FRI Natural language processing course 2022/2023
Final report

University of Ljubljana
Faculty of Computer and Information Science

Literacy situation models knowledge base creation

Jaša Frelih Črv, Jurij Kolenik, and Žan Jonke

Abstract

The paper strives to build a knowledge base based on situation models from selected English short stories. We focused on exploring a subset of literary analysis, focusing on fictional character analysis. A simple pipeline is proposed which consists of character extraction, sentiment analysis of characters and protagonist and antagonist detection. These methods are applied and evaluated on newly annotated corpus of fables.

Keywords

Story entities, Family relationship extraction, Fables, Sentiment analysis, Named entity recognition

Advisors: Slavko Žitnik

Introduction

Every story we read contains a lot of information that can be extracted using literacy situation models. We can extract different characters, how they are connected between each other and the general sentiment about them. We can go even further by looking at events in the story and analyzing the space and time around them. By analyzing the sentiments, co-occurrence, and importance of characters in a story, we can quickly grasp the content and meaning of the story as a whole. This is especially useful for people who may not have a lot of time to read or analyze lengthy texts. This project aims to develop a comprehensive framework for character analysis and sentiment analysis in storytelling. By employing literacy situation models, the project seeks to extract essential information from stories, including characters and the overall sentiment surrounding them. We propose a pipeline for building the literacy situation model. Firstly, named entity recognition (NER) was utilized to extract characters from the text, with the option of incorporating coreference resolution (COREF) techniques. Subsequently, character sentiment analysis was conducted to determine the moral orientation of each character. Additionally, the project aimed to identify protagonists and antagonists based on their sentiment scores and their entity importance scores. Our implementation of methods and annotated corpus is available on our Github [1].

1. Existing Solutions & Related work

The problem of automatic sentiment analysis (SA) is a growing research topic. Although SA is an important area and already has a wide range of applications, it clearly is not a straightforward task and has many challenges related to natural language processing (NLP) [2]. The simple idea behind computing an emotional figure profile is that the strength of semantic associations between a character (name) and the prototypical “emotion words” contained in the label list gives us an estimate of their emotion profile [3]. In article [4] character sentiment analysis is conducted using a simple word-list based approach where the “emotion” of characters such as “Miss Elizabeth Bennet” from Jane Austen’s Pride and Prejudice by counting all emotional words in paragraphs that featured only one character and adding them to the character’s total. Further building on top of this, [5] exploited text structure and knowledge-based SA to track the emotional trajectories of interpersonal relationships rather than of a whole text or an isolated character. To extract these relationships, they mined for character-to-character sentiment by summing the valence values (provided by the AFINN sentiment lexicon [6]) over each instance of continuous speech and then assumed that sentiment was directed towards the character that spoke immediately before the current speaker. This assumption however does not always hold, as a conversation between two characters might be centered around expressing sentiment about someone offstage.

In the article [7], the authors examined the network of named entity co-occurrences in written texts, which coincides directly with our proposed work on our project. They pointed out that the use of methods borrowed from statistics and physics to analyze written texts has allowed the discovery of unprecedented patterns of human behavior and cognition by establishing links between models features and language structure. While current models have been useful to unveil patterns via analysis of syntactical and semantic networks, only a few works have probed the relevance of investigating the structure arising from the relationship between relevant entities such as characters, locations and organizations. They focused on the representation of entities appearing in the same context as a co-occurrence network, where links are established according to a null model based on random, shuffled texts in their paperwork. With some simulations performed in novels, they discovered that their proposed model displayed interesting topological features, such as the small world feature, characterized by high values of clustering coefficient. The effectiveness of their model was later also verified in a practical pattern recognition task in real networks. When they compared their own model with traditional word adjacency networks, it displayed optimized results in identifying unknown references in texts. They concluded that the proposed representation plays a complementary role in characterizing unstructured documents via topological analysis of named entities. They finally thought that their method could be used to improve the characterization of written texts (and related systems), especially if it was combined with traditional approaches based on statistical and deeper paradigms.

The next article [8] focused on named Entity Recognition. Named Entity Recognition serves as the basis for many other areas in Information Management. However, it is unclear what the meaning of Named Entity is, and yet there is a general belief that Named Entity Recognition is a solved task. The authors of this paper we analyzed the evolution of the field from a theoretical and practical point of view. They argued that the task is actually far from solved, and also presented the consequences for the development and evaluation of tools. They also discussed topics for further research with the goal of bringing the task back to the research scenario. We also believe that some ideas that the authors established in their work could be beneficial for our research work.

