Accurate semantic MT evaluation for any output language”. In: *Proceedings of the Second Conference on Machine Translation*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 589–597. DOI: [10.18653/v1/W17-4767](https://doi.org/10.18653/v1/W17-4767). URL: <https://www.aclweb.org/anthology/W17-4767>.
- Logan, Robert, Nelson F. Liu, Matthew E. Peters, Matt Gardner, and Sameer Singh (2019). “Barack’s Wife Hillary: Using Knowledge Graphs for Fact-Aware Language Modeling”. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 5962–5971. DOI: [10.18653/v1/P19-1598](https://doi.org/10.18653/v1/P19-1598). URL: <https://www.aclweb.org/anthology/P19-1598>.
- Lorenzo, Martínez, Pere Lluís Huguet Cabot, and Roberto Navigli (2023). “AMRs Assemble! Learning to Ensemble with Autoregressive Models for AMR Parsing”. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Toronto, Canada: Association for Computational Linguistics, pp. 1595–1605. URL: <https://aclanthology.org/2023.acl-short.137>.
- Lorenzo, Martínez, Marco Maru, and Roberto Navigli (2022). “Fully-Semantic Parsing and Generation: the BabelNet Meaning Representation”. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, pp. 1727–1741.

- DOI: [10.18653/v1/2022.acl-long.121](https://doi.org/10.18653/v1/2022.acl-long.121). URL: <https://aclanthology.org/2022.acl-long.121>.
- Lugini, Luca and Diane Litman (2018). “Argument Component Classification for Classroom Discussions”. In: *Proceedings of the 5th Workshop on Argument Mining*. Brussels, Belgium: Association for Computational Linguistics, pp. 57–67. DOI: [10.18653/v1/W18-5208](https://doi.org/10.18653/v1/W18-5208). URL: <https://aclanthology.org/W18-5208>.
- Luhn, Hans Peter (1957). “A statistical approach to mechanized encoding and searching of literary information”. In: *IBM Journal of research and development* 1.4, pp. 309–317.
- Lyu, Chunchuan and Ivan Titov (2018). “AMR Parsing as Graph Prediction with Latent Alignment”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 397–407. DOI: [10.18653/v1/P18-1037](https://doi.org/10.18653/v1/P18-1037). URL: <https://www.aclweb.org/anthology/P18-1037>.
- Ma, Qingsong, Ondrej Bojar, and Yvette Graham (2018). “Results of the WMT18 Metrics Shared Task: Both characters and embeddings achieve good performance”. In: *Proceedings of the Third Conference on Machine Translation: Shared Task Papers, WMT 2018, Belgium, Brussels, October 31 - November 1, 2018*. Ed. by Ondrej Bojar et al., pp. 671–688. ISBN: 978-1-948087-81-0. URL: <https://www.aclweb.org/anthology/W18-6450/>.
- Ma, Qingsong, Johnny Wei, Ondřej Bojar, and Yvette Graham (2019a). “Results of the WMT19 Metrics Shared Task: Segment-Level and Strong MT Systems Pose Big Challenges”. In: *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*. Florence, Italy: Association for Computational Linguistics, pp. 62–90. DOI: [10.18653/v1/W19-5302](https://doi.org/10.18653/v1/W19-5302). URL: <https://www.aclweb.org/anthology/W19-5302>.
- Ma, Qingsong, Johnny Wei, Ondřej Bojar, and Yvette Graham (2019b). “Results of the WMT19 Metrics Shared Task: Segment-Level and Strong MT Systems Pose Big Challenges”. In: *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*. Florence, Italy: Association for Computational Linguistics, pp. 62–90. DOI: [10.18653/v1/W19-5302](https://doi.org/10.18653/v1/W19-5302). URL: <https://aclanthology.org/W19-5302>.
- Mager, Manuel, Ramón Fernandez Astudillo, Tahira Naseem, Md Arafat Sultan, Young-Suk Lee, Radu Florian, and Salim Roukos (2020a). “GPT-too: A Language-Model-First Approach for AMR-to-Text Generation”. In: *Proc. of ACL*. Online: Association

- for Computational Linguistics, pp. 1846–1852. DOI: [10.18653/v1/2020.acl-main.167](https://doi.org/10.18653/v1/2020.acl-main.167). URL: <https://www.aclweb.org/anthology/2020.acl-main.167>.
- Mager, Manuel, Ramón Fernandez Astudillo, Tahira Naseem, Md Arafat Sultan, Young-Suk Lee, Radu Florian, and Salim Roukos (2020b). “GPT-too: A Language-Model-First Approach for AMR-to-Text Generation”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 1846–1852. DOI: [10.18653/v1/2020.acl-main.167](https://doi.org/10.18653/v1/2020.acl-main.167). URL: <https://www.aclweb.org/anthology/2020.acl-main.167>.
- Mann, William C. (1983). “An Overview of the Penman Text Generation System”. In: *Proceedings of the Third AAAI Conference on Artificial Intelligence*. AAAI’83. Washington, D.C.: AAAI Press, pp. 261–265. URL: <http://dl.acm.org/citation.cfm?id=2886844.2886899>.
- Mann, William C and Sandra A Thompson (1988). “Rhetorical structure theory: Toward a functional theory of text organization”. In: *Text-interdisciplinary Journal for the Study of Discourse* 8.3, pp. 243–281.
- Manning, Emma and Nathan Schneider (2021). “Referenceless Parsing-Based Evaluation of AMR-to-English Generation”. In: *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*. Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 114–122. URL: <https://aclanthology.org/2021.eval4nlp-1.12>.
- Manning, Emma, Shira Wein, and Nathan Schneider (2020). “A Human Evaluation of AMR-to-English Generation Systems”. In: *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, pp. 4773–4786. DOI: [10.18653/v1/2020.coling-main.420](https://doi.org/10.18653/v1/2020.coling-main.420). URL: <https://aclanthology.org/2020.coling-main.420>.
- Marelli, Marco, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli (2014). “A SICK cure for the evaluation of compositional distributional semantic models”. In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*. Reykjavik, Iceland: European Languages Resources Association (ELRA), pp. 216–223. URL: http://www.lrec-conf.org/proceedings/lrec2014/pdf/363_Paper.pdf.

