Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 4923–4932
Marseille, 20-25 June 2022

Semantic Relations between Text Segments for Semantic Storytelling: Annotation Tool – Dataset – Evaluation

Michael Raring, Malte Ostendorff, Georg Rehm
DFKI GmbH, Alt-Moabit 91c, 10559 Berlin, Germany
michael.raring@gmail.com, {malte.ostendorff, georg.rehm}@dfki.de

Abstract

Semantic Storytelling describes the goal to automatically and semi-automatically generate stories based on extracted, processed, classified and annotated information from large content resources. Essential is the automated processing of text segments extracted from different content resources by identifying the relevance of a text segment to a topic and its semantic relation to other text segments. In this paper we present an approach to create an automatic classifier for semantic relations between extracted text segments from different news articles. We devise custom annotation guidelines based on various discourse structure theories and annotate a dataset of 2,501 sentence pairs extracted from 2,638 Wikinews articles. For the annotation, we developed a dedicated annotation tool. Based on the constructed dataset, we perform initial experiments with Transformer language models that are trained for the automatic classification of semantic relations. Our results with promising high accuracy scores suggest the validity and applicability of our approach for future Semantic Storytelling solutions.

Keywords: Semantic storytelling, Text classification, Discourse parsing, Wikinews

1. Introduction

The ever-increasing size of information available digitally, and especially the growth of unstructured information online, leads to the development of technologies to curate, process and understand digital data (Moreno Schneider et al., 2017). Due to its semi-structured nature, it is a challenge for the machine to process the information automatically. Content creators, scientists, and journalists write and publish articles and literary works based on researched information from a variety of (often digital) sources. They need domain-specific knowledge to verify content and facts, discuss different perspectives, and combine different sources. For journalists in particular, the ever-growing incoming stream of heterogeneous information, such as news articles, social media and press statements, is a major challenge. To better cope with this, knowledge workers rely on curation technologies to help them process, analyze, skim, sort, summarize, evaluate and present large amounts of digital content (Bois et al., 2017; Caselli and Vossen, 2017; van Meersbergen et al., 2017; Rehm et al., 2019; Rehm et al., 2021; Rehm et al., 2018; Linscheid et al., 2021).

The long-term Semantic Storytelling vision describes the automatic and semi-automatic generation of stories based on extracted, processed, classified and annotated information from large content resources (Rehm et al., 2019; Rehm et al., 2020b; Rehm et al., 2021). Storytelling can be understood as a technique to order a series of events in the world or to recognize meaningful patterns in natural language (Bruner, 1991). Hereby, Semantic Storytelling is one element of our long-term efforts to develop curation technologies as part of the QURATOR project (Rehm et al., 2020a). Our Semantic Storytelling approach is characterized by a stronger focus on extraction and presentation, in contrast to the much more established field Natural Language Generation (NLG) (Fan et al., 2018; Fan et al., 2019). Our goal is to support content curators in creating new storylines through the relevant information extracted and presented by a corresponding tool (Rehm et al., 2021). This enables knowledge workers to explore datasets fast, efficiently and intuitively (Rehm et al., 2020b). Essential for Semantic Storytelling is the automated processing of extracted text segments from different content resources or news streams. In order to process this information faster for users, the relevance of a text segment to a certain topic, the importance and the semantic relation to other text segments need to be estimated automatically (Rehm et al., 2020b).

For this purpose, we train a new classifier based on current machine learning approaches that can determine the semantic relation between extracted sentences from different news articles (Section 4). We use pre-trained language models based on the Transformer (Vaswani et al., 2017), BERT (Devlin et al., 2019) and DeBERTa (He et al., 2020) architecture. The most severe challenge for Semantic Storytelling is the lack of available datasets. There are only very few annotated corpora for cross-document semantic relations like CSTBank (Radev et al., 2004). For our vision of Semantic Storytelling, no such dataset exists, which is why we create our own corpus based on semantic relation classes derived from the discourse relation frameworks CST (Radev, 2000), PDTB (Prasad et al., 2008) and RST (Mann and Thompson, 1987) (Section 3.4). To support the annotation process, we develop a custom annotation tool tailored to our use case (Section 3.3). We make the dataset and the annotation tool publicly available.[1]

[1] https://github.com/DFKI-NLP/semantic-storytelling

2. Related Work

The study of semantic relations between text segments from narratives such as news articles is related to the analysis of discourse relations, which has mostly been considered in the context of text coherence.

This has led to established theories and frameworks such as Rhetorical Structure Theory (RST) (Mann and Thompson, 1987), which analyses the rhetorical structure of a text into hierarchical functional blocks that can be related to one another. For this purpose, 23 relation classes are defined as well as two unit types: the *nucleus* carries the main information, and the *satellite* adds supporting information. Carlson et al. (2003) present an RST corpus with 385 annotated Wall Street Journal articles from the Penn Treebank (Marcus et al., 1993). The largest corpus with annotated discourse relations, Penn Discourse Treebank (PDTB), includes Wall Street Journal articles with more than one million words (Miltsakaki et al., 2004; Prasad et al., 2008). Discourse relations are encoded using a theory-neutral lexical approach (Webber et al., 2003). Discourse relations are classified according to hierarchical senses, which are captured by marking explicit lexical items called discourse connectives. This approach makes it difficult to transfer the PDTB framework to our scenario, which is not *in-document* but *cross*-document. Rehm et al. (2020b) adopt this approach for classifying semantic relations between text segments from *different* documents and fine-tuned two DistilBERT language models in a siamese architecture on the PDTB2 corpus for the four top-level senses *temporal*, *contingency*, *comparison*, *expansion* (and *none*). The authors qualitatively evaluate the transfer to the cross-document scenario and emphasise the problem of dependency on lexical markers.

