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

Literacy situation models knowledge base creation

Nina Velikajne, Jer Pelhan

Abstract

Keywords
sentiment analysis, named entity recognition

Advisors: Slavko Žitnik

Introduction

Literature is a rich and diverse field that contains a vast set of characters and relationships. From novels and short stories to plays and poems, literature offers an endless variety of human experiences, emotions, and perspectives. Each literary work presents a unique set of characters with their own personalities, backgrounds, and relationships with each other. These characters interact with each other in complex ways, creating intricate webs of relationships and social dynamics. Understanding these elements might be easy for a human, but it can present a difficult task for natural language processing (NLP) since natural language contains ambiguities and unclear references that might be difficult to understand for machines.

In this paper, we describe our approach for creating a knowledge base of literacy situation models using a combination of named entity recognition (NER), character graph co-occurrence, and character sentiment analysis. Our approach is based on the analysis of short stories that most often have very contrasting characters and more superficial sentence structures. By analysing character interactions in our knowledge base, we will gain insights into how their interactions shape the story, who is protagonist and antagonist of the story, and how the characters are portrayed.

The first step of our approach is applying NER to identify and classify different characters in stories. The entities identified through NER techniques might be noisy, ambiguous, and lack clear semantics. Consequently, identifying connections between related entities in a given dataset, as well as with existing knowledge might enrich the data, and increase disambiguation and data consolidation. Dataset providers often seek to enhance a particular dataset by adding links to comprehensive reference datasets in order to achieve mentioned goals [1]. Herein, we plan to analyse the text to determine which characters interact with each other and how frequent their interactions are and construct a character graph co-occurrence graph. Then, we apply sentimental analysis algorithms to determine character morality and character-to-character sentiment. In the following, we will gain information about the morality of characters and from this in combination with character graph concurrence, conclude which characters are protagonists and antagonists, which hold side roles, etc. Furthermore, we plan to analyse the sentiment of whole stories, in order to label where bad or good events occur, and present them chronologically throughout the story.

Our corpus consists of X shorty stories, fables collected from [1]. Our method is used to automatically create a knowledge base on whole corpora. We evaluate the performance of our pipeline on a smaller subcorpora consisting of 10 tales, for which we write all the expected outputs, and basically annotate the data.

Related works

There are several approaches to NER in NLP. The simplest one are the rule-based systems. One of the first research papers in the field was presented by Lisa F. Rau [2]. It relies on heuristics and handcrafted rules. System was tested on over one million words and it successfully extracted company names with over 95% accuracy. Newer approaches use machine learning in order to classify NER. Mikheev *et al.* [3] proposed a supervised method based on maximum entropy. They report a recall of 76% for locations, 49% for organizations and 26% for persons with precision ranging from 70% to 90%. The techniques used in (NER) generally depend on lexical resources (such as WordNet), lexical patterns, and statistics derived from analyzing large unannotated corpora.

[1] https://www.gutenberg.org/

Akbik *et al.* [4] presented contextual string embedding. They use LSTM and Bi-LSTM language models and achieve F1 score of 93.09%. A similar approach is also used in the Stanza Python library, which we use.

The analysis of character interactions is a critical aspect of NLP, and relation extraction plays a key role in this process. Lehmann *et al.* [1] developed RelFinder, a tool that utilizes a BFS algorithm to identify semantic connections among multiple entities in an RDF dataset. This algorithm is responsible for finding all related entities in the tripleset. Rather than focusing on finding connections between two given entities, paper [5] aims to identify the most connected entities in relation to a given relationship and entity. Although their approach offers an intriguing perspective on the problem, it differs from our problem. They search for connected entities based on known relationships, while we will try to uncover such connections between known entities. Leskovec *et al.* [6] presented a technique suggesting positive and negative relationships between people in a social network. This might also be useful in detecting the negative and positive characters in our stories.

Sentiment analysis is a widely researched topic used in commercial applications to predict user opinions on product reviews, detect harmful speech on social media, etc. Without using massive amounts of texts with the annotated sentiment, knowledge-based methods such as the method presented by Andreevskaia [7] rely on sentiment lexicons. Mentioned lexicons are lists of words that typically express sentiment, negative or positive, with the corresponding label (usually an integer), that represents positive to negative sentiment. But the performance of these methods heavily relies on each sentimental word being in the lexical base. Furthermore, some words in different contexts have different sentimental values, which cannot be distinguished using knowledge-based methods. Strapparava [8] a lexical database WordNet-Affect, that extends WordNet for sentiment analysis applications. WordNet-Affect provides a way to identify the emotional tone of the text by linking the words in the text to their corresponding affective information in the WordNet-Affect database. This way, the method can achieve more detailed information about the emotional meaning, ”happy” has a positive valence, even though the word ”mourning” is next to it in the method presented by Andreevskaia [7], but not with WordNet.

