FRI Natural language processing course 2022
Project report

University of Ljubljana
Faculty of Computer and Information Science

Literacy situation models knowledge base creation

Jan Krivec, Klemen Klemar, Nino Brezac

Abstract

The paper strives to build a knowledge base based on situation models from selected English short stories. A simple pipeline is proposed - it consists of named entity extraction, character sentiment analysis and character co-occurrence and relation. Fictional characters are extracted from short literature excerpts. A simple sentiment analysis on all characters follows, and finally the co-occurrence of recognized characters is studied. In short the goal is to, as accurately as possible, provide meaningful short story character analysis.

Keywords
named entity recognition, sentiment analysis, literature, text processing, character analysis

Advisor: Slavko Žitnik

Introduction

Our motivation for this project was to build a knowledge base based on situation models from selected English short stories. The output of natural language processing methods on such stories can be used to quickly grasp the content of the story and either build a better model of the story in our minds before reading, or solidify the content we've read. There is a lot going on in these stories and therefore a lot of data and different representations to extract, for example we can extract different characters, general sentiment about them, how they are connected with each other and so on. We can also extract the events in the stories, analyse the space and time around these events and build a model that can represent the story according to this data in a concise way. Because there is so much to extract from these stories, we decided to focus only on a subset - the extraction of characters and sentiment analysis based on the context they appear in, building a model and implementing its representation which will reveal the connections between them.

Related work

There is plenty of literature covering the complex task of extracting data from various sources, but the implementation of knowledge base creation on English short stories has not been yet established, as longer texts were used for data extraction and focus was directed on only a subset of the problems we intend to solve. One of similar works was done by Jacobs, 2019[1], where the entire collection of Harry Potter books was made into a corpus and analyzed by a novel tool called `SentiArt` and then compared to some other deep neural network approaches. It used vector space models(VSM) to compute the valence of each word in a text by locating its position in a 2D emotion potential space spanned by the words of the vector space model. The tool produces plausible predictions regarding the emotional and personality profile of the characters and calculates so called "big5" personality traits, which can then be further classified into the "good" or "bad" category. SentiArt achieved 100% accuracy in predicting the sentiment category of 120 excerpts from the Harry Potter books, outperforming the standard sentiment analysis tool, VADER (Hutto and Gilbert, 2014 [2])

We can split the sentiment analysis into three classes: dictionary (word list) based, Value Steam Mapping (VSM) based, neural and a hybrid between the three. The first one determines the valence of a word from text by looking it up in a reference list (Also used to compute the "emotion" of characters as discussed in Elsner 2012 [3]), while the second uses unsupervised learning approach by implementing either the Latent Semantic Analysis (LSA) discussed by the Laundauer et al.[4] or Pointwise Mutual Information (PMI) and Information Retrieval (IR) to measure the similarity of pairs of words.

Of course our goal is to also create a character network and not only the sentiment analysis of individual characters, which was vaguely adressed in Labatut 2019[5]. One of the relevant works that touched multiple aspects of our task is the LitBank [6] corpus which provides an annotated dataset (Entity annotations, Event annotations, Coreference annotations, Quotation annotations) of 100 works of English-language fiction. This corpus is as close as we can get to our target data and even contains a tagger, which can be used to tag entities and events in new text.

We have also reviewed some other papers that gave us the initial understanding on how people construct mental models of narrative stories such as Zwaan 1995[7] and 1998[8] but they did not help us form the initial idea.

Methods

The general pipeline for our task was imagined as follows:

1. Named entity recognition (NER)
2. Character sentiment analysis
3. Character co-occurrence
4. Visualizing the data and final remarks

Named entity recognition (NER)

Our corpus is made up of 3 subcorpora:

- short stories - a set of 81 very short classic Aesop fables. Very simple plots and few characters. Characters are often recurring cliches (cunning fox, evil wolf, brave lion etc.).
- medium stories - a set of 7 early 20th century American literature short stories. Plots are more developed, and the characters are more detailed compared to our short subcorpus.
- litbank - a curated set of 100 varied shorter stories from Anglo-Saxon literature. They range in complexity, from childrens stories to delicate and mature passages.

This way we could examine the model performance in different scenarios. Table 1 represents a primitive corpus analysis - the character count, word count, sentence count, for each subcorpora along with some mean values (letters per word, words per sentence, sentences per story). Such numbers along with our short descriptions give a rough outline of the nature of the corpus.

	short_stories_corpus	medium_stories_corpus	litbank_corpus
characters	61483	161421	1034539
words	13823	35176	216213
sentences	499.00	2035.00	8687.00
letters per word	4.32	4.37	4.7
words per sentence	28.79	18.37	28.21
sentences per story	6.09	290.71	86.87

Table 1. Corpus analysis (characters, words, sentences, letters/word, words/sentence, sentences/story).

