Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019)

Extraction and Classification of Speech, Thought, and Writing in German Narrative Texts

Luise Schricker and Manfred Stede
Applied Computational Linguistics
Dept. of Linguistics
University of Potsdam / Germany
schricker|stede@uni-potsdam.de

Peer Trilcke
Literary Studies
Dept. of German Studies
University of Potsdam / Germany
trilcke@uni-potsdam.de

Abstract

For various purposes of narrative text analysis, it is helpful to identify *speech and thought events*: material that is uttered or imagined by some protagonist. This task commonly distinguishes between direct and indirect speech, but we will also consider free indirect and reported speech here. Specifically, we build upon earlier work by Brunner (2015), who presented an annotated German corpus of narrative texts as well as an automatic analysis system. We propose a variety of extensions and are able to substantially improve on the original results for all four categories.

1 Introduction

Identifying speech, thought and writing (henceforth STWR) of characters is a central task in automatically understanding narrative text. It is commonly divided into several categories, the most widely-researched (on the computational side) being *direct speech*. In the following example from Fontane's *Effi Briest*, the speech content is marked by italics and the so-called 'inquit formula' by boldface:

(1) "*Warum lacht ihr?*", **sagte Effi** pikiert. "*Was soll das heißen?*"
("*Why are you laughing?*", **said Effi** slightly offended. "*What is it supposed to mean?*")

The next example, taken from the same novel, illustrates *indirect speech*, which is commonly expressed with dependent clauses (italics) and a framing clause (boldface):

(2) In diesem Augenblick trat Wilke vom Saal her ein und **meldete**, *dass er alles schon nachgezählt und alles vollzählig gefunden habe;*
(At this point Wilke entered from the hall and **announced** *that he has counted everything and found it to be complete;*)

In *free indirect speech* the character speaks with the narrators voice, which we also illustrate with an example from *Effi Briest*, with Effi's thought process in italics:

(3) Sie hatte Mühe, sich zurechtzufinden. *Wo war sie? Richtig, in Kessin, im Hause des Landrats von Innstetten, und sie war seine Frau, Baronin Innstetten.*
(She had trouble with orientation. *Where was she? Right, in Kessin, in the house of Instetten, and she was his wife, Baroness Innstetten.*)

Finally, *reported speech* has the most distance to the actual words produced by the character, which the narrator may summarize, as in this example from the same source:

(4) Sie sprachen noch eine Weile so weiter, wobei sie sich ihrer gemeinschaftlichen Schulstunden und einer ganzen Reihe Holzapfelscher Unpassendheiten mit Empörung und Behagen erinnerten.
(They talked like that for another while, remembering with indignation and with pleasure the common school hours and a number of Holzapfel's inappropriate moments.)

More detailed descriptions of the STWR types can be found in narratological studies, such as Genette (1998), Jessing et al. (2007), or Martinez and Scheffel (2012).

From the computational viewpoint, attributing speech, thought and writing to the characters producing them is a fundamental step in following the storyline of a narrative, and the first central step is to be able to *identify* STWR material. This, in turn has two steps: making the distinction between direct/indirect/free ind./reported, and that between speech, thought and writing. For German, there has not been a lot of work on these tasks, and only one sizable annotated corpus was available at the time of our system's development.[1] It was com-

[1] In the meantime, a bigger corpus with STWR annotations

piled by Brunner (2015), and her work constitutes the starting point of our study. Our main contribution is to propose an extended set of features used by an automatic classifier, as well as a few other improvements that substantially improve on Brunner's original results and thus can be regarded as a new state of the art for this dataset.

The paper is structured as follows: After introducing the corpus and briefly reviewing earlier research in Section 2, we explain our experiments in Section 3, report and discuss the results in Section 4, and then conclude in Section 5.

2 Corpus and Related Work

2.1 Corpus

The corpus annotated by Brunner (2015) is made up of thirteen short German narratives. The texts were written between 1787 and 1913 and consist of overall roughly 57,000 tokens. They were annotated for the 12 categories that result from combining speech, thought and writing each with the four different types direct, indirect, free_indirect and reported. Several attributes, mostly signaling border cases of annotation, were also included.

The corpus has several strengths and weaknesses. It is not a balanced dataset, insofar as the number of instances for each class differ considerably. This is most pronounced in the case of *free_indirect* STWR which only amounts to 110 instances compared to 1038 instances for *direct* STWR. Furthermore, the whole corpus was annotated by a single person, Brunner herself, thus making the annotations susceptible to subjectivity.

On the other hand, the texts of the corpus were carefully chosen to represent a wide selection of narrative texts. Complete short texts were selected, rather than excerpts (with two exceptions). The texts have diverse dates of origin spanning over a 100 years, and authors of both genders are represented. Also, care was taken to ensure a mix of different narrative perspectives, first person and third person. The texts differ in the punctuation schema used as well as in orthography. We propose that these characteristics provide a realistic set-up for developing a system for practical use in literary studies.

has been beta-released by the *Redewiedergabe-Projekt* (Brunner et al., 2019). This newly released corpus could not be used in the present study though because of time constraints.

