University *of Ljubljana*
Faculty *of Computer and Information Science*

Literacy situation models knowledge base creation

Jaša Frelih Črv, Jurij Kolenik, and Žan Jonke

Abstract

Keywords
Story entities, Family relationship extraction, Short stories

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.

For our project, we will focus on character analysis and sentiment analysis of these characters. We will build a pipeline, which will perform the tasks listed below:

- Named entity recognition (with and without prior COREF) to extract characters.
- Character sentiment analysis to determine whether a character is good or bad.
- Calculate entity importance score and determine protagonist and antagonist based on the sentiment score.

For NER we will use Stanza and Stacy libraries, where we will also perform coreference resolution. With Stanza we will perform COREF using AllenNLP library and with SpaCy we will use Neuralcoref library. The performance of these methods will be calculated and compared. To determine whether a character is good or bad, we will look at the general sentence occurrence sentiment, where we will analyze the general theme of sentences that a character occurs in and determine the sentiment based on the average context of the sentences. For entity importance score we will use EOIS (Entity Occurrence Importance Score) approach.

0.1 What has been done

We have done the following tasks:

- Annotate 148 short Aesop fables.
- Used 2 different approaches for COREF+NER with Stanza+AllenNLP and SpaCy+Neuralcoref on Aesop fables.
- Sentiment analysis on Aesop fables.

0.2 Future work

Our future work plan:

- Use entity importance score and sentiment analysis score to determine protagonist and antagonist.
- Calculate performance metrics (precision, recall, f1 score) of results of used libraries compared to annotated data.
- Annotate and apply pipeline on the additional corpus (7 mid sized stories).

1. Existing Solutions & Related work

When we started working on this project, we firstly looked at some existing papers that have the same topic or are connected to the same field of research as ours.
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) [1]. 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 [2]. In article [3] 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, [4] 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 [5]) 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 [6], 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. Their 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 [7] 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 [8] 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 [9] 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 consisted of 148 short Aesop fables stories. We added annotations to all the stories, which include the characters that appeared 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 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 behaviours: 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 behaviours: be kind, be sacrificial, help others, strive to improve the situation. If a character did not commit or exhibit any of the aforementioned behaviours, 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.

1.2 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.

1.2.1 Coreference Resolution (COREF) when performing NER with spaCy

When we did the coreference resolution using *spaCy* and *neuralcoref* we followed the pipeline shown in figure 1.

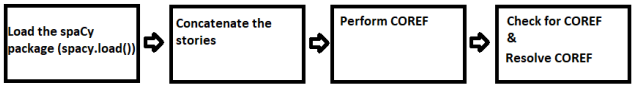

Figure 1. *COREF* pipeline using spaCy.

We implemented the procedure shown in the pipeline image above and then saved the results to *.json* format so we could late use them together with named entity resolution procedure.

Firstly we loaded the spaCy's pipeline with spacy.load() command and added the loaded pipeline to neuralcoref with neuralcoref.add() command. After that we read our selected stories one by one and concatenated each of them, so they are properly represented. After that we called spaCy's coreference resolution function. SpaCy's coreference resolution system needs to look for mentions of entities and then tries to cluster the mentions of the same entity together. Each mention is a span of text, or in other words a contiguous sequence of words. Systems learn from gold standard annotations, which group these spans into entity clusters. The clustering is evaluated by comparing the grouping of mentions produced by the system to the ground truth clustering provided in the test data. Then we checked if our result even has coreference resolution done the right way. Furthermore for positive coreference resolution results we resolved the coreference resolution and also visualized the results using the Neuralcoref Visualizer online tool, and example output can be seen in figure 2 below.

Lastly we saved the resolved text back to a .json file that we used later in our combined Named entity recognition and coreference resolution function.

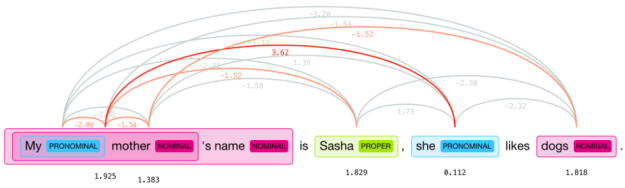

Figure 2. Neuralcoref Visualizer online tool example.

