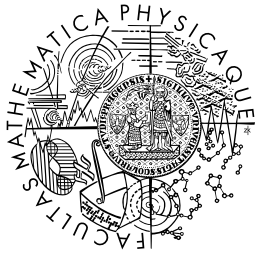

FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

RIGOROUS THESIS

Mgr. Kateřina Macková

Question Answering in Czech via Machine Translation and Cross-lingual Transfer

Institute of Formal and Applied Linguistics

Supervisor of the rigorous thesis: RNDr. Milan Straka, PhD.

Study programme: Computer Science

Study branch: Artificial Intelligence

Prague 2022

I declare that I carried out this rigorous thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In ..... date ....

Kateřina Macková

I would like to thank my Master's studies supervisor Milan Straka and my Doctoral studies supervisor Martin Pilát for their oversight and guidance. I would also like to thank my family and close friends for their support, especially my grandmother and my boyfriend for their unrelenting psychological support. I would also like to give special thanks to Lukáš Kyjánek for his expert linguistic assistance and unceasing mutual support in our co-suffering during the studies.

Title: Question Answering in Czech via Machine Translation and Cross-lingual Transfer

Author: Mgr. Kateřina Macková

Department: Institute of Formal and Applied Linguistics

Supervisor: RNDr. Milan Straka, PhD., Institute of Formal and Applied Linguistics

Abstract: Reading comprehension and question answering are computer science disciplines in the field of natural language processing and information retrieval. Reading comprehension is the ability of the model to read text, process it and understand its meaning. One of its applications is in question answering tasks, which is concerned with building a system that can automatically find an answer in the text to a certain question relied on the content of the text. It is a well-studied task, with huge training datasets in English. However, there are no Czech datasets and models for this task.

This work focuses on building reading comprehension and question answering systems for Czech, without requiring any manually annotated Czech training data. Our main focus is to create Czech training and development datasets, create the models for the Czech question answering system using Czech data, and create the models for the Czech question answering system using English data and cross-lingual transfer and compare the results and select the best model. First of all, we translated freely available English question answering datasets SQuAD 1.1 and SQuAD 2.0 to Czech to create training and development datasets. We then trained and evaluated several BERT and XLM-RoBERTa baseline models used for the question answering task in English. The best results were obtained XLM-RoBERTa model trained on English and evaluated directly on Czech. This model achieved very good results, similar to the model trained on the translated Czech data. However, we consider the result obtained from XLM-Roberta to be overperforming the models trained on Czech because the model does not require any Czech data during training. This also proves that the cross-lingual transfer approach is very flexible and provides reading comprehension in any language, for which we have enough monolingual raw texts to pretrain the language model.

Keywords: Question answering, Reading Comprehension, Natural language processing, Crosslingual Transfer, SQuAD, XLM-RoBERTa, Transformer, BERT

Contents

1. Introduction

Question answering is a computer science discipline within the fields of natural language processing and information retrieval. The target is to build a system that can automatically find the answer to a question related to a certain text. To answer correctly, the computer needs to understand the question, answer, the whole text and all the relations among words and sentences. Computer understanding of the texts is achieved by such numerical internal representation of sentences which preserves language semantics and word relations thoroughly. Syntactical and semantical analysis of the question, answer and corresponding text are also necessary to answer the question correctly.

Many English datasets have been developed for English question answering tasks, some of them are very large. In this work, we consider the frequently used SQuAD 1.1 [Rajpurkar et al., 2016], an English question answering dataset with around 100,000 question-answer pairs, which is widely used to train many different question answering models with relatively good accuracy. We also utilize SQuAD 2.0 [Rajpurkar et al., 2018], which combines SQuAD 1.1 dataset with 50,000 unanswerable questions linked to already existing paragraphs, making this dataset more challenging for question answering systems.

The target of this thesis is to produce such datasets and train question-answering models also for the Czech language as there is not a dataset or a model for this task, although Czech is one of the best processable languages in Natural Language Processing and one of the languages with the greatest coverage of corpora and other language data. We change this deficiency by creating the Czech QA dataset and developing and comparing several Czech QA systems. This thesis covers not only the creation of a Czech question-answering dataset but also the building, evaluation and comparison of similar models in the Czech language by reusing English models and English datasets. This thesis was created as an extension of the author's Master Thesis [Macková, 2020] and of the author's published article [Macková and Straka, 2020].

1.1 Our Contribution

In this thesis, we describe the creation of a question answering system for Czech without having any manually annotated Czech training data. This is achieved by reusing English models and English datasets. Our contributions are following. We translated both SQuAD 1.1 and SQuAD 2.0 to Czech by CUB-

BITT [Popel et al., 2020] translator and relocated the answers in the translated text using MorphoDiTa [Straková et al., 2014] and DeriNet [Vidra et al., 2019]. We released the translated dataset at LINDAT/CLARIAH-CZ, `http://hdl.handle.net/11234/1-3249`.

