

# Prompt Design and Answer Processing for Knowledge Base Construction from Pre-trained Language Models (KBC-LM)

Xiao Fang<sup>1</sup>, Alex Kalinowski<sup>1</sup>, Haoran Zhao<sup>1</sup>, Ziao You<sup>1</sup>, Yuhao Zhang<sup>1</sup> and Yuan An<sup>1</sup>

<sup>1</sup> College of Computing and Informatics, Drexel University, Philadelphia, PA 19104, USA

## Abstract

Prompt-Design on Pre-trained large Language Models (prompt-design&PLM) has become an emerging paradigm for a range of NLP tasks. Although an increased effort has been put into reformulating many classic NLP problems as prompt-based learning, less explored areas include knowledge base construction from PLMs. The ISWC-2022 challenge on Knowledge Base Construction from Pre-trained Language Models (KBC-LM) provides 12 pre-defined relations each of which is equipped with a number of train and dev triples. In participating in the challenge, we manually developed relation-specific prompt templates to probe BERT-related LMs. Given a (SubjectEntity, relation) pair, we predicted none, one, or many ObjectEntities to complete the pair as a triple. The test results on unseen (SubjectEntity, relation) pairs showed our prompt design achieved 49% overall macro average F1-score, a 48% improvement from the baseline's 31% F1-score. The insights we learned about the "knowledge" of a language model would lead us to select appropriate LMs for future knowledge base construction tasks.

## 1. Introduction

Pre-trained large Language Models (PLM) such as BERT [1], RoBERTa [2], GPT-3 [3], and T5 [4] have attracted a significant attention in AI and NLP communities. A recent emerging paradigm leveraging the Pre-trained LMs (PLM) is to use textual prompts to solve a range of NLP tasks. For example, for Sentiment Analysis, if we have a piece of text "This is a boring movie.", we can use a textual prompt "This is a boring movie. The review is \_\_" to ask a pre-trained language model (PLM) to fill up the blank with a positive or negative label. The downstream applications using this paradigm are reformulated in a way as if we need to predict a missing or next word using a pre-trained LM. We dub this paradigm as *prompt-design on pre-trained large language models* or *prompt-design&PLM* for short. To effectively apply *prompt-design&PLM*, one needs to address several critical issues including selecting a relevant LM, designing appropriate prompts, and extracting the final predictions. Typically, one develops *templates* to generate prompts such that a template processes the original text with some extra tokens. For example, the template "[TEXT] The review is [MASK]" generates the prompt we used earlier for Sentiment Analysis, where "[TEXT]" corresponds to the original sentence,


---

LM-KBC'22: Knowledge Base Construction from Pre-trained Language Models, Challenge at ISWC 2022

✉ xfang@drexel.edu (X. Fang); ajk437@drexel.edu (A. Kalinowski); hz454@drexel.edu (H. Zhao); zy364@drexel.edu (Z. You); hz446@drexel.edu (Y. Zhang); ya45@drexel.edu (Y. An)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

and the token “[MASK]” stands for a blank to be filled up. Optionally, one may further develop a *verbalizer* to project original labels to words in the vocabulary of the LM for final prediction. For example, the verbalizer for our Sentiment Analysis example is {“positive”:“interesting”, “negative”:“boring”}.

A flurry of studies have been reported using *prompt-design&PLM* to solve text classification [3, 5, 6, 7], named-entity recognition [8], natural language inference [5, 6, 7], sentiment analysis [9], relation extraction [10], text summarization [11], and parsing [12]. Despite the efforts, an under-explored area is to directly extract structured knowledge from PLMs to construct a knowledge base. The ISWC-2022 Challenge for Knowledge Base Construction from Pre-trained Language Models (KBC-LM) aims to explore the capability of various pre-trained language models (PLMs) for constructing a knowledge base with a set of given predicates/relationships. The problem is formally defined as follows:

**Definition 1.** *Given a set of inputs each of which contains a SubjectEntity(s) and a relation(r). Predict the set of correct ObjectEntitys  $\{o_1, o_2, \dots, o_k\}$  using a LM probing method for each input.*