Methods

Corpus Analysis

Our corpus consists of 44 short stories. The data was taken from a project of a group from last year. They annotated 55 stories. We decided to enlarge the corpus so we added X additional stories and annotated them in the same format. The dataset and the annotations are publicly available in our GitHub repository[2]. The ground truth is annotated in JSON format for each story separately. It contains a list of all present characters, protagonist and antagonist labels, and sentiments. Sentiments include pairwise sentiments between all the characters. Sentiment 1 annotates a positive relationship between the two characters, -1 a negative one, and 0 a neutral relationship.

Our corpus is analysed in Table 1 and consists of 63 Grimm fairy tales, with an average length of approximately 1900 words. We find that on average the vocabulary size is 472 without any text preprocessing, which shows superficial sentence structures.

Table 1. Basic corpus analysis.

number of stories	63
shortest (#words)	286
longest (#words)	4696
avg. num. of words	1915,17
avg. num. of sentences	44.57
avg. num. of words in sentence	42.97
vocabulary size	5723
avg. vocabulary size	472

Coreference Resolution

Coreference resolution is an essential step of text processing as it identifies pronouns or nouns that refer to the same entity, Furthermore, it also removes named entity deduplicates. We use coreference resolution from AlenNLP [9], which is a strong tool as it employs a network-based approach to link entities across a text. It is trained on large corpora of annotated texts, enabling it to identify co-referent phrases, pronouns and nouns based on their linguistic features such as gender, number and semantic similarity. We use a trained model to identify and link co-refernt entities, which enables a more accurate and detailed analysis of relationships between the characters in a literary work. As is demonstrated in Figure 1, our preliminary study has revealed that NeuralCoref's performance was unsatisfactory. Consequently, we have made the decision to exclude it from the subsequent stages of our method's pipeline.

In the preliminary study, we also found that AllenNLP [9] always uses the first mention of the character as the cluster name. Thus if the first mention is ”the lucky beatufiul girl called Red Hat”, it replaces all occurances with this long confusing ”character name”. Thus, we redesign the character cluster naming procedure. Within each cluster we removed pronouns, definite and indefinite articles, and stopwords from the clustered duplicates of the character. The cluster head was then chosen as the most frequently occuring name for each character. In the preliminary study, this approach yielded better results than the .

Named-Entity Recognition

NER is a subfield of NLP that deals with identifying and classifying named entities in text. Named entities are specific

[2] https://github.com/UL-FRI-NLP-Course-2022-23/nlp-course-mataviuc/tree/main/data/aesop

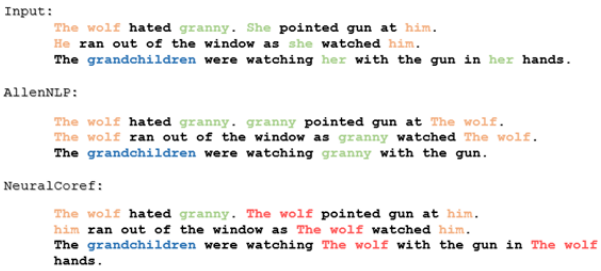

Figure 1. Comparison of AlenNLP [9] and NeuroCoref [10] coreference resolution performance on a challenging paragraph.

entities that are referenced by proper names, such as people, organizations, locations, dates, and other miscellaneous entities like products and events. It involves training machine learning models to identify and classify these named entities within a given text. Once the model is learned it can be used to automatically identify and extract named entities from new texts [**?**]. We use two methods in order to detect NER in our dataset, namely the AllenNLP NER [3] and the Stanza NER [4].

AllenNLP NER uses a bidirectional LSTM-CRF model. Each word is represented as a vector pre-trained word embeddings, such as GloVe or FastText. These vectors are the input to a bi-directional LSTM layer, which captures the context and other dependencies of each word with respect to other neighboring words. This is then inputted into Conditional Random Field (CRF) layer. It applies a probability distribution over all possible NER tags for each word. The model is optimized using a maximum likelihood objective function, that tries to maximize the probability of the correct tag for each input [11].

Stanza NER uses a Bidirectional Encoder Representation from Transformers (BERT). BERT is a powerful language model that has been pre-trained on massive amounts of text data. Stanza NER first tokenizes text and then produces a probability distribution for all possible NER tags for each word. The model uses word embeddings, attention mechanisms, and convolutional neural networks to analyze the context surrounding each token and make predictions about the entity tag. After prediction, the model uses a post-processing step to combine adjacent tokens with the same entity label into a single entity. This improves the overall accuracy of the NER [12].