2.2 Earlier research

Most earlier work on speech event identification targets a different genre, viz. news text. For instance, Krestel et al. (2008) develop a rule-based system for extracting *direct* and *indirect* speech from newspaper articles. Their system consists of two components: a reporting verb marker and a reported speech finder, which uses six patterns combining a reporting verb, a speech source, and one or two clauses containing the speech content. The authors evaluate their system on seven newspaper articles from the Wall Street Journal corpus with a total of 6100 words and report a recall of 0.83 and a precision of 0.98 on this test set.

Sarmento and Nunes (2009) develop a system, which they call *verbatim*, for the extraction of *direct* and *indirect* quotes from Portuguese news texts. Their system uses 19 patterns and a list of 35 speech words to extract quotes along with the respective speakers and speech acts. The system is evaluated manually: of 570 extracted quotes, 68 are considered errors, yielding a precision of 0.88; recall is not reported.

Pareti et al. (2013) train and evaluate two machine learning (ML) approaches to extract *direct*, *indirect* and mixed quotes from two news corpora. In order to avoid compiling a list of common speech verbs, which cannot account for all verbs used as cues for quotations, the authors train a classifier to identify attribution verb cues. They experiment both with a token-based approach and with a system that classifies parse nodes. These experiments result in F1 measures of up to 0.60 for indirect and 0.94 for direct quotations with exact boundary matching.

Turning to narrative, Brunner (2015) (see also (Brunner, 2019)) implemented both a rule-based and an ML based system for the automatic annotation of the STWR categories in German narrative. The best F1 scores that were achieved with her systems range from 0.40 for *free indirect* STWR to 0.87 for *direct* STWR. For the types *reported* and *indirect* STWR the best F1 values are 0.58 resp. 0.71. These results were achieved with sentences as a basic unit: Brunner considers a sentence as belonging to a certain STWR class, if at least one STWR instance belonging to this class appears somewhere within it. If a sentence contains instances of several different STWR classes, it receives multiple labels. In addition, Brunner also experimented with segments of sentences, which

roughly correspond to clauses, but she discarded that approach because it lowered the performance of her system. Brunner presented her implementations as prototypes and suggested that further features be explored for potentially improving the systems. Our experiments reported below build directly on Brunner's work.

3 Experiments

3.1 Approach

While Brunner's work focuses on sentence-based classification, our work targets the automatic recognition of sub-sentential boundaries. Hence, we train our classifiers on the segments of sentences Brunner used for her "extra" experiments. In order to also allow the annotation of unseen texts, we reimplemented the tokenization algorithm described by Brunner and slightly extended it toward a broader quotation recognition method by adding heuristics for disambiguating apostrophes and quotation marks, following Percillier (2017).

3.2 Training and balancing

We follow Brunner in training a binary classifier for each STWR type, to allow multiple labels per segment, as STWR instances can naturally occur in nested form. After extracting STWR instances from the corpus per type, each training data set is transformed into feature representations. At this point the segments are still in the order as they appear in the texts, allowing us to also build sequence-based features (and we will specifically test their impact on the results). For the SVM classifier, the feature representations are also being scaled for efficiency reasons. The transformed data set is then split into stratified train and test sets. This step differs from Brunner, who did not set aside a dedicated test set, but rather evaluated the ML classifiers via cross validation (CV). on the whole training set. As she for the most part abstains from parameter tuning, the CV results are probably not biased.[2] For our study, on the other hand, we want to run parameter tuning in order to find the optimal configurations for the ML methods. Thus, 25% of the data for each class are set aside for testing. Due to this difference in evaluation data, some of our results presented in Sct. 4 cannot be directly compared to Brunner's results.

Experiments and parameter tuning were performed via stratified ten-fold cross validation on the training set. For feature scaling, the test-train split and cross validation, implementations provided by Scikit-learn 0.20.2 (Pedregosa et al., 2011) are used. In every fold the train set is further adjusted to tackle the class imbalance problem, which is especially pronounced in the case of the free indirect class. Brunner used the technique of *oversampling* on the training data to achieve an equal class distribution. For this technique instances are drawn randomly from the smaller class until an equal class distribution is achieved. While the training data can be adjusted to achieve equal class distribution, data set aside for evaluation or validation stays untouched to ensure reliable results.

Specifically, we use two alternative strategies to counter the class imbalance problem and compare them to the oversampling technique used by Brunner. The *Synthetic minority over-sampling technique (SMOTE)* (Chawla et al., 2002) uses a combination of undersampling the majority class and oversampling the minority class. Furthermore, the oversampled instances are synthetically created by adjusting features in the direction of one of the k nearest neighbors of the instance. For this, we use the imbalanced-learn package (Lemaître et al., 2017), version 0.4.3.

