FRI Natural language processing course 2022
Project report

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

Cross lingual question answering

Martin Božič, Matic Šincek, and Jakob Maležič

Abstract

The abstract goes here. The abstract goes here.

Keywords

question-answering, cross-lingual, natural-language-processing

Advisors: Slavko Žitnik

Introduction

Reading Comprehension (RC), or the ability to read text and then answer questions about it, is a challenging task for machines. There has been s lot of examples of systems that achieved good results in this area in the past. One example is simple system Quarc [1] from year 2000 which does not use a lot of syntactic analysis but uses part-of-speech tagging, semantic class tagging, and entity recognition. It differentiates between who, when, where, why and what questions and looks for keywords that are useful for identifying the person, time, place, or intent in sentences. It has the most problems with answering what questions, since there is a variety of different ways to answer them. The system was used on reading comprehension tests for children. It achieved 40% accuracy on the given dataset. Another example is Watson [2], the question answering system developed by IBM in 2010, which was built to try compete with the top human competitors on the well-known U.S. TV quiz Jeopardy. IBM devised the ”DeepQA atchitecture” which combines many different algorithms that address many different problems in question answering and now performs at human expert levels in terms of precision and confidence. It classifies The knowledge for the answering process was extracted from a wide range of encyclopedias, dictionaries, thesauri, newswire articles, literary works and more, as the system is not connected to the internet during the show. The process that took 2 hours to answer a single question on a single cpu with 70% accuracy at first was then highly parallelized by the IBM team and can now answer 80% of the questions in under 5 seconds.

In recent years transformer models significantly outperform traditional deep neural networks on various NLP tasks. Perhaps one of more important steps in world of NLP was introduction of BERT language model, which consists of encoder part of transformer architecture.

BERT language model has proven successful at most machine learning comprehension (RC) dataset. Wang, Ng, Ma, Nallapati and Xiang in [3] want to extend BERT models from RC task, where model only needs to find an answer from a given paragraph and which is simplified version of QA task, to open-domain question answering system, which is able to pinpoint answers from a massive article collection, that can often include entire web. They show that global normalization makes QA model more stable while pinpointing answers from large number of paragraphs. They get 4% improvements by splitting articles into passages with the length of 100 words. They manage to get extra 2% improvements by leveraging a BERT-based passages ranker and they find out that explicit inter-sentence matching is not helpful for BERT.

In [4] a Stanford Question Answering Dataset (SQuAD) is presented, that consists of 100,000+ questions posed by crowdworkers on a set of Wikipedia articles, where the answer to each question is a segment of text from the corresponding reading passage. For retrieving high-quality articles they used Wikipedia's internal PageRanks to obtain top 1000 articles of English Wikipedia, from which they sampled 546 articles uniformly at random. From these they extracted paragraphs and discarded those, that were shorter than 500 characters. The result was 23,215 paragraphs for the 536 articles covering a wide range of topics. They created a collection of questions and answers by employing crowdworkers. For each paragraph, crowdworkers had to prepare up to 5 questions and answers on the content of that paragraph. They were encouraged to ask the questions in their own words, without copying word phrases from the paragraph. For the baseline, they implemented a sliding window approach and the distance-based extensions for the sliding window approach, as described by Richardson et al. in [5]. Then they implemented a logistic regression model and compare its accuracy with that of the baseline methods.

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.

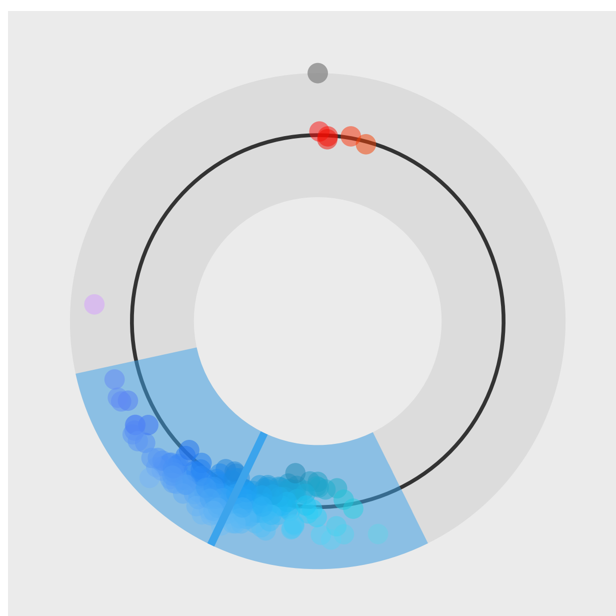

Figure 1. A random visualization. This is an example of a figure that spans only across one of the two columns.