Every dataset becomes noisy after the machine translation. In our case, we have lost the information about the position of the real answer to the question in the context paragraphs. The position is specified by the starting index of the answer in the text. We had to relocate the starting indices of all answers in the text as follows: We lemmatized the translated text and answer using MorphoDiTa. We replaced the lemmas with roots of their word-formation relation trees according to the DeriNet 2.0 lexicon. Then, we found all continuous subsequences of the text with the same DeriNet roots as the answer, but with any word order. Finally, if several occurrences were located, we chose the one with the relative position in the text being the most similar to the relative position of the original answer in the original text. We selected the starting index of such an answer as the new starting index and updated this information in the translated dataset. We believe this algorithm has a sufficiently high precision after manually verifying several of the located answers. In the SQuAD 2.0 training dataset, we have preserved 82.2% of the English questions and in the development dataset, we kept 91.3% of the questions of the original dataset. The ratio of the kept data in SQuAD 1.1 is lower because unanswerable questions of SQuAD 2.0 are always preserved.

After the translation of the datasets, we trained several baseline systems using BERT [Devlin et al., 2018] and XLM-RoBERTa [Liu et al., 2019] architectures. Firstly, we trained and evaluated a system on the translated Czech data. Secondly, we trained and evaluated a system which first translates a text and a question from Czech to English, uses an English model and translates the answer back to Czech. Thirdly, we trained and evaluate cross-lingual systems based on BERT and XLM-RoBERTa, which are trained on English and then evaluated directly on Czech. We report that such systems have very good performance despite not using any Czech data nor Czech translation system. As to the models, XLM-RoBERTa significantly overperformed all other BERT-based models. With XLM-RoBERTa trained on English and evaluated directly on Czech, we have reached 73.64% exact match and 84.07% F1 score on SQuAD 1.1 dataset, and 73.50% exact match and 77.58% F1 score on SQuAD 2.0 dataset. With such good results, this model can be reused for any language for which only raw monolingual data are available while still reaching very good performance.

2. Question Answering

Question answering (QA) is a computer science discipline in the area of artificial intelligence within the Natural Language Processing (NLP) and Information Retrieval (IR) fields. QA concerns with building a system that can automatically find an answer to a certain question related to a certain text, which is posed by the human in natural language. To find the correct answer, the computer needs to understand the meaning of the whole text and the meaning of the question properly. It is achieved by using reading comprehension methods that allow to decompose and analyse of the text beginning with the single words and whole sentences and up to the overall meaning. The fundamental application of QA systems is to assist human-machine interactions and help with querying a structured database and automatically extracting important information from large texts. QA is an important task often implemented as an extension of other tasks processing natural language. The most common ones are the text search and dialogue systems.

There are two types of question answering tasks: closed-domain and open-domain. **Closed-domain QA** system can answer only questions related to a specific domain, for example, medicine, sports or literature. **Open-domain QA** systems can answer arbitrary questions without specification of the particular domain. Those systems can rely only on general ontologies and general world knowledge that they have learned from unstructured data. They are less accurate and need significantly larger datasets for the training of deduction of answers and to obtain general world knowledge. Therefore, it is more complicated to train them. On the other hand, they can be used for arbitrary questions from any area [Cimiano et al., 2014].

2.1 Related Disciplines

There are three main disciplines related to QA: natural language processing, reading comprehension and information retrieval.

Natural language processing (NLP) is a sub-area of linguistic, artificial intelligence and computer science, which analyses natural language tasks that require natural language comprehension. It investigates the interaction between human and machine language intending to represent and work with data from natural language. The real-world applications of NLP are for example speech recognition, information retrieval, natural language generation and recognition,

text to speech processing and the question answering which is described in this thesis.

Reading comprehension (RC) is a subarea of the NLP. It is concerned with the understanding of the meaning of a text, which is crucial for particular tasks of NLP working with natural language. Fundamental skills for reading comprehension compose from

- understanding of the meaning of single words
- understanding of the meaning of whole sentences,
- understanding of the relations among words and sentences,
- ability to make links to previous words or sentences in the text,
- understanding of the meaning of the whole text in general,
- extract the main idea of the text,
- capabilities of other deductions from the text,
- determination the other situation in fluences (mood, intonation, environment, context).

Computer understanding of a text can be described as a text preprocessing that can reuse the context and meaning of words and sentences to create an internal numerical representation of input text which preserves language semantics and word relations. The system uses such a representation to return a relevant response to a posed query.

Information retrieval (IR) is the task of retrieving desired pieces of information from a collection of documents. IR systems are used for searching in the database of documents for the particular information in the document, or for the document itself. The correct response for the given query is obtained by enabling indexing in the collection of documents and selecting the particular parts of the documents by keyword extraction. This is important for the question answering task where an answer for a given question is to be found in the collection of texts.