While SMOTE is domain agnostic, our second balancing strategy is a domain-specific *data augmentation* method. Here, features that were expected to be variable in naturally occurring data were manually selected for each class. As with oversampling, instances are drawn from the minority class until an even class distribution is achieved. For each drawn instance, a random subset of the augmentation features for this class are chosen and the instance's respective feature values are changed: Boolean features are negated and real-valued features are substituted by a random value within the interval of possible values for this instance.

3.3 Classifiers

Our reporting type classifier consists of four separate binary classifiers that determine whether an instance contains at least one occurrence of the class *direct, indirect, free indirect* and *reported*, respectively. We do not use a multiclass approach because an instance can contain nested or sequentially

[2] She mentions experiments with different ML methods and ways to remove class imbalances, but does not give concrete results. Her reported results were all achieved by the same configuration: a Random Forest with 500 trees, which can each inspect eight features when adding a new node.

appearing STWRs that belong to different classes. Furthermore, the STWR annotations which have the *ambiguous* attribute belong to two classes.

We evaluate three different ML techniques on our task: Random Forest, Support Vector Machines (SVM) and Multi-Layer Perceptron (MLP). For all three ML techniques implementations provided by Scikit-learn 0.20.2 (Pedregosa et al., 2011) are used: *RandomForestClassifier*, *SVC* and *MLPClassifier*. Some parameters, for example the number and size of the hidden layers (MLP) or the minimal number of samples per leaf (Random Forest), were adjusted via grid search and cross validation on the training data. For the others, we largely adopted the default parameter options, but the MLP's *early_stopping* parameter was set to True instead of False (default) and the *tolerance for the optimization (tol)* was set to 0.01 instead of 0.0001 (default). These settings achieve an early stop of the training process when the loss or validation score does not improve by at least *tol* for ten consecutive iterations. This is a regularization measure to prevent overfitting, which is important given the small number of training samples. The SVC's *tol* parameter was also set to 0.01 for the same reasons.

Following the annotations performed by the reporting type classifier, a simple rule-based system adds a further layer of annotation: The detected STWR instances are categorized as either speech, thought or writing. The *direct* and *free indirect* classes are always annotated with their majority class. For the *indirect* and *reported* classes, we search the respective segment for occurrences of words in a *reporting word list* (see below). If exactly one such word is found in the segment, it is marked correspondingly as speech, thought or writing. In the absence of a reporting word, we use the majority class.

3.4 Features

We now describe our extensive feature set, which was developed on the basis of Brunner (2015) – henceforth: B15 – and was extended with features adapted from other literature or devised by us.

In order to keep track of sequential features, we use a backlog with information about the previous ten segments' labels. We see this as potentially useful because certain STWR types tend to appear in blocks, for example when a conversation between characters is reported. B15 also reported this observation, but did not use corresponding features.

Our set-up of the data flow is such that only gold labels can be used for the sequential features instead of real predictions by the system, which can only be made after the system has been trained, i.e. after the dataset is already split into stratified train- and test-sets. Those sets lose segment order information, as they are constrained to contain the same distribution of classes. More sophisticated experiments with system-generated sequential features are left for future work. However, to ascertain their general influence, we evaluated our classifier both with and without the gold label features, thus providing upper and lower bounds for performance.

In the course of feature extraction the following tools and packages are used: *pandas* 0.21.0 (McKinney and others, 2010) for data handling, *numpy* 1.13.3 (Walt et al., 2011) for vector operations, *spaCy* 2.0.12 (Honnibal and Montani, 2019) for most linguistic processing tasks (e.g. part of speech (POS) tagging, and named entity recognition), *scipy* 1.0.0 (Jones et al., 2001) for computing cosine distances, *gensim* 3.1.0 (Řehůřek and Sojka, 2010) for the handling of word vectors and the *RFtagger*[3] (Schmid and Laws, 2008) for morphological analysis. As spaCy's German lemmatizer turned out to not work well, *GermaLemma* 0.1.1 (Konrad, 2019) was used as an alternative.

3.4.1 Token features

POS: distribution of POS tags, using both the *TIGER Treebank tags* and *Google Universal POS tag set*. B15 used the STTS tagset and a self-made broader tagset that combined some related tags of the STTS .

Named Entities: Presence of NEs and their types (person, location, organization, misc). Similar features were also used by Fernandes et al. (2011) and Pareti et al. (2013).

Special tokens: Presence of colons (also used by B15), question marks (Cohn (1978) mentions interrogations as one of the characteristic traits of *free indirect* thought) and quotation marks (generally used in the literature for finding *direct* speech).[4] Like B15, we also use segment-end fea-

[3] `http://www.cis.uni-muenchen.de/~schmid/tools/RFTagger/`

[4] We use the number of opening and closing quotation marks within the segment; whether the segment is in quotes;

tures: comma (as indicator of a following inquit formula for direct STWR), emphatic punctuation marks (*? ! - –*), and special combinations associated with the *direct* class.