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] Ellen Riloff and Michael Thelen. Rule-based question answering system for reading comprehension tests. *ANLP/NAACL-2000 Workshop on Reading Comprehension Tests as Evaluation for Computer-Based Language Understanding Systems*, 6, 05 2000.

[2] David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A. Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John Prager, Nico Schlaefer, and Chris Welty. Building watson: An overview of the deepqa project. *AI Magazine*, 31(3):59–79, Jul. 2010.

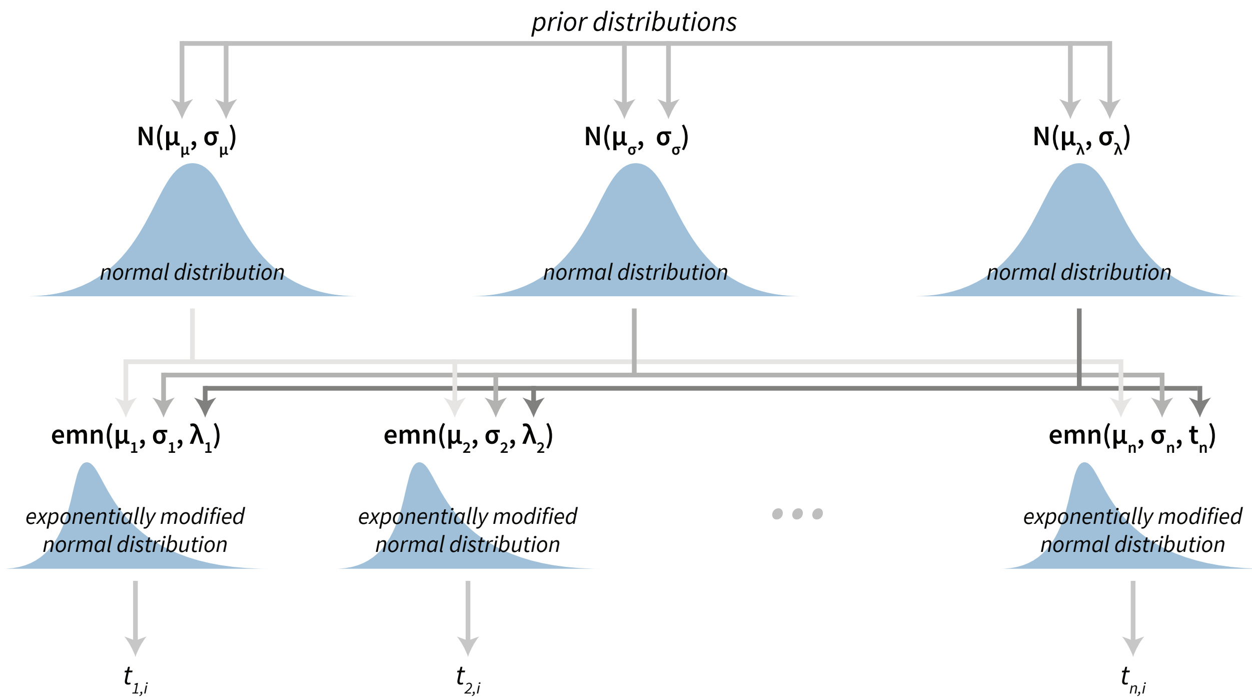

Figure 2. Visualization of a Bayesian hierarchical model. This is an example of a figure that spans the whole width of the report.

[3] Zhiguo Wang, Patrick Ng, Xiaofei Ma, Ramesh Nallapati, and Bing Xiang. Multi-passage bert: A globally normalized bert model for open-domain question answering. *arXiv preprint arXiv:1908.08167*, 2019.

[4] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics.

[5] Matthew Richardson, Christopher JC Burges, and Erin Renshaw. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 193–203, 2013.