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

Cross lingual question answering

Martin Božič, Matic Šincek, and Jakob Maležič

Abstract

The goal of the project is to prepare an English, Slovene and multilingual English and Slovene model for question answering system. Stanford Question Answering Dataset (SQuAD 2.0) [1] is used as the main corpora. For training a Slovene model, SQuAD corpora is automatically translated to Slovene, using the EK translator [2].

Keywords

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

Advisors: Slavko Žitnik

Introduction

Question answering consists of text reading and system question answering based on read knowledge. We call this process Reading Comprehension (RC). RC is usually a challenging task for machines. In this article we mostly focus on question answering performed with texts of limited scope. For the whole development process we use data collected from Open-SQuAD dataset [3], which is a collection of text paragraphs and answers in English. First we develop a system that is able to read English paragraphs and answer them in English, then we translate dataset using EK translator and fine-tune and apply the same model on Slovene texts. Here we face some difficulties, e.g. we have to translate paragraphs, questions and paragraphs and sometimes it happens, that in the process of translation, correct answers to translated paragraphs are lost. We tackle these issues using additional preprocessing steps.

There has been s lot of examples of systems that achieved good results on various RC tasks. One example is simple system Quarc [4] 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 [5], 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 [6] 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 [3] 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 [7]. Then they implemented a logistic regression model and compare its accuracy with that of the baseline methods.

In [8] the TriviaQA a reading comprehension question answering dataset is described. It contains over 650.000 question-answer-evidence pairings. Some of the data is provided by trivia enthusiasts as question-answer pairs are collected from 14 trivia websites while the evidence part is extracted from documents from wikipedia and the web. It contains more than three times as many questions that require the knowledge of more than one sentence than SQuAD, making it a challenge for QA models. TriviaQA is the first dataset where full-sentence questions are authored organically (i.e. independently of an NLP task). This decoupling of question generation from evidence collection allows us to control for potential bias in question style or content, while offering organically generated questions from various topics.

The Natural Questions dataset, presented in [9], is a dataset consisting of real anonymized queries issued to the Google search engine. Becuse the questions are considered natural because they were asked by real people. Natural Questions contains (question, wikipedia page, long answer, short answer) quadruples. Human annotators were given the question together with a Wikipedia page from the top-5 search results, and marked a long answer (a paragraph) and a short answer (the target span) in the article, or null if the answer was not present in the article. Questions were excluded if the annotators classified them as ambigous, incomprehensible or otherwise misleading. Annotation quality was constantly monitored. It consists of over 300.000 question-answer pairs along with the evidence documents.

The Children's Book Test (CBT) dataset, described in [10] is a reading comprehension dataset consisting of 108 children's books that are freely available thanks to Project Gutenberg. An example consists of context, query, answer candidates and the actual answer. One word is removed from each sentence, making it a question of the removed word. The same word is the solution to the question that is formed. The answer candidates list is a list of words from the context of the same word type as the answer. It consists of more than 680.000 (context, question) pairs.

MS MARCO [11] is another big dataset which contains search queries submitted via Bing or Cortana. For each query, the dataset contains text passages from documents that are provided by Bing as a result for the given query. It consists of more than one million anonymized questions making it more than ten times larger than SQuAD. In addition, the dataset contains more than 3 million web documents retrieved by Bing—that provide the information necessary for handling the natural language answers. Questions may have one answer, multiple answers or no answers at all. The text is produced by real users and so it, like the Natural Questions, includes typos and abbreviations.

When fine tuning multilingual BERT models on squad dataset, using other non-English languages, we have to firstly translate original dataset and the fine tune models on translated dataset. This can be a tricky part, cause automatic translations usually aren't on the same quality level as original dataset, which can lead to drop in model performance.

One such example is described in [12], where they firstly translate squad dataset into Spanish and fine tune BERT multilingual model on translated dataset. For translation purposes, they develop and use Translate Align Retrieve (TAR) method. They evaluate their QA models with the recently proposed MLQA and XQuAD benchmarks for cross-lingual Extractive QA and manage to obtain 68.1 F1 points on the Spanish MLQA corpus and 77.6 F1 points on the Spanish XQuAD corpus.

When looking for models fine tuned on English squad dataset, we can find models surpassing even human level performance. These state of the art models are constructed using ensamble of different fine tuned BERT family models, with current state of the art model achieving F1 score of 93.214. This surpasses human performance, which can be set around 89.452 F1 points.