A significant difference between the ISWC-2022 KBC-LM challenge and the existing baseline, e.g., LAMA presented in [13], is that there is no constraint on the number of ObjectEntitys that can participate in a (SubjectEntity, relation) pair. Specifically, a SubjectEntity can join zero, one, or many ObjectEntitys in a relation. There are two tracks in this challenge. Track 1 explores the pre-trained BERT-related LMs [1] such as BERT Base Cased Model (BERT-base) and BERT Large Cased Model (BERT-large). Track 2 explores other LMs including RoBERTa, Transformer-XL, GPT-2, BART, etc. The outputs are evaluated using the established F1-score KB metric. We participated in the Track 1 challenge using BERT-related LMs. This paper reports our prompt design, answering processing steps, and test results on the unseen test data hold back by the challenge organizers.

The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 presents the relation-specific prompts for LM probing. Section 4 presents test results. Section 5 discusses the prompt design and lessons learned. Section 6 describes the structure of the implementation. Finally, Section 7 concludes the paper.

## 2. Related Work

**Pre-trained Language Model (PLM).** Standard language models are trained to predict text in an autoregressive fashion, that is, predicting the tokens in the sequence one at a time. This is usually done from left to right, but can be done in other orders as well. Representative examples of modern pre-trained left-to-right autoregressive LMs include GPT-3 [3]. A disadvantage of autoregressive language models is its directionality in processing text. To predict text based on surrounding text, masked language models (MLM) have been developed that use bidirectional objective function. Representative pre-trained models using MLMs include BERT [1], ERNIE [14] and many variants. The prefix LM is a left-to-right LM that decodes a target text  $y$  conditioned on a prefixed sequence  $x$  as for translation. Example prefix LMs include UniLM 1-2 [15, 16] and ERNIE-M [17]. The encoder-decoder model uses a left-to-right LM to decode a target text  $y$  conditioned on a separate encoder for text  $x$  with a fully-connected mask. Example encoder-decoder pre-trained models include T5 [4], BART [18], MASS [19] and their variants.

**Prompt Design.** In general, there are two types of prompts, cloze prompts which fill in blanks in a textual string and prefix prompts which continue a string prefix. Prompts can be designed manually based on human intuition [13, 3, 5] or automatically through mining [20], paraphrasing [21, 22], gradient-based search [23], generation [24], and scoring [25]. In addition to discrete hard prompts, researchers have also developed continuous soft prompts that interact directly with LMs in the embedding space. Soft prompts have their own parameters that can be tuned through different strategies including prefix tuning [26], hard-prompt initialized tuning [27], and hybrid tuning [28].

**Answer Processing.** An answer can take (1) one of the tokens in the PLM’s vocabulary, (2) a short multi-token span, or (3) a sentence or document. Answer processing aims to extract the correct answers from the output space of a PLM. Researchers have developed manual approaches using verbalizers [13, 29, 30, 8] and automatic methods through paraphrasing [20], pruning [31], and label decomposition [32].

**Few-shot Learning.** In addition to zero-shot prompt-design&PLM setting, there are also methods that use training data to optimize parameters in prompts or PLMs. Few-shot learning methods have been developed for tuning prompts only [33, 34] and tuning both prompts and PLMs [24, 28, 10].

**Knowledge Base Construction from LM Probing.** The seminal work on KBC-LM is LAMA [13] which manually created cloze templates to probe knowledge in PLMs. Few-shot learning on the original LAMA datasets has also been evaluated [35]. More studies have been reported on probing PLMs for complicated knowledge [36], temporal knowledge [37], and domain specific knowledge [38, 39].

### 3. Relation-Specific Prompt Design and Answer Processing

For participating in this challenge, we chose the BERT Large Cased model to construct the knowledge base through PLM probing. The challenge uses a diverse set of 12 relations, each covering a different topic equipped with a set of (SubjectEntity, relation, ObjectEntity) triples as ground truth. Table 1 lists the relations along with their descriptions and ground truth examples. We describe our prompt design for each individual relation below. Notice that for the KBC-LM task, we usually do not have a verbalizer to project prediction labels.