The article [9] focuses on data-driven relation discovery from unstructured texts. The authors proposed a data driven methodology for the extraction of subject-verb-object triplets from a text corpus. Previous works on the field solved the problem by means of complex learning algorithms requiring hand-crafted examples. They stated that their proposal completely avoids learning triplets from a data-set, and they built it on top of a well-known baseline algorithm. The baseline algorithm uses only syntactic information for generating triplets and is characterized by a very low precision, meaning that very few triplets are meaningful. Their idea is to integrate the semantics of the words with the aim of filtering out the wrong triplets, thus increasing the overall precision of the system. The algorithm has been thoroughly tested and has shown good performance with respect to the baseline algorithm for triplet extraction. We also believe some ideas presented in this paper to be useful during our own research work.

In [10] the authors focused on automatic Extraction of Causal Relations from Text using Linguistically Informed Deep Neural Networks. The author proposed a linguistically informed recursive neural network architecture for automatic extraction of cause-effect relations from text. These relations can be expressed in arbitrarily complex ways. The architecture they used relies on word level embedding and other linguistic features to detect causal events and their effects mentioned within a sentence. They used the extracted events and their relations to build a causal-graph after clustering and appropriate generalization, which is then used for predictive purposes. They have evaluated the performance of the proposed extraction model with respect to two baseline systems. Furthermore, they also compared their final results with related work reported in the past by other authors on *SEMEVAL* data set, and found that their proposed model works better.

Methods

1.1 Corpus

Our corpus consists of 148 short Aesop fables stories, which is included in our repository together with annotations in directory `/data/annotations` in file `AesopFablesCharacterSentiment.json`. We added annotations to all the stories, which include the characters that appear in each individual story and the sentiment of each character. In order to keep the same pattern of sentiment annotations through the stories, we used the following rules. Each character was classified into one of the following classes: negative, neutral or positive. For a character to have been classified as negative, he had to exhibit or commit any set of the following behaviors: lie, deceive, steal, be hateful or envious or strive to worsen the situation. Additionally, since all of our stories are fables and contain interactions between animals, which more often exhibit perceived acts of violence, we did not annotate such characters necessarily as negative. Such a character would need to have had a motive which could have been derived from one of the aforementioned principles. In opposition, a positive character had to exhibit or commit the following behaviors: be kind, be sacrificial, help others, strive to improve the situation. If a character did not commit or exhibit any of the aforementioned behaviors, he was classified as neutral.

We acknowledge that applying such set of rules might be ambiguous, a subject of interpretation or even controversial. To mitigate this issue, we cross-checked all the annotations and resolved all conflicting annotations through majority vote. Below, we can see an example of a story with the added annotations of characters and character sentiments. An example of a story and its associated annotation can be seen below.


```
{
     "number": "10",
     "title": "The Kid and the Wolf",
     "story":[
         "A frisky young Kid had been left by the
              herdsman on the thatched roof of a
              sheep shelter to keep him out of harm's
               way.",
         "The Kid was browsing near the edge of the
              roof, when he spied a Wolf and began to
               jeer at him, making faces and abusing
              him to his heart's content.",
         "I hear you, said the Wolf, and I haven't
              the least grudge against you for what
              you say or do.",
         "When you are up there it is the roof that's
               talking, not you."
     ],
     "moral": "Do not say anything at any time that
         you would not say at all times.",
     "characters": [
       "The young Kid", "The herdsman", "The Wolf"
     ],
     "character_sentiment": {
       "The young Kid": -1,
       "The herdsman": 0,
       "The Wolf": 1
     }
 }
```


Additionally, figures 1 and 2 show some additional information about our corpus.

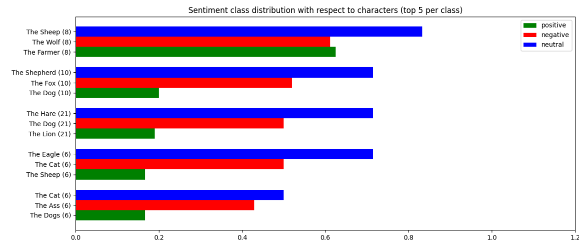

Figure 1. Sentiment class distribution with respect to characters. Our stories are fables, where the same characters occur in different stories, taking on different roles. Here we present the top 5 most commonly appearing characters with respect to sentiment class. This plot shows that the Wolf, which appeared in 8 different stories, was annotated as a negative character in 62.5% of the stories.

