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

Cross-lingual question answering

Nace Gorenc, Jernej Vrhunc in Žan Pečovnik

Abstract

In the article we take a look at question answering, an important area of natrual language processing. Look at the related work that has been done on the very widly spread area. Later we analize choosen dataset, which in our case is SQuAD 2.0. Because the data in its form is not in a form in which we want it to be, we pre-process it and then translate it to target language. Because the data is not aligned with the use of translator, we implemented different methods to align such data. Using BERT we then create three different models and compare the results. The most accurate is the one where the data is aligned directly from finding the word in context with 65%, followed by tokenized version of alignment with 62%, and the worst in our case was the one aligned with the use of Levenshtein distance in original sentence with 57%. At the end we talk about further ideas and discus our reults.

Keywords

QA, BERT, Levenshtein, Tokenization, SQuAD 2.0

Advisors: Slavko Žitnik

Introduction

In this project we will take a closer look at natural language processing, more precisely the area of question answering and machine translation. We begin with the overview of related work and short presentation of potentially viable solutions.

Lewis et al. [1] tried to evaluate different question answering strategies, they came to the idea that it would be interesting when searching the web for answers, that different target languages could increase the number of possible answers. Eventhough it is good that more potential answers is overall better, it is still preferable to have answers in the same language as the question was. They focused on bilingual question answering systems, processing questions in both French and English. They approached the problem from 2 angles. The first one was to translate each question to the target language and then process the question analysis in the target language. The second approach is a term-by-term translation meaning that the question is analysed in the same language and the answer is then translated to the target language. Both approaches have a major flaw which is that a lot of valuable information can be lost during the translation.

Ligozat et al. [2] created a new multi-way aligned extractive question answering evaluation benchamrk dataset. QA models have shown rapid progress due to new high quality benchmark datasets eventhough most of them are in English, since they are very costly to collect. This dataset has over 12000 instances in English language and between 5000 and 6000 instances in Arabic, German, Spanish, Hindi, Vietnamese and Simplified Chinese. The sentences gathered from Wikipedia articles were translated by professionals and annotated in the aligned contexts for target languages. Each instance has an aligned equivalent in multiple other languages (always including English). Comparison was made between multilingual BERT and XLM models, both trained in English on SQuAD as training dataset. It turns out that zero-shot XLM model transfers best, but the transfer results are far worse that training-language performance.

Ojokoh and Adebisi [3] reviewed multiple known frameworks for question answering, passage retrieval and answer extraction. They addressed some important issues associated with QA systems, which include question processing, question classes, cross-lingual and real time question answering. Finally, they classified QA systems based on some identifying criteria like application domain, question type and data source.

McCann et al. [4] introduced the Natural Language Decathlon, which is a challenge that spans ten tasks: question answering, machine translation, summarization, natural language inference, sentiment analysis, semantic role labeling, relation extraction, goal-oriented dialogue, semantic parsing, and commonsense pronoun resolution. They casted all tasks as question answering over a context and proposed a multitask

question answering network (MQAN) which jointly learns all tasks in their challenge.

In the paper [5], the authors presented the Natural Question corpus, a question answering data set. It is said to be natural, because the dataset consists of real anonymous questions collected from the Google search engine. They set three goals in their research:

- provide large scale end-to-end training data,
- provide dataset that drives research in natural language understanding,
- to study human performance in providing QA annotations.

The process of the method is as follows. An annotator is presented with a pair of question and the Wikipedia page. It returns also a pair of short answer and a long answer, which is usually a paragraph or a table from the Wikipedia page in HTML format. Both answers can be returned as null, if the annotator hasn't found any answer for the question that has been asked. They present the metrics that can be used with natural questions, for the purposes of evaluating the performance of question answering systems. The NQ corpus is designed to provide a benchmark with which we can evaluate the performance of the QA system. They represent the upper bound on these metrics and show that existing methods currently do not approach this upper bound.

Dataset analysis

SQuAD 2.0 is an improvement of the SQuAD 1.1 dataset which had 107.785 question-answer pairs on 536 Wikipedia articles. In the SQuAD 1.1 dataset the correct answer to the question always existed somewhere, the models just had to pick the most probable answer instead of checking that the answer is entailed by the text, so they extended the new dataset with 53775 new unanswerable questions about the same paragraphs as in SQuAD 1.1, which were crafted by people in such way that they are relevant to the paragraph and that this paragraph contains a plausible answer. New dataset is significantly more challenging since the state-of-the-art models had a 66.3% F1 score on this dataset, while the same models had a 85.8% F1 score on the previous SQuAD 1.1 dataset, whereas human F1 score is 89.5% meaning that the new dataset is approximately 20% more challenging. Together there are now 151.054 questions on 505 articles, approximately a third of that is unanswerable.

For development purposes we used the *dev-v2.0.json* dataset, which we translated with the help of eTranslation webservice [6] into slovenian language and is approximately 10 times smaller than the bigger *train-v2.0.json* dataset.