In [13] they fine tune multiple variations of BERT models, where using BERT-base PyTorch Implementation model, they mamage to obtain F1 score of 76.70, which they further improve to 79.443 F1 points, using BERT large case model.

Methods

Translation

First we translated the whole SQuAD to Slovene using online automatic translator EK translator [2]. Since the data was too big, we split it into multiple parts and translated each part individually and then combined the parts back together.

Translator translated answers and context independently of each other and therefore produced answers that are not exactly the same as they appear in the context. That is why we had to implement an algorithm to fix the answers and the answers' starting index so they are exactly the same as they appear in the context. We did this by implementing an algorithm which is similar to the weighted n-gram approach that is described in [14].

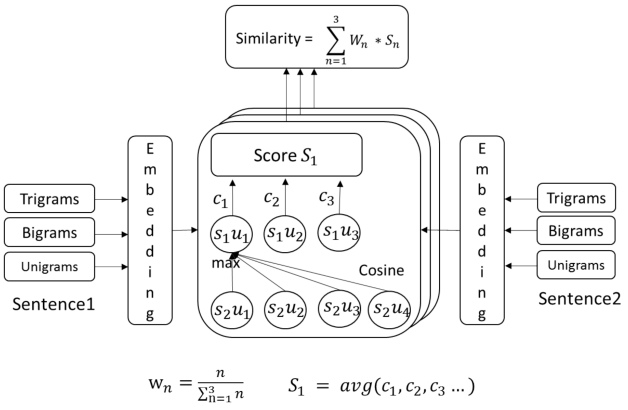

Figure 1. Weighted N-gram approach. Diagram for algorithm with weighted N-grams [14].

First we clean the text by converting all characters to lower case and remove characters that would worsen the result when calculating similarity between word embeddings. Such characters are: apostrophes, dots, braces etc. To find the answer in the context, we first reduce the context based on the index from the original dataset. We do this by taking 200 characters before and after the original index as the reduced context in which we will try to find the answer. We then split the context into *n*-grams. *N*-grams are consecutive string of *n* words. For example, 3-grams are all possible three word long sub-strings of a given sentence. If answer contains *m* words, we split the context into *n*-grams of length 1 to *m*. We then convert *n*-grams to word embeddings using pre-trained FastText [15] model and calculate cosine similarity between word embeddings for every *n*-gram of context and answer. We then take *n*-grams with highest similarity for each length from 1 to *m* and calculate the weighted similarity. If the weighted similarity is smaller than 0.75, we remove the answer, otherwise, the n-gram of length *m* with the best similarity is used as the corrected answer. The following script shows an example of finding the answer *francija* in the context.

```
# indexed context:
17: dali 18: ime 19: normandiji 20: regiji 21: v
22: franciji 23: bili 24: so 25: potomci 26: nordijskih

# original answer:
francija

# new answer:
franciji (index: 22, similarity: 0.66)
```

One of the drawbacks of using this approach is the need to clean the text beforehand. Especially converting all characters to lower case might have worsen the result of the model. However, this had to be done, because cosine similarity between word embeddings that were calculated FastText gave bad results without it. Even after the cleaning, the similarity doesn't work as well as we would expect. Even from the previous example, we can see that although the original and new answer differ only by one character, the similarity is not very high. However, there are also examples, where the answers are not similar at all, but the similarity is high. That's why had to set a pretty high threshold for removing the answers which lead to shrinking the corpus by quite a lot.

We tested the model on fixed dataset, that is the dataset where we just fixed the answers to the most similar answer from the context and on the reduced dataset, that had 20313 less answers.

Dataset preparation

We use translated dataset and construct training, testing and development data sets. We take original translated text and transform it into structure, where each sample of data set consists of context, single question and one or multiple answers to this question.

Our final prepared data set consists of 92748 such data samples. Out of which we use 86820 data samples for training, 1000 for development and 4928 for test data set.

We further prepare our dataset, in order to use it for model fine tuning. We combine each question with context and merge them in one single vector. We then tokenize this vector using pretrained BERT tokenizer.

Some of these vectors can be very long, sometimes even longer than 512 tokens, which is maximum input length to our BERT model. In these cases we split context into multiple parts, where each two different context parts have a shared part, length of which we set to 128 tokens. We then look for each question, to which part of context it belongs, and construct final input vectors.

If single question has multiple answers, we always use best fitting answer, when evaluating and fine tuning the model.

Model fine tuning

Firstly we fine tune bert-base-multilingual-uncased model, which is a base model originally trained on top of 102 languages with the largest Wikipedia, using a masked language modeling (MLM) objective. Model is originally described in [16]. Slovene is also part of the 102 language corpus.