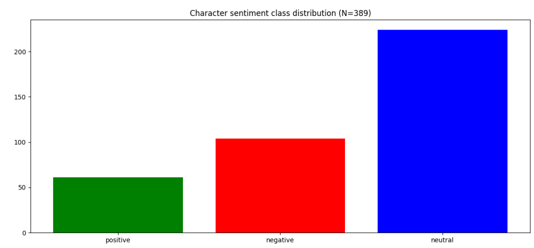

Figure 2. Sentiment class distribution.

1.1.1 Corpus Analysis

We firstly roughly analyzed our corpus and calculated some statistical and other parameters that we thought would be useful for better understanding of what kind of stories we were dealing with. We calculated some of the statistical data by writing the methods ourselves, furthermore we used the *nltk* library to further help us visualize the generated statistical data. Firstly we calculated some of the most common statistics we could think of, the results can be seen in the table 1 below.

Average letters per word	5.111
Average words per sentence	15.447
Average sentences per story	10.612
Characters count	123174
Words count	24098
Sentences count	1560

Table 1. Statistical analysis of the corpus.

Later, we also decided to use some of the inbuilt *nltk* methods and plot some graphs using those metrics. Firstly we plotted the graph for the words with the highest occurrences in our corpus, the figure 3 with the graph can be seen below.

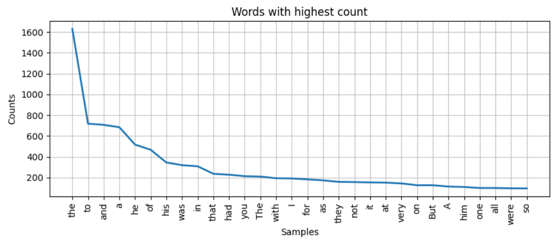

Figure 3. Highest occurrences of words in corpus

Lastly, we also found a function of the *nltk* library that let us discover the top used non-stopwords, the results of this function can be seen in the figure 4 with the graph below.

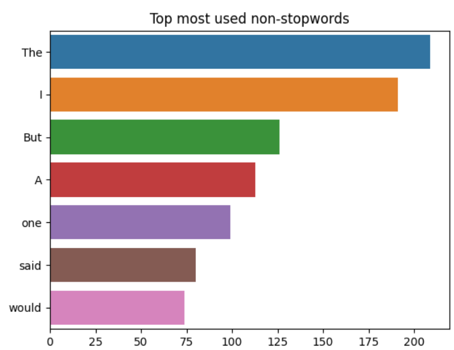

Figure 4. Most used non-stopwords in corpus

1.2 Named entity recognition (NER)

Named entity recognition (NER) — sometimes referred to as entity chunking, extraction, or identification — is the task of identifying and categorizing key information (entities) in text. An entity can be any word or series of words that consistently refers to the same thing. Every detected entity is classified into a predetermined category.

NER is a form of natural language processing (NLP), a sub-field of artificial intelligence. *NLP* is concerned with computers processing and analyzing natural language, i.e., any language that has developed naturally, rather than artificially, such as with computer coding languages. For NER we used 2 libraries, namely Stanza and Spacy with intention to compare their results later.

The NER module in Stanza [11] is based on a neural architecture called a bidirectional LSTM-CRF, which is a combination of a bidirectional Long Short-Term Memory (LSTM) model and a Conditional Random Field (CRF) model.

SpaCy is a natural language processing library that provides pre-trained models for various NLP tasks, including named entity recognition (NER). The NER module in SpaCy is based on a combination of rule-based and statistical techniques.

Figure 5 shows the general pipeline we constructed and used with both approaches.