Information type	Result
Total # of articles	35
Total # of questions	11873
# of unanswerable questions	5945
Average # of questions per article	339.23
Average # of answers per question	1.71
Length of shortest question (characters)	37
Length of longest question (characters)	181
Length of shortest answer (characters)	1
Length of longest answer (characters)	181

Table 1. Additional information about the SQuAD dev-v2 dataset

Methods

Data pre-processing

Data pre-processing is very important in natural language processing, especially in question answering since it is really important that the data is formatted and tokenized [7] so it can be used as correct input [8].

The SQuAD v2.0 dataset is originally in a json format, where each article has an array of paragraphs and each paragraph contains an array of questions with answers. First of all we had to clean out the non-necessary data and delete it, since it will not be used and is thus obsolete. We then iterated over all of the articles and extracted the context of each paragraph, all of the questions for each paragraph and joined all of the possible answers for a certain question in one array, containing the text, starting index and index at the end of the answer. We can see the example of how the final, cleaned-up and fixed-up json file would look like below in listing 1.

Listing 1. Example of how the translated dataset looks like.

```
{
 "version": "v2.0",
 "data": [
  {
   "title": "Normani",
   "paragraphs": [
    {
     "qas": [
      {
       "question": "V␣kateri␣drzavi␣se␣
           ↪ nahaja␣Normandija?",
       "id": "56ddde6b9a695914005b9628",
       "answers": [
        {
         "text": "Francija",
         "answer_start": 147,
         "answer_end": 155
        },
        ...
       ]
      },
      ...
     ]
    },
```

```
      ...
    ]
   },
   ...
  ]
}
```

Each text that is fed into the BERT model then has to be tokenized, which is described more in detail below. We used the BertTokenizerFast implementation of the tokenizer and not the regular BertTokenizer, because it has the ability to locate the token which is at the beginning or at the end of the answer in some text by using the *char_to_token*() function.

Data translating

We translated the whole SQuAD dataset with the help of eTranslation webservice [6]. There were some minor errors in the process of the translation which we then fixed by hand (e.g. missing paranthesis), but a much bigger problem occured that we did not expect. Since the translation is not using the same exact words and phrasing as the original text, this means that the location of answers to certain questions in the text also changes, meaning that in some cases the starting position of the answer is not correct anymore. We tried solving this by aligning the texts using the Hamming distance and then changing the starting index of the answer if the alignment is good enough.

BERT

BERT is shorter for Bidirectional Encoder Representations from Transformers, in the recent years it has become one of the most popular and widely used natural language processing model. BERT models have the ability to consider the broader context of a certain word by looking at the words which appear before and after this certain word, which is very useful for understanding the intent of a query being asked. BERT is actually a very big neural network, having 24 hidden layers with approximately 340 million parameters, which ends up in a model of a size of more than 1.3 GB. BERT has a unique way of processing inputs, since the input has to be tokenized by using the wordpiece tokenization [7]. Meaning that each word has an assigned token id and more complex words get split down into subwords, where also each subword has its own token id. E. g. the word "running" would be split into two parts "run" and "##ning", where "##" is used as a delimiter, so BERT would know that this is not a standalone word but rather a suffix to some less complex words. BERT also uses two addiditonal special tokens [CLS] and [SEP], where [CLS] is used for classification on sentence-level classification and therefore not very important in our case and [SEP] is used as a separator between two pieces of texts (question-answer). Apart from token embeddings, BERT also internally uses position embeddings for specifying the position of words in the text and segment embeddings for differentiating a question from the rest of the text.

There are quite some pre-trained BERT models for question answering, text classification and many more in the python library transformers [9], from where we loaded such a pre-trained model and then used it to answer question on the SQuAD dataset.

Methods for aligning the data

Directly

/

Tokenization

/

Levenshtein distance in original sentence

If the word was not found directly in the sentence we upgraded our method, with searching for it in the exactly the same sentence as the original (source) context. When we found the original sentence, we used Levenshtein distance [10] and by moving through the sentence searched for the most similar context word. For the method to found the actual answer we set the upper boundary of distance to be less then half the length of actual answer. If any part of the sentence did not match the criteria, we marked that the sentence does not contain answer to certain question. The problem came with the sentences, where there are more than one almost similar answers with different locations. Our algorithm then took the first answer it found and as there is a chance that this answer was the wrong one (in a wrong location), the algorithm then performed worse and got less accurate results.

Results

We tested the HuggingFace's BERT implementation for question answering on different data sets described above. Tokenization was done with CroSloEngual BERT. The learning curves and the validation accuracy are presented in figures 1 and 2[11].

As we can observe, the train loss in all cases declines and the validation loss increases while accuracy slightly rises. This could be due to the overfitting of the network.

