FRI Natural language processing course 2023
Project report

University of Ljubljana
Faculty of Computer and
Information Science

Literacy situation models knowledge base creation

Aljaž Sebastjan Ahtik, Jaša Kerec, Luka Kuzman

Abstract

The abstract goes here. The abstract goes here.

Keywords
Keyword1, Keyword2, Keyword3 ...

Advisors: Slavko Žitnik

Introduction

Text comprehension is a difficult task in the field of natural language processing, as it requires solving multiple sub tasks. In this project, we strive to build a knowledge base for a number of Slovenian and English novels which focuses on exploring the relationship sentiment between the characters in the stories, namely the antagonist/protagonist relationship. Using it, we can quickly grasp the contents and ideas of the books, even if we have not read it before.

For the purposes of our project, we constructed a corpus which consists of:

- **7 English novels** to test the correctness of our solutions,
- **33 Slovenian short stories** with simple plot lines using which we will determine the correctness of our solution on Slovenian texts and
- **21 Slovenian novels** which are more advanced texts and will be used to test the performance of our implementations on more difficult cases.

Due to a large amount of information we can extract from the corpus we limit ourselves to a small subset of them. Our initial plan consists of named entity recognition, relationship recognition, relationship sentiment analysis, antagonist and protagonist detection and visualization of the data.

Related work

When analysing stories the first task is often named entity recognition. One approach to solving this problem is by using large databases of language specific known entity names, however in most languages and domains there is very few training data available. Lample et al. [1] present two different LSTM-based models for named entity recognition that capture ortographic and distrubutional evidence without resorting to any language-specific resources.

The next important part of the story analysis is relation extraction. This problem is generally solved either through supervised or unsupervised learning algorithms. For using supervised algorithms we need a text corpus for which the entities and their relation types are known, which is not always the case. So the authors of [2] proposed a hybrid approach. This approach identifies the main characters and collects the sentences related to them. The collected sentences are then processed and classified to extract relationships between characters.

Sentiment analysis deals with understanding emotional tone behind a string of text. Sentiment analysis approaches can be split into three categories: machine learning based (such as Naive Bayes, Support Vector Machine and Maximum Entropy), lexicon/corpus based (which employs a sentiment dictionary, one such being NRC Emotion Lexicon [3], the drawback being that a human must be present) and hybrid methods. Article [4] present some existing solutions regarding sentiment analysis.

In order for the model to better understand the text, it needs to employ some common sense reasoning. Because of this many databases of commonsense knowledge were built, however, the data is spread over many sources with different foci. Ilievski et al. [5] attempt to combine these different sources into one knowledge graph which comes with three key challenges: (1) the different knowledge modeling approaches, (2) imprecise descriptions of entities, and (3) the sparse overlap between the sources. They achieve this by constructing a commonsense knowledge graph linking seven key sources.

Another important concept in natural language processing is the concept of causality, which can informally be described as a relationship between two events such that one event causes the other. Shingo Nahatame [6] investigates the properties of global and local causality, semantic text relations and their effect on L2 readers' memory, finding that global structure of the text rather local has a stronger impact on how well the subject remembers the text. This is in contrast to semantic relations, where local relations have a larger impact. In light of this information, we investigate a work by Tirthankar et al. [7] which propose a linguistically informed deep neural network in order to extract casual relations from documents, finding that a bi-directional LSTM performs well on the task. Sendong Zhao et al.[8] employ an approach using Restricted Hidden Naive Bayes model to extract causality. The advantage of this approach is in its ability to cope with partial interaction amongst features, which helps avoid overfitting present with Hidden Naive Bayes model. Besides better text comprehension, causality is also useful when predicting medical results, future natural disasters and their aftereffects etc.

Caselli, T. and P. Vossen in [9] presented a new dataset for training and evaluating models for causal and temporal relation extraction. They also presented three baseline systems with their performance on the dataset which showed how complex the task is and gave directions for the development of more robust systems. The dataset (corpus) is meant to provide an intrinsic evaluation benchmark for the StoryLine Extraction task. The task is composed of three basic parts. (1) Event detection and classification - detect and classify events (which compose a topic) in each document. (2) Temporal anchoring of events - Anchor each event mention with the time in which it happened. (3) Explanatory Relation Identification and Classification - classify the storyline relation type based on the selection of event pairs that are temporally and logically connected.