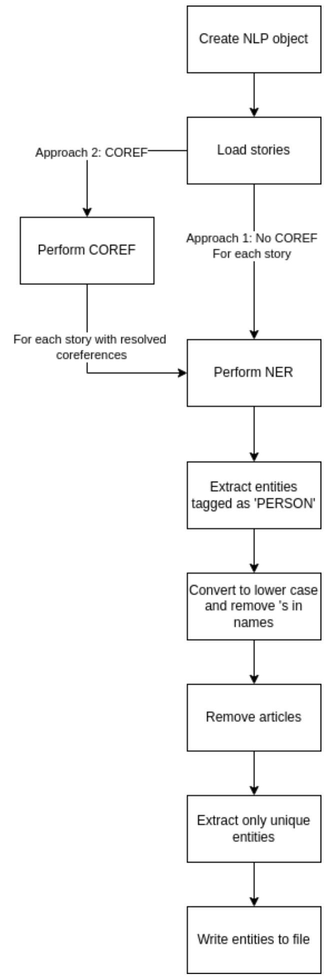

Figure 5. General *NER* pipeline.

1.2.1 Coreference Resolution (COREF)

Coreference resolution (CR) is the task of finding all linguistic expressions (called mentions) in a given text that refer to the same real-world entity. After finding and grouping these mentions, we can resolve them by replacing, as stated above, pronouns with noun phrases. It is an important step for a lot of higher level NLP tasks that involve natural language understanding such as document summarization, question answering, and information extraction. We used Coreference resolution before performing NER to try to improve the result of NER. When performing NER with Spacy we used Neuralcoref library to perform coreference resolution. For NER with Stanza, we used AllenNLP library to perform coreference resolution.

Before we started implementing our coreference resolution methods, we used an online coreference resolution visualization tool [12] which can be used to better understand the procedure of coreference resolution and how the author's State-of-the-art neural coreference resolution system performs. An example output from one of our corpus stories titled "The Fisherman and the Little Fish" can be seen below in the figure 6.

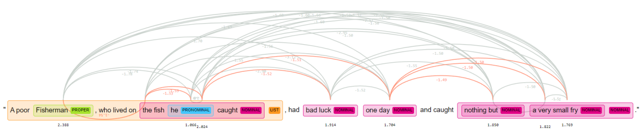

Figure 6. Coreference resolution visualization of "The Fisherman and the Little Fish" story.

For the sake of transparency, we only used the first sentence of the story so the coreference resolution visualization tool's output is as clear and as understandable as it can be.

1.3 Sentiment analysis

For sentiment analysis of characters, we used the methods which are described in this section. These methods were used to provide sentence based sentiment predictions. To obtain a character based sentiment prediction, several approaches were used for extraction of sentences i.e. using all the sentences where the character is referenced, using only the first sentence where the character is referenced, using only the last sentence where the character is referenced, using both the first and last sentences where the character is referenced, using all the sentences until the character was referenced and using all the sentences after the last reference to the character. To consolidate these sentence predictions into character based sentiment prediction, the average was taken. In order to obtain a final prediction, the average was rounded to the nearest integer.

1.3.1 AFINN

As described above, AFINN (Affective Norms for English Words) is a lexicon-based approach to sentiment analysis, which uses a pre-built list of words with associated sentiment scores to determine the overall sentiment of a piece of text [6]. To obtain a sentence level sentiment prediction using AFINN a sum over of valences of all words contained within that sentence, was used.

1.3.2 CNN based

The CNN (convolutional neural network) based approach uses learned word vector representations through neural language models [13] to transform words into vectors, which are then stacked into a matrix and fed into a CNN. The CNN can then be trained for any NLP classification task. In [14] a CNN was trained using this approach and achieved 81.5% classification accuracy on sentiment prediction of movie reviews with one sentence per review.

1.3.3 SiEBERT

The RoBERTa model was first proposed in [15] and it builds on BERT and modifies key hyperparameters, removing the next-sentence pretraining objective and training with much larger mini-batches and learning rates [16]. At that point, using that approach yielded state-of-the-art performance on the GLUE (General Language Understanding Evaluation) benchmark. SiEBERT, a fine-tuned checkpoint of RoBERTa, was trained and evaluated on 15 data sets from diverse text sources to enhance generalization across different types of texts (reviews, tweets, etc.).

1.4 Protagonist and antagonist prediction

To determine the protagonist and antagonist for a story, we used the results of sentiment analysis and calculated an entity importance score for each entity by counting the number of appearances in the story. The character with positive and neutral sentiment that occurs the most is treated as a protagonist, and the character with the negative sentiment that has the most appearances is treated as antagonistic. We decided to count neutral sentiment to allow for a more balanced, nuanced, and inclusive approach to protagonist prediction. This method was implemented such that we perform NER with Stanza and count all character occurrences. For entity sentiments, we used results which we got with the CNN method because that was the one that performed the best. Then we used data from both those methods to predict the protagonist and antagonist. The results along with character occurrences were saved to a file.