#### 3.1. ChemicalCompoundElement

At first glance, the semantics of this relation seems to be ambiguous, because chemical components can be an object at the molecular, ionic, or atomic level. After analyzing the training and development datasets, we found that more than 98% of the object entities were chemical elements, and only less than 2% of the objects were in ionic groups. The limited number of chemical elements allows us to filter out the LM outputs by only keeping chemical elements for filling up the blanks.

In addition, we noticed that names of some object entities follow simple linguistic rules. For example, the entity named “Zinc phosphate” has chemical compound elements “zinc” and “phosphorus”. The first four characters of each token in “Zinc phosphate” are respectively the same as that of “zinc” and “phosphorus”. In terms of prompt design, we noticed that the names

#	Relation	Description	Example
1	ChemicalCompoundElement	chemical compound (s) consists of an element (o)	(Water, ChemicalCompoundElement, [Hydrogen, Oxygen])
2	PersonCauseOfDeath	person (s) died due to a cause (o)	(Neil deGrasse Tyson, PersonCauseOfDeath, [])
3	CompanyParentOrganization	company (s) has another company (o) as its parent organization	(Hitachi, CompanyParentOrganization, [])
4	PersonInstrument	person (s) plays an instrument (o)	(Chester Bennington, PersonInstrument, [])
5	PersonEmployer	person (s) is or was employed by a company (o)	(Susan Wojcicki, PersonEmployer, [Google])
6	PersonPlaceOfDeath	person (s) died at a location (o)	(Elvis Presley, PersonPlaceOfDeath, [Graceland])
7	RiverBasinsCountry	country (s) basins in a country (o)	(Drin, RiverBasinsCountry, [Albania])
8	PersonLanguage	person (s) speaks in a language (o)	(Bruno Mars, PersonLanguage, [Spanish, English])
9	PersonProfession	person (s) held a profession (o)	(Nicolas Sarkozy, PersonProfession, [Lawyer, Politician, Statesperson])
10	CountryBordersWithCountry	country (s) shares a land border with another country (o)	(Canada, CountryBordersWithCountry, [[USA, United States of America]])
11	CountryOfficialLanguage	country (s) has an official language (o)	(Belarus, CountryOfficialLanguage, [Belarusian, Russian])
12	StateSharesBorderState	state (s) of a country shares a land border with another state (o)	(Florida, StateSharesBorderState, [Georgia])

**Table 1**  
ISWC-2022 KBC-LM Challenge Relation Names, Descriptions, and Triple Examples

of some subject entities comprised of two or more words such as “acid” and “hydroxide”. The basic knowledge in Chemistry indicates that “acid” must contain hydrogen, and “hydroxide” must contain hydrogen and oxygen. We hypothesized that if we split the compound names into individual tokens and feed the single tokens to the language model, we might obtain more correct answers from the LM. Using “Chloric acid” as an example, not only we can ask language model “[MASK] is a chemical compound element of Chloric acid”, we can also ask “[MASK]... of Chloric” and “[MASK]... of acid”. By implement this idea, we observed improved recall metrics.

### 3.2. PersonCauseOfDeath

For this relation, we analyzed the training data set and found 50% of the SubjectEntity in the data set are still alive with an empty “cause of death”. Therefore, we split the probing into 2 steps. First, we probed the LM about the life status of the SubjectEntity, i.e., determining whether the SubjectEntity is “dead” or “alive”. Second, we probed the LM about the cause of death under the premise that the SubjectEntity has already deceased.

To address the first problem, we began with the prompts corresponding to the direct question: “Did XXX die?”. Unfortunately, we found that a small perturbation in the prompts would cause significantly different results. Using the following set of prompts: “Did XXX really die? [MASK].” and “Is XXX still alive? [MASK].”, we noticed that the results would be more skewed towards “No”s. Sometimes, we received the answer “No” to both questions for the same subject. This is unreasonable from human understanding, because nobody can be “not alive” and “not dead” at the same time. In a slightly different way, if we asked “Did XXX die? [MASK].” or “Is XXX alive? [MASK].”, the results generally contains more “Yes”s. The propensity of answers can be easily affected by the mood words in the prompts, such as “really”, “still”. On the contrary, it is not sensitive to the keyword itself. Asking “dead” and “alive” will even get trend-aligned answers. If we design prompt in this way, the answer extraction step would be extremely unreliable and unstable.