- Mathur, Nitika, Timothy Baldwin, and Trevor Cohn (2020a). “Tangled up in BLEU: Reevaluating the Evaluation of Automatic Machine Translation Evaluation Metrics”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 4984–4997. DOI: [10.18653/v1/2020.acl-main.448](https://doi.org/10.18653/v1/2020.acl-main.448). URL: <https://www.aclweb.org/anthology/2020.acl-main.448>.
- Mathur, Nitika, Johnny Wei, Markus Freitag, Qingsong Ma, and Ondřej Bojar (2020b). “Results of the WMT20 Metrics Shared Task”. In: *Proceedings of the Fifth Conference on Machine Translation*. Online: Association for Computational Linguistics, pp. 688–725. URL: <https://aclanthology.org/2020.wmt-1.77>.
- Maturana, Humberto R (1988). “Reality: The search for objectivity or the quest for a compelling argument”. In: *The Irish journal of psychology* 9.1, pp. 25–82.
- May, Jonathan and Jay Priyadarshi (2017). “SemEval-2017 Task 9: Abstract Meaning Representation Parsing and Generation”. In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Vancouver, Canada: Association for Computational Linguistics, pp. 536–545. DOI: [10.18653/v1/S17-2090](https://doi.org/10.18653/v1/S17-2090). URL: <https://www.aclweb.org/anthology/S17-2090>.
- Meer, Michiel van der, Myrthe Reuver, Urja Khurana, Lea Krause, and Selene Baez Santamaria (2022). “Will It Blend? Mixing Training Paradigms & Prompting for Argument Quality Prediction”. In: *Proceedings of the 9th Workshop on Argument Mining*. Online and in Gyeongju, Republic of Korea: International Conference on Computational Linguistics, pp. 95–103. URL: <https://aclanthology.org/2022.argmining-1.8>.
- Mendes, Pablo, Max Jakob, and Christian Bizer (2012). “DBpedia: A Multilingual Cross-domain Knowledge Base”. In: *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC’12)*. Istanbul, Turkey: European Language Resources Association (ELRA), pp. 1813–1817. URL: http://www.lrec-conf.org/proceedings/lrec2012/pdf/570_Paper.pdf.
- Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean (2013). “Efficient estimation of word representations in vector space”. In: *arXiv preprint arXiv:1301.3781*.
- Misra, Amita, Brian Ecker, and Marilyn Walker (2016). “Measuring the Similarity of Sentential Arguments in Dialogue”. In: *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Los Angeles: Association for Computational Linguistics, pp. 276–287. DOI: [10.18653/v1/W16-3636](https://doi.org/10.18653/v1/W16-3636). URL: <https://aclanthology.org/W16-3636>.

- Mkrtychian, Nadezhda, Evgeny Blagovechtchenski, Diana Kurmakaeva, Daria Gnedykh, Svetlana Kostromina, and Yury Shtyrov (2019). “Concrete vs. abstract semantics: from mental representations to functional brain mapping”. In: *Frontiers in human neuroscience* 13, p. 267.
- Moeller, Lucas, Dmitry Nikolaev, and Sebastian Padó (2023). “An Attribution Method for Siamese Encoders”. In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, pp. 15818–15827. DOI: [10.18653/v1/2023.emnlp-main.980](https://doi.org/10.18653/v1/2023.emnlp-main.980). URL: <https://aclanthology.org/2023.emnlp-main.980>.
- Montague, Richard (1973). “The proper treatment of quantification in ordinary English”. In: *Approaches to natural language*. Springer, pp. 221–242.
- Montella, Sebastien, Alexis Nasr, Johannes Heinecke, Frederic Bechet, and Lina M Rojas-Barahona (2023). “Investigating the Effect of Relative Positional Embeddings on AMR-to-Text Generation with Structural Adapters”. In: *arXiv preprint arXiv:2302.05900*.
- Müller, Almuth and Achim Kuwertz (2022). “Evaluation of a Semantic Search Approach based on AMR for Information Retrieval in Image Exploitation”. In: *2022 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, pp. 1–6.
- Naseem, Tahira, Abhishek Shah, Hui Wan, Radu Florian, Salim Roukos, and Miguel Ballesteros (2019). “Rewarding Smatch: Transition-Based AMR Parsing with Reinforcement Learning”. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 4586–4592. DOI: [10.18653/v1/P19-1451](https://doi.org/10.18653/v1/P19-1451). URL: <https://www.aclweb.org/anthology/P19-1451>.
- Nema, Preksha and Mitesh M. Khapra (2018). “Towards a Better Metric for Evaluating Question Generation Systems”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 3950–3959. DOI: [10.18653/v1/D18-1429](https://doi.org/10.18653/v1/D18-1429). URL: <https://www.aclweb.org/anthology/D18-1429>.
- Neuhaus, Michel, Kaspar Riesen, and Horst Bunke (2006). “Fast suboptimal algorithms for the computation of graph edit distance”. In: *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*. Springer, pp. 163–172.
- Niven, Timothy and Hung-Yu Kao (2019). “Probing Neural Network Comprehension of Natural Language Arguments”. In: *Proceedings of the 57th Annual Meeting of the*

- Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 4658–4664. DOI: [10.18653/v1/P19-1459](https://doi.org/10.18653/v1/P19-1459). URL: <https://aclanthology.org/P19-1459>.
- Nivre, Joakim et al. (2016). “Universal Dependencies v1: A Multilingual Treebank Collection”. In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*. Portorož, Slovenia: European Language Resources Association (ELRA), pp. 1659–1666. URL: <https://aclanthology.org/L16-1262>.
- Noord, Rik van and Johan Bos (2017a). “Dealing with Co-reference in Neural Semantic Parsing”. In: *Proceedings of the 2nd Workshop on Semantic Deep Learning (SemDeep-2)*. Montpellier, France: Association for Computational Linguistics, pp. 41–49. URL: <https://www.aclweb.org/anthology/W17-7306>.
- Noord, Rik van and Johan Bos (2017b). “Neural semantic parsing by character-based translation: Experiments with abstract meaning representations”. In: *arXiv preprint arXiv:1705.09980*.
- Novikova, Jekaterina, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser (2017). “Why We Need New Evaluation Metrics for NLG”. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 2241–2252. DOI: [10.18653/v1/D17-1238](https://doi.org/10.18653/v1/D17-1238). URL: <https://www.aclweb.org/anthology/D17-1238>.
- Opitz, Juri (2019). “Argumentative Relation Classification as Plausibility Ranking”. In: *Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019): Long Papers*. Erlangen, Germany: German Society for Computational Linguistics & Language Technology, pp. 193–202.
- Opitz, Juri (2020). “AMR Quality Rating with a Lightweight CNN”. In: *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*. Suzhou, China: Association for Computational Linguistics, pp. 235–247. URL: <https://aclanthology.org/2020.aacl-main.27>.
- Opitz, Juri (2023a). “Gzip versus bag-of-words for text classification with KNN”. In: *arXiv preprint arXiv:2307.15002*.
- Opitz, Juri (2023b). *How to hack an AMR Parsing evaluation, and what to do about it*. URL: <https://www.juriopitz.com/2023/10/04/hacking-a-metric.html> (visited on 12/27/2023).

- Opitz, Juri (2023c). “SMATCH++: Standardized and Extended Evaluation of Semantic Graphs”. In: *Findings of the Association for Computational Linguistics: EACL 2023*. Dubrovnik, Croatia: Association for Computational Linguistics, pp. 1595–1607. URL: <https://aclanthology.org/2023.findings-eacl.118>.
- Opitz, Juri and Sebastian Burst (2019). “Macro F1 and Macro F1”. In: *arXiv preprint arXiv:1911.03347*.
- Opitz, Juri, Angel Daza, and Anette Frank (2021a). “Weisfeiler-Leman in the Bamboo: Novel AMR Graph Metrics and a Benchmark for AMR Graph Similarity”. In: *Transactions of the Association for Computational Linguistics* 9, pp. 1425–1441. ISSN: 2307-387X. DOI: 10.1162/tacl_a_00435. eprint: https://direct.mit.edu/tacl/article-pdf/doi/10.1162/tacl_a_00435/1979290/tacl_a_00435.pdf. URL: https://doi.org/10.1162/tacl_a_00435.
- Opitz, Juri and Anette Frank (2019a). “An Argument-Marker Model for Syntax-Agnostic Proto-Role Labeling”. In: *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 224–234. DOI: 10.18653/v1/S19-1025. URL: <https://aclanthology.org/S19-1025>.
- Opitz, Juri and Anette Frank (2019b). “Automatic Accuracy Prediction for AMR Parsing”. In: *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 212–223. DOI: 10.18653/v1/S19-1024. URL: <https://aclanthology.org/S19-1024>.
- Opitz, Juri and Anette Frank (2019c). “Dissecting Content and Context in Argumentative Relation Analysis”. In: *Proceedings of the 6th Workshop on Argument Mining*. Florence, Italy: Association for Computational Linguistics, pp. 25–34. DOI: 10.18653/v1/W19-4503. URL: <https://aclanthology.org/W19-4503>.
- Opitz, Juri and Anette Frank (2021). “Towards a Decomposable Metric for Explainable Evaluation of Text Generation from AMR”. In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online: Association for Computational Linguistics, pp. 1504–1518. URL: <https://aclanthology.org/2021.eacl-main.129>.
- Opitz, Juri and Anette Frank (2022a). “Better Smatch = Better Parser? AMR evaluation is not so simple anymore”. In: *Proceedings of the 3rd Workshop on Evaluation*