3.4.2 Morphological features

Person: The change from third to first/second person is often considered an indicator of a change from narration to characters' words (Cohn, 1978). We use the frequency of first and second person pronouns in the segment as separate features (whereas B15 and Mamede and Chaleira (2004) put both pronoun types into the same category); frequency of third person pronouns (cf. B15); Boolean features indicating whether only third person, only first person or a mixture of first and third person was used in the previous five segments. These features give a broader picture of the usage of person in the text.

Mode: Partly adapted from B15, we use Boolean features indicating whether any verb and whether all verbs in the segment are in indicative or in subjunctive mode.[5]

Tense: The basic tense of narration is often past tense while characters' words mostly use present tense (Cohn, 1978). *Free indirect* STWR is an exception, as it displays features of the narrator's voice, e.g. by using the same tense (Jessing et al., 2007; Martinez and Scheffel, 2012). We use Boolean features indicating whether any verb and whether all verbs in the segment are in a form of present tense and past tense, respectively.

3.4.3 Grammatical features

This group comprises a set of Boolean features indicating whether

(i) the previous segment ended with a comma and this segment contains a verb, suggesting an embedded sentence. Embedded sentences, according to B15, are frequently part of *indirect* STWR, especially if they contain verbs in subjunctive mode.

(ii) the segment contains a form of the verb *würden* in combination with an infinitive verb or by itself. Fabricius-Hansen (2002) observes the combination of *würde* ('would') with an infinitive verb as a possible indicator of *free indirect* STWR. B15 also adapted this observation in her system.

(iii) any reporting word within the segment has a noun or prepositional complement, which B15 treated as an indicator of *reported* STWR. (Reporting words are further described below.)

(iv) any reporting word within the segment has a sentence or infinitive complement, which B15 saw as an indicator of *indirect* STWR.

3.4.4 Candidate speaker features

Candidate speakers are usually extracted with the aim to attribute quotes to their speakers, e.g. (Elson and McKeown, 2010; Mamede and Chaleira, 2004). Here, the idea is that the appearance of a candidate speaker might also be useful for detecting STWR. We regard pronouns, named entities of the person type and head nouns that belong to the *Person* category as possible speakers. The list of nouns that belong to the *Person* category was gathered by recursively extracting the hyponyms of all synsets of the word *Person* from GermaNet (Henrich and Hinrichs, 2010). The process for extracting candidate speakers is adapted from Elson and McKeown (2010) and Jannidis (2017).

Our feature set consists of a Boolean feature indicating whether the segment's subject is a candidate speaker; the number of candidate speakers in the segment; and the candidate speaker features of the previous segment.

3.4.5 Reporting word features

We used the list of reporting words that B15 compiled by combining three sources: a linguistically motivated list of reporting verbs, words from a corpus that were extracted via pattern matching, and related words (verbs and nouns) mined from a thesaurus. Each word was given a penalty value representing the degree of "typicality" for a reporting word. If a word can refer to exactly one type of reporting act (speech, thought or writing), this type is recorded in the list — examples are *sagen* ('say') for the speech category and *denken* ('think') for the thought category. For some words, a marker is set to indicate that the word can not be used as a framing clause of *direct* or *indirect* STWR but only for the reporting type. Examples of such words are *plaudern* ('chat') and *befragen* ('interrogate').

Reporting word list features (i) For each penalty value, Boolean features indicate whether at least one reporting word of this value or of lower value appears in the segment. (Adapted from B15.)

(ii) Boolean features indicating whether the segment contains any reporting word that is a verb and

number of contiguous previous segments in quotes (meant to prevent errors caused by missing closing quotation marks).

[5] Martinez and Scheffel (2012) mention the verb form of first person indicative with present tense as one of the typical features of *direct* speech.

whether it contains any that is a noun.

(iii) For each penalty value, Boolean features indicate whether at least one reporting word with lower or equal penalty value with a *reporting marker* is contained within the segment. (B15 distinguishes these reporting words in her rule-based approach, but not in the ML implementation.)

(iv) For each penalty value, the number of reporting words of this (or a lower) value that are contained in the segment.

(v) Numbers of reporting verbs and reporting nouns contained in the segment.

(vi) For each penalty value the number of reporting words of this (or a lower) value with a *reporting marker* within the segment.

(vii) The reporting word features of the previous segment.[6]

Word vector features: Lists of reporting words are often used for quotation extraction, see e.g. (Krestel et al., 2008; Clergerie et al., 2009; Sarmento and Nunes, 2009; Brunner, 2015), but their inherent inflexibility poses a problem. STWR instances can be framed very differently. Especially literary texts can convey reporting acts with potentially infinite variance. Pareti et al. (2013) try to circumvent this problem by implementing a separate classifier for detecting reporting words. Glass and Bangay (2007) use a list of features to determine whether a verb is a reporting verb. Here, we implemented a different approach to the problem.