2.2 History of Question Answering

The history of QA starts in the 1950s not so long after the beginning of NLP. The first systems invented to answer the question were BASEBALL [Green et al.,

1961] and LUNAR [Woods, 1972]. BASEBALL was the first domain-based QA system made in America in 1960 and able to answer questions about baseball leagues in the US. LUNAR was another domain-based system able to answer the questions about the geological analysis of rocks from the Apollo mission to the Moon. Both of the systems were similarly efficient. They were not like today's QA systems as they did not understand the text properly, but they were only decomposing the question into single words, extracting the keywords and using deduction in the knowledge database to find correct answers.

During the following years, other domain-based QA systems were developed. The biggest boom was in the 1970s when a lot of knowledge-based domains for plenty of different areas with relatively good accuracy were released. They all were similar to today's ones but they all were based only on deduction. As the LUNAR and BASEBALL, they all had a huge knowledge database about the particular domain from which they deduced answers without understanding the meaning of the text. One of the examples of such a system was ELIZA [Weizenbaum, 1966]. It was a computer program for the study of natural language communication between man and machine based on the keywords extraction and deduction from a knowledge database.

In the 1970s-1980s, more complex development of statistical linguistic theories started. That motivated humans to teach machines how to understand the meaning of the texts and not only to deduce some information without a deeper understanding of it. One of the first systems was Unix Consultant [Chin, 1983] developed by the end of the 1980s. It was able to answer specific questions about the Unix operating system and it was used as an assistant for Linux users. LILOG [Herzog and Rollinger, 1991] was a similar system for understanding the text concerning about tourism industry in one of the German cities. All of those systems helped in the development of computer deduction and reading and text comprehension.

Nowadays, specialized systems based on deep neural network language models such as Transformers [Wolf et al., 2019] to answer questions in natural language automatically are developed. They can understand the meaning of the text and the question and easily find the answer in the text for the posed question. Wolfram Alpha [Hoy, 2010] is an online computation machine that can be considered one of the examples of such a system. Another example can be EAGLi [Bauer and Berleant, 2012], which answers questions about liver health. With the bigger and bigger boom of deep neural networks, more and more systems are trained to solve QA problems.

2.2.1 Application

QA systems are important and useful systems with applications in a wide variety of tasks in different areas, especially in information retrieval. There are plenty of huge unstructured and unlabeled data in the world freely accessible by anybody. Extraction of important information from such data can be an exhaustive and time-consuming task. Originally, the main information from the data was extracted manually by humans. Nowadays, it is not possible anymore with the constantly increasing amount of available data. Therefore, QA systems allow us to facilitate such tasks and do the exhaustive work automatically by computers.

QA has applications in many common areas and it is used in the background of many other NLP systems.

1. **Search Engines** Common search engines are based on simple indexing and matching of single keywords and they are not able to analyse the meaning of the request. Search engines based on QA have a stronger capacity to analyse the query, understand its meaning, and return the most relevant response. These systems can facilitate searching for particular information in educational and general information retrieval systems.

2. **Chatbots and Dialogue Systems** Dialogue systems and chatbots are designed to simulate human conversation and help people to solve their problems and receive useful information. Question answering in these systems enables chatbots to analyse the humans' questions and answer them correctly. These systems have practical utilisation for example in customer support on shop websites, education, and online assistance services.

2.3 Question Answering Systems

Question Answering Systems are computer programs used for solving QA tasks. They should return the correct answer to the posed question. The question is posed by the human in natural language. The computer needs to process the question, process the related text, understand the meaning of both and then reply appropriately. The answering process in basic QA systems is based on finding and marking the correct answer in the text, or in advanced QA systems, announcing that the question is not present.

2.3.1 Creation of Question Answering Systems

The basic approach to the creation of a QA system consists of the 1) definition of a model that will be able to answer the questions and 2) the creation of the dataset for model training. In earlier approaches, keywords extraction and deduction from the knowledge database were used. Nowadays, deep learning neural network-based models that exploit the definitions and datasets are widely used with surprisingly good results for this task.

2.3.2 Model Definition

QA model is a machine learning or deep learning-based structure for solving QA tasks. Standard models have several layers where each layer has its specified function. Every model has an input layer, which reads the input text and question and an output layer which returns the span of the text with the correct answer. Between these two layers, there are several hidden layers. The number and type of hidden layers depend on the concrete model but they should process input text and the question, decompose them to single words, analyse the relations between the words and whole sentences, understand their meaning extract the main keywords and the type and target of the question, analyse the relationship between the question and return the corresponding part of the text as the answer for the question.

To enable this a suitable internal numerical representation of text need to be created. Such a representation called text embeddings is widely used for all the NLP tasks. It consists of converting words into vectors in multidimensional space so that words with similar properties are close to each other. This approach can be generalized to entire sentences and documents. Word and text embeddings are used to represent and preserve word and whole sentence relations to enable NLP models to solve human-language related tasks.