A language model captures massive statistics about word co-occurrences in a context. To probe more effectively, a prompt should reproduce the context in which the SubjectEntity and ObjectEntity tend to co-occur. For example, if a person XXX is dead, it is more likely that the following phrase appears in the corpus: “XXX is dead”. Based on the co-occurrence statistics, we designed prompts in the format “[SubjectEntity] (is|has) [MASK]”, where “(is|has)” means choosing one of the strings separated by |. We treated the tokens predicted by the language model as an answer space. We detected the “alive” or “dead” status of the SubjectEntity based on the presence of “die” and its variants. For example, if the token “die” exists in the answer space, we consider the SubjectEntity is dead, otherwise is alive. This strategy indeed achieved improved results tested on both the training and development sets. To address the second problem, we followed our intuition and experimented with different prompts and answer thresholds with moderate improvements compared to the baseline.

### 3.3. CompanyParentOrganization

For this relation, we probed in 2 steps: (1) does the SubjectEntity has a parent organization? and (2) if yes, which one in the answer list is likely a parent organization. A challenge for addressing the first problem is to distinguish the relations of subsidiary and parent organizations. We found that the LM under-performed for the problem no matter how the prompts were designed.

### 3.4. PersonInstrument

The prompt is: “[SubjectEntity] (loves|likes playing) [MASK], which is a instrument.” The prompt uses the article “a” instead “an” in front of “instrument” for a better performance. We also noticed that the verb phrase (loves|likes playing) improved the model performance.

### 3.5. PersonEmployer

We have tried many different prompts and adjusted thresholds for top\_k answers. However, we still obtained the lowest performance for this relation. The current prompt is: “[SubjectEntity] joined and work at [MASK] as an employer, which is a company”.

### 3.6. PersonPlaceOfDeath

In the same way for probing answers for the relation PersonCauseOfDeath, we probed for PersonPlaceOfDeath in two steps: (1) checking whether the SubjectEntity is “dead” or “alive” and (2) discovering the place of death if the SubjectEntity has deceased. Assuming we found out that the SubjectEntity is dead, the prompt for detecting the place of death is: “[SubjectEntity] died at home or hospital in [MASK].”

### 3.7. RiverBasinsCountry

The prompt is: “[SubjectEntity] river basins in [MASK].” Other prompts did not achieve better performance than the prompt above. The F1-score was improved from 0.38 of using BERT-base to 0.55 of using BERT-large.

### 3.8. PersonLanguage

The prompt is: “[SubjectEntity] speaks in [MASK], which is a language.” As with many other relations, we found that a prompt containing a subordinate clause such as “which is a language” improved performance. The F1-score was improved from 0.43 of using BERT-base to 0.70 of using BERT-large.

### 3.9. PersonProfession

The prompt is: “[SubjectEntity] is (a or an) [MASK], which is a profession.” Using the phrase “(a or an)” which includes both forms of the article “a” improved the performance. The F1-score was improved from 0.0 of using BERT-base to 0.25 of using BERT-large. As an intermediate summary, Table 2 shows the probing parameters and validation results of these three relation: RiverBasinsCountry, PersonLanguage, and PersonProfession.

	<b>RiverBasinsCountry</b>	<b>PersonLanguage</b>	<b>PersonProfession</b>
model	BERT-large	BERT-large	BERT-large
top_k	4	1	5
threshold	0.071	0.184	0.010
Precision	0.643	0.840	0.365
Recall	0.590	0.654	0.202
F1	0.546	0.701	0.249

**Table 2**

Probing Parameters and Validation Results of RiverBasinsCountry, PersonLanguage, PersonProfession

### 3.10. CountryBordersWithCountry

The prompt is: “[SubjectEntity] and [MASK] are neighboring country. They share the border.” We experimented with more than twenty prompts and finally we found joining the “[SubjectEntity]” and the “[MASK]” with “and” as the subject in the prompt will perform better than the prompt “[SubjectEntity] is neighboring with [MASK]. They share the border.” In particular, the recall was improved from 8.7% to 66.2%, and the F1-score was from 12.2% to 54.8%. The advantage of our prompt is that it strengthens the relationship between the “[SubjectEntity]” and the “[MASK]” as neighboring countries. The model takes the entire string “[SubjectEntity] and [MASK]” as a whole to probe the LM.

