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

Literacy situation models knowledge base creation

Jan Bajt, Anže Habjan, and Tadej Stanonik

Abstract

The abstract goes here. The abstract goes here.

Keywords

keyword extraction, natural language processing

Advisors: Slavko Žitnik

Introduction

Today, we have access to a vast amount of literature that provides us with the information we need. To help us quickly distinguish which literature has the information that we need, scientific articles have the most important keywords listed. Keyword extraction is a crucial technique for automatically identifying the most important words in a text document, and has become a popular research area in the field of natural language processing (NLP). By identifying the key topics of a document, and extracting the most relevant and informative words that represent those topics, keyword extraction can help us quickly gather literature with desired information.

Graph based text representation does not only enable us to utilize the power of graph theory and network analysis as a scientific field, but also unfolds us new aspects of text *e. g.* relationship between words, missing from traditional NLP text representations. It is for that reason, the graph-based keyword extraction should be exploited to a further extent. Another notable benefit of employing such a technique is the non-necessity of annotated dataset exploitation, for which manual annotation in our case for short stories would be a tedious task. In addition using models pre-trained on data emerging from other domains would not prove to yield satisfactory level results.

As per Sonawane et. al [1] there are numerous ways text can be represented as a graph. More precisely edge relations *i. e.* relation between two terms (nodes) can be derived from different term relations *e. g.* term occurring together, common terms, terms with similar meaning, opposite terms etc. Additionally setting node and edge weights, a diversity of methods are available, perhaps a naive one could be term frequency. Aforementioned divergent modalities in graph-based textual representation have crucial aspect of network analysis.

In our work we construct a graph from text, and then using network analysis approaches such as centrality measures (closeness, betweenness), degree distribution, clustering coefficient etc. try to extract keywords of graph represented text. We aim to use previously stated distinct variations of graph constructing from text. Our goal is to then use network analysis for keyword extraction on a variation of graphs constructed from same text. Following this method we could compare different graph representation methods to obtain the most appropriate graph-based text representation for our use case.

Related work

In [2] the authors propose a novel unsupervised graph based keyword extraction method called Keyword Extraction using Collective Node Weight (KECNW) which determines the importance of a keyword by collectively taking various influencing parameters. The KECNW is based on Node Edge rank centrality with node weight depending on various parameters. The model is validated with five datasets: Uri Attack, American Election, Harry Potter, IPL and Donald Trump. The result of KECMW is compared with three existing models. It is observed from the experimental results that the proposed method is far better than the others. The performances are

shown in terms of precision, recall and F-measure.

In article [3] authors proposed and evaluated two graph-based approaches: supervised and unsupervised, for the cross-lingual keyword extraction to be used in extractive summarization of text documents. In the supervised approach, they trained classification algorithms on a summarized collection of documents with the purpose of inducing a keyword identification model. In the unsupervised approach, they ran the HITS algorithm on document graphs under the assumption that the top-ranked nodes should represent the document keywords. Their results show that given a set of summarized training documents, the supervised classification provides the highest keyword identification accuracy, while the highest F-measure is reached with a simple degree-based ranking.

In article [4] authors compared different keyword extraction systems for Slovene text. On Slovene news corpora they compared methods from two main groups of such systems: supervised and unsupervised. For supervised approaches they evaluated TNT-KID and BERT and for the unsupervised systems they evaluated five different methods from three different groups of approaches like statistical, graph-based and embedding-based. Their results show that supervised models perform significantly better than unsupervised approaches.

Methods

Use the Methods section to describe what you did an how you did it – in what way did you prepare the data, what algorithms did you use, how did you test various solutions ... Provide all the required details for a reproduction of your work.

Below are LaTeXexamples of some common elements that you will probably need when writing your report (e.g. figures, equations, lists, code examples ...).

Equations

You can write equations inline, e.g. $\cos\pi = -1$, $E = m \cdot c^2$ and α, or you can include them as separate objects. The Bayes's rule is stated mathematically as:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}, \tag{1}$$

where A and B are some events. You can also reference it – the equation 1 describes the Bayes's rule.

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.