2.3.3 Training Dataset

For training of the model, a huge labelled training set is necessary to teach the model how to find the answer to the questions related to certain context paragraphs correctly. The dataset has to consist of paragraphs with linked questions and answers to them marked in the texts. Usually, the dataset has to be created manually to ensure its quality for the model learning.

The training dataset is split into two disjoint parts. The first part is used for training the model and the second part is used for evaluation of the accu-

racy of the model. Ordinarily, the size of the training dataset is around three times larger than the testing dataset to balance the training and evaluation parts of the process [Lin, 2002].

2.3.4 Training Procedure

The basic training process consists of the preprocessing of the dataset by tokenizing each paragraph with content, question and answer into single words. Then, relations among the words and whole sentences are found using reading comprehension and NLP techniques and the most important words are marked. According to it, the most suitable input data representation is created and the training of the model is started.

Training is the process of learning mappings from the questions to the text, from the questions to the answers and from the answers to the text and vice versa. After the model is trained, its accuracy is evaluated. If it is accurate enough, it can be used to answer questions in previously unseen data from which it is necessary to extract particular information.

The trained model works as follows. It reads input questions, decomposes them into single words and performs keyword extraction using word tagging. Keywords such as 'who', 'where', 'which', etc. are crucial for deciphering the target of the question and finding then the correct answer in the text. After analysing the most relevant keywords, information retrieval from the text can be used to obtain the answer to a given question.

The purpose of this thesis is to produce such datasets and train question answering models also for the Czech language as there is not a dataset or a model for this task although Czech is one of the best processable languages in NLP and one of the languages with the greatest coverage of corpora and other language data. In this thesis, we change this deficiency by developing Czech QA systems. This thesis covers not only the creation of a Czech question answering dataset but also building similar models in the Czech language without having any Czech training datasets by reusing English models and English datasets [Das et al., 2018], [Prager et al., 2000].

3. Question Aswering Datasets

There are several English datasets for the English question answering task. In this chapter, the most common ones are explored. SQuAD datasets, which were used as the training datasets in this thesis, are described in more detail.

3.1 Existing Datasets

Reading comprehension is the ability to read and understand a text and then eventually answer the question posed to the particular text. It is a big challenge for the machines as it requires natural language understanding and basic knowledge of the world. There exist several English datasets for the question answering task. They vary in size, difficulty, and collection methodology. Unfortunately, the high-quality datasets require to be created by humans and thus they are often too small. The datasets generated automatically by machines are usually larger but they are not so suitable for training as the questions are not posed in human language and for that, they may not test natural language comprehension directly. We begin with a brief survey and comparison of available datasets. We describe SQuAD [Rajpurkar et al., 2016], MCTest [Richardson et al., 2013], TREC-QA [Voorhees and Tice, 2000] Wiki-QA [Yang et al., 2015], News-QA [Trischler et al., 2016], CNN Daily [Chen et al., 2016] and CBT [Hill et al., 2015].

3.1.1 MCTest

Machine Comprehension Test (MCTest) [Richardson et al., 2013] is a freely available dataset that consists of 660 elementary-level children's stories with associated questions for the machine comprehension of the text. This dataset was created by crowd-workers with 4 questions per paragraph and 4 different choices of answers for each question. The stories and questions were carefully limited by reducing the world knowledge that is required to be known for the task. The questions are designed to require a basic level of reasoning. That makes the dataset quite challenging. Moreover, the data set is not large enough to produce a good result. Therefore, it was not used for the training of question answering models in this thesis.

3.1.2 Wiki-QA

Wiki-QA [Yang et al., 2015] is a freely available dataset for open-domain question answering created by crowd-workers. This dataset contains 3047 questions originally sampled from real-life Bing queries based on Wikipedia articles. The creation process of this dataset is similar to the SQuAD dataset. The only difference is that the whole sentences were used for the answer selection in Wiki-QA. The SQuAD dataset only requires selecting a specific span in the sentence in the text as the answer.

3.1.3 TREC-QA

Text retrieval Conference (TREC) [Voorhees and Tice, 2000] has been focusing on the creation of different question answering datasets since 1999. Since that time, several different QA datasets were released. The very last dataset contains 1479 question-answer pairs. The dataset is not large enough to be used for training new question answering models.

3.1.4 News-QA

News-QA [Trischler et al., 2016] is a freely available reading comprehension dataset containing almost 120,000 human-generated question-answer pairs based on more than 10,000 news articles from CNN. Answers in this dataset are spans of text in corresponding articles as in the SQuAD dataset. The SQuAD dataset is the most closely related comprehension dataset but the questions and answers are more realistic than in News-QA. Some of the questions in News-QA have no answer in the corresponding article, which makes this dataset more challenging.