In addition to looking up words in the reporting word list, similarity values of word vectors were used to achieve a more general indication of the appearance of reporting words. To this end, prototypical word vectors for reporting words and for reporting words with the *reporting marker* were computed. This was done by averaging all word vectors representing the words in the reporting word list with penalty 0 (i.e. the most typical words) and those with penalty 0 and a *reporting marker*. These average vectors can be considered as prototypical representations of the general reporting word group. All lemmata of the words contained in a segment were compared in turn to the two prototypical word vectors using the cosine similarity measure. The maximum similarities to each of the vectors were then added as features. This way the appearance of a word which is not contained in the reporting word list, but which is nonetheless sometimes used as a reporting word, can be detected.

Two word vector models were compared to each other in the experiments: a model trained on the KOLIMO corpus[7] (Herrmann and Lauer, 2017) and a model trained on the German Wikipedia and a corpus of German news articles. The Wikipedia model is available as a pretrained model[8] (Müller, 2018). Its suitability for the present task is not clear because of the genre difference to our task (contemporary non-narrative texts versus narrative written between 1787 and 1913). The data of the KOLIMO corpus is in principle better suited for the task because it contains German narrative with a focus on Modernism, a period which roughly spans the years 1880 to 1930. KOLIMO is a broad corpus though, comprising also many texts from earlier periods, as well as non-narrative. Furthermore, the KOLIMO distribution is a beta release, which still contains some noise and does not have a consistent format (Herrmann and Lauer, 2017). Therefore, both models have their merits and disadvantages.

For preprocessing and training of the KOLIMO word vectors, the toolkit provided by Müller (2018) was used. This toolkit builds on Google's word2vec tool as described by Mikolov et al. (2013).

3.4.6 Other word features

(i) The percentage of **deictic words** in the segment. The list of deictic expressions was taken from B15: *heute*, *morgen*, *gestern*, *jetzt*, *hier* (today, tomorrow, yesterday, now, here). Deictic words can indicate the speaker's perspective, i.e. whether the speaker is likely a character, who is situated in the narrative's here and now, or whether the speaker is a narrator whose frame of reference can be different from the intratextual one (Cohn, 1978).

(ii) A Boolean feature indicating whether the segment begins with a **conjunction** that can indicate the *indirect* class. The list is taken from B15: *dass* ('that'), *ob* ('whether'), *wo* ('where'), etc.

(iii) The percentage of **modal particles** in the segment, whose appearance can be an indicator of character speech. The list of modal particles was also taken from B15: *ja*, *nein*, *wohl*, *schon*, *eigentlich*, *sowieso*, *eben* ('yes', 'no', 'already',

[6] As B15 stated in her description of the segment-based ML experiments, instances of *indirect* reporting are almost always split up into two segments: the framing clause and the dependent clause. Adding the reporting word features of the previous segment should reestablish the connection between the two segments.

[7] https://kolimo.uni-goettingen.de

[8] https://devmount.github.io/GermanWordEmbeddings/

'anyway' and various particles with no English equivalent).

(iv) The percentage of words within the segment that are associated with **negation**. Our list contains *nein* ('no'), *nicht* ('not'), *kein* ('no'). This feature was added because B15 observed in her description of the STWR corpus that the classes *indirect* and *reported* often display markers of negation and non-factuality. She did not use respective features in her own implementations.

(v) Boolean features indicating the appearance of words describing **facial expressions, gestures and voice**. These features were inspired by Tenchini (2010)'s narratological study on the function and relevance of coverbal language in literature. The reasoning is that the appearance of words indicating coverbal language might be an alternative way to recognize the context of an STWR instance in the absence of reporting words. Word lists were manually crafted for each category:

- facial expressions: *Gesicht* ('face'), *Mund* ('mouth'), *Augenbraue* ('brow'), *Auge* ('eye'), *Stirn* ('forehead'), *Lippe* ('lip'), *Nase* ('nose'), *Nasenflügel* ('side of nose')
- gestures: *Hand* ('hand'), *Arm* ('arm'), *Handfläche* ('palm'), *Finger* ('finger'), *Schulter* ('shoulder'), *Faust* ('fist')
- voice: *Stimme* ('voice'), *Ton* ('sound'), *Tonhöhe* ('pitch'), *Tonfall* ('tone'), *Stimmlage* ('register'), *Atem* ('breath')

(vi) A Boolean feature indicating whether any word in the segment is repeated. According to Cohn (1978) the **repetition** of words can be a feature of characters' language.