The functional cross-document structure theory (CST) (Radev, 2000) is related to RST. CST analyses rhetorical relations *between* thematically related *documents*. Rather than relying on a tree representation, cube and graph representations are used to illustrate connections within and between documents. Relationships are classified using 18 domain-independent, linguistically motivated relationship types, including comparative bidirectional relations such as *identity*, *equivalence*, *reader profile*, or *change of perspective*. Radev et al. (2004) created a corpus of document clusters manually annotated with CST relations from various news sources (e. g., BBC, CNN, MSNBC, ABC News, USA Today, FOX News, Penn Treebank). Zhang et al. (2003) first attempted to classify seven sentence-level CST relations across document boundaries from this corpus using a *booster* machine learning approach that combines multiple weak hypotheses (classifiers) into one strong hypothesis. Comparative lexical, syntactic, and semantic features were used to process the sentence pairs. The multi-label and multi-class classifier achieved an accuracy of 88% and a macro F1 score of 38%. Maziero and Pardo (2011) was able to improve the F1-score to 44% using a decision tree (J48) classifier on the Portuguese-language CSTNews corpus.[2] The authors used similar lexical and syntactic features and linked them to external knowledge resources. This was done by adding the number of possible common synonyms and named entities as features. Kumar et al. (2013) were able to increase the F1 score to 62% by creating an SVM classifier for only four CST relations, which was trained on a set of 477 sentence pairs from the CST Bank corpus and tested on 205 sentence pairs. Only lexical metrics were used as features, e. g., cosine similarity, word overlap and sentence length.

3. A Corpus of Semantic Relations between Text Segments

In the following, we first describe the annotation task (Section 3.1) as well as the relations and annotation guidelines developed specifically for this work (Section 3.2). Afterwards, Section 3.3 describes the annotation tool we developed and Section 3.4 gives a brief overview of the constructed dataset.

3.1. Annotation Task

For annotating discourse relations between text segments *within* a document, annotation guidelines, software, and annotated datasets exist, see, e. g., Miltsakaki et al. (2004). In the context of Semantic Storytelling, we are especially interested in semantic relations between extracted sentences from *different* documents, especially news articles. Our source documents are English-language Wikinews articles from 2004 to 2020. We select a large number of sentence pair candidates from articles of similar category such that there is a high probability of the sentences being semantically related. Here, semantic relatedness refers to any possible relation with respect to topic, discourse or any other similarity that allows a story to be created out of sentence pairs.

For selecting candidate pairs, we apply three different pre-matching strategies: random matching, cosine similarity, and next sentence prediction. With random matching any two sentences, which are not from the same article, are randomly sampled from the whole corpus. With cosine similarity, the sentences are first converted into vector representations and then candidates are selected from vector representations with a high cosine similarity. For next sentence prediction, we make use of BERT's language model objective (Devlin et al., 2019) that is the model being trained for predicting whether one sentence follows another sentence. We use the BERT model in the same fashion and select sentence pairs with a high prediction value as candidates. After automatically selecting candidate pairs, we use our dedicated annotation tool to manually annotate the semantic relations between the sentence pair candidates (both directions).

[2] http://nilc.icmc.usp.br/CSTNews/

3.2. Relations and Annotation Guidelines

We annotate relation labels for each sentence pair (both directions) using the following relation classes:

- **None** indicates no semantic relation from one text segment to the other.
- **Attribution** is a uni-directional connection that exists if segment A presents an attributed version of information in segment B, e. g. using "According to CNN". It is derived from the CST classes *attribution*, *citation* and *modality*. Example:

  (A) *According to a top Bush advisor, the President was alarmed at the news.*

  (B) *The President was alarmed to hear of his daughter's low grades.*

- **Causal** is a bi-directional relation that indicates that both segments are causally influenced, but not in a conditional way. Similar relation classes exist in PDTB (*Contingency.Cause*) and RST (*evidence*, *justify*, *solutionhood*, *(non)volitional cause*, *(non)volitional result*). Example:

  (A) *By 11:59 p.m. tonight, President Bush must order $16 billion of automatic, across-the-board cuts in government spending to comply with the Gramm-Rudman budget law.*

  (B) *The cuts are necessary because Congress and the administration have failed to reach agreement on a deficit-cutting bill.*

- **Conditional** is a uni-directional connection present if an unrealized situation in segment A leads to the situation described in segment B. The relation also exists in PDTB (*Contingency.Condition*) and RST (*condition*). Example:

  (A) *Call Jim Wright's office in downtown Fort Worth, Texas, these days.*

  (B) *The receptionist still answers the phone, "Speaker Wright's office".*

- **Contrast** is a bi-directional relation type that highlights conflicting information and important differences between segments regarding falsehood (first example), different aspects (second example) and different point of views (third example). It is included in CST (*contradiction*) and PDTB (*Comparison.Contrast*) and relates to the RST relations *antithesis* and *concession*. First example:

  (A) *There were 122 people on the downed plane.*

  (B) *126 people were aboard the plane.*

  Second example:

  (A) *After all, gold prices usually soar when inflation is high.*

  (B) *Utility stocks, on the other hand, thrive on disinflation.*

  Third example:

  (A) *Mr. Edelman said the decision "has nothing to do with Marty Ackerman".*

  (B) *Mr. Ackerman contended that it was a direct response to his efforts to gain control of Datapoint.*

- **Description** is a uni-directional relation that applies if segment B describes an entity from segment A. The relation is derived from CST and PDTB (*EntRel*). Example:

  (A) *Mr. Greenfield appeared in court yesterday.*

  (B) *Greenfield, a retired general and father of two, has declined to comment.*

- **Equivalence** is another bi-directional relation that indicates that both segments describe the same situation from different perspectives, including personal, political and other dimensions. It covers several relations from CST (*equivalence*, *reader profile*, *change of perspective*), PDTB (*Expansion.Equivalence*, *Comparison.Similarity*) and RST (*restatement*). Example:

  (A) *Chairman Krebs says the California pension fund is getting a bargain price that wouldn't have been offered to others.*

  (B) *In other words: The real estate has a higher value than the pending deal suggests.*

- **Fulfillment** is a uni-directional class that states that an event predicted in segment A is asserted in segment B. This relation is only present in CST. Example:

  (A) *Mr. Green will go to Austria Thursday.*

  (B) *After traveling to Austria Thursday, Mr. Green returned home to New York.*

- **Identity** is a bi-directional relation that indicates that both segments provide the same information. This relation is available in CST only. Example:

  (A) *Tony Blair was elected for a second term today.*

  (B) *Today, Tony Blair won the election and is preparing for a second term.*

- **Purpose** is a uni-directional connection from A to B if segment A presents an action that an *agent* undertakes with the purpose of the *goal* conveyed by segment B being achieved. This class is derived from PDTB (*Contingency.Purpose*) and RST (*enablement*, *motivation*, *purpose*). Example:

(A) *Skilled ringers use their wrists to advance or retard the next swing,*

(B) *so that one bell can swap places with another in the following change.*

- **Summary** is a uni-directional relation class applied if segment B summarizes segment A. This relation covers multiple classes in CST (*overlap*, *subsumption*, *summary*), PDTB (*Expansion.Instantiation*, *Expansion.Level-of-detail*, *Expansion.Manner*, *Expansion.Substitution*) and RST (*background*, *interpretation*, *summary*). Example:

  (A) *After a grueling first six games, the Mets came from behind tonight to take the Title.*

  (B) *The Mets won the Title in seven games.*

- **Temporal** can connect segments uni- or bi-directionally if one segment presents additional information which has happened after (first example) or at the same time (second example). Temporal classes are also present in CST (*follow-up*) and PDTB (*Temporal.Asynchronous*, *Temporal.Synchronous*). First Example:

  (A) *So far, no casualties from the quake have been confirmed.*

  (B) *102 casualties have been reported in the earthquake region.*

  Second Example:

  (A) *Then, in late-afternoon trading, hundred-thousand-share buy orders for UAL hit the market, including a 200,000-share order through Bear Stearns that seemed to spark UAL's late price surge.*

  (B) *Almost simultaneously, PaineWebber began a very visible buy program for dozens of stocks.*

The definitions and examples of these relation classes are taken from the original paper on RST (Mann and Thompson, 1987), a follow-up paper of the original publication of CST (Zhang et al., 2002) and version 3 of PDTB.[3] We grouped and filtered their relation classes in order to create an initial inventory of relations that we can experiment with towards the goal of further developing our Semantic Storytelling approach.

3.3. Annotation Tool and Data Format

The annotation process was performed using the open-source web-based annotation platform INCEpTION[4] (Klie et al., 2018), which supports the annotation of various NLP-related features and setting up multiple annotation layers, label classes, annotators, documents, knowledge resources and recommender systems. In order to be able to annotate relations of individual text segments from different source documents, a separate editor was developed using the modular platform. INCEpTION's internal data structure is based on Apache UIMA[5] (Unstructured Information Management Architecture) framework and the associated Common Analysis System (CAS) data structure (Ferrucci et al., 2008). The data structure allows the annotation model to be represented by merging the extracted text segments into one document, adding the source documents as metadata to each segment, and using placeholder relations between them to represent pre-matching. We preprocessed the data in Python using the open source library DKPro cassis[6] (Eckart de Castilho and Gurevych, 2014) and imported the resulting UIMA CAS XMI file in INCEpTION. After the annotation process, we exported the project in the same format for further processing.

Figure 1 shows the user interface of our relation editor. It allows the annotator to quickly assess the text segments, retrieve their source documents for reference, and select directed relations using a drop-down menu. The annotator can navigate through the sentence pairs either in a pair-wise fashion or individually for each segment, to ensure an efficient annotation process. The navigation bar shows additional information about the progress and distribution of label classes in the dataset.

3.4. Overview of the Dataset

We annotated a total of 2,501 sentence pairs from 2,638 different articles with semantic relations. The sentences result in a corpus of 291,146 words (376,002 words if we add the titles of the original articles). In terms of pre-matching (Section 3.1), 616 sentence pairs were matched randomly, 529 using BERT's NSP model, and 1,356 using cosine similarity. Figure 2 shows the final distribution of labels in the dataset. It should be noted that each sentence pair includes *two* relations (both directions) and, thus, two labels. However, this is an imbalanced classification problem. The *none* relation (no semantic relation) is by far the most represented with 63.4% (3,172 cases). It is followed by the *temporal* with 11.8% (590 cases), *causal* with 8.7% (434 cases) and *equivalence* with 6.8% (340 cases) and *contrast* with 3.3% (165 cases). The seven other relation labels account for less than 2% each.