- and Comparison of NLP Systems*. Online: Association for Computational Linguistics, pp. 32–43. DOI: [10.18653/v1/2022.eval4nlp-1.4](https://doi.org/10.18653/v1/2022.eval4nlp-1.4). URL: <https://aclanthology.org/2022.eval4nlp-1.4>.
- Opitz, Juri and Anette Frank (2022b). “SBERT studies Meaning Representations: Decomposing Sentence Embeddings into Explainable Semantic Features”. In: *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*. Online only: Association for Computational Linguistics, pp. 625–638. URL: <https://aclanthology.org/2022.aacl-main.48>.
- Opitz, Juri, Philipp Heinisch, Philipp Wiesenbach, Philipp Cimiano, and Anette Frank (2021b). “Explainable Unsupervised Argument Similarity Rating with Abstract Meaning Representation and Conclusion Generation”. In: *Proceedings of the 8th Workshop on Argument Mining*. Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 24–35. DOI: [10.18653/v1/2021.argmining-1.3](https://doi.org/10.18653/v1/2021.argmining-1.3). URL: <https://aclanthology.org/2021.argmining-1.3>.
- Opitz, Juri, Philipp Meier, and Anette Frank (2023a). “SMARAGD: Learning SMatch for Accurate and Rapid Approximate Graph Distance”. In: *Proceedings of the 15th International Conference on Computational Semantics*. Ed. by Maxime Amblard and Ellen Breitholtz. Nancy, France: Association for Computational Linguistics, pp. 267–274. URL: <https://aclanthology.org/2023.iwcs-1.28>.
- Opitz, Juri, Letitia Parcalabescu, and Anette Frank (2020). “AMR Similarity Metrics from Principles”. In: *Transactions of the Association for Computational Linguistics* 8, pp. 522–538. DOI: [10.1162/tacl_a_00329](https://doi.org/10.1162/tacl_a_00329). URL: <https://aclanthology.org/2020.tacl-1.34>.
- Opitz, Juri, Shira Wein, Julius Steen, Anette Frank, and Nathan Schneider (2023b). “AMR4NLI: Interpretable and robust NLI measures from semantic graphs”. In: *Proceedings of the 15th International Conference on Computational Semantics*. Ed. by Maxime Amblard and Ellen Breitholtz. Nancy, France: Association for Computational Linguistics, pp. 275–283. URL: <https://aclanthology.org/2023.iwcs-1.29>.
- Padó, Sebastian and Mirella Lapata (2007). “Dependency-Based Construction of Semantic Space Models”. In: *Computational Linguistics* 33.2, pp. 161–199. ISSN: 0891-2017. DOI: [10.1162/coli.2007.33.2.161](https://doi.org/10.1162/coli.2007.33.2.161). eprint: <https://direct.mit.edu/coli/article-pdf/33/2/161/1798388/coli.2007.33.2.161.pdf>. URL: <https://doi.org/10.1162/coli.2007.33.2.161>.

- Palmer, Martha, Daniel Gildea, and Paul Kingsbury (2005). “The Proposition Bank: An Annotated Corpus of Semantic Roles”. In: *Computational Linguistics* 31.1, pp. 71–106. DOI: [10.1162/0891201053630264](https://doi.org/10.1162/0891201053630264). URL: <https://www.aclweb.org/anthology/J05-1004>.
- Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu (2002). “Bleu: a Method for Automatic Evaluation of Machine Translation”. In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, pp. 311–318. DOI: [10.3115/1073083.1073135](https://doi.org/10.3115/1073083.1073135). URL: <https://www.aclweb.org/anthology/P02-1040>.
- Paul, Debjit, Juri Opitz, Maria Becker, Jonathan Kobbe, Graeme Hirst, and Anette Frank (2020). “Argumentative relation classification with background knowledge”. In: *Computational Models of Argument*. IOS Press, pp. 319–330.
- Pavlova, Siyana, Maxime Amblard, and Bruno Guillaume (2023). “Structural and Global Features for Comparing Semantic Representation Formalisms”. In: *Proceedings of the Fourth International Workshop on Designing Meaning Representations*. Ed. by Julia Bonn and Nianwen Xue. Nancy, France: Association for Computational Linguistics, pp. 1–12. URL: <https://aclanthology.org/2023.dmr-1.1>.
- Pedersen, Ted (2007). “Unsupervised corpus-based methods for WSD”. In: *Word sense disambiguation*, pp. 133–166.
- Pelletier, Francis Jeffry (1994). “The principle of semantic compositionality”. In: *Topoi* 13.1, pp. 11–24.
- Pennington, Jeffrey, Richard Socher, and Christopher Manning (2014a). “Glove: Global Vectors for Word Representation”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pp. 1532–1543. DOI: [10.3115/v1/D14-1162](https://doi.org/10.3115/v1/D14-1162). URL: <https://www.aclweb.org/anthology/D14-1162>.
- Pennington, Jeffrey, Richard Socher, and Christopher D. Manning (2014b). “GloVe: Global Vectors for Word Representation”. In: *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543. URL: <http://www.aclweb.org/anthology/D14-1162>.
- Peters, Matthew E., Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer (2018). “Deep Contextualized Word Representations”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*

- (*Long Papers*). New Orleans, Louisiana: Association for Computational Linguistics, pp. 2227–2237. DOI: [10.18653/v1/N18-1202](https://doi.org/10.18653/v1/N18-1202). URL: <https://aclanthology.org/N18-1202>.
- Plenz, Moritz, Juri Opitz, Philipp Heinisch, Philipp Cimiano, and Anette Frank (2023). “Similarity-weighted Construction of Contextualized Commonsense Knowledge Graphs for Knowledge-intense Argumentation Tasks”. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Toronto, Canada: Association for Computational Linguistics, pp. 6130–6158. URL: <https://aclanthology.org/2023.acl-long.338>.
- Poliak, Adam, Yonatan Belinkov, James Glass, and Benjamin Van Durme (2018). “On the Evaluation of Semantic Phenomena in Neural Machine Translation Using Natural Language Inference”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 513–523. DOI: [10.18653/v1/N18-2082](https://doi.org/10.18653/v1/N18-2082). URL: <https://www.aclweb.org/anthology/N18-2082>.
- Pourdamghani, Nima, Kevin Knight, and Ulf Hermjakob (2016). “Generating English from Abstract Meaning Representations”. In: *Proceedings of the 9th International Natural Language Generation conference*. Edinburgh, UK: Association for Computational Linguistics, pp. 21–25. DOI: [10.18653/v1/W16-6603](https://doi.org/10.18653/v1/W16-6603). URL: <https://www.aclweb.org/anthology/W16-6603>.
- Puccetti, Giovanni, Alessio Miaschi, and Felice Dell’Orletta (2021). “How Do BERT Embeddings Organize Linguistic Knowledge?” In: *Proceedings of Deep Learning Inside Out (DeeLIO): The 2nd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*. Online: Association for Computational Linguistics, pp. 48–57. DOI: [10.18653/v1/2021.deelio-1.6](https://doi.org/10.18653/v1/2021.deelio-1.6). URL: <https://aclanthology.org/2021.deelio-1.6>.
- Pustejovsky, James, Ken Lai, and Nianwen Xue (2019). “Modeling quantification and scope in Abstract Meaning Representations”. In: *Proceedings of the first international workshop on designing meaning representations*, pp. 28–33.
- Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever (2019). “Language models are unsupervised multitask learners”. In: *OpenAI blog* 1.8, p. 9.
- Rae, Jack W., Anna Potapenko, Siddhant M. Jayakumar, Chloe Hillier, and Timothy P. Lillicrap (2020). “Compressive Transformers for Long-Range Sequence Modelling”. In:

- International Conference on Learning Representations*. URL: <https://openreview.net/forum?id=SylKikSYDH>.
- Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu (2020). “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer”. In: *Journal of Machine Learning Research* 21.140, pp. 1–67. URL: <http://jmlr.org/papers/v21/20-074.html>.
- Rago, A, O Cocarascu, C Bechlivanidis, D Lagnado, and F Toni (2021). “Argumentative explanations for interactive recommendations”. In: *Artificial Intelligence* 296, pp. 1–22. DOI: [10.1016/j.artint.2021.103506](https://doi.org/10.1016/j.artint.2021.103506). URL: <http://dx.doi.org/10.1016/j.artint.2021.103506>.
- Ravi, Sujith, Kevin Knight, and Radu Soricut (2008). “Automatic prediction of parser accuracy”. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pp. 887–896.
- Reimers, Nils and Iryna Gurevych (2019). “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 3982–3992. DOI: [10.18653/v1/D19-1410](https://doi.org/10.18653/v1/D19-1410). URL: <https://aclanthology.org/D19-1410>.
- Reimers, Nils, Benjamin Schiller, Tilman Beck, Johannes Daxenberger, Christian Stab, and Iryna Gurevych (2019). “Classification and Clustering of Arguments with Contextualized Word Embeddings”. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 567–578. DOI: [10.18653/v1/P19-1054](https://doi.org/10.18653/v1/P19-1054). URL: <https://aclanthology.org/P19-1054>.
- Reisinger, Drew, Rachel Rudinger, Francis Ferraro, Craig Harman, Kyle Rawlins, and Benjamin Van Durme (2015). “Semantic Proto-Roles”. In: *Transactions of the Association for Computational Linguistics* 3, pp. 475–488. DOI: [10.1162/tacl_a_00152](https://doi.org/10.1162/tacl_a_00152). URL: <https://www.aclweb.org/anthology/Q15-1034>.
- Ribeiro, Leonardo F. R., Claire Gardent, and Iryna Gurevych (2019). “Enhancing AMR-to-Text Generation with Dual Graph Representations”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 3183–3194. DOI:

- 10.18653/v1/D19-1314. URL: <https://aclanthology.org/D19-1314>.
- Ribeiro, Leonardo F. R., Yue Zhang, and Iryna Gurevych (2021). “Structural Adapters in Pretrained Language Models for AMR-to-Text Generation”. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Ed. by Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 4269–4282. DOI: 10.18653/v1/2021.emnlp-main.351. URL: <https://aclanthology.org/2021.emnlp-main.351>.
- Ribeiro, Marco Tulio, Tongshuang Wu, Carlos Guestrin, and Sameer Singh (2020). “Beyond Accuracy: Behavioral Testing of NLP Models with CheckList”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 4902–4912. DOI: 10.18653/v1/2020.acl-main.442. URL: <https://aclanthology.org/2020.acl-main.442>.
- Rijsbergen, C. J. van (1979). *Information Retrieval*. Butterworth. ISBN: 0-408-70929-4.
- Rissland, Edwina L, David B Skalak, and M Timur Friedman (1993). “BankXX: a program to generate argument through case-base search”. In: *Proceedings of the 4th international conference on Artificial intelligence and law*, pp. 117–124.
- Saadat-Yazdi, Ameer, Xue Li, Sandrine Chaussou, Vaishak Belle, Björn Ross, Jeff Z. Pan, and Nadin Kökciyan (2022). “KEViN: A Knowledge Enhanced Validity and Novelty Classifier for Arguments”. In: *Proceedings of the 9th Workshop on Argument Mining*. Online and in Gyeongju, Republic of Korea: International Conference on Computational Linguistics, pp. 104–110. URL: <https://aclanthology.org/2022.argmining-1.9>.
- Sai, Ananya B, Akash Kumar Mohankumar, and Mitesh M Khapra (2020). “A Survey of Evaluation Metrics Used for NLG Systems”. In: *arXiv preprint arXiv:2008.12009*.
- Sanh, Victor, Lysandre Debut, Julien Chaumond, and Thomas Wolf (2019). “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter”. In: *arXiv preprint arXiv:1910.01108*.
- Sanyal, Soumya and Xiang Ren (2021). “Discretized Integrated Gradients for Explaining Language Models”. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 10285–10299. DOI: 10.18653/v1/2021.