3.1.5 CNN/Daily

The CNN/Daily Mail corpus [Chen et al., 2016] consists of 1.4 million question-answer pairs from the news articles from CNN newspapers. This dataset was created automatically by taking articles as a source text. For each article, questions were generated synthetically by deleting a single entity from abstract summary texts, which follow each article. Finding the correct answer is therefore mostly achieved by recognizing the contextual link between the article and the question. The process of creation of this dataset is automatic and it is not difficult to create a huge amount of data. Unfortunately, the question answering task on this dataset needs only a limited amount of reasoning steps and the accuracy

of the best model is not so high [Chen et al., 2016]. Therefore, this dataset does not have enough quality to be used for the question answering task in this thesis.

3.1.6 Children's Book Test

The Children's Book Test (CBT) [Hill et al., 2015] was created by a similar process as CNN/Daily Mail. Instead of news articles, 20-sentence excerpts were selected from children's books. Questions were generated by deleting a single word in each 21^{st} sentence of the text. The dataset was generated automatically. There are 4 splits of the dataset and each split contains over 100,000 stories. The dataset is large enough to train deep learning models for the question answering task, but it is not considered a real QA and reading comprehension task, because there it requires no questioning, but only filling a missing word in each 21^{st} sentence.

3.1.7 Summary

Several datasets were presented, described and compared here. Reading comprehension is a complex and difficult task which requires a huge and high-quality dataset to allow to train the models precisely. These two requirements are of the same importance for the datasets to achieve good results while training the question answering models. Therefore, the SQuAD datasets were selected and used in this thesis as they fulfil both of these requirements for the size and the quality of the dataset the most, see Section 3.2.

3.2 SQuAD Datasets

3.2.1 SQuAD 1.1

The Stanford Question Answering Dataset (SQuAD) version 1.1 [Rajpurkar et al., 2016] is a freely available reading comprehension dataset. It was created by crowdworkers on a rich set of Wikipedia article and it consists of 107,785 question-answer pairs based on 536 articles. Unlike the other datasets, every answer to a question is a segment of text from the corresponding reading paragraph. It was not the first dataset for reading comprehension tasks, but it was the first huge and high-quality one.

3.2.2 Data Collection

This dataset was collected in three stages:

1. **Article selection**

   The top 10,000 articles from the English Wikipedia website were taken to obtain high-quality data. From this amount of data, 536 articles were randomly sampled. These articles were divided into individual paragraphs and all non-textual data were erased. Also, the articles and paragraphs shorter than 500 characters were removed as they do not contain enough information to pose questions. The resultant 23,215 paragraphs were split into a training, test and development set, while the training set is 80% size of the original dataset, the test set is 10% and the development set is remaining 10%.

2. **Question-answer collection**

   On each paragraph, crowd-workers made manually up to 5 questions asking about its content. Each answer to each question was required to be a part of the text.

3. **Additional answers collection**

   To make the evaluation more robust, at least 2 additional answers were created for each question in the development and test set. If some questions were unanswerable in the text, crow-workers created answers without marking them in the text.

3.2.3 Dataset Analysis

It is necessary to analyze the questions and the answers to understand the properties of the whole dataset. Three main aspects were analyzed for measuring how difficult the answer is for the system [Rajpurkar et al., 2016]:

1. **Diversity of answer types**

   Diversity in answers is an automatical categorisation of the answers into numerical or non-numerical. Non-numerical answers are subsequently separated according to the word class called a POS tag. In POS tags, there are several categories and one of them is nouns, which are further tagged accordingly to the place, time, person, etc. These tags are called NER tags, see Figure 3.1.

2. **Reasoning required to answer questions**

   The reasoning is necessary for selecting the correct answer. The difficulty of answering questions based on the reasoning can be measured to verify dataset quality. Several questions were sampled from each article and they were manually labelled with the categories mentioned above. The results showed, that each of the answers has some syntactical or lexical deviation between the question and answer in the text. These deviations are described more in [Rajpurkar et al., 2016].

3. **Degree of syntactic divergence between the question and answer**

   Stratification by syntactic divergence is an automatic method for quantification of syntactical divergence between answer and question which measures the difficulty of the answer. It was established as a minimum distance between all possible words which belong to the answer, see Figure 3.2.

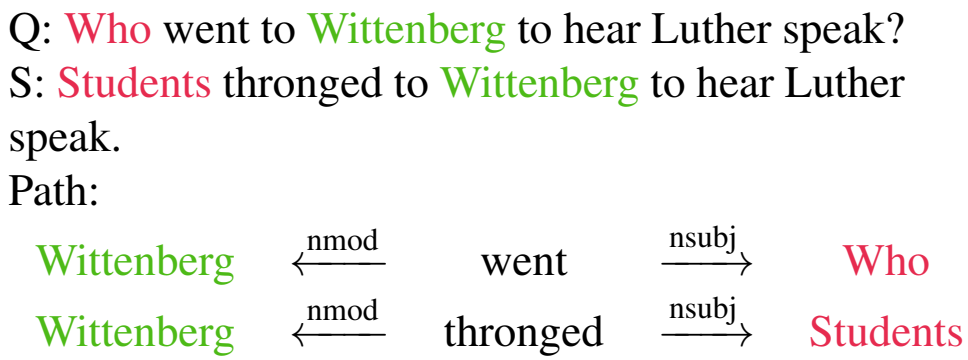

Figure 3.1: An example showing the keyword selection and dependencies modelling between answer and question. Source [Rajpurkar et al., 2016].