The methods presented reach beyond the scopes of this project. For example, sentiment analysis finds its use in medicine. One such overview is presented by Kerstin Denecke and Yihan Deng [10].

Methods

We evaluate our methods on three different datasets. In this section we will describe our entire dataset and all the methods we use. We will also define and explain every step in our pipeline for protagonist and antagonist detection.

Datasets

Our corpus consists of three datasets. First one is IMapBook project dataset in english. It consists 7 short stories:

- **The Ransom of Red Chief** (about 4.160 words).
- **Hills Like White Elephants** (about 1.470 words).
- **Leiningen Versus the Ants** (about 8.610 words).
- **The Lady, or The Tiger** (about 2.710 words).
- **The Most Dangerous Game** (about 7.950 words).
- **The Tell-Tale Heart** (about 2.200 words).
- **The Gift of the Magi** (about 1.860 words).

These stories are considered classics in English literature and have been widely read and studied by scholars, students and enthusiasts alike. The dataset is suitable for use in various natural language processing tasks. The stories were originally published in the late 19th and early 20th centuries, and offer a glimpse into the socio-cultural and historical context of that period. They showcase a variety of literary styles and themes, including suspense, irony, humor, and romance. Overall, this dataset provides a rich and diverse source of literary content for researchers and practitioners working in the field of natural language processing and related areas.

In the second dataset are 33 Slovenian short folktales. A folktale is a traditional story that has been passed down orally from generation to generation within a particular culture or community. These stories typically have anonymous authors and are often part of a larger body of oral tradition. Folktales may involve animals, supernatural creatures, or human characters and are usually used to teach moral lessons, explain natural phenomena, or simply entertain. On average, the stories from this dataset are a little bit shorter than the stories from IMapBook project. They also include less characters as the previous mentioned stories, since there are rarely more than 4 in one story. In many stories there are no specific names of the characters or the main characters are animals.

The last dataset consists of a collection of Slovenian novels, each with a word count ranging from 1000 to 10000 words. The novels cover a wide range of genres, including romance, adventure, and mystery, and were written by a diverse group of authors. Overall, this dataset provides a rich and varied source of Slovenian literary content for analysis and exploration.

Named entity recognition

Named entity recognition (NER) is a process for determining which of the input tokens represent a named entity and is a fundamental step in identifying the relationship between characters in stories [2]. NER is a very challenging problem, especially in less common and complicated languages like Slovenian, where there is only a small amount of supervised training data available and there are few constraints on the word form of named entities. ”As a result, carefully constructed orthographic features and language-specific knowledge resources, such as gazetteers, are widely used for solving this task” [1], however these are costly to develop and adapt.

More recently, semi-supervised machine learning methods are being used to learn the word grouping, orthographical, and distributional evidence of named entities. In this paper we will use the built in NER preprocessors from Stanza [11] for the English language and from Stanza based Classla [12] for the Slovenian language.

Coreference resolution

The important step before named entity extracion is coreference resolution. Because often characters are described with pronouns instead of full names. Coreference resolution (CR) is a task that involves identifying all expressions in a text that refer to the same entity. It is a crucial step in many NLP applications, including information extraction, machine translation, question answering, and sentiment analysis. The goal of CR is to enable NLP systems to correctly identify and resolve coreferent expressions, which can greatly improve the accuracy and performance of downstream tasks. The task of CR is challenging because coreferent expressions can have different forms and can appear in different parts of the text. For example, consider the sentence ”John went to the store. He bought some milk.” The pronoun ”he” refers back to the noun phrase ”John”, and therefore, the two expressions are coreferent. In a given example we can see that in the second sentence ”John” is replaces with ”he” so the NER algorithm will not detect that as a ”John” entity. But coreference resolution does exactly that, it replaces pronouns with actual entities. For the coreference resolution in Python we use the original implementation of Fastcoref method described in [13]. It is based on a neural network architecture that uses a combination of syntactic and semantic features to identify and resolve coreferent expressions in text. One of the key features of Fastcoref [13] is its speed. It can process large amounts of text in a matter of seconds, making it suitable for use in real-time applications. This is achieved through the use of efficient data structures and optimized algorithms, as well as a parallel processing pipeline that takes advantage of multi-core CPUs and GPUs. For the Slovenian language we used the BERT based coreference model [14].