Using our annotation tool, we were able to annotate between 10 and 30 sentence pairs per hour, depending on the complexity of the articles and the pre-matching algorithm on which the chances of potential semantic links depended. The annotation of all 2,501 sentence pairs by one annotator took about 150 hours. The annotation process was performed by one of the authors, so the annotation guidelines explained in 3.2 were suf-

[3] https://catalog.ldc.upenn.edu/docs/LDC2019T05/PDTB3-Annotation-Manual.pdf

[4] https://inception-project.github.io

[5] https://uima.apache.org

[6] https://github.com/dkpro/dkpro-cassis

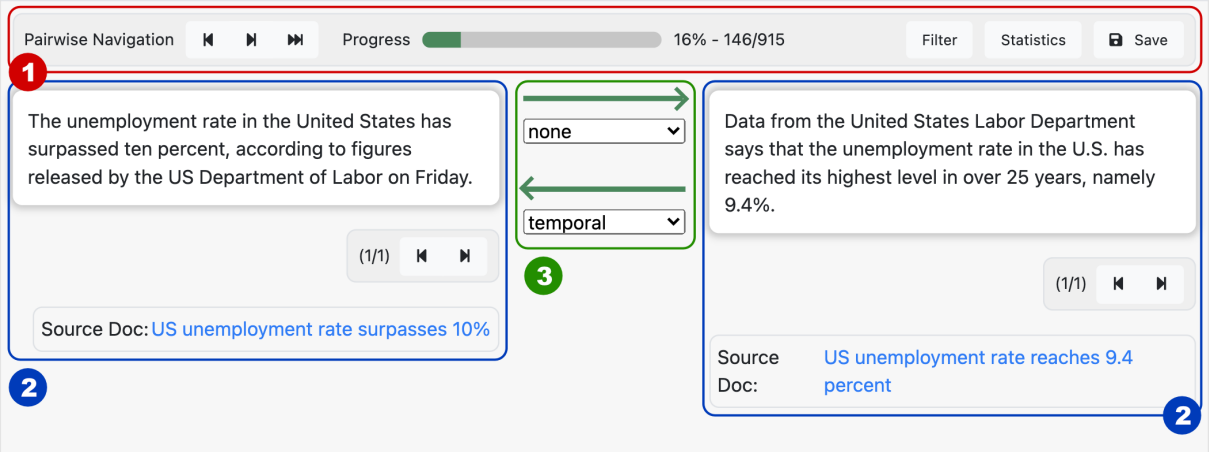

Figure 1: User interface of the annotation tool, developed using INCEpTION, with navigation bar (1), segment items with meta data (2), and the relation label selector (3).

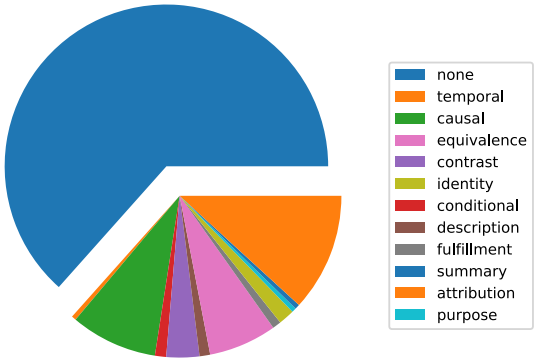

Figure 2: Label distribution in the annotated dataset

ficient and an inter-annotator agreement could be omitted.

4. Experiments

In order to train a multi-class classifier for semantic relations between sentence pairs, it is necessary to exploit the annotated dataset in the best possible way. We utilise pre-trained language models and fine-tune them for the classification task on the corpus. As an input sequence, we investigate different input strategies (Table 1) and evaluate whether a language model can benefit from adding more metadata such as the title of the article (TS, TSD, STD) or publication date (TSD, STD) in the input to be classified and whether the order of title and extracted sentence have an impact on the focus of the language model (TSD or STD).

For training and testing, we apply 4-fold cross validation. The training process involves four runs in which the validation dataset consists of one alternating fold and the training data consists of the remaining three folds. To evaluate the results, we consider the mean and standard deviation of the four folds. Using scikit-learn[7] (Buitinck et al., 2013), we employ a stratified k-fold method that produces equally distributed folds regarding the relation labels. Thus, low standard deviations can be expected.

To assess the impact of the imbalance in the distribution of label classes on the results and to ensure better comparability of the results with other experiments that also have a smaller number of classes, we also run all experiments with an adjusted relation class inven-

	Input	Example
S	Sentence.	However preliminary results based on 95% of the votes cast give Hamas' Change and Reform Party 76 seats, leaving Fatah with 43 seats.
TS	Title. Sentence.	Hamas wins Palestinian election. However preliminary results based on 95% of the votes cast give Hamas' Change and Reform Party 76 seats, leaving Fatah with 43 seats.
TSD	Title. Sentence. Date.	Hamas wins Palestinian election. However preliminary results based on 95% of the votes cast give Hamas' Change and Reform Party 76 seats, leaving Fatah with 43 seats. January 29, 2005
STD	Sentence. Title. Date.	However preliminary results based on 95% of the votes cast give Hamas' Change and Reform Party 76 seats, leaving Fatah with 43 seats. Hamas wins Palestinian election. January 29, 2005

Table 1: Tested input strategies for elements of the input pair of the classifier (Sentence: the extracted sentence; Title: the title of the article; Date: the publication date of the article).