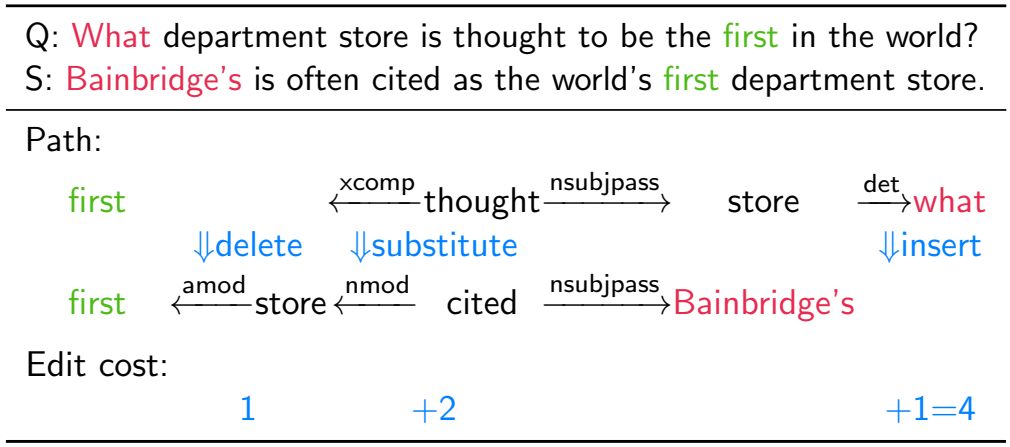

Figure 3.2: An example showing the computation of syntactic divergence between answer and question. Source [Rajpurkar et al., 2016].

3.2.4 Dataset Answers Selection

The dataset answer selection process is as follows. For selecting the correct answer to the question, the system must get through all possible spans in the text and find the one that is the most matching to the posed question. It generates a high number of possible candidates which must be compared and evaluated. Then, the best one must be chosen as the required answer. For that, special techniques based on distances and dependency trees are used. In SQuAD 2.0, the model has to verify whether the answer for the question is present in the paragraph by checking whether the possible answer similarity is above a certain limit [Rajpurkar et al., 2016].

Methods for Analysis

A logistic regression model [Rajpurkar et al., 2016] was created and compared to the candidate answer generation method and sliding window method. Generating candidate answers consists of passing character by character and generating all possible answers and finding the best one. The sliding window method is based on computing unigram/bigram overlap between sentence containing answer and question and by using sliding-window select the best answer.

In the logistic regression model, several types of features for each candidate question were selected. They were devised according to the linguistic analysis and they are matching of word frequency, match of bigram frequency, match of the root, a span of word frequency, lexicalization, parsing and path in the dependency tree. Loss is computed by AdaGrad with an initial learning rate of 0.1.

3.2.5 Evaluation

Several metrics for evaluation process are commonly used.

1. **Exact match**

   An exact match between every word in the real answer and the predicted answer is computed. To obtain a point, both compared words must be the same.

2. **F1 score**

   F1 score is computed as

   $$F1 = \frac{2 \cdot p \cdot r}{p + r}, \tag{3.1}$$

   where p is the precision, which is the number of correct positive results divided by the number of all positive results returned by the classifier, and r is the recall, which is the number of correct positive results divided by the number of all samples that should have been identified as positive. The F1 score is the harmonic mean of precision and recall. It has values between 0 and 1.

3.2.6 SQuAD 2.0

The SQuAD version 2.0 [Rajpurkar et al., 2018] combines existing SQuAD data with over 53,776 unanswerable questions written adversarially by crowd workers to look similar to answerable ones. The questions are relevant to the original SQuAD 1.1 paragraphs but the answers are not present. Moreover, the paragraphs contain plausible answers to the questions. It means that it contains something of the same type as what the question asks for. The relevance and the plausibility of the questions for the paragraphs are crucial. Otherwise, simple heuristics based on word overlap and type-matching could distinguish answerable and unanswerable questions and there will not be any pressure for understanding the meaning of the texts.

The unanswerable questions were shuffled together with the original ones to ensure the dataset's generality. Models trained on SQuAD 2.0 must not only answer questions when possible but also determine when there is no suitable answer in the paragraph for a given answer. Therefore, SQuAD 2.0 is a more challenging dataset for natural language understanding tasks.

