

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 occurred 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 additional 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.

## Preliminary results

As we had a lot of problems with the translation of the text, as the json file was somewhat corrupted when we got it back from the eTranslation web-service, we then had to fix it by hand (which took really a lot of time) and additionally re-align the text so we would get the correct location of the answers in the text, we ran out of time to train and fine-tune the BERT model, since the training process is very long as we do not have any super good GPUs that could shorten this process and make it faster.

## 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.

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

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.

## 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.