### 3.11. CountryOfficialLanguage

The prompt is: “[SubjectEntity]’s official language is [MASK].” Firstly, we experimented with changing the order of the “[SubjectEntity]” and “[MASK]” in the prompt sentence. For example, we tried “[MASK] is the official language of [SubjectEntity].” We also tried to place different adjectives such as “national”, “official”, and “country” in front of the word “language”. Finally, we determined to use the template “[SubjectEntity]’s (adjective) language is [MASK].”, and use “official” to describe “language”. Overall, we improved the recall by 6.5% and improved the F1-score from 78.6% to 81.2% from the default baseline.

### 3.12. StateSharesBorderState

This is a relatively difficult relationship to deal with, because “a state” could refer to different geographical entities in different countries. It would be difficult to retrieve a correct answer if the location of the state cannot be determined in the probing. We took two steps to address the problem. First, we query the LM to discover a list of possible countries where a state is located by using the first prompt: “[SubjectEntity] is a state in [MASK], which is a country.” Second, we embed a possible country name in next prompt sentence to probe bordering states. The second prompt is: “[SubjectEntity] and [MASK] are neighboring states in [ObjectEntity]”, where the “[ObjectEntity]” is replaced by a result of probing with the first prompt. This strategy effectively narrowed the scope of the prompt query, and successfully improved the precision and recall metrics. We ended up with improving the F1-score from the baseline’s 0.01% to 31%.

## 4. Test Results

Table 3 shows the test results using the CodaLab live leaderboard<sup>1</sup> provided by the organizers. Our probing results are list in the columns with the header “Drexel”. We also list the probing

---

<sup>1</sup><https://codalab.lisn.upsaclay.fr/competitions/5815>

results of the “Baseline”<sup>2</sup>. The comparison shows that we improved the overall average F1-score from the Baseline’s 31% to 49%.

#	Relation	Macro Average Precision		Macro Average Recall		Macro Average F1-score	
		Drexel	Baseline	Drexel	Baseline	Drexel	Baseline
1	ChemicalCompoundElement	0.5381	0.9800	0.7213	0.0690	0.5870	0.0980
2	PersonCauseOfDeath	1.0000	0.8600	0.5000	0.5000	0.5000	0.3600
3	CompanyParentOrganization	1.0000	0.9000	0.7400	0.7400	0.7400	0.6400
4	PersonInstrument	0.4067	1.0000	0.7505	0.3600	0.3553	0.3600
5	PersonEmployer	0.9600	0.9800	0.0200	0.0200	0.0200	0.0200
6	PersonPlaceOfDeath	1.0000	0.9800	0.5000	0.5000	0.5000	0.4800
7	RiverBasinsCountry	0.5400	0.9600	0.5557	0.4040	0.5009	0.4290
8	PersonLanguage	0.9400	1.0000	0.6652	0.3757	0.7363	0.4280
9	PersonProfession	0.4977	1.0000	0.2281	0.0000	0.2966	0.0000
10	CountryBordersWithCountry	0.5738	0.9800	0.6627	0.1046	0.5479	0.1187
11	CountryOfficialLanguage	0.9600	0.9800	0.7652	0.7185	0.8120	0.7860
12	StateSharesBorderState	0.3951	0.9000	0.2858	0.0057	0.3160	0.0100
	<b>Average</b>	0.7343	<b>0.9600</b>	<b>0.5329</b>	0.3165	<b>0.4927</b>	0.3108

**Table 3**  
Evaluation results by the CodaLab live leaderboard.

## 5. Discussion

All the prompts were manually designed based on human intuition and trial-and-error. At the end, we also manually aggregate the results and removed stop words. It would be more useful and helpful if prompts could be developed in a systematic and general way for future KBC-LM tasks. We will investigate automatic methods that can learn appropriate prompts by matching training triples to text corpora.

By participating in this challenge, we have learned valuable insights about the “knowledge” of a language model, in particular, the BERT Large Cased Model. We found that it was relatively easier to probe scientific knowledge from the LM than to retrieve facts about social events such as the cause of the death of a famous person. A possible reason could be that the text corpora used for training the LM contained noisier information about social events than about scientific facts and rules.