The precondition to extract any sort of value from our work was to extract the characters in the story - to perform named entity recogniton.

We evaluated 2 different NER solutions - *Stanza* and *Spacy*.

Stanza is a collection of accurate and efficient tools for the linguistic analysis of many languages.

We decided to use Stanza because it provided a complete NN network pipeline (tokenization, POS tagging, dependency parsing, and of course, named entity recognition) and contained some already pretrained neural models. Moreover, it is a native Python implementation with minimal set-up hassle.

Spacy is a similar toolkit, again allowing swift and varied natural language processing (tokenization, visualizing, pretrained transformers). SpaCy is by default faster than Stanza as it is optimized under Cython (Python and C mixture).

The process was as follows:

- We use $spacy\_stanza.load\_pipeline()$ to create an nlp object to process a text with a Stanza pipeline and create a spaCy Doc object.
- Extract all entities tagged ”PERSON” and ”ORG”. In the process we ignore stopwords such as ”and” and ”a”, and join tagged entities that are in sequence (e.g. ”Leonardo da Vinci”, not just ”Leonardo”).
- We also postprocess the extracted entities - remove all occurrences of ’ś’, ’the ’, ’–’, ’/’, ’.’. Any instances of multiple whitespace are also removed.

We compared the performance of the two above stated NER solutions on three different corpora. First, we compared the running times of both implementations, which can be seen from table 2.

	Short Stories	Medium Stories	Litbank
Spacy	0.03 s	1.23 s	0.41 s
Stanza	2.95 s	52.33 s	30.76 s

Table 2. Comparison of average running times of entity extraction algorithms.

Next we compared the number and quality of recognized entities of the two above stated NER solutions, results of which can be seen in table 3. To explain it a little further, ”Equal” shows how many times, was the output of both, Stanza and Spacy, identical. That could indicate, that on those examples, that all the entities have been found, and that these are in fact correct. In case of ”Equal with different output”, the number of recognized entities was the same, but at least one of the entities was different.

On manual inspection of the outputs on a couple of randomly chosen stories from all corpora, given by both implementations, we could conclude, that Stanza gave higher quality results, i.e. is more reliable than Spacy. That also shows, that in table 3 in case of ”Stanza=0” has a lower number, which means, that there were fewer cases, given an input where Stanza did not find any entities. Another sign of better quality and reliability is that in majority of short stories, Stanza was able to extract more entities than Spacy. It can also be seen, that in longer texts, there were no cases, where either of the implementations would not be able to extract any entity.

Because we reason, that the reliability of Spacy is worse than Stanza, it is expectable, that Spacy would find more entities given longer text, where their quality would be worse, despite the data was preprocessed and postprocessed in the same manner in both implementations. After manual review of the output of both implementations on a couple of randomly chosen stories, this proved to be in fact true.

	Short Stories	Medium Stories	Litbank
Equal	27	0	7
Equal with different output	2	0	0
Stanza	35	0	16
Spacy	18	7	77
Stanza=0	6	0	0
Spacy=0	14	0	0

Table 3. Comparison of number of extracted entities using Spacy and Stanza.

Character sentiment analysis

To obtain the general consensus of a fiction character, we had to perform sentiment analysis on given stories. A lot of tools that can calculate overall sentiment of the story already exist and can be found in different Python libraries out of the box, but existing implementations of sentiment analysis only for parts (aspects) of text - called aspect based sentiment analysis are very rare and and limited in nature. We have attempted to solve this problem with few different approaches with varying success.

POS tagging and Dependency Parsing

Our first idea was to extract the words describing our characters, and then classify our characters by calculating the sentiment of these words. To extract characters (aspects) and the corresponding opinions we used POS tagging and Dependency Parsing as described in the paper [9]. We used the Stanza pipeline to extract Parts of Speech Tagging (POS) to parse sentences into constituent elements and then Dependency parsing to determine the relationships between words. After that, rule based analysis can be applied to the extracted opinion words to determine the general sentiment about the aspect. We have managed to implement the method from the paper, but further testing revealed that it only works reliably on texts where the sentiment is expressed directly (Example 1 below), unlike the fiction stories where sentiment is usually hidden in more complex ways (Example 2).