3.4.7 Sequential label features

For each reporting type, we compute (i) Boolean features that indicate whether the previous segment and whether any of the previous five segments was labeled with this type, and (ii) the number of segments among the previous ten segments which were labeled with the respective STWR type (this should help to detect blocks of STWR). Our final feature in this group is the overall number of STWR instances of all reporting types annotated within the previous ten segments.

3.4.8 Length/position features

In the last group, we use these features: (i) Token- and character-based lengths of the current and the previous segment, as an approximation of the style of the segment by indicating whether longer or shorter words are used. (B15 only used the token-based sentence/segment length.)

(ii) The sum of the token-based lengths of the current and the previous segments and the sum of the character-based lengths.

(iii) Boolean features indicating whether the current or the previous segment constitutes the end of a paragraph (taken from B15).

4 Results and discussion

Below we report our results on overall performance, in comparison to the segment-based results of B15 as far as possible. We will explore the difference between using sequential label features and ignoring them, and between the two word vector models. Regarding the various *balancing* techniques, we found that oversampling and our own data augmentation method outperformed the SMOTE method. As for the different ML approaches, Random Forest yielded the best performance for the overall best configurations (BEST-ALL). These were evaluated on the test set, results are shown in Table 2. Note that the BEST-ALL configurations are not always the same as the best configurations within each individual class (BEST-IND), which were derived by CV on the training data, and whose results are listed in Table 1, sometimes stemming from other classifiers than Random Forest (for reasons of space, we do not provide the detailed comparison). The BEST-ALL configurations were picked by adding the F1 values for each of the four classes and choosing the combination which achieves the highest sum.

The BEST-ALL configuration with sequential label features uses the Random Forest model with KOLIMO word vectors and oversampling. Both KOLIMO vectors and oversampling were chosen as frozen parameters for the ML parameter adjustment phase, which may have influenced the result. For the *indirect* and *reported* classes considered individually (BEST-IND), the Random Forest model with KOLIMO word vectors and data augmentation slightly outperforms the BEST-ALL configuration, indicating untapped potential in data augmentation.

Without sequential label features the BEST-ALL configuration is the Random Forest model with Wikipedia word vectors and oversampling. Three out of four BEST-IND models also use the Wikipedia word vectors. This result is somewhat

surprising, as noted earlier, and may indicate that word vector models trained on non-narrative data can be used for work on narrative texts.

For all four classes, the BEST-IND F1 scores achieved with CV and gold sequential label features outperform the models without sequential label features. The *free indirect* class profits most from the sequential label features, with an increase of 52.17 points F1 score.[9]

The baseline of Brunner's system was only compared to the results of the BEST-IND models without sequential features, derived via CV, in order to ensure a fair comparison. The system developed in this study outperforms Brunner's for every class,[10] with improvements of 17.02 points F1 score for the *indirect* class and 15.81 points for the *direct* class. The *reported* and *free indirect* classes show the least improvement with an additional 9.70 points resp. 11.39 points F1 score.

The BEST-ALL configurations with and without sequential label features were evaluated on the test set. A simple baseline was also evaluated to provide an additional point of comparison. This baseline uses the same configuration as the STWR classifiers, but it only has one feature, viz. the segment's character-based length. Results on the test set are listed in Table 2. The results of the simple baseline are considerably lower than the ML model's results, both with and without sequential label features. For all STWR types except the *free indirect* class the results achieved on the test set are within a range of two points on either side of those achieved with CV on the training set. Results that can be reproduced on the test set are considered reliable. For the *free indirect* class, the F1 scores on the test set, with and without sequential label features, are more than 10 points lower than those achieved via CV. This is likely a consequence of overfitting to the training set, which is difficult to avoid for such a small dataset.

Overall, the results of the models with sequential label features are promising, with an F1 score of 95.01 for the *direct* class, 79.48 for the *indirect* and 70.13 for the *free indirect* class. Only the *reported* class, with a score of 49.28, is still not recognized well. The difference of the scores achieved with and without sequential label features, i.e. between the upper and lower bounds of evaluation, are similar to those observed for CV. *Free indirect* is the class that suffers the most severe decline in accuracy with the loss of the sequential label features, i.e. 58.37 points F1 score, compared to a loss of 52.17 points F1 score for CV. This dependency on the gold labels shows that the class by itself is not well recognized by the classifier. The problem might be alleviated by using a bigger training data set. The *reported* class on the other hand is the class that loses least accuracy when the sequential label features are removed, indicating that *reported* STWR does not often occur blockwise within the STWR corpus.