## 6. Implementation

Our implementation is available in a github repository here: <https://github.com/anyuanay/KBC-LM-Drexel><sup>3</sup>. The implementation directory contains the following content:

<sup>2</sup><https://github.com/lm-kbc/dataset/blob/main/baseline.py>

<sup>3</sup><https://github.com/anyuanay/KBC-LM-Drexel>



- `main.py`: the main entry
- `MyTools.py`: prompts and other middle processes
- `Processors.py`: optimizing parameters such as `top_k` and thresholds
- `MyHelpers.py`: help functions on some logics
- `baseline.py`: Script provided by the organizers; called by our program
- `file_io.py`: Script provided by the organizers; called by our program
- `README.txt`
- `data/`
  - `test.jsonl`
  - `predictions.jsonl`

The README file in the directory contains instructions to run the system.

## 7. Conclusion

This challenge provides multiple types of relations for LM probing. We designed and tested relation-specific prompts and answer processing steps. The test results showed our probing significantly improved the baseline from 31% to 49% in terms of macro average F1-score. The insights we learned from this challenge would lead us to select appropriate LMs for future knowledge base constructions.

## Acknowledgments

This project is partially supported by the Drexel Office of Faculty Affairs' 2022 Faculty Summer Research awards #284213.

## References

- [1] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: NAACL, Minneapolis, Minnesota, 2019, pp. 4171–4186. URL: <https://aclanthology.org/N19-1423>. doi:10.18653/v1/N19-1423.
- [2] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, RoBERTa: A Robustly Optimized BERT Pretraining Approach, arXiv e-prints (2019) arXiv:1907.11692. arXiv:1907.11692.
- [3] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, volume 33, 2020, pp. 1877–1901. URL: <https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfc4967418bfb8ac142f64a-Paper.pdf>.

- [4] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, *Journal of Machine Learning Research* 21 (2020) 1–67. URL: <http://jmlr.org/papers/v21/20-074.html>.
- [5] T. Schick, H. Schütze, Exploiting cloze-questions for few-shot text classification and natural language inference, in: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*, Online, 2021, pp. 255–269. URL: <https://aclanthology.org/2021.eacl-main.20>. doi:10.18653/v1/2021.eacl-main.20.
- [6] T. Schick, H. Schütze, It’s not just size that matters: Small language models are also few-shot learners, in: *NAACL*, Online, 2021, pp. 2339–2352. URL: <https://aclanthology.org/2021.naacl-main.185>. doi:10.18653/v1/2021.naacl-main.185.
- [7] T. Gao, A. Fisch, D. Chen, Making pre-trained language models better few-shot learners, in: *ACL*, Online, 2021, pp. 3816–3830. URL: <https://aclanthology.org/2021.acl-long.295>. doi:10.18653/v1/2021.acl-long.295.
- [8] L. Cui, Y. Wu, J. Liu, S. Yang, Y. Zhang, Template-based named entity recognition using BART, in: *ACL-IJCNLP*, Online, 2021, pp. 1835–1845. URL: <https://aclanthology.org/2021.findings-acl.161>. doi:10.18653/v1/2021.findings-acl.161.
- [9] C. Li, F. Gao, J. Bu, L. Xu, X. Chen, Y. Gu, Z. Shao, Q. Zheng, N. Zhang, Y. Wang, Z. Yu, SentiPrompt: Sentiment Knowledge Enhanced Prompt-Tuning for Aspect-Based Sentiment Analysis, *arXiv e-prints* (2021) arXiv:2109.08306. arXiv:2109.08306.
- [10] X. Han, W. Zhao, N. Ding, Z. Liu, M. Sun, Ptr: Prompt tuning with rules for text classification, *arXiv preprint arXiv:2105.11259* (2021).
- [11] A. Aghajanyan, D. Okhonko, M. Lewis, M. Joshi, H. Xu, G. Ghosh, L. Zettlemoyer, HTML: Hyper-Text Pre-Training and Prompting of Language Models, *arXiv e-prints* (2021) arXiv:2107.06955. arXiv:2107.06955.
- [12] D. K. Choe, E. Charniak, Parsing as language modeling, in: *EMNLP*, Austin, Texas, 2016, pp. 2331–2336. URL: <https://aclanthology.org/D16-1257>. doi:10.18653/v1/D16-1257.
- [13] F. Petroni, T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu, A. Miller, Language models as knowledge bases?, in: *EMNLP-IJCNLP*, Hong Kong, China, 2019, pp. 2463–2473. URL: <https://aclanthology.org/D19-1250>. doi:10.18653/v1/D19-1250.
- [14] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, Q. Liu, ERNIE: Enhanced language representation with informative entities, in: *ACL*, Florence, Italy, 2019, pp. 1441–1451. URL: <https://aclanthology.org/P19-1139>. doi:10.18653/v1/P19-1139.
- [15] L. Dong, N. Yang, W. Wang, F. Wei, X. Liu, Y. Wang, J. Gao, M. Zhou, H.-W. Hon, *Unified Language Model Pre-Training for Natural Language Understanding and Generation*, Curran Associates Inc., Red Hook, NY, USA, 2019.
- [16] H. Bao, L. Dong, F. Wei, W. Wang, N. Yang, X. Liu, Y. Wang, S. Piao, J. Gao, M. Zhou, H.-W. Hon, Unilmv2: Pseudo-masked language models for unified language model pre-training, in: *Proceedings of the 37th International Conference on Machine Learning, ICML’20*, JMLR.org, 2020.
- [17] X. Ouyang, S. Wang, C. Pang, Y. Sun, H. Tian, H. Wu, H. Wang, ERNIE-M: Enhanced multilingual representation by aligning cross-lingual semantics with monolingual corpora, in: *EMNLP*, Online and Punta Cana, Dominican Republic, 2021, pp. 27–38. URL: <https://aclanthology.org/2021.emnlp-main.3>. doi:10.18653/v1/2021.emnlp-main.3.
- [18] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettle-