- emnlp-main.805. URL: <https://aclanthology.org/2021.emnlp-main.805>.
- Sato, Misa, Kohsuke Yanai, Toshinori Miyoshi, Toshihiko Yanase, Makoto Iwayama, Qinghua Sun, and Yoshiki Niwa (2015). “End-to-end argument generation system in debating”. In: *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, pp. 109–114.
- Schenker, Adam, Horst Bunke, Mark Last, and Abraham Kandel (2005). *Graph-Theoretic Techniques for Web Content Mining*. USA: World Scientific Publishing Co., Inc. ISBN: 9789812563392.
- Schiller, Benjamin, Johannes Daxenberger, and Iryna Gurevych (2020). “Aspect-controlled neural argument generation”. In: *arXiv preprint arXiv:2005.00084*.
- Sellam, Thibault, Dipanjan Das, and Ankur Parikh (2020). “BLEURT: Learning Robust Metrics for Text Generation”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 7881–7892. DOI: [10.18653/v1/2020.acl-main.704](https://doi.org/10.18653/v1/2020.acl-main.704). URL: <https://www.aclweb.org/anthology/2020.acl-main.704>.
- Selvaraju, Ramprasaath R, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra (2017). “Grad-cam: Visual explanations from deep networks via gradient-based localization”. In: *Proceedings of the IEEE international conference on computer vision*, pp. 618–626.
- Shapley, Lloyd Stowell (1951). *Notes on the n-Person Game—II: The Value of an n-Person Game*.
- Shervashidze, Nino, Pascal Schweitzer, Erik Jan Van Leeuwen, Kurt Mehlhorn, and Karsten M Borgwardt (2011). “Weisfeiler-lehman graph kernels.” In: *Journal of Machine Learning Research* 12.9.
- Sheth, Janaki, Young-Suk Lee, Ramon Fernandez Astudillo, Tahira Naseem, Radu Florian, Salim Roukos, and Todd Ward (2021). “Bootstrapping Multilingual AMR with Contextual Word Alignments”. In: *arXiv preprint arXiv:2102.02189*.
- Shimorina, Anastaisa, Claire Gardent, Shashi Narayan, and Laura Perez-Beltrachini (2017). *The WebNLG challenge: report on human evaluation*. Tech. rep. Université de Lorraine, Nancy (France).
- Shou, Ziyi and Fangzhen Lin (2023). “Evaluate AMR Graph Similarity via Self-supervised Learning”. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Toronto, Canada: Association for

- Computational Linguistics, pp. 16112–16123. URL: <https://aclanthology.org/2023.acl-long.892>.
- Singh, Keshav, Farjana Sultana Mim, Naoya Inoue, Shoichi Naito, and Kentaro Inui (2021). “Exploring Methodologies for Collecting High-Quality Implicit Reasoning in Arguments”. In: *Proceedings of the 8th Workshop on Argument Mining*, pp. 57–66.
- Slonim, Noam, Yonatan Bilu, Carlos Alzate, Roy Bar-Haim, Ben Bogin, Francesca Bonin, Leshem Choshen, Edo Cohen-Karlik, Lena Dankin, Lilach Edelstein, et al. (2021). “An autonomous debating system”. In: *Nature* 591.7850, pp. 379–384.
- Song, Linfeng (2019). “Tackling Graphical NLP problems with Graph Recurrent Networks”. In: *CoRR* abs/1907.06142. arXiv: 1907.06142. URL: <http://arxiv.org/abs/1907.06142>.
- Song, Linfeng and Daniel Gildea (2019). “SemBleu: A Robust Metric for AMR Parsing Evaluation”. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 4547–4552. DOI: 10.18653/v1/P19-1446. URL: <https://www.aclweb.org/anthology/P19-1446>.
- Song, Linfeng, Yue Zhang, Zhiguo Wang, and Daniel Gildea (2018). “A Graph-to-Sequence Model for AMR-to-Text Generation”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 1616–1626. DOI: 10.18653/v1/P18-1150. URL: <https://www.aclweb.org/anthology/P18-1150>.
- Soricut, Radu and Sushant Narsale (2012). “Combining quality prediction and system selection for improved automatic translation output”. In: *Proceedings of the Seventh Workshop on Statistical Machine Translation*. Association for Computational Linguistics, pp. 163–170.
- Spall, James C (1987). “A stochastic approximation technique for generating maximum likelihood parameter estimates”. In: *1987 American control conference*. IEEE, pp. 1161–1167.
- Spall, James C (1998). “An overview of the simultaneous perturbation method for efficient optimization”. In: *Johns Hopkins apl technical digest* 19.4, pp. 482–492.

- Spaulding, Elizabeth, Gary Kazantsev, and Mark Dredze (2023). “Joint End-to-end Semantic Proto-role Labeling”. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Toronto, Canada: Association for Computational Linguistics, pp. 723–736. URL: <https://aclanthology.org/2023.acl-short.63>.
- Specia, Lucia, Kashif Shah, Jose G. C. De Souza, Trevor Cohn, and Fondazione Bruno Kessler (2013). “QuEst - A Translation Quality Estimation Framework”. In: *In Proceedings of the 51th Conference of the Association for Computational Linguistics (ACL), Demo Session*.
- Speer, Robyn, Joshua Chin, and Catherine Havasi (2017). “Conceptnet 5.5: An open multilingual graph of general knowledge”. In: *Thirty-first AAAI conference on artificial intelligence*.
- Stab, Christian and Iryna Gurevych (2017). “Parsing Argumentation Structures in Persuasive Essays”. In: *Computational Linguistics* 43.3, pp. 619–659. DOI: [10.1162/COLI_a_00295](https://doi.org/10.1162/COLI_a_00295). URL: <https://aclanthology.org/J17-3005>.
- Steen, Julius, Juri Opitz, Anette Frank, and Katja Markert (2023). “With a Little Push, NLI Models can Robustly and Efficiently Predict Faithfulness”. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, pp. 914–924. DOI: [10.18653/v1/2023.acl-short.79](https://doi.org/10.18653/v1/2023.acl-short.79). URL: <https://aclanthology.org/2023.acl-short.79>.
- Stengel-Eskin, Elias, Aaron Steven White, Sheng Zhang, and Benjamin Van Durme (2020). “Universal Decompositional Semantic Parsing”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 8427–8439. DOI: [10.18653/v1/2020.acl-main.746](https://doi.org/10.18653/v1/2020.acl-main.746). URL: <https://aclanthology.org/2020.acl-main.746>.
- Strubell, Emma, Ananya Ganesh, and Andrew McCallum (2019). “Energy and Policy Considerations for Deep Learning in NLP”. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 3645–3650. DOI: [10.18653/v1/P19-1355](https://doi.org/10.18653/v1/P19-1355). URL: <https://aclanthology.org/P19-1355>.
- Teichert, Adam, Adam Poliak, Benjamin Van Durme, and Matthew R Gormley (2017). “Semantic proto-role labeling”. In: *Thirty-First AAAI Conference on Artificial Intelligence*.