Finally, we evaluated the second classifier (for distinguishing speech, thought and writing) on the test set. Results are listed in Table 3. Note that only positively classified instances (real predictions, not gold labels) are further annotated with speech, thought and writing. The classification scheme based on the majority class works well for the classes *direct* and *free indirect*. The *direct* class is classified correctly as speech with an F1 score of over 90 points. Thought and writing are minority classes with counts of up to eleven instances, and thus can be ignored. The *free indirect* class only consists of instances that are correctly labeled with the majority class thought, the F1 score reaches 90 points with sequential label features and 60 points without. This shows the influence of the first classifier's performance, because instances that do not belong to a class but are incorrectly recognized as such, are also incorrectly classified with one of the types speech, thought or writing. The results for the *indirect* and *reported* classes are mixed. These types are more heterogeneous than the other two, i.e. their majority class covers a smaller percentage of the instances. Before defaulting to the majority class, the classifier attempts to classify *reported* and *indirect* instances by searching for reporting words. The F1 scores for the classes with more than zero members range between 28.57 and 69.46. This shows that the classification methods do not completely fail in the case of minority classes but that there is still room for improvement.

[9] This substantial increase could be attributed to the fact that most of the *free indirect* samples in the STWR corpus can be found in one text, *Der Irre* by Georg Heym. Therefore, if a sample's sequential label features contain instances of *free indirect*, there is a high probability that the sample belongs to this particular text.

[10] The BEST-IND results were used as the point of comparison to Brunner's baseline, but the BEST-ALL configuration also improves upon Brunner's baseline in every class. Brunners sentence-based results are not reported here, as they cannot be compared to the segment-based results of the present system.

Model	STWR	Seq	Prec.	Recall	F1
Brunner	direct	-	71.00	66.00	69.00
STWR	direct	-	83.25	86.58	84.81
STWR	direct	+	96.47	95.19	95.80
Brunner	indirect	-	40.00	38.00	39.00
STWR	indirect	-	65.46	49.12	56.02
STWR	indirect	+	84.50	74.80	79.21
Brunner	free ind	-	28.00	15.00	20.00
STWR	free ind	-	23.79	47.15	31.39
STWR	free ind	+	87.79	80.46	83.56
Brunner	reported	-	39.00	31.00	34.00
STWR	reported	-	39.61	49.63	43.70
STWR	reported	+	42.56	62.93	50.51

Table 1: Results (BEST-IND) of the parameter tuning process via cross validation compared to Brunner's segment-based results.

Model	STWR	Seq	Prec.	Recall	F1
Baseline	direct	-	28.39	59.21	38.37
STWR	direct	-	87.09	83.26	85.13
STWR	direct	+	96.54	93.51	95.01
Baseline	indirect	-	10.85	43.84	17.40
STWR	indirect	-	61.82	50.25	55.43
STWR	indirect	+	84.07	75.37	79.48
Baseline	free ind	-	03.27	72.73	6.26
STWR	free ind	-	42.86	06.82	11.76
STWR	free ind	+	81.82	61.36	70.13
Baseline	reported	-	12.00	39.56	18.41
STWR	reported	-	41.10	49.45	44.89
STWR	reported	+	43.64	56.59	49.28

Table 2: Results (BEST-ALL) of the evaluation of the STWR classifiers on the test set.

5 Conclusion

We were able to substantially improve the original classification results on a German corpus annotated for speech, thought and writing (Brunner, 2015). This is largely due to changes in the feature set, which we described in detail. In addition, we tested techniques for handling class imbalance: oversampling, SMOTE and our own (domain-specific) data augmentation method. Also, we demonstrated the utility of sequence based features (experiments with predicted values are left for future work, though), and we compared the contributions of two different word vector models.

STW	STWR	Seq	Co	Prec.	Rec.	F1
S	direct	-	389	85.12	100.0	91.96
T	direct	-	9	0.00	0.00	0.00
W	direct	-	1	0.00	0.00	0.00
S	direct	+	436	94.17	100.0	96.70
T	direct	+	11	0.00	0.00	0.00
W	direct	+	1	0.00	0.00	0.00
S	indirect	-	42	31.43	26.19	28.57
T	indirect	-	60	42.31	91.67	57.89
S	indirect	+	62	41.18	22.58	29.17
T	indirect	+	91	56.08	91.21	69.46
T	free ind	-	3	42.86	100.0	60.00
T	free ind	+	27	81.82	100.0	90.00
S	rep	-	66	39.02	96.97	55.65
T	rep	-	23	28.85	65.22	40.00
W	rep	-	3	33.33	33.33	33.33
S	rep	+	67	34.74	98.51	51.36
T	rep	+	33	31.82	42.42	36.36
W	rep	+	5	50.00	20.00	28.57

Table 3: Results of the evaluation of the speech (S), thought (T) and writing (W) classifier on the test set. The count column (Co) represents those instances identified by the STWR classifier, which are further annotated with the classes speech, thought and writing.

References

Annelen Brunner, Lukas Weimer, Ngoc Duyen Tanja Tu, Stefan Engelberg, and Fotis Jannidis. 2019. Das Redewiedergabe-Korpus. Eine neue Ressource. In Patrick Sahle, editor, *Digital Humanities: multimedial multimodal. 6. Tagung des Verbands Digital Humanities im deutschsprachigen Raum e.V. (DHd 2019)*, pages 103–106.