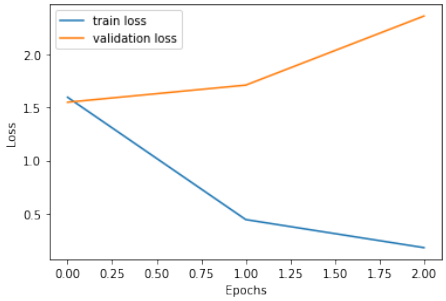
Finally, the table 2 shows the accuracy of the network on different test sets. As we can observe, the best results were obtained with "directly" test set.

Data set	Accuracy
Directly	0.626
Tokenized	0.653
Levenshtein distance in orig. sentence	0.567

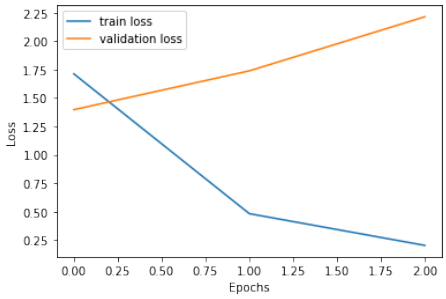
Table 2. Accuracy on the test set.

Future directions and ideas

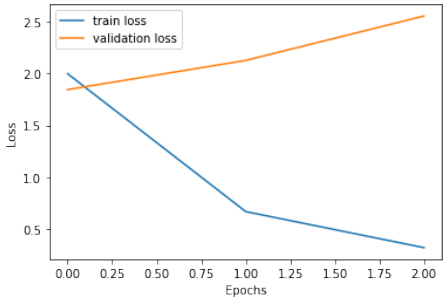
We would like to finish the training/fine-tuning process of the current BERT implementation on the translated Slovenian data, so we could then compare it to the other proposed method of translating the text on the fly.



(a) Directly.



(b) Tokenized.



(c) Levenshtein distance in original sentence.

Figure 1. Learning curves for different data sets.

Since we had quite a lot of issues with the translation of the whole SQuAD dataset, we believe that the other proposed method would be a lot more accurate, better and robust, since there would be no problems with locating the answer in text. The eTranslation web-service [6] also provides an API endpoint, which we will use for translating the text on the fly, meaning we would translate the question from Slovenian (or any other language) to English, find the answer in the original English corpus and then translate the answer back to Slovenian (or any other language).

We will also translate the bigger *train-v2.0.json* dataset to Slovenian, align it again with the same method as we did now, try to train and fine-tune BERT on this dataset and then use the smaller *dev-v2.0.json* dataset only for testing/validation purposes.

Discussion

We can see that searching for the word directly in sentence gives us the best results. It is kind of strange, since this

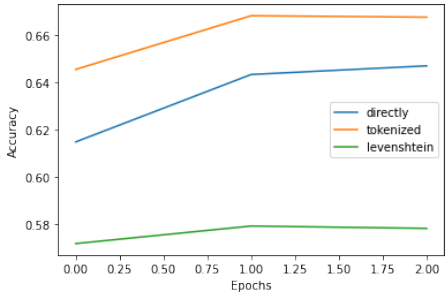


Figure 2. Validation accuracy.

was the first method we tried out. Next methods were our upgrades of the initial method, but turned out to be worse then we imagined. We think that there can be a lot of work done to increase the accuracy of the tokenized method and method using Levenshtein distance in the original sentence. We are satisfied with the results anyway, since we learned a lot working on the exercise and testing different methods on data translation.

References

[1] Patrick Lewis, Barlas Oğuz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. Mlqa: Evaluating cross-lingual extractive question answering. *arXiv preprint arXiv:1910.07475*, 2019.

[2] Anne-Laure Ligozat, Brigitte Grau, Isabelle Robba, and Anne Vilnat. Evaluation and improvement of cross-lingual question answering strategies. In *Proceedings of the Workshop on Multilingual Question Answering at EACL Conference*, pages 23–30, Trento, Italy, May 2006.

[3] Bolanle Ojokoh and Emmanuel Adebisi. A review of question answering systems. *Journal of Web Engineering*, 17(8):717–758, 2018.

[4] Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*, 2018.

[5] 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.

[6] eTranslation documentation. https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/How+to+retrieve+the+list+of+available+language+pairs+and+domains. Accessed: 2022-04-29.

[7] How to Build a WordPiece Tokenizer For BERT. https://towardsdatascience.com/how-to-build-a-wordpiece-tokenizer-for-bert-f505d97dddbb. Accessed: 2022-04-29.

[8] Question Answering with a fine-tuned BERT. https://towardsdatascience.com/question-answering-with-a-fine-tuned-bert-bc4dafd45626. Accessed: 2022-04-29.

[9] HuggingFace Transformers. https://huggingface.co/docs/transformers/index. Accessed: 2022-04-29.

[10] Implementing The Levenshtein Distance for Word Autocompletion and Autocorrection. https://blog.paperspace.com/implementing-levenshtein-distance-word-autocomplete-autocorrect/. Accessed: 2022-05-20.

[11] Matej Ulčar and Marko Robnik-Šikonja. Finest bert and crosloengual bert: less is more in multilingual models, 2020.