3.2.7 SQuAD 1.1 and 2.0 Datasets Comparison

The datasets have 3 parts: train, test and development part. The total number of examples in the train part in SQuAD 1.1 is 87,599 and in SQuAD 2.0 it is 130,319 from which 43,498 examples are unanswerable questions. These questions were posed to in total of 442 articles from which 285 articles contain also unanswerable questions in SQuAD 2.0. The development part has in total of 10 questions in SQuAD 1.1. and 11,873 in SQuAD 2.0 from which 5,945 questions are unanswerable. These questions were posed in articles 48 and 35 which contain also unanswerable questions. In the test dataset, there are in total of 9,533 questions in SQuAD 1.1. and 8,862 in SQuAD 2.0 from which 4,332 questions are unanswerable. These questions were posed in articles 46 and 28 which contain also unanswerable questions. The test dataset is not publicly available and therefore, train and development datasets are used for training models. Moreover, in the SQuAD 2.0 development and test sets, the articles without unanswerable questions were removed. This resulted in approximately one to one ratio of answerable and unanswerable questions, whereas the train data has approximately twice more answerable questions than unanswerable.

Three models were trained on SQuAD 1.1 and 2.0 datasets to compare their difficulty: BiDAF-No-Answer [Seo et al., 2016], DocQA [Clark and Gardner, 2017], and DocQA+ELMo [Peters et al., 2018]. Average exact matches and F1 scores were compared also to human accuracy to measure overall quality and difficulty. The best model on the SQuAD 2.0 test set DocQA + ELMo achieved a 66.3% F1 score which is 23.2 percentage points lower than the human accuracy of 89.5% F1 scores. The best model on the SQuAD 1.1 test set, DocQA + ELMo, achieved an 85.8% F1 score on the test set, which is 5.4 percentage points lower than the human accuracy of 91.2% F1 scores. The results show that there is a much larger gap between humans and machines on SQuAD 2.0 compared to SQuAD 1.1, which confirms that SQuAD 2 is the much harder dataset for the training of existing models. The best model on the SQuAD 2.0 development set DocQA + ELMo achieved a 67.6% F1 score which is 21.4 percentage points lower than the human accuracy of 89.0% F1 score but the results on the SQuAD 1.1 development set were not evaluated and therefore, could not be compared [Rajpurkar et al., 2018], see Table 3.1.

Model	F1 DocQA + ELMo [%]	F1 human[%]
SQuAD 2.0 test	66.3	89.5
SQuAD 1.1 test	85.8	91.2
SQuAD 2.0 dev	67.6	89.0
SQuAD 1.1 dev	-	-

Table 3.1: Comparison of results of English QA models evaluated on the SQuAD 1.1 dataset.

3.3 Conclusion

The SQuAD 1.1 dataset was selected as the best one for the QA task in this thesis as it is high-quality and large enough for training deep learning neural networks. However, it does not handle the problem of unanswerable questions. Therefore, SQuAD 2.0 dataset was also used to train the models to not only find the answer for a given question in the text but also to verify, whether the answer is present there. Models trained on SQuAD 2.0 have in general lower accuracy than models trained on SQuAD 1.1 because unanswerable questions are more challenging and require deeper reading comprehension and natural language understanding.

4. Question Answering Models

In this chapter, several question answering models are selected and compared. BERT based models which were selected to create a question answering model in this thesis are described in more detail.

4.1 Existing Models

Question answering models are machine learning or deep learning-based models that can answer questions given some context. Usually, they are trained to extract and mark the answers in the context paragraphs. The accuracy of such a model depends on the dataset which was used for training and the overall model architecture and used technologies.

The question answering model should understand the structure of the language, understand the meaning of the context and the questions and it should have the ability to locate the position of an answer in the context paragraph. Training such a model is a difficult task and there are several commonly used approaches. At first, recurrent neural network-based (RNN) or convolution neural network-based (CNN) models were trained. However, the best results were achieved with neural network models based on an attention mechanism called Transformer. The most commonly used model based on Transformer architecture called BERT is recently used to train question answering models with the highest accuracy. Before Transformers and BERT language models, there were other approaches for solving the problem of Question Answering. We begin with a brief survey and comparison of available models trained and evaluated on SQuAD dataset and their results comparisons. We describe BiAttFlow [Seo et al., 2016], DrQA [Chen et al., 2017], jNet [Zhang et al., 2017] and QANet [Yu et al., 2018].

4.1.1 BiDirectional Attention Flow

BiDirectional Attention Flow [Seo et al., 2016] (BiDAF) was used for the question answering task before BERT-based models. It was one of the first models using the attention mechanism. Therefore, it outperformed all previously used models. BiDAF has a hierarchical multi-stage architecture which is modelling representation of context in several layers as character layer, word layer and contextual layer with the usage of attention mechanism which works in both direction context-query and query-context. It allows both sides to share information

about the contexts among words and reduces the loss of information and increases the model accuracy. By the time BiDAF was released, it was trained on SQuAD dataset and it has outperformed all already known models. After BERT based models were released, they overperformed the BiDAF model and replaced them. Therefore, BiDAF models are not used anymore for the question answering task.