Lists

We can insert numbered and bullet lists:

1. First item in the list.
2. Second item in the list.
3. Third item in the list.

- First item in the list.
- Second item in the list.
- Third item in the list.

We can use the description environment to define or describe key terms and phrases.

Word What is a word?.

Concept What is a concept?

Idea What is an idea?

Random text

This text is inserted only to make this template look more like a proper report. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam blandit dictum facilisis. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Interdum et malesuada fames ac ante ipsum primis in faucibus. Etiam convallis tellus velit, quis ornare ipsum aliquam id. Maecenas tempus mauris sit amet libero elementum eleifend. Nulla nunc orci, consectetur non consequat ac, consequat non nisl. Aenean vitae dui nec ex fringilla malesuada. Proin elit libero, faucibus eget neque quis, condimentum laoreet urna. Etiam at nunc quis felis pulvinar dignissim. Phasellus turpis turpis, vestibulum eget imperdiet in, molestie eget neque. Curabitur quis ante sed nunc varius dictum non quis nisl. Donec nec lobortis velit. Ut cursus, libero efficitur dictum imperdiet, odio mi fermentum dui, id vulputate metus velit sit amet risus. Nulla vel volutpat elit. Mauris ex erat, pulvinar ac accumsan sit amet, ultrices sit amet turpis.

Phasellus in ligula nunc. Vivamus sem lorem, malesuada sed pretium quis, varius convallis lectus. Quisque in risus nec lectus lobortis gravida non a sem. Quisque et vestibulum sem, vel mollis dolor. Nullam ante ex, scelerisque ac efficitur vel, rhoncus quis lectus. Pellentesque scelerisque efficitur purus in faucibus. Maecenas vestibulum vulputate nisl sed vestibulum. Nullam varius turpis in hendrerit posuere.

Figures

You can insert figures that span over the whole page, or over just a single column. The first one, Figure 1, is an example of a figure that spans only across one of the two columns in the report.

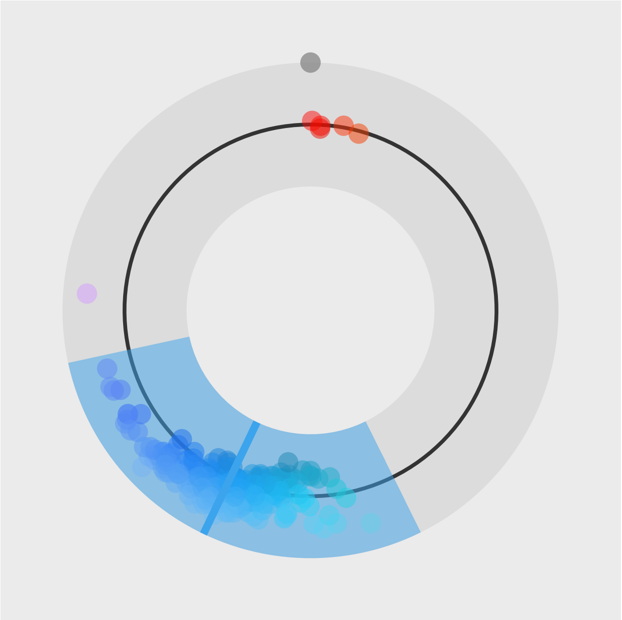

Figure 1. A random visualization. This is an example of a figure that spans only across one of the two columns.

On the other hand, Figure 2 is an example of a figure that spans across the whole page (across both columns) of the report.

Tables

Use the table environment to insert tables.

Table 1. Table of grades.

Name		
First name	Last Name	Grade
John	Doe	7.5
Jane	Doe	10
Mike	Smith	8

Code examples

You can also insert short code examples. You can specify them manually, or insert a whole file with code. Please avoid inserting long code snippets, advisors will have access to your repositories and can take a look at your code there. If necessary, you can use this technique to insert code (or pseudo code) of short algorithms that are crucial for the understanding of the manuscript.

Listing 1. Insert code directly from a file.

```
import os
import time
import random

fruits = ["apple", "banana", "cherry"]
for x in fruits:
  print(x)
```

Listing 2. Write the code you want to insert.

```
import(dplyr)
import(ggplot)

ggplot(diamonds,
           aes(x=carat, y=price, color=cut)) +
  geom_point() +
  geom_smooth()
```

Results

Use the results section to present the final results of your work. Present the results in a objective and scientific fashion. Use visualisations to convey your results in a clear and efficient manner. When comparing results between various techniques use appropriate statistical methodology.