[7] https://scikit-learn.org

tory with a total of seven classes. For this purpose, we group the six least represented classes *conditional* (56 samples), *description* (52), *fulfillment* (44), *attribution* (24), *summary* (24), and *purpose* (24) to *others* (sum of 224 samples). With this adjustment we expect a strong improvement in the macro class observations.

4.1. Systems

BERT (Devlin et al., 2019, Bidirectional Encoder Representations from Transformers) is a pre-trained language model based on the architecture of the multi-layer encoder of a Transformer by Vaswani et al. (2017). BERT achieved the state of the art in the GLUE benchmark (Wang et al., 2018) and in eleven sentence-level and token-level NLP tasks including Multi-Genre Natural Language Inference (Williams et al., 2018, p. 1), a classification task with similar properties to classifying semantic relations between text segments. The BERT architecture is very well suited for our experiments due to its pre-training with two text segments as our prior work has shown (Ostendorff et al., 2020a; Ostendorff et al., 2020b). Hence, we expect to achieve a high level of text comprehension by the classifier for each sentence by simultaneously processing the left and right context of a token in the input sequence in each layer (bi-directionality), enabled by the pre-training strategy of Masked Language Modeling. In addition, we expect the BERT model to infer (semantic) relations between a sentence pair through the pre-training training strategy of Next Sentence Prediction (NSP), which predicts if the second input is the successor of the first in the source document.

DeBERTa (He et al., 2020, Decoding-enhanced BERT with disentangled attention) is an optimization of the BERT architecture processing the content and position embeddings of the input as separate vectors in the attention mechanism (disentangled attention) and using absolute token positions in the last layer of the decoder (Enhanced Mask Decoder). The authors argue that this enables DeBERTa to better identify important tokens in the text segment and understand syntactical nuances like subject and object contexts. We hypothesise that the understanding of news excerpts and the identification of semantic relations benefit from this.

Implementation Details We perform and evaluate the experiments with the pre-trained cased and uncased $\text{BERT}_{\text{base}}$ models with approx. 110 million parameters, the cased and uncased $\text{BERT}_{\text{large}}$ models with approx. 335 million parameters, and the $\text{DeBERTa}_{\text{base}}$ model with 140 million parameters. For all experiments, we use a batch size of 8, a learning rate of 0.00002, and a weight decay of 0.01. We found that the best accuracy was achieved at 10 training epochs. Two NVIDIA Quadro RTX 6000 GPUs with 24GB memory each are used. The fine-tuning of the four folds conducted with Hugging Face's Transformers library[8] with GPU support by the PyTorch[9] backend took, for the $\text{BERT}_{\text{large}}$ cased model with ten training epochs and the STD input strategy, up to 2h 52min under these conditions. The same experiment with the $\text{BERT}_{\text{base}}$ cased model took only 1h 11min. The main metric for the evaluation in every training step and for evaluating the best hyperparameters (language model, input strategy, number of label classes) is accuracy. For a deeper understanding of the results and a better comparison to similar experiments, we also consider the macro and micro averages of precision, recall and F1-score.

Relation	Prec.	Rec.	F1	Support
none	**87.0**	**86.0**	**86.5**	794
identity	78.3	77.5	77.6	20
equivalence	78.5	71.9	74.6	85
causal	57.3	67.0	61.6	109
contrast	55.0	66.1	59.9	41
temporal	45.9	50.3	48.0	147
conditional	24.1	21.4	21.5	14
description	26.5	13.1	17.4	13
attribution	20.8	8.3	11.8	6
fulfillment	0.0	0.0	0.0	11
summary	0.0	0.0	0.0	6
purpose	0.0	0.0	0.0	6
Micro avg.	74.6	75.0	74.6	1251
Macro avg.	39.5	38.5	38.2	1251

Table 2: Training results of the best classifier ($\text{DeBERTa}_{\text{base}}$ with TS input strategy) for all classes sorted by F1-score (average of four folds and rounded support)

4.2. Results

For the 12 relations the best classifier is based on $\text{DeBERTa}_{\text{base}}$ using the TS input strategy. It has an average accuracy of four folds of 75% with a standard deviation of 1.74 percentage points, which is in the upper middle range compared to the other classifiers. The micro precision, recall and F1-score are also the best results in the classification of the 12 label classes with about 75%. As shown in Table 4, the macro averages of precision, recall and F1-score are rather low with 39% and 38%, which is due to the high number of classes and their uneven distribution in the data set. Table 2 shows that especially the underrepresented unidirectional classes *conditional*, *description*, *attribution*, *fulfillment*, *summary* and *purpose* have low F1-scores. In the experiment with the subset of seven relations, these labels are grouped together in the class *others*. In this setting, the best classifier is also based on the $\text{DeBERTa}_{\text{base}}$ model using the STD input strategy. Table 3 shows the difference in metrics for each class. Except for *equivalence*, all classes benefit in terms of F1-score. The new class *others* performs the worst, but

[8] https://huggingface.co/transformers/

[9] https://pytorch.org

Class	Prec.	Rec.	F1	Support
none	+0.8	+1.8	+1.3	0
identity	+11.8	-1.3	+4.6	0
equivalence	-4.3	-3.5	-3.6	0
contrast	+11.0	-3.6	+4.2	0
causal	+2.6	+0.7	+1.8	0
temporal	+4.6	+3.2	+3.9	0
*others**	+41.6	+33.5	+36.9	+55
Micro avg.	+2.7	+2.2	+2.6	0
Macro avg.	+27.7	+25.7	+27.11	0

Table 3: Difference in training results of best classifier (DeBERTa$_{base}$ with STD input strategy) for the subset of seven classes compared to the full classification (rounded, * is the added class *others*).