2. Results

Results for all the methods we tested are available in the repository in the directory `/results`, where results for a specific method are stored in ther respectable subdirectory.

2.1 NER metrics

We calculated precision, recall and F1 Score for both the approaches to NER with and without coreference resolution over the entire corpus. The results are shown in table 2. As we can see, Spacy benefited a lit-

Method	Precision	Recall	F1-score
Stanza NER	0.697	0.842	0.763
Spacy NER	0.508	0.811	0.625
Stanza NER+COREF	0.687	0.825	0.75
Spacy NER+COREF	0.523	0.734	0.611

Table 2. Performance metrics for NER.

tle bit by using Neuralcoref library for coreference resolution before doing named entity recognition. Stanza on the other hand got worse performance when doing coreference resolution with AllenNlp. Overall, the best approach was just using Stanza library. All the approaches had higher recall score, which means the libraries are effective at identifying positive instances, but lower precision means we have a higher rate of false positives.

2.2 Sentiment analysis

Here we present the results of the character based sentiment analysis. The results are split into two parts, one part focuses on reporting the results of different approaches of extracting relevant sentences for character based sentiment prediction and the other focuses on reporting performance of individual methods for sentiment prediction. Performance of all the approaches and all the sentiment prediction methods can be seen on figures 7 and 8 respectively. To obtain these metrics, each sentence extraction approach was tested with each sentiment prediction method and vice versa. To consolidate the performance of either the sentence extraction approach or sentiment prediction method, the average was taken. Comparing the methods for sentiment prediction, it can be observed that the only method that achieved better than random performance (on all sentiment classes) is the CNN based method. Comparing the approaches for sentence extraction, it can be seen that taking the first and last sentence achieved overall best performance.

2.3 Protagonist and antagonist prediction

We performed protagonist and antagonist prediction over the entire corpus as described in the previous chapter. With the described procedure we finally obtained 147 determined protagonists and 29 antagonists.

3. Discussion

We performed named entity recognition both with and without prior coreference resolution and shown that plain Stanza method performed the best. When applying coreference resolution in Spacy, improved

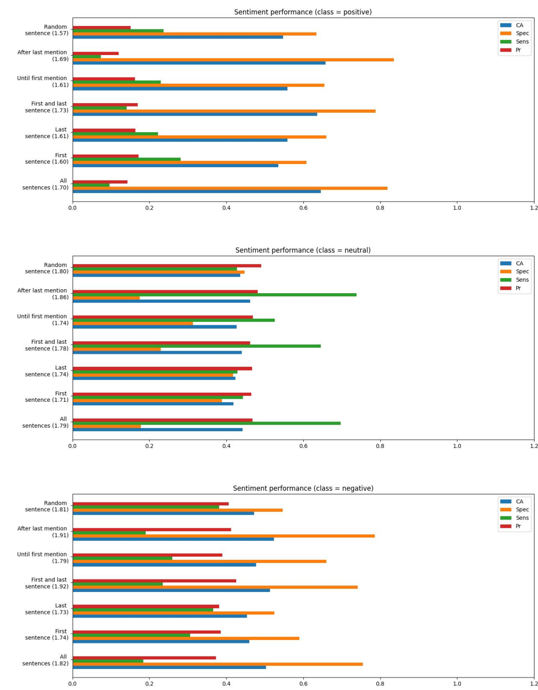

Figure 7. Performance of sentence extraction approaches (Pr=Precision, Sens=Sensitivity, Spec=Specificity, CA=Classification accuracy). Numbers in brackets next to the labels, present the sum of all the metrics. For each approach the average of all sentiment prediction method was used.