Character Morality and Character-to-Character Sentiment analysis

For sentiment analysis of stories, we use the VADER [13] rule-based approach for sentiment analysis. It uses a lexicon-based approach to determine the polarity (positive, neutral or negative) of the input text. For character-to-character sentiment analysis, we find all sentences and blocks of sentences where two characters interact with each other and analyse the polarity of the language towards each other. This way we determine their relationship and sentiment towards each other. For character morality analysis VADER is used to analyse the actions and words of a character in a story. By analysing the polarity of sentences or actions that are used/performed by a character, we portray the character as good or bad.

Co-occurrence graph

For constructing the co-occurrence graph we implemented our own method. First, we applied coreference resolution to uniform all character occurrences. Next, we extracted the named entities, namely all the characters in the story. For extracting the NER we used the already presented AllenNLP NER. Once we extract the characters, we initiate an adjacency matrix, where columns and rows represent one specific character. We iterate through the sentences and count how many times two characters occur together. The obtained adjacency matrix is symmetric. Based on the adjacency matrix we construct an undirected graph. Each character is represented as a node. Two nodes are connected if two characters co-occur. Based on the counts in the adjacency matrix, we also determine the weights of links.

Results

Named-Entity Recognition

We tested AllenNLP and Stanza NER on a short story – fable database. We first applied coreference resolution to uniform all character occurrences. Next, we extracted characters in every story using the two mentioned NER models. Once the characters were extracted, we compared the obtained predicted characters with the ground truth characters. For each model, we calculated the precision, recall, and F1 score and averaged them over all stories (see Table 2). We can see that Stanza has higher precision, meaning that it detects more true positives – right characters, compared to the AllenNLP. However, the recall and F1 score for the Stanza model are much lower compared to the AllenNLP. This means that the Stanza model in general detects more right characters – true positives. But is not able to identify all positive instances, characters present in the story (false negatives). AllanNLP on the other hand detects more characters, but not all are relevant. This reflects in lower accuracy. But it detects fewer false negatives, herein higher recall.

One of the biggest issues was detecting characters without a name e.g., queen, wolf, possibly written in lowercase. Several models do not recognize this as a named entity. So we will try to train our own BERT model on the labeled dataset in order to obtain more accurate results.

Co-occurence graph

For constructing the co-occurrence graph we built the adjacency matrix as presented in Subsection . We constructed a graph for each story in the dataset. We tested the accuracy of

[3] https://demo.allennlp.org/named-entity-recognition/named-entity-recognition

[4] https://stanfordnlp.github.io/stanza/ner.html

NER model	Precision	Recall	F1 score
AllenNLP	0.74	0.77	0.72
Stanza	0.80	0.65	0.68

Table 2. Obtained precision, recall, and F1 score for AllenNLP and Stanza NER models on fables dataset.

the method by reading the story, manually constructing the co-occurrence graph, and comparing it to the automatically constructed graph. We decided to represent only the most interesting examples.

The story The Hare With Many Friends contains the most characters – 7. The hare is the main character and the other animals are her friends. This can be nicely seen from the constructed graph (see Fig. 2). The hare is connected to all other animals, and the weights on the links are the heaviest, as one would expect.

Another example was chosen randomly, namely The Cat Maiden (see Fig. 3). Jupiter changes a cat into a maiden and together with Venus they observe her behavior. The constructed graph is correct, and so are the links and their weights. The link between Jupiter and Venus is stronger, and so is the link between the cat and the maiden.

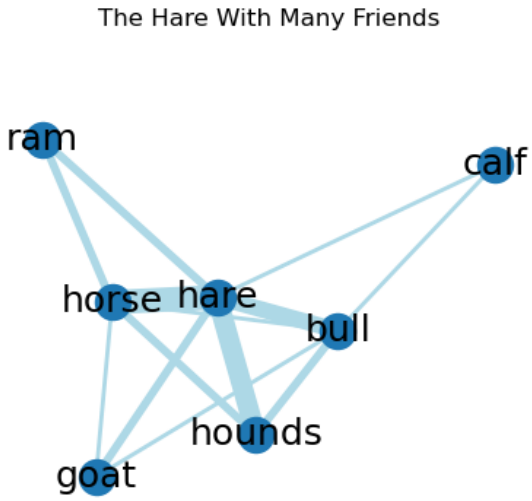

Figure 2. Co-occurence graph for The Hare With Many Friends. Thicker links between nodes symbolize higher weights, and more co-occurrences of the two characters.