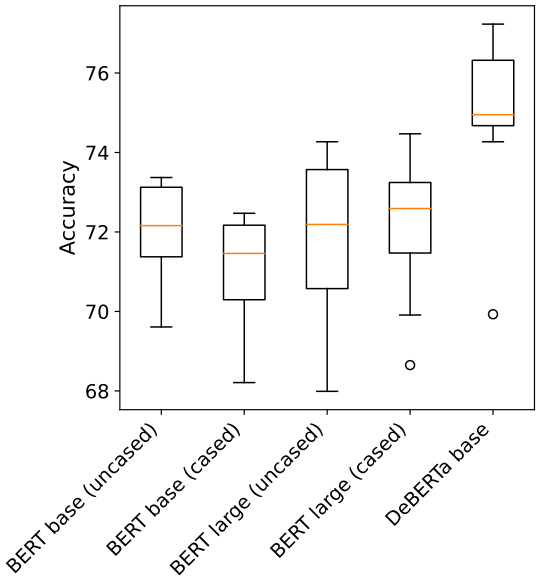

Figure 3: Distribution of accuracy of all fine-tuned classifiers depending on Transformer models

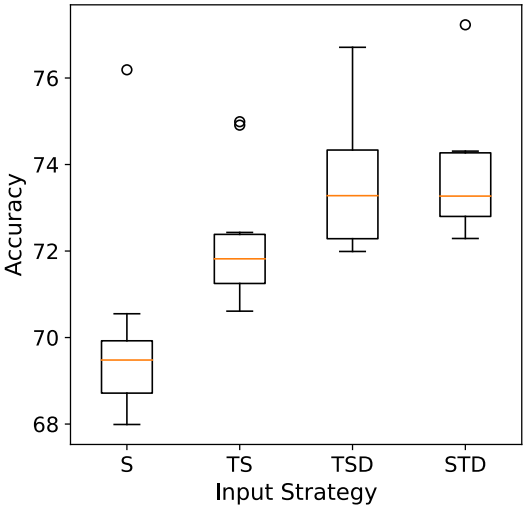

Figure 4: Distribution of accuracy of all fine-tuned classifiers depending on input strategy

far better than any replaced class. The macro averages of precision, recall, and F1-score benefit from the reduction of class labels, with gains of over 26 percentage points. The micro averages also improve, just like the general accuracy from 75% to 77%.

Compared to the experiments of Zhang et al. (2003), Maziero and Pardo (2011) and Kumar et al. (2013) with five to seven CST relation classes, our classifier with seven of our own relation classes achieves new best scores in macro precision and F1-score and a similarly high macro recall as in the experiment by Kumar et al. (2013). However, the accuracy of the multi-label classifier of Zhang et al. (2003) could not be improved upon. This classifier tested whether the searched label was present among the top ranked labels. Thus, it had more chances for a correct classification, which is why the accuracy is only partially comparable.

4.3. Analysis

We performed all experiments with different hyperparameters regarding the choice of language model and input strategy: the DeBERTa architecture is superior to BERT in the classification task, both on average (Figure 3) and in the best classifiers. DeBERTa$_{base}$ outperforming the other Transformer models is consistent with the findings by (He et al., 2020) and due to the deeper syntactical understanding of the input by disentangled attention (separating content and position embeddings) and the Enhanced Mask Decoder which includes absolute positions for tokens. Furthermore, it can be observed that the uncased BERT$_{base}$ model performs better than the cased one. The expansion of the vocabulary in the cased version seems to be counterproductive here and to interfere with the deeper understanding of the input. In the BERT$_{large}$ model variants, on the other hand, this observation cannot be made. The larger models seem to be able to process and generalize the larger vocabulary better due to the multitude of parameters. Except in the cased version, in general no strong improvement of accuracy is observed by the multitude of parameters in the BERT$_{large}$ models, which means that the extra computational effort is not always justified. Only the best classifiers can improve. The BERT$_{large}$ models also appear comparatively unstable, with high standard deviation of accuracy within the four folds and a wide range in Figure 3. However, on average, the DeBERTa$_{base}$ models are subject to larger variations within the four folds, too. This seems to be due to the disproportion between the high number of parameters and comparatively low amount of training data.

The evaluation of the individual input strategies is not

Model	Cl. Fr.	# Cl.	Datas.	Acc.	Precision		Recall		F1-score	
					Macro	Micro	Macro	Micro	Macro	Micro
$DeBERTa_{base}$ (I.S. TS)	Custom	12	2501	75	39	75	38	75	38	75
$DeBERTa_{base}$ (I.S. STD)	Custom	7	2501	77	**67**	77	**64**	77	**65**	77
Related Experiments										
Zhang et al. (2003)	CST	7	3942	**82**	47	–	33	–	38	–
Maziero and Pardo (2011)	CST	7	*1511	–	44	–	44	–	44	–
Kumar et al. (2013)	CST	5	682	–	66	–	**64**	–	62	–

Table 4: Comparison of full-classification and subset-classification results to related experiments (Cl. Fr.: Class framework, # Cl.: Number of classes, Datas.: Number of sentence pairs, *: Sum of class support, Acc.: Accuracy, I.S.: Input Strategy)

entirely clear. The distribution of the accuracy of all classifiers with respect to the input strategy (Figure 4) shows a tendency that both the addition of the article title (TS versus T) and appending the publication date (TSD and STD versus TS) significantly increases the average accuracy. However, there are also outliers for the input strategies T and TS. One of these is our best classifier for the 12 relation classes. One reason could be that some relations in this setup do not benefit (as much) from adding the publication date and the attention mechanism of DeBERTa might be disturbed by the formatted date at the end of the input since the other models benefit more from the publication date. Although there are more uni-directional classes among the 12 relation classes than in the reduced set of classes, they are not as dependent on the order of publication as the *temporal* relation. In general, adding the article title seems to be useful and also the publication date, if the uni-directional relations can benefit from it or when a classic BERT architecture is used.