Example 1:
’The pool is great but the water is very cold.’
returns [[’pool’, [’great’]], [’water’, [’cold’]]

Example 2:
’There was Soon the Lion was let loose from his den, and rushed bounding and roaring towards his victim.’
returns [[’lion’, []]

Afinn

For our second idea, we first tried a more basic approach, using Afinn [10] - a list of words rated for valence with an integer between minus five (for negative sentiment) and plus five (for positive sentiment). We first extracted the general sentiment of the story by calculating sentiment score on the whole text. Then we also calculated the sentiment for each character by calculating the average sentiment score of the each sentence in which this character appears. We then normalized the results in order for them to be comparable with results we get from sentiment analysis using Stanza. Of course, even though such a naive approach is easy to implement it has some obvious faults. Sentiment analysis in such a way is in the end, just a mathematical evaluation. For example, such an approach can be overwhelmed by repetition even though the overall sentiment is negative. A sentence such as ”John is a happy happy murderer.” could be characterized as positive, as the positive value of two ”happy”’s together outweigh the negative value of ”murderer”.

General sentence occurrence sentiment

The third idea was to calculate the general theme of the sentences the character occurs in and then draw the conclusions about the character from that. The idea is, that on average, positive characters will more often appear in positive context and negative characters will appear in a negative context. We used the Stanza pipeline with the pre-trained model to get the sentiment of each sentence (-1 for negative, 0 for neutral and +1 for positive sentiment) and then identify if the character appears in the given sentence. We kept a list of sentiments for each character occurrence and calculated the average sentiment. We quickly noticed that some stories are overwhelmingly negative and others positive. To counter this effect, we normalised the average sentiment score of the character with the average sentiment score of all sentences. The equation 1 below shows the formula to calculate the sentiment of each character, where $V_{sentiment}$ represents the list of sentiments in which the character appears and $SEN_{sentiment}$ represents the list of sentiments for all sentences.

$$X_{sentiment} = \frac{\sum V^i_{sentiment}}{|V_{sentiment}|} - \frac{\sum SEN^i_{sentiment}}{|SEN_{sentiment}|} \quad (1)$$

In the Listing 1 below, we can see the output of our program before and after normalization. We can see that the general sentiment of the story is negative, which skews how we rate our characters. Androcles is is overall a positive character, but the negativity of the enire text skews the perception, which is fixed in the normalized version.

Listing 1. Comparison of sentiment analysis with and without normalization

```
## Without normalization
"Androcles.txt": {
        "general": -0.2857142857142857,
        "entities": {
            "androcles": -0.1111111111111111,
            "lion": -0.6666666666666663
        }
    },
## Normalized
"Androcles.txt": {
        "general": -0.2857142857142857,
        "entities": {
            "androcles": 0.1746031746031746,
            "lion": -0.38095238095238093
        }
    },
```

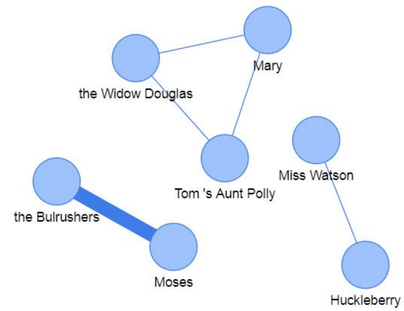

Figure 1. Co-occurrence network in Huckleberry Finn excerpt

Character co-occurrence

After mapping sentiment values to all characters, we decided to determine which characters actually interact. This can be determined by studying co-occurence.

The process was as follows:

- extract entities (characters)
- extract all sentences from story and filter them
- create bigrams from filtered sentences
- create network from bigrams

The prerequisite for character co-occurrence is naturally knowledge which characters actually exist (the process was explained in detail in the first section). It was then necessary to study their coexistence and interaction, which is why the next step consisted of extracting all sentences but only such where our characters are directly mentioned. A thorough bigram generation followed for each filtered sentence. The final step was constructing a co-occurrence network of all characters. A simple iteration over all bigrams sufficed. If the bigram contained two already recognized characters - it was added as an edge in the network. More frequent bigrams consequently have a larger weight. An example co-occurrence network output can be seen on figure 1.

Of course, such an approach leaves plenty of room for improvement. For example, detecting character co-occurrence on an excerpt from The Picture of Dorian Gray was not as successful. Dorian, the main character, was not present in the output network at all! This is because throughout the excerpt he is mentioned frequently but almost exclusively indirectly (through different pronouns, nouns and descriptions). A possible solution might be not to filter the sentences so strictly, to look at the wider context, neighboring bigrams etc.