Sentiment analysis

We performed character sentiment analysis or character morality. We analysed the sentiment of sentences and actions that are used/performed by the character, and took the most occuring polarity of the character as the morality of the character. The morality is classified as 1, 0 and -1, which stands for positive, neutral, and negative character, respectively. As visible in Table 3, Vader outperforms the Stanza method.

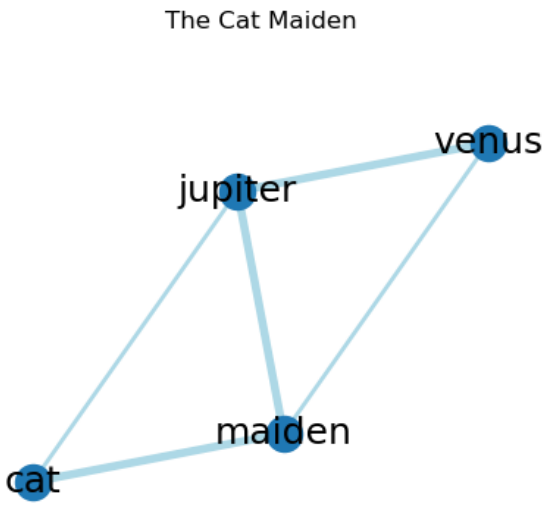

Figure 3. Co-occurence graph for The Cat Maiden. Thicker links between nodes symbolize higher weights, and more co-occurrences of the two characters.

Sentiment model	Precision	Recall	F1 score
Stanza	0.77	0.63	0.68
Vader	0.78	0.70	0.70

Table 3. Obtained precision, recall, and F1 score for Vader and Stanza pipelines on fables dataset for character morality classification.

In the advancement we added character-to-character sentiment to our knowledge base. Within this part of our method, we searched for subject-verb-object triplets using the Spacy [14] language processing tool. In each such sentence, we performed sentiment analysis. The resulting value was put in affinity matrix between the subject to object, with both being the characters. The results are presented in Table 4. The Vader outperforms Stanza on sentiment prediciton task with respect to all used metrics.

Sentiment model	Precision	Recall	F1 score
Stanza	0.66	0.61	0.61
Vader	0.71	0.60	0.63

Table 4. Obtained precision, recall, and F1 score for Vader and Stanza pipelines on fables dataset for character-to-character sentiment analysis.

References

[1] Philipp Heim, Sebastian Hellmann, Jens Lehmann, Steffen Lohmann, and Timo Stegemann. Relfinder: Revealing relationships in rdf knowledge bases. *SAMT*, 5887:182–187, 2009.

[2] Lisa F Rau. Extracting company names from text. In *Proceedings the Seventh IEEE Conference on Artificial Intelligence Application*, pages 29–30. IEEE Computer Society, 1991.

[3] Andrei Mikheev, Marc Moens, and Claire Grover. Named entity recognition without gazetteers. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 1–8, 1999.

[4] Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th international conference on computational linguistics*, pages 1638–1649, 2018.

[5] Yong-Jin Han, Seong-Bae Park, Sang-Jo Lee, Se Young Park, and Kweon Yang Kim. Ranking entities similar to an entity for a given relationship. In *PRICAI 2010: Trends in Artificial Intelligence: 11th Pacific Rim International Conference on Artificial Intelligence, Daegu, Korea, August 30–September 2, 2010. Proceedings 11*, pages 409–420. Springer, 2010.

[6] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Predicting positive and negative links in online social networks. In *Proceedings of the 19th international conference on World wide web*, pages 641–650, 2010.

[7] Alina Andreevskaia and Sabine Bergler. CLaC and CLaC-NB: Knowledge-based and corpus-based approaches to sentiment tagging. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 117–120, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[8] Carlo Strapparava and Alessandro Valitutti. Wordnet-affect: an affective extension of wordnet. *Vol 4.*, 4, 01 2004.

[9] Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. Allennlp: A deep semantic natural language processing platform. *arXiv preprint arXiv:1803.07640*, 2018.

[10] Thomas Wolf. State-of-the-art neural coreference resolution for chatbots, Sep 2020.

[11] Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. Allennlp: A deep semantic natural language processing platform. *arXiv preprint arXiv:1803.07640*, 2018.

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

[13] Clayton Hutto and Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the international AAAI conference on web and social media*, volume 8, pages 216–225, 2014.

[14] Explosion AI. spacy: Industrial-strength natural language processing in Python. https://spacy.io, 2020.