5. Conclusion

Our experiments yielded promising results in the classification of cross-document semantic relations between text segments using pre-trained language models. The performance of the $DeBERTa_{base}$ model for the subset of seven classes surpassed previous results by CST-based experiments with a similar setting. The results show that the use of pre-trained language models is suitable for the task and yet leave room for improvement. It turned out that an annotation model must fit the dataset and should not contain too many underrepresented classes. In the future, our annotation guidelines with 12 relation classes must be adapted to this finding, underrepresented classes must be questioned and the individual classes must be made more distinct. Using a custom annotation model is a surmountable challenge in this regard. The process of constructing a sufficiently large and qualitative dataset was made possible using the tools we developed. The process could be improved in the future by further adjustments to the editor, such as displaying more metadata like the article's publication date or an article preview that would, in many cases, avoid opening external links. Also, topic-based source article selection and preprocessing of articles and extracted sentences using Named Entity Recognition, Event Detection and Coreference Resolution could enable higher quality pre-matching and support the annotators in their work. As a pre-matching algorithm, cosine similarity produced the best results in our experiment. For future experiments, the DeBERTa architecture and the use of different input string strategies adding more metadata have proven particularly useful.

6. Acknowledgements

The research presented in this article is partially funded by the German Federal Ministry of Education and Research (BMBF) through the projects QURATOR (Unternehmen Region, Wachstumskern, no. 03WKDA1A) and PANQURA (no. 03COV03E).

7. Bibliographical References

Bois, R., Gravier, G., Jamet, E., Robert, M., Morin, E., Sébillot, P., and Robert, M. (2017). Language-based construction of explorable news graphs for journalists. In *Proceedings of the 2017 EMNLP Workshop on Natural Language Processing meets Journalism*, pages 31–36. ACL.

Bruner, J. (1991). The narrative construction of reality. *Critical inquiry*, 18(1):1–21.

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., and Varoquaux, G. (2013). API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122.

Carlson, L., Marcu, D., and Okurowski, M. E. (2003). Building a discourse-tagged corpus in the framework of rhetorical structure theory. In *Current and new directions in discourse and dialogue*, pages 85–112. Springer.

Caselli, T. and Vossen, P. (2017). The event storyline corpus: A new benchmark for causal and temporal relation extraction. In *Proceedings of the Events and Stories in the News Workshop*, pages 77–86. ACL.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding.

Eckart de Castilho, R. and Gurevych, I. (2014). A broad-coverage collection of portable NLP components for building shareable analysis pipelines. In *Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT*, pages 1–11, Dublin, Ireland, August. Association for Computational Linguistics and Dublin City University.

Fan, A., Lewis, M., and Dauphin, Y. (2018). Hierarchical neural story generation. In *Proc. of the 56th Annual Meeting of the Assoc. for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898.

Fan, A., Lewis, M., and Dauphin, Y. (2019). Strategies for structuring story generation. arXiv preprint arXiv:1902.01109.

Ferrucci, D., Lally, A., Verspoor, K., and Nyberg, E., (2008). *Unstructured Information Management Architecture (UIMA) Version 1.0*. Oasis.

He, P., Liu, X., Gao, J., and Chen, W. (2020). DeBERTa: Decoding-enhanced BERT with disentangled attention.

Klie, J.-C., Bugert, M., Boullosa, B., de Castilho, R. E., and Gurevych, I. (2018). The inception platform: Machine-assisted and knowledge-oriented interactive annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9. Association for Computational Linguistics, 06.

Kumar, Y. J., Salim, N., Osman, A. H., and Abuobieda, A. (2013). Using SVMs for classification of cross-document relationships. *Editorial Board*, pages 239–246.

Linscheid, P., Bourgonje, P., and Rehm, G. (2021). Parsing Discourse Structures for Semantic Storytelling: Evaluating an efficient RST Parser. In Adrian Paschke, et al., editors, *Proceedings of QURATOR 2021 – Conference on Digital Curation Technologies*, Berlin, Germany, 02. CEUR Workshop Proceedings, Volume 2836. 11/12 February 2021.

Mann, W. C. and Thompson, S. A. (1987). *Rhetorical structure theory: A theory of text organization*. University of Southern California, Information Sciences Institute Los Angeles.

Marcus, M., Santorini, B., and Marcinkiewicz, M. A. (1993). Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2).

Maziero, E. G. and Pardo, T. A. S. (2011). Multi-document discourse parsing using traditional and hierarchical machine learning. In *Proceedings of the 8th Brazilian Symposium in Information and Human Language Technology*.

Miltsakaki, E., Prasad, R., Joshi, A. K., and Webber, B. L. (2004). The penn discourse treebank. In *LREC*. Citeseer.