Results

In this section we will provide some metrics on 15 randomized short stories. As our corpus is not already analyzed/tagged we opted to, as a start, manually analyze these 15 in terms of entity recognition and sentiment analysis and compare the algorithm results with our baseline values.

Detailed results of the named entity recognition (in terms of true positives, false negatives, false positives) are seen in table 4. The results are solid, the algorithm occasionally misses an entity, and almost never misinterprets a false entity.

We can find similar results for sentiment analysis in table 5, where two different approaches were evaluated and compared with manually obtained results. Both algorithms performed similarly - with around average classification success. While evaluating those results, we have to take into consideration, that the sentiment of a character is a subjective opinion based on ones interpretation of the story and some other person can potentially classify the character sentiment in a different way and obtain different results.

Story	TP	FN	FP
Androcles	2	0	0
Avaricious and Envious	1	2	0
Belling the Cat	1	1	0
Hercules and the Waggoner	2	0	0
The Ant and the Grasshopper	2	0	0
The Four Oxen and the Lion	1	2	0
The Fox and the Cat	3	2	0
The Fox and the Crow	4	0	0
The Fox and the Goat	1	1	0
The Fox and the Grapes	0	1	0
The Wind and the Sun	1	2	0
The Wolf and the Crane	2	0	0
The Wolf and the Kid	1	1	0
The Wolf and the Lamb	1	1	1
The Tortoise and the Birds	3	0	0
Overall	*25*	*13*	*1*

Table 4. Comparison of detected entities vs. actual entities

Corresponding precision and recall metrics for NER (from equations 2 and 3) are calculated below.

$$Precision = \frac{\sum TP_i}{\sum TP_i + \sum FP_i} = \frac{25}{26} = 0.96 \quad (2)$$

$$Recall = \frac{\sum TP_i}{\sum TP_i + \sum FNi} = \frac{25}{38} = 0.66 \quad (3)$$

Story	Afinn(correct/all)	Stanza(correct/all)
Androcles	0/2	1/2
Avaricious and Envious	1/1	1/1
Belling the Cat	1/1	1/1
Hercules and the Waggoner	2/2	2/2
The Ant and the Grasshopper	2/2	0/2
The Four Oxen and the Lion	0/1	1/1
The Fox and the Cat	0/3	1/3
The Fox and the Crow	3/4	2/4
The Fox and the Goat	1/1	0/1
The Fox and the Grapes	1/1	0/1
The Wind and the Sun	1/1	0/1
The Wolf and the Crane	0/2	0/2
The Wolf and the Kid	0/1	0/1
The Wolf and the Lamb	1/2	2/2
The Tortoise and the Birds	1/3	1/3
Overall	*14/27*	*13/27*

Table 5. Comparison of detected sentiments vs. actual sentiments

Discussion

Possible ideas for future work include: adding visual cues to co-occurrence graphs - nodes that represent good characters have green colors, while those that represent bad ones are colored red. Moreover, the edges could also have visual cues. An edge that connects the antagonist with the protagonist could be darker in color, while one connecting friends would be lighter.

Some way of mitigating indirect mentions could prove productive in all stages of the pipeline. That way all pronouns that refer to entities are actually taken into consideration in subsequent sentiment and co-occurrence analysis.

References

[1] Arthur M Jacobs. Sentiment analysis for words and fiction characters from the perspective of computational (neuro-) poetics. *Frontiers in Robotics and AI*, 6:53, 2019.

[2] 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.

[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, 2012.

[4] Thomas K Landauer, Peter W Foltz, and Darrell Laham. An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284, 1998.

[5] Vincent Labatut and Xavier Bost. Extraction and analysis of fictional character networks: A survey. *ACM Computing Surveys (CSUR)*, 52(5):1–40, 2019.

[6] David Bamman, Sejal Popat, and Sheng Shen. An annotated dataset of literary entities. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2138–2144, 2019.

[7] Rolf A. Zwaan, Joseph P. Magliano, and Arthur C. Graesser. Dimensions of situation model construction in narrative comprehension. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 21(2):386–397, March 1995.

[8] Rolf A. Zwaan. Situation models. *Current Directions in Psychological Science*, 8(1):15–18, February 1999.

[9] Nachiappan Chockalingam. Simple and effective feature based sentiment analysis on product reviews using domain specific sentiment scores. *POLIBITS*, 57:39–43, 2018.

[10] F. Å. Nielsen. Afinn, mar 2011.