More random text

This text is inserted only to make this template look more like a proper report. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam blandit dictum facilisis. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Interdum et malesuada fames ac ante ipsum primis in faucibus. Etiam convallis tellus velit, quis ornare ipsum aliquam id. Maecenas tempus mauris sit amet libero elementum eleifend. Nulla nunc orci, consectetur non consequat ac, consequat non nisl. Aenean vitae dui nec ex fringilla malesuada. Proin elit libero, faucibus eget neque quis, condimentum laoreet urna. Etiam at nunc quis felis pulvinar dignissim. Phasellus turpis turpis, vestibulum eget imperdiet in, molestie eget neque. Curabitur quis ante sed nunc varius dictum non quis nisl. Donec nec lobortis velit. Ut cursus, libero efficitur dictum imperdiet, odio mi fermentum dui, id vulputate metus velit sit amet risus. Nulla vel volutpat elit. Mauris ex erat, pulvinar ac accumsan sit amet, ultrices sit amet turpis.

Phasellus in ligula nunc. Vivamus sem lorem, malesuada sed pretium quis, varius convallis lectus. Quisque in risus nec lectus lobortis gravida non a sem. Quisque et vestibulum sem, vel mollis dolor. Nullam ante ex, scelerisque ac efficitur vel, rhoncus quis lectus. Pellentesque scelerisque efficitur purus in faucibus. Maecenas vestibulum vulputate nisl sed vestibulum. Nullam varius turpis in hendrerit posuere.

Nulla rhoncus tortor eget ipsum commodo lacinia sit amet eu urna. Cras maximus leo mauris, ac congue eros sollicitudin ac. Integer vel erat varius, scelerisque orci eu, tristique purus. Proin id leo quis ante pharetra suscipit et non magna. Morbi in volutpat erat. Vivamus sit amet libero eu lacus pulvinar pharetra sed at felis. Vivamus non nibh a orci viverra rhoncus sit amet ullamcorper sem. Ut nec tempor dui. Aliquam convallis vitae nisi ac volutpat. Nam accumsan, erat eget faucibus commodo, ligula dui cursus nisi, at laoreet odio augue id eros. Curabitur quis tellus eget nunc ornare auctor.

Discussion

Use the Discussion section to objectively evaluate your work, do not just put praise on everything you did, be critical and exposes flaws and weaknesses of your solution. You can also explain what you would do differently if you would be able to start again and what upgrades could be done on the project in the future.

Acknowledgments

Here you can thank other persons (advisors, colleagues ...) that contributed to the successful completion of your project.

References

[1] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition, 2016.

[2] V. Devisree and Reghu P.C. A hybrid approach to relationship extraction from stories. *Procedia Technology*, 24:1499–1506, 12 2016.

[3] Saif M Mohammad and Peter D Turney. Nrc emotion lexicon. *National Research Council, Canada*, 2:234, 2013.