- Togninalli, Matteo, Elisabetta Ghisu, Felipe Llinares-López, Bastian Rieck, and Karsten Borgwardt (2019). “Wasserstein Weisfeiler-Lehman Graph Kernels”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc., pp. 6436–6446. URL: <https://proceedings.neurips.cc/paper/2019/file/73fed7fd472e502d8908794430511f4d-Paper.pdf>.
- Toulmin, Stephen E (2003). *The uses of argument*. Cambridge university press.
- Uhrig, Sarah, Yoalli Garcia, Juri Opitz, and Anette Frank (2021). “Translate, then Parse! A Strong Baseline for Cross-Lingual AMR Parsing”. In: *Proceedings of the 17th International Conference on Parsing Technologies and the IWPT 2021 Shared Task on Parsing into Enhanced Universal Dependencies (IWPT 2021)*. Online: Association for Computational Linguistics, pp. 58–64. DOI: 10.18653/v1/2021.iwpt-1.6. URL: <https://aclanthology.org/2021.iwpt-1.6>.
- Van Gysel, Jens EL, Meagan Vigus, Jayeol Chun, Kenneth Lai, Sarah Moeller, Jiarui Yao, Tim O’Gorman, Andrew Cowell, William Croft, Chu-Ren Huang, et al. (2021). “Designing a uniform meaning representation for natural language processing”. In: *KI-Künstliche Intelligenz* 35.3, pp. 343–360.
- Vanroy, Bram (2023). “Smatch is non-deterministic and does not yield score=1 for the same input/output graph”. In: *GitHub issues*. Note: The issue refers to Smatch with hill-climbing solver. URL: <https://github.com/snowblink14/smatch/issues/43>.
- Vassiliades, Alexandros, Nick Bassiliades, and Theodore Patkos (2021). “Argumentation and explainable artificial intelligence: a survey”. In: *The Knowledge Engineering Review* 36.
- Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin (2017). “Attention is all you need”. In: *Advances in neural information processing systems* 30.
- Vasylenko, Pavlo, Pere Lluís Huguet Cabot, Abelardo Carlos Martínez Lorenzo, and Roberto Navigli (2023). “Incorporating Graph Information in Transformer-based AMR Parsing”. In: *Findings of the Association for Computational Linguistics: ACL 2023*. Toronto, Canada: Association for Computational Linguistics, pp. 1995–2011. URL: <https://aclanthology.org/2023.findings-acl.125>.
- Vinyals, Oriol, Łukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton (2015). “Grammar as a Foreign Language”. In: *Advances in Neural Information Processing Systems* 28. Ed. by C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama,

- and R. Garnett. Curran Associates, Inc., pp. 2773–2781. URL: <http://papers.nips.cc/paper/5635-grammar-as-a-foreign-language.pdf>.
- Wachsmuth, Henning, Martin Potthast, Khalid Al Khatib, Yamen Ajour, Jana Puschmann, Jiani Qu, Jonas Dorsch, Viorel Morari, Janek Bevendorff, and Benno Stein (2017). “Building an argument search engine for the web”. In: *Proceedings of the 4th Workshop on Argument Mining*, pp. 49–59.
- Wachsmuth, Henning, Shahbaz Syed, and Benno Stein (2018). “Retrieval of the Best Counterargument without Prior Topic Knowledge”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 241–251. DOI: [10.18653/v1/P18-1023](https://doi.org/10.18653/v1/P18-1023). URL: <https://aclanthology.org/P18-1023>.
- Wagemans, Jean HM (2011). “The assessment of argumentation from expert opinion”. In: *Argumentation* 25.3, pp. 329–339.
- Walton, Douglas (2005). “Justification of argumentation schemes”. In: *The Australasian Journal of Logic* 3.
- Walton, Douglas, Christopher Reed, and Fabrizio Macagno (2008). *Argumentation schemes*. Cambridge University Press.
- Wang, Chaojun and Rico Sennrich (2020). “On Exposure Bias, Hallucination and Domain Shift in Neural Machine Translation”. In: *arXiv preprint arXiv:2005.03642*.
- Wang, Chen (2020). “An overview of SPSA: recent development and applications”. In: *arXiv preprint arXiv:2012.06952*.
- Wang, Chuan, Sameer Pradhan, Xiaoman Pan, Heng Ji, and Nianwen Xue (2016). “CAMR at SemEval-2016 Task 8: An Extended Transition-based AMR Parser”. In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. San Diego, California, pp. 1173–1178. DOI: [10.18653/v1/S16-1181](https://doi.org/10.18653/v1/S16-1181). URL: <https://www.aclweb.org/anthology/S16-1181>.
- Wang, Chuan, Nianwen Xue, and Sameer Pradhan (2015). “A Transition-based Algorithm for AMR Parsing”. In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Denver, Colorado: Association for Computational Linguistics, pp. 366–375. DOI: [10.3115/v1/N15-1040](https://doi.org/10.3115/v1/N15-1040). URL: <https://aclanthology.org/N15-1040>.