- moyer, BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, in: ACL, Online, 2020, pp. 7871–7880. URL: <https://aclanthology.org/2020.acl-main.703>. doi:10.18653/v1/2020.acl-main.703.
- [19] K. Song, X. Tan, T. Qin, J. Lu, T.-Y. Liu, Mass: Masked sequence to sequence pre-training for language generation, in: International Conference on Machine Learning, 2019, pp. 5926–5936.
- [20] Z. Jiang, F. F. Xu, J. Araki, G. Neubig, How Can We Know What Language Models Know?, Transactions of the Association for Computational Linguistics 8 (2020) 423–438. URL: [https://doi.org/10.1162/tacl\\_a\\_00324](https://doi.org/10.1162/tacl_a_00324). doi:10.1162/tacl\_a\_00324.
- [21] W. Yuan, G. Neubig, P. Liu, Bartscore: Evaluating generated text as text generation, in: M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, J. W. Vaughan (Eds.), NeurIPS, volume 34, 2021, pp. 27263–27277. URL: <https://proceedings.neurips.cc/paper/2021/file/e4d2b6e6fdeca3e60e0f1a62fee3d9dd-Paper.pdf>.
- [22] A. Haviv, J. Berant, A. Globerson, BERTese: Learning to speak to BERT, in: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, Online, 2021, pp. 3618–3623. URL: <https://aclanthology.org/2021.eacl-main.316>. doi:10.18653/v1/2021.eacl-main.316.
- [23] E. Wallace, S. Feng, N. Kandpal, M. Gardner, S. Singh, Universal adversarial triggers for attacking and analyzing NLP, in: EMNLP-IJCNLP, Hong Kong, China, 2019, pp. 2153–2162. URL: <https://aclanthology.org/D19-1221>. doi:10.18653/v1/D19-1221.
- [24] E. Ben-David, N. Oved, R. Reichart, PADA: Example-based Prompt Learning for on-the-fly Adaptation to Unseen Domains, Transactions of the Association for Computational Linguistics 10 (2022) 414–433. URL: [https://doi.org/10.1162/tacl\\_a\\_00468](https://doi.org/10.1162/tacl_a_00468). doi:10.1162/tacl\_a\_00468.
- [25] J. Davison, J. Feldman, A. Rush, Commonsense knowledge mining from pretrained models, in: EMNLP-IJCNLP, Hong Kong, China, 2019, pp. 1173–1178. URL: <https://aclanthology.org/D19-1109>. doi:10.18653/v1/D19-1109.
- [26] X. L. Li, P. Liang, Prefix-tuning: Optimizing continuous prompts for generation, in: ACL, Online, 2021, pp. 4582–4597. URL: <https://aclanthology.org/2021.acl-long.353>. doi:10.18653/v1/2021.acl-long.353.
- [27] Z. Zhong, D. Friedman, D. Chen, Factual probing is [mask]: Learning vs. learning to recall, in: NAACL, 2021.
- [28] X. Liu, Y. Zheng, Z. Du, M. Ding, Y. Qian, Z. Yang, J. Tang, GPT understands, too, CoRR abs/2103.10385 (2021). URL: <https://arxiv.org/abs/2103.10385>. arXiv:2103.10385.
- [29] Z. Jiang, A. Anastasopoulos, J. Araki, H. Ding, G. Neubig, X-FACTR: Multilingual factual knowledge retrieval from pretrained language models, in: EMNLP, Online, 2020, pp. 5943–5959. URL: <https://aclanthology.org/2020.emnlp-main.479>. doi:10.18653/v1/2020.emnlp-main.479.
- [30] W. Yin, J. Hay, D. Roth, Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach, in: EMNLP-IJCNLP, Hong Kong, China, 2019, pp. 3914–3923. URL: <https://aclanthology.org/D19-1404>. doi:10.18653/v1/D19-1404.
- [31] T. Shin, Y. Razeghi, R. L. Logan IV, E. Wallace, S. Singh, AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts, in: Proceedings of EMNLP, Online, 2020, pp. 4222–4235. URL: <https://aclanthology.org/2020.emnlp-main.346>. doi:10.