Annelen Brunner. 2015. *Automatische Erkennung von Redewiedergabe: ein Beitrag zur quantitativen Narratologie*. Walter de Gruyter.

Annelen Brunner. 2019. Redewiedergabe–Schritte zur automatischen Erkennung. *Zeitschrift für germanistische Linguistik*, 47(1):216–248.

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.

Éric de la Clergerie, Benoît Sagot, Rosa Stern, Pascal Denis, Gaëlle Recourcé, and Victor Mignot. 2009. Extracting and visualizing quotations from news wires. In *Language and Technology Conference*, pages 522–532. Springer.

Dorrit Cohn. 1978. *Transparent minds: Narrative modes for presenting consciousness in fiction*. Princeton University Press.

David K Elson and Kathleen R McKeown. 2010. Automatic attribution of quoted speech in literary narrative. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*.

Cathrine Fabricius-Hansen. 2002. Nicht-direktes Referat im Deutschen – Typologie und Abgrenzungsprobleme. In Cathrine Fabricius-Hansen, Oddleif Leirbukt, and Ole Letnes, editors, *Modus, Modalverben, Modalpartikeln*, volume 25 of *Linguistisch-philologische Studien*, pages 6–29. Wissenschaftlicher Verlag Trier, Trier.

William Paulo Ducca Fernandes, Eduardo Motta, and Ruy Luiz Milidiú. 2011. Quotation extraction for

portuguese. In *Proceedings of the 8th Brazilian Symposium in Information and Human Language Technology*.

Gérard Genette. 1998. *Die Erzählung*. Wilhelm Fink Verlag, 2 edition.

Kevin Glass and Shaun Bangay. 2007. A naive salience-based method for speaker identification in fiction books. In *Proceedings of the 18th Annual Symposium of the Pattern Recognition Association of South Africa (PRASA '07)*, pages 1–6.

Verena Henrich and Erhard Hinrichs. 2010. Gernedit - the germanet editing tool. In *Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC 2010)*, pages 2228–2235, Valletta, Malta.

Berenike Herrmann and Gerhard Lauer. 2017. Kolimo. a corpus of literary modernism for comparative analysis. https://kolimo.uni-goettingen.de/about.html (2019-3-10).

Matthew Honnibal and Ines Montani. 2019. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*.

Fotis Jannidis. 2017. Netzwerke. In Fotis Jannidis, Hubertus Kohle, and Malte Rehbein, editors, *Digital Humanities*, pages 147–161. Springer.

Benedikt Jessing, Ralph Koehnen, and Horst Weber. 2007. *Einfuehrung in die Neuere deutsche Literaturwissenschaft*. Springer.

Eric Jones, Travis Oliphant, Pearu Peterson, et al. 2001. SciPy: Open source scientific tools for Python. http://www.scipy.org/.

Markus Konrad. 2019. Germalemma. https://github.com/WZBSocialScienceCenter/-germalemma.

Ralf Krestel, Sabine Bergler, and René Witte. 2008. Minding the source: Automatic tagging of reported speech in newspaper articles. *Reporter*, 1(5):4.

Guillaume Lemaître, Fernando Nogueira, and Christos K Aridas. 2017. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5.

Nuno Mamede and Pedro Chaleira. 2004. Character identification in children stories. In *International Conference on Natural Language Processing*, pages 82–90, Berlin, Heidelberg. Springer.

M. Martinez and M. Scheffel. 2012. *Einführung in die Erzähltheorie*. C.H.Beck, 9 edition.

Wes McKinney et al. 2010. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, volume 445, pages 51–56. SciPy Austin, TX.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*.

Andreas Müller. 2018. GermanWordEmbeddings. https://devmount.github.io/GermanWordEmbeddings/ (2018-06-20).

Silvia Pareti, Tim O'Keefe, Ioannis Konstas, James R Curran, and Irena Koprinska. 2013. Automatically detecting and attributing indirect quotations. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 989–999.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Michael Percillier. 2017. Creating and analyzing literary corpora. In Shalin Hai-Jew, editor, *Data Analytics in Digital Humanities*, pages 91–118. 05.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May. ELRA.

Luis Sarmento and Sérgio Nunes. 2009. Automatic extraction of quotes and topics from news feeds. In *DSIE'09-4th Doctoral Symposium on Informatics Engineering*.

Helmut Schmid and Florian Laws. 2008. Estimation of conditional probabilities with decision trees and an application to fine-grained pos tagging. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 777–784, Manchester, Great Britain. Association for Computational Linguistics.

Maria Paola Tenchini. 2010. Reporting gesture and voice in reporting speech: Co-verbal language in literature. *CI-DIT*, 2010(2).

Stéfan van der Walt, S Chris Colbert, and Gael Varoquaux. 2011. The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30.