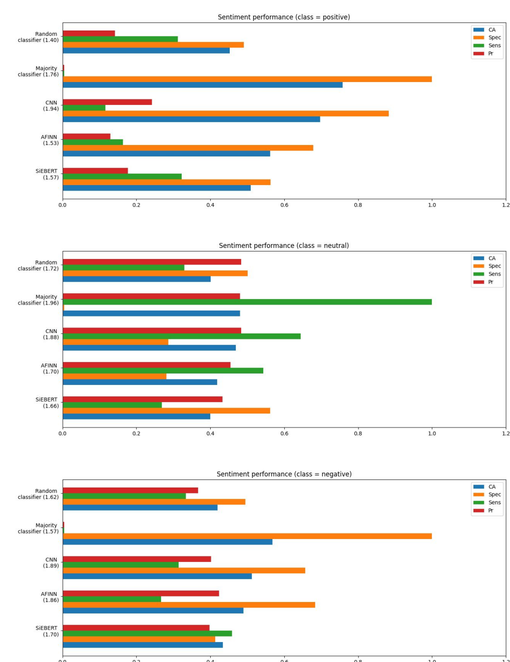

Figure 8. Performance of sentiment prediction methods. For each method the average of all sentence extraction approaches was used.

results were observed in terms of accuracy and precision for Named Entity Recognition (NER). This can be attributed to the enhanced understanding of context facilitated by coreference resolution. Contrarily, in the case of Stanza, better results were obtained when coreference resolution was not employed. AlenNLP's coreference resolution algorithm may not be optimized for shorter texts like fables, where the context and reference relationships are less intricate. Coreference resolution might introduce additional complexity, leading to a slight degradation in performance, which produces worse text for Stanza to perform NER on.

We have conducted character sentiment analysis with respect to different methods for sentence sentiment evaluation and different methods for extraction of appropriate sentences. We have shown, that only the CNN based approach achieved better than random performance and that taking only the first and last sentence that references the character in consideration yields the best performance. The SiEBERT model did not perform better than a random classifier, which is most likely due to the fact of being trained only on a binary sentiment sentence classification problem and the corpus used also considers neutral characters. The AFINN model's performance also suffered as it scores each word's sentiment without considering sentence semantics. The corpus which was used contained short stories and a low number of characters per story, which impacts the utility of each sentence that references a given character. As such, choosing all sentences where the character is mentioned, did not achieve as good performance since in some cases it meant that the whole story was considered for certain characters. Future research should explore longer stories and character sentiment evolution.

When predicting protagonists and antagonists, the results were somewhat acceptable. Given that we allowed positive and neutral sentiment for protagonist detection, there were a lot more protagonists determined by our algorithm. On the other hand, there were only 29 antagonists detected. We think that the biggest factor for that was that fables often do not include characters with clear negative sentiment, because they are mainly focused on simple and positive concepts.

References

[1] Paper Authors. Project github. https://github.com/UL-FRI-NLP-Course-2022-23/nlp-course-team-100.

[2] DMilind Soam and Sanjeev Thakur. Sentiment analysis using deep learning: A comparative study. In *2022 Second International Conference on Computer Science, Engineering and Applications (ICCSEA)*, pages 1–6, 2022.

[3] Arthur M. Jacobs. Sentiment analysis for words and fiction characters from the perspective of computational (neuro-) poetics. Frontiers in Robotics and AI, 2019.

[4] Micha Elsner. Character-based kernels for novelistic plot structure. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 634–644, Avignon, France, April 2012. Association for Computational Linguistics.

[5] Eric T. Nalisnick and Henry S. Baird. Character-to-character sentiment analysis in shakespeare's plays. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 479–483, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[6] Finn Årup Nielsen. A new ANEW: Evaluation of a word list for sentiment analysis in microblogs, 2011.

[7] Diego Raphael Amancio. Network analysis of named entity co-occurrences in written texts. *Europhysics Letters*, 114(5):58005, jul 2016.

[8] Mónica Marrero, Julián Urbano, Sonia Sánchez-Cuadrado, Jorge Morato, and Juan Miguel Gómez-Berbís. Named entity recognition: Fallacies, challenges and opportunities. *Computer Standards Interfaces*, 35(5):482–489, 2013.

[9] Marilena Ditta, Fabrizio Milazzo, Valentina Raví, Giovanni Pilato, and Agnese Augello. Data-driven relation discovery from unstructured texts. In *2015 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K)*, volume 01, pages 597–602, 2015.

[10] Tirthankar Dasgupta, Rupsa Saha, Lipika Dey, and Abir Naskar. Automatic extraction of causal relations from text using linguistically informed deep neural networks. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 306–316, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[11] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. Stanza: A python natural language processing toolkit for many human languages. *CoRR*, abs/2003.07082, 2020.

[12] Web source. Neuralcoref visualizer. https://huggingface.co/coref/.

[13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

[14] Yoon Kim. Convolutional neural networks for sentence classification, 2014.

[15] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.