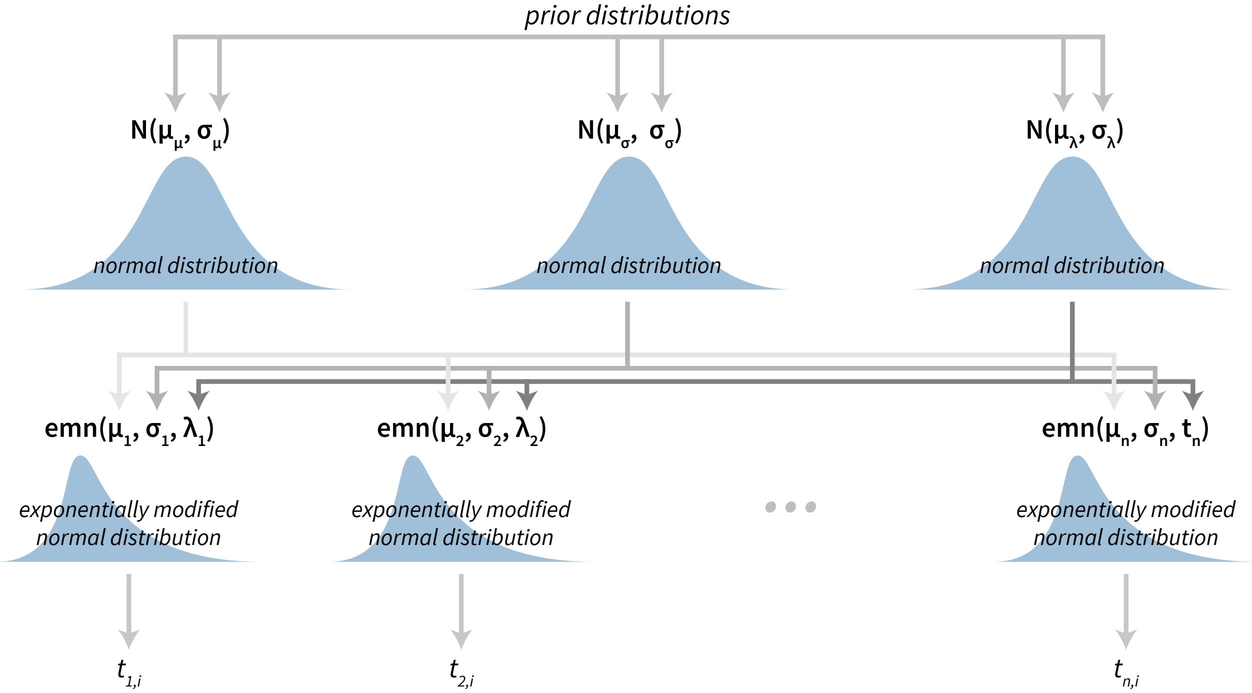
prior distributions
$N(\mu_\mu, \sigma_\mu)$
normal distribution
$N(\mu_\sigma, \sigma_\sigma)$
normal distribution
$N(\mu_\lambda, \sigma_\lambda)$
normal distribution
$emn(\mu_1, \sigma_1, \lambda_1)$
exponentially modified normal distribution
$emn(\mu_2, \sigma_2, \lambda_2)$
exponentially modified normal distribution
$emn(\mu_n, \sigma_n, t_n)$
exponentially modified normal distribution
$t_{1,i}$
$t_{2,i}$
$t_{n,i}$

Figure 2. Visualization of a Bayesian hierarchical model. This is an example of a figure that spans the whole width of the report.

[4] Doaa Mohey El-Din Mohamed Hussein. A survey on sentiment analysis challenges. *Journal of King Saud University-Engineering Sciences*, 30(4):330–338, 2018.

[5] Filip Ilievski, Pedro A. Szekely, and Bin Zhang. CSKG: the commonsense knowledge graph. *CoRR*, abs/2012.11490, 2020.

[6] Shingo Nahatame. Revisiting second language readers' memory for narrative texts: the role of causal and semantic text relations. *Reading Psychology*, 41(8):753–777, 2020.

[7] 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, 2018.

[8] Sendong Zhao, Ting Liu, Sicheng Zhao, Yiheng Chen, and Jian-Yun Nie. Event causality extraction based on connectives analysis. *Neurocomputing*, 173:1943–1950, 2016.

[9] Tommaso Caselli and Piek Vossen. The event StoryLine corpus: A new benchmark for causal and temporal relation extraction. In *Proceedings of the Events and Stories in the News Workshop*, pages 77–86, Vancouver, Canada, August 2017. Association for Computational Linguistics.

[10] Kerstin Denecke and Yihan Deng. Sentiment analysis in medical settings: New opportunities and challenges. *Artificial intelligence in medicine*, 64(1):17–27, 2015.

[11] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2020.

[12] Nikola Ljubešić and Kaja Dobrovoljc. What does neural bring? analysing improvements in morphosyntactic annotation and lemmatisation of Slovenian, Croatian and Serbian. In *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*, pages 29–34, Florence, Italy, August 2019. Association for Computational Linguistics.

[13] Shon Otmazgin, Arie Cattan, and Yoav Goldberg. F-coref: Fast, accurate and easy to use coreference resolution. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 48–56, Taipei, Taiwan, November 2022. Association for Computational Linguistics.

[14] Matej Klemen and Slavko Žitnik. Neural coreference resolution for slovene language. *Computer Science and Information Systems*, 19(2):495–521, 2022.