- Wang, Junlin, Jens Tuyls, Eric Wallace, and Sameer Singh (2020a). “Gradient-based Analysis of NLP Models is Manipulable”. In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, pp. 247–258. DOI: [10.18653/v1/2020.findings-emnlp.24](https://doi.org/10.18653/v1/2020.findings-emnlp.24). URL: <https://aclanthology.org/2020.findings-emnlp.24>.
- Wang, Tianming, Xiaojun Wan, and Hanqi Jin (2020b). “AMR-To-Text Generation with Graph Transformer”. In: *Transactions of the Association for Computational Linguistics* 8.0, pp. 19–33. ISSN: 2307-387X. URL: <https://transacl.org/ojs/index.php/tacl/article/view/1805>.
- Wang, Tianming, Xiaojun Wan, and Shaowei Yao (2020c). “Better AMR-To-Text Generation with Graph Structure Reconstruction”. In: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*. Ed. by Christian Bessiere. Main track. International Joint Conferences on Artificial Intelligence Organization, pp. 3919–3925. DOI: [10.24963/ijcai.2020/542](https://doi.org/10.24963/ijcai.2020/542). URL: <https://doi.org/10.24963/ijcai.2020/542>.
- Wang, Xian Chuan, Xian Chao Wang, Shi Bing Wang, Xiu Ming Chen, and Zong Tian Liu (2020d). “Neo-Davidsonian-Based Event Class Semantic Representation”. In: *Procedia Computer Science* 166, pp. 120–124.
- Warstadt, Alex et al. (2019). “Investigating BERT’s Knowledge of Language: Five Analysis Methods with NPIs”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 2877–2887. DOI: [10.18653/v1/D19-1286](https://doi.org/10.18653/v1/D19-1286). URL: <https://aclanthology.org/D19-1286>.
- Wein, Shira and Nathan Schneider (2022). “Accounting for Language Effect in the Evaluation of Cross-lingual AMR Parsers”. In: *Proceedings of the 29th International Conference on Computational Linguistics*. Gyeongju, Republic of Korea: International Committee on Computational Linguistics, pp. 3824–3834.
- Wein, Shira and Nathan Schneider (2023). “Assessing the Cross-linguistic Utility of Abstract Meaning Representation”. In: *Computational Linguistics*, pp. 1–55. ISSN: 0891-2017. DOI: [10.1162/coli_a_00503](https://doi.org/10.1162/coli_a_00503). eprint: https://direct.mit.edu/coli/article-pdf/doi/10.1162/coli_a_00503/2198446/coli_a_00503.pdf. URL: https://doi.org/10.1162/coli_a_00503.

- Weisfeiler, Boris and Andrei Leman (1968). “The reduction of a graph to canonical form and the algebra which appears therein”. In: *NTI, Series 2.9*, pp. 12–16.
- Whitney, Hassler (1932). “Congruent graphs and connectivity of graphs”. In: *Amer. J. Math.* 58, pp. 150–168.
- Wiegrefe, Sarah and Yuval Pinter (2019). “Attention is not not Explanation”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 11–20. DOI: [10.18653/v1/D19-1002](https://doi.org/10.18653/v1/D19-1002). URL: <https://aclanthology.org/D19-1002>.
- Xu, Dongqin, Junhui Li, Muhua Zhu, Min Zhang, and Guodong Zhou (2020). “Improving AMR Parsing with Sequence-to-Sequence Pre-training”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pp. 2501–2511. DOI: [10.18653/v1/2020.emnlp-main.196](https://doi.org/10.18653/v1/2020.emnlp-main.196). URL: <https://www.aclweb.org/anthology/2020.emnlp-main.196>.
- Yanardag, Pinar and SVN Vishwanathan (2015). “Deep graph kernels”. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1365–1374.
- Yuan, Jian, Zhongyu Wei, Donghua Zhao, Qi Zhang, and Changjian Jiang (2021). “Leveraging Argumentation Knowledge Graph for Interactive Argument Pair Identification”. In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Online: Association for Computational Linguistics, pp. 2310–2319. DOI: [10.18653/v1/2021.findings-acl.203](https://doi.org/10.18653/v1/2021.findings-acl.203). URL: <https://aclanthology.org/2021.findings-acl.203>.
- Zeidler, Laura, Juri Opitz, and Anette Frank (2022). “A Dynamic, Interpreted CheckList for Meaning-oriented NLG Metric Evaluation – through the Lens of Semantic Similarity Rating”. In: *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*. Seattle, Washington: Association for Computational Linguistics, pp. 157–172. DOI: [10.18653/v1/2022.starsem-1.14](https://doi.org/10.18653/v1/2022.starsem-1.14). URL: <https://aclanthology.org/2022.starsem-1.14>.
- Zenker, Frank (2013). “Bayesian argumentation: The practical side of probability”. In: *Bayesian Argumentation*. Springer, pp. 1–11.
- Zhang, Tianyi, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi (2020). “BERTScore: Evaluating Text Generation with BERT”. In: *International Conference*

on Learning Representations. URL: <https://openreview.net/forum?id=SkeHuCVFDr>.

Zhou, Wangchunshu and Ke Xu (2020). “Learning to Compare for Better Training and Evaluation of Open Domain Natural Language Generation Models”. In: *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, pp. 9717–9724. URL: <https://aaai.org/ojs/index.php/AAAI/article/view/6521>.

Zhu, Wanzheng and Suma Bhat (2020). “GRUEN for Evaluating Linguistic Quality of Generated Text”. In: *arXiv preprint arXiv:2010.02498*.

Zhu, Yaoming, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu (2018). “Texygen: A benchmarking platform for text generation models”. In: *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 1097–1100.