18653/v1/2020.emnlp-main.346.

- [32] X. Chen, N. Zhang, X. Xie, S. Deng, Y. Yao, C. Tan, F. Huang, L. Si, H. Chen, Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction, in: Proceedings of the ACM Web Conference 2022, WWW '22, New York, NY, USA, 2022, p. 2778–2788. URL: <https://doi.org/10.1145/3485447.3511998>. doi:10.1145/3485447.3511998.
- [33] X. L. Li, P. Liang, Prefix-tuning: Optimizing continuous prompts for generation, in: ACL, Association for Computational Linguistics, Online, 2021, pp. 4582–4597. URL: <https://aclanthology.org/2021.acl-long.353>. doi:10.18653/v1/2021.acl-long.353.
- [34] B. Lester, R. Al-Rfou, N. Constant, The power of scale for parameter-efficient prompt tuning, in: EMNLP, Online and Punta Cana, Dominican Republic, 2021, pp. 3045–3059. URL: <https://aclanthology.org/2021.emnlp-main.243>. doi:10.18653/v1/2021.emnlp-main.243.
- [35] T. He, K. Cho, J. R. Glass, An empirical study on few-shot knowledge probing for pretrained language models, CoRR abs/2109.02772 (2021). URL: <https://arxiv.org/abs/2109.02772>. arXiv:2109.02772.
- [36] N. Poerner, U. Waltinger, H. Schütze, E-BERT: Efficient-Yet-Effective Entity Embeddings for BERT, arXiv e-prints (2019) arXiv:1911.03681. arXiv:1911.03681.
- [37] B. Dhingra, J. R. Cole, J. M. Eisenschlos, D. Gillick, J. Eisenstein, W. W. Cohen, Time-aware language models as temporal knowledge bases, Transactions of the Association for Computational Linguistics 10 (2022) 257–273. URL: <https://aclanthology.org/2022.tacl-1.15>. doi:10.1162/tacl\_a\_00459.
- [38] M. Sung, J. Lee, S. Yi, M. Jeon, S. Kim, J. Kang, Can language models be biomedical knowledge bases, in: EMNLP, 2021.
- [39] Z. Meng, F. Liu, E. Shareghi, Y. Su, C. Collins, N. Collier, Rewire-then-probe: A contrastive recipe for probing biomedical knowledge of pre-trained language models, in: 60th ACL, 2022.