Moreno Schneider, J., Srivastava, A., Bourgonje, P., Wabnitz, D., and Rehm, G. (2017). Semantic storytelling, cross-lingual event detection and other semantic services for a newsroom content curation dashboard. In *Proceedings of the 2017 EMNLP Workshop: Natural Language Processing meets Journalism*, pages 68–73. ACL.

Ostendorff, M., Ruas, T., Blume, T., Gipp, B., and Rehm, G. (2020a). Aspect-based document similarity for research papers. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6194–6206, Barcelona, Spain (Online), December. International Committee on Computational Linguistics.

Ostendorff, M., Ruas, T., Schubotz, M., and Gipp, B. (2020b). Pairwise multi-class document classification for semantic relations between wikipedia articles. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, Aug.

Prasad, R., Dinesh, N., Lee, A., Miltsakaki, E., Robaldo, L., Joshi, A. K., and Webber, B. L. (2008). The penn discourse treebank 2.0. In *LREC*. Citeseer.

Radev, D. R., Otterbacher, J., and Zhang, Z. (2004). CST bank: A corpus for the study of cross-document structural relationships. In *LREC*.

Radev, D. (2000). A common theory of information fusion from multiple text sources step one: cross-document structure. In *1st SIGdial workshop on Discourse and dialogue*, pages 74–83.

Rehm, G., Schneider, J. M., Bourgonje, P., Srivastava, A., Fricke, R., Thomsen, J., He, J., Quantz, J., Berger, A., König, L., Räuchle, S., Gerth, J., and Wabnitz, D. (2018). Different Types of Automated and Semi-Automated Semantic Storytelling: Curation Technologies for Different Sectors. In Georg Rehm et al., editors, *Language Technologies for the Challenges of the Digital Age: 27th International Conference, GSCL 2017, Berlin, Germany, September 13-14, 2017, Proceedings*, number 10713 in Lecture Notes in Artificial Intelligence (LNAI), pages 232–247, Cham, Switzerland, 1. Gesellschaft für Sprachtechnologie und Computerlinguistik e.V., Springer. 13/14 September 2017.

Rehm, G., Zaczynska, K., and Moreno-Schneider, J. (2019). Semantic storytelling: Towards identifying storylines in large amounts of text content. In *Proceedings of Text2Story – Second Workshop on Narrative Extraction From Texts co-located with 41th European Conference on Information Retrieval (ECIR 2019)*, pages 63–70.

Rehm, G., Bourgonje, P., Hegele, S., Kintzel, F., Schneider, J. M., Ostendorff, M., Zaczynska, K., Berger, A., Grill, S., Räuchle, S., Rauenbusch, J., Rutenburg, L., Schmidt, A., Wild, M., Hoffmann, H., Fink, J., Schulz, S., Seva, J., Quantz, J., Böttger, J., Matthey, J., Fricke, R., Thomsen, J., Paschke, A.,

Qundus, J. A., Hoppe, T., Karam, N., Weichhardt, F., Fillies, C., Neudecker, C., Gerber, M., Labusch, K., Rezanezhad, V., Schaefer, R., Zellhöfer, D., Siewert, D., Bunk, P., Pintscher, L., Aleynikova, E., and Heine, F. (2020a). QURATOR: Innovative Technologies for Content and Data Curation. In Adrian Paschke, et al., editors, *Proceedings of QURATOR 2020 – The conference for intelligent content solutions*, Berlin, Germany, 02. CEUR Workshop Proceedings, Volume 2535. 20/21 January 2020.

Rehm, G., Zaczynska, K., Moreno-Schneider, J., Ostendorff, M., Bourgonje, P., Berger, M., Rauenbusch, J., Schmidt, A., and Wild, M. (2020b). Towards discourse parsing-inspired semantic storytelling.

Rehm, G., Zaczynska, K., Bourgonje, P., Ostendorff, M., Moreno-Schneider, J., Berger, M., Rauenbusch, J., Schmidt, A., Wild, M., Böttger, J., Quantz, J., Thomsen, J., and Fricke, R. (2021). Semantic Storytelling: From Experiments and Prototypes to a Technical Solution. In Tommaso Caselli, et al., editors, *Computational Analysis of Storylines: Making Sense of Events*, Studies in Natural Language Processing, pages 240–259. Cambridge University Press, Cambridge, November.

van Meersbergen, M., Vossen, P., van der Zwaan, J., Fokkens, A., van Hage, W., Leemans, I., and Maks, I. (2017). Storyteller: Visual analytics of perspectives on rich text interpretations. In *Proceedings of the 2017 EMNLP Workshop: Natural Language Processing meets Journalism*, pages 37–45, Copenhagen, Denmark, September. Association for Computational Linguistics.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. (2018). GLUE: A multi-task benchmark and analysis platform for natural language understanding.

Webber, B., Stone, M., Joshi, A., and Knott, A. (2003). Anaphora and discourse structure. *Computational linguistics*, 29(4):545–587.

Williams, A., Nangia, N., and Bowman, S. R. (2018). A broad-coverage challenge corpus for sentence understanding through inference.

Zhang, Z., Blair-Goldensohn, S., and Radev, D. R. (2002). Towards CST-enhanced summarization. In *Aaai/Iaai*, pages 439–446.

Zhang, Z., Otterbacher, J., and Radev, D. (2003). Learning cross-document structural relationships using boosting. In *Proceedings of the twelfth international conference on Information and knowledge management*, pages 124–130.