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

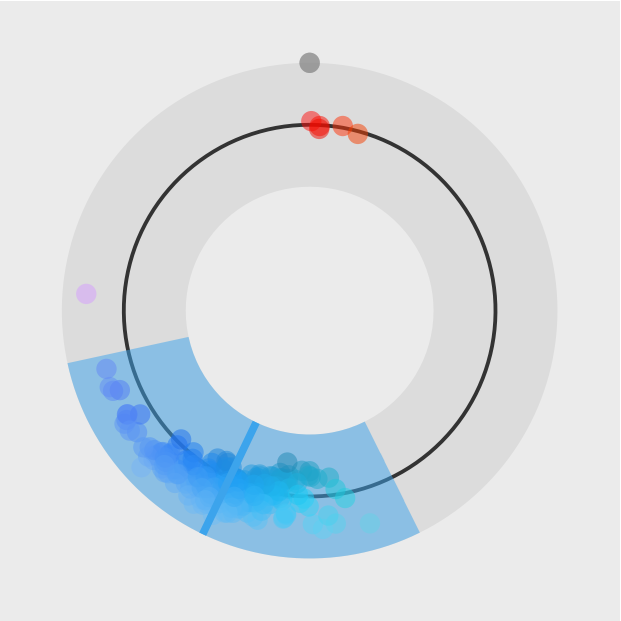

Figure 1. A random visualization. This is an example of a figure that spans only across one of the two columns.

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] Sheetal Sonawane and P. Kulkarni. Graph based representation and analysis of text document: A survey of techniques. *International Journal of Computer Applications*, 96:1–8, 06 2014.

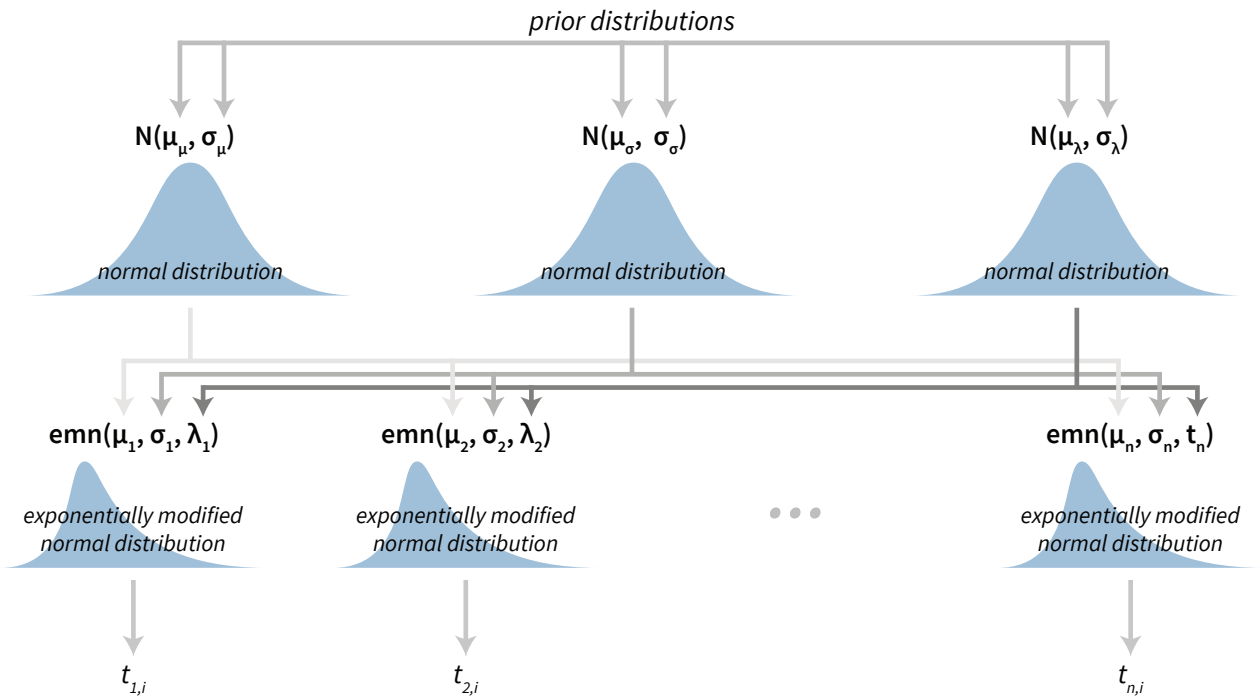

Figure 2. Visualization of a Bayesian hierarchical model. This is an example of a figure that spans the whole width of the report.

[2] Saroj Kr Biswas, Monali Bordoloi, and Jacob Shreya. A graph based keyword extraction model using collective node weight. *Expert Systems with Applications*, 97:51–59, 2018.

[3] Marina Litvak and Mark Last. Graph-based keyword extraction for single-document summarization. In *Coling 2008: Proceedings of the workshop multi-source multilingual information extraction and summarization*, pages 17–24, 2008.

[4] Boshko Koloski, Senja Pollak, and Matej Martinc. What works for slovenian? a comparative study of different keyword extraction systems. 09 2022.