1.2.2 Coreference Resolution (COREF) when performing NER with Stanza

We used AllenNLP library to perform coreference resolution before performing NER with Stanza. It is a natural language processing library that provides pre-trained models for various NLP tasks, including coreference resolution. The coreference resolution module in AllenNLP is based on a neural network architecture called the SpanBERT model, which is a variant of the BERT model. The library performs coreference resolution in the following steps:

1. Tokenization: The input text is first tokenized into individual words, punctuation marks, and other symbols.
2. Encoding: The tokens are then mapped to high-dimensional vectors using a pre-trained contextual embedding model.
3. Mention Extraction: The SpanBERT model is used to predict the probability of each contiguous span of tokens being a mention. The model generates a mention score for each span, indicating the likelihood that the span refers to an entity.
4. Coreference Resolution: The model then performs coreference resolution by using the mention scores to cluster the mentions that refer to the same entity. The model outputs a coreference cluster for each entity in the input text.
5. Post-processing: Finally, the coreference clusters are post-processed to generate a more coherent output, such as replacing pronouns with their antecedents in the text.

The general pipeline of using the library was the same as with Neuralcoref.

1.3 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.

1.3.1 Named entity recognition (NER) using spaCy

When we did the named entity recognition using *spaCy* we followed the pipeline shown in figure 3.

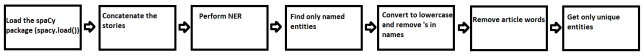

Figure 3. *NER* pipeline using spaCy.

We implemented the procedure shown in the pipeline image and then saved the results to *.json* format so we could process them later.

Firstly we loaded the spaCy's pipeline with spacy.load() command. SpaCy's trained pipelines can be installed as Python packages. This means that they're a component of your application, just like any other module. After that we read our selected stories one by one and concatenated each of them, so they are properly represented. After that we called spaCy's named entity recognition function. Spacy has the 'ner' pipeline component that identifies token spans fitting a predetermined set of named entities. These are available as the 'ents' property of a Doc object. Then we selected only named entities from the output of the 'ner' function. Furthermore we converted the entities to lowercase and remove all the 's from names. Later we also removed article words and finally extract only unique entities.

1.3.2 Named entity recognition (NER) using Stanza

The NER module in Stanza [10] 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. The process of NER in Stanza is made of following steps:

1. Tokenization: The text is segmented into individual tokens (words, punctuation marks, etc.).
2. Embedding: Each token is then mapped to a high-dimensional vector using a pre-trained word embedding model. This captures the semantic meaning of the token.
3. BiLSTM Encoding: The token embeddings are fed into a bidirectional LSTM model, which learns to represent the context of each token.
4. NER Tagging: The outputs of the BiLSTM model are then fed into a CRF model, which predicts the most likely sequence of named entity tags (such as person, organization, or location) for the input text.
5. Post-processing: Finally, the NER tags are post-processed to generate a more coherent output, such as merging contiguous tokens with the same tag into a single named entity.

The general pipeline of using the library was the same as with SpaCy.

1.4 Coreference Resolution (COREF) + Named entity recognition (NER)

In this section we performed named entity recognition on outputs of already processed stories text which have already gone through the process of coreference resolution. In the end we compared the results of our named entity recognition to the combined results of coreference resolution + named entity recognition.

1.5 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 where the character in consideration appeared. To consolidate these sentence predictions, the average was taken. To obtain a final prediction, the average was rounded to the nearest integer.

1.5.1 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 [5]. To obtain a sentence level sentiment prediction using AFINN a sum over of valences of all words contained within that sentence, was used.

1.5.2 CNN based

The CNN (convolutional neural network) based approach uses learned word vector representations through neural language models [11] 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 [12] a CNN was trained using this approach and achieved 81.5% classification accuracy on sentiment prediction of movie reviews with one sentence per review.

1.5.3 SiEBERT

The RoBERTa model was first proposed in [13] 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 [14]. 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.).

1.6 Protagonist and antagonist prediction

To determine the protagonist and antagonist for a story we will use the results of sentiment analysis and entity importance score. First we will normalize the sentiment score, and then multiply that by the entity importance score. The idea is

that the antagonist will have the highest positive score and protagonist will have the highest negative score.

References

[1] 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.

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

[3] 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.

[4] 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.

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

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

[7] 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.

[8] 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.

[9] 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.

[10] 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.

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

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

[13] 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.

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