Further we fine tune the XLM-RoBERTa-base model, which is a base model originally trained on top of 100 languages and is based on Facebook's RoBERTa model, released in 2019. Model is originally described in [17].

We also fine tune XLM-RoBERTa-large model, which is similar to base model, with the main difference in the number of layers. Large model consists of 24 layers, while base model consists of 12 layers.

We take each pretrained model, together with it's tokenizer, from hugging face library. For model fine tuning we use the Pytorch library.

Lastly we fine tune model on original English squad dataset. When we are satisfied with results obtained by testing on the English squad dataset, we test the model using the translated dataset and compare the results.

Results

In total we fine tuned 6 different models:

Model	Precision	Recall	F-score	Cos. sim.
A	0.63	0.66	0.64	0.60
B	0.63	0.66	0.64	0.61
C	0.66	0.68	0.67	0.63
D	0.70	0.72	0.71	0.65
E	0.77	0.86	0.82	0.80
F	0.57	0.59	0.58	0.54

Table 1. A table for comparison of model performance.

- A - ert-base-multilingual-uncased model, fine tuned on the whole dataset, in 1 epoch
- B - xlm-roberta-base, fine tuned on the whole dataset, in 1 epoch
- C - xlm-roberta-base, fine tuned on the whole dataset, in 3 epochs
- D - xlm-roberta-large, fine tuned on smaller part of dataset, in 1 epoch
- E - xlm-roberta-base, base model fine tuned on English text and tested on English, in 1 epoch
- F - xlm-roberta-base, base model fine tuned on English text and tested on Slovene, in 1 epoch

We fine tuned models in batches, with each batch of size 24, and with learning rate set to 2e-5. Every prediction, that we got from a model, we tokenized and evaluated against the tokenized ground-truth answer. For each prediction we calculated precision, recall and f-score and calculated cosine similarity. We then defined the four aforementioned measures for a model as the average from all of the predictions from the model. The results of model evaluation can be seen in the table 1. Our fine-tuned models reach up to a f-score of 71. We can also see that the English model, trained on the original English corpora and tested on the original English corpora, gives the best results due to others having corruption and data loss in the translations.

Discussion

In this paper we took the original squad data set and translated it into Slovene using EK translator. We then took six different multilingual BERT models and fine tuned them for Slovene question answering task.

When translating text to Slovene we tried to evaluate translated answers and their matching references in original text using similarity score calculation. We figured out, that if we took only answers with high enough similarity score, we were able to construct better training data sets. Here we also had to be careful, that we haven't placed similarity score barrier too high, because in this case we constructed too small training data set, which again resulted in poorer performance.

When fine tuning models, we found out, that if we fine tune them in more epochs, this usually doesn't contribute much to model's overall performance, while if we fine tune them using more data, this much improves model overall performance.

When fine tuning xlm-roberta-large model, we were constrained with model overall size and we couldn't train it on the whole data set. So we think with more resources we would be able to obtain even better results.

We also fine tuned multilingual xml model using English squad data set. We tested it using Slovene and English test set. Without suprise, when we tested it with the English dataset the results were better than when using the Slovene dataset, but we were surprised that even though we fine tuned the model using only the English data set, we still managed to obtain sufficient results, when testing it using the Slovene dataset.

References

[1] The stanford question answering dataset. https://rajpurkar.github.io/SQuAD-explorer/. (Accessed on 03/23/2022).

[2] etranslation. https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/eTranslation. (Accessed on 03/23/2022).

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.

[9] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle

Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

[10] Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. The goldilocks principle: Reading children's books with explicit memory representations. *arXiv preprint arXiv:1511.02301*, 2015.

[11] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. Ms marco: A human generated machine reading comprehension dataset. In *CoCo@ NIPS*, 2016.

[12] Casimiro Pio Carrino, Marta R Costa-jussà, and José AR Fonollosa. Automatic spanish translation of the squad dataset for multilingual question answering. *arXiv preprint arXiv:1912.05200*, 2019.

[13] Yuwen Zhang and Zhaozhuo Xu. Bert for question answering on squad 2.0. *Stanford University Report*, 2019.

[14] Shashavali Dudekula, V. Vishwjeet, Rahul Kumar, Gaurav Mathur, Nikhil Nihal, Siddhartha Mukherjee, and Suresh Patil. Sentence similarity techniques for short vs variable length text using word embeddings. volume 23, 10 2019.

[15] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[17] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*, 2019.