[Seo et al., 2016] trained the model on the SQuAD dataset with an F1 score of 81.1% and an EM score of 73.3% on the development dataset. These results were overcome by the best BERT models by 11.8 percentage points in the F1 score and in 10.7 percentage points in the EM score.

4.1.2 Document Reader Question Answering

Document Reader Question Answering (DrQA) [Chen et al., 2017] is a model that combines document retrieval, which means finding the relevant articles, with machine comprehension of text, which means identifying the answer spans from selected articles. This approach searches for the main components to find relevant articles using the TF-IDF score and then trains a multi-layer recurrent neural network model to detect answers in selected paragraphs. It is one of the approaches based on classical RNNs which were used before the attention mechanism. This model trained and evaluated on the SQuAD 1.1 dataset reaches 69.5% EM and 78.8% F1 score. These results were beaten by attention-mechanism-base approaches and they are not so commonly used anymore.

4.1.3 Neural-Network-Based Question Answering: jNet

jNet [Zhang et al., 2017] is another RNN-based approach for QA. Unlike the classic RNNs using chain-structured LSTM, this model uses TreeLSTM that captures long-distance interaction in a tree structure and enables the training of the model faster and preserves more relations among words and texts. The baseline model is composed of five components: word embedding layer, input encoder layer, a text alignment layer, aggregation layer, and prediction layer where each component has its specific function to train the model and preserve semantic relations over given syntactic structures. This model trained and evaluated on the SQuAD 1.1 dataset reaches 68.73% EM and 77.39% F1 score which are insufficient results compared to other approaches.

4.1.4 Question Answering Net

Question Answering Net (QANet) [Yu et al., 2018] is an architecture which does not exploit recurrent networks. Its encoder consists only of convolution and self-attention layers. Convolutions ensure local interactions and self-attention models global interactions. The structure of the QANet model is similar to most existing reading comprehension models. It contains five major layers: an embedding layer, an embedding encoder layer, a context-query attention layer, a model encoder layer and an output layer. The difference is that both the embedding and modelling encoders use only convolutions and attention. This model enabled several times faster training than previous RNN based approaches while achieving equivalent accuracy. This speed-up allows training the model on much larger data. On the SQuAD 1.1 dataset, the QANet model achieves an 84.6 F1 score on the test set, which is slightly better than BiDAF but still 7.2 percentage points worse than BERT.

4.1.5 Summary

Overall results show that the most suitable models for Question Answering tasks are obtained by BERT-based models, see Table 4.1. Therefore, the BERT model was selected as the best model for the training question answering system in this thesis, see Section 4.2.

System	**EM** [%]	**F1**[%]
BERT	**85.1**	**91.8**
BiDAF	73.3	81.1
QANet	-	84.6
DrQA	69.5	78.8
jNet	68.73	77.39

Table 4.1: Comparison of results of English QA models evaluated on the SQuAD 1.1 dataset.

4.2 Bidirectional Encoder Representations from Transformers

Bidirectional Encoder Representations from Transformers (BERT) [Devlin et al., 2018] is a new universal language representation model based on the Transformer [Wolf et al., 2019] language model designed to pretrain deep bidirectional representations from the unlabeled text. It can be finetuned with only one additional output layer to create a specific model for a certain task, for example, for a question answering task.

Previous general language processing models only used unidirectional language models to learn general language representations from left to right. [Wolf et al., 2019] proved that this restricts the power of the pretrained representations because it limits the choice of architectures as each token can only attend to the previous one in self-attention layers. This restriction can be very harmful when applying to finetune for certain tasks. For example, in question answering, it is very important to consider the context in both directions in the question, the answer and the whole context paragraph.

BERT mitigates this deficiency by using a language model, which randomly masks some of the tokens from the input and tries to predict the original words based on the context. It allows to link left and right context and to pretrain bidirectional Transformer, which subsequently creates contextual representation. It also uses next sentence prediction to achieve better results. In the next sentence prediction, a model receives an input document and the pair of sentences and tries to predict whether the second sentence is following the first sentence in the documents. Finally, the whole sentence context is represented in the embedding of each word, see Figure 4.1.

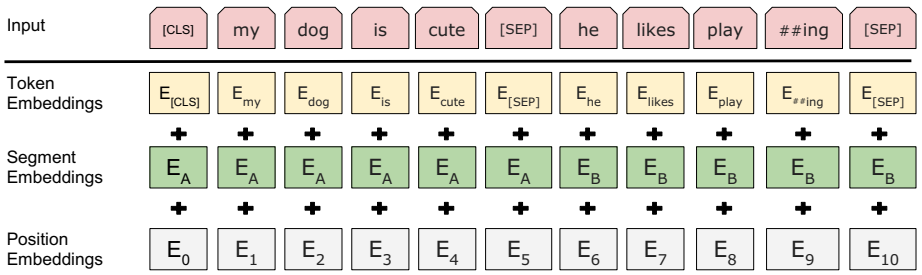

Figure 4.1: BERT input representation. Source [Devlin et al., 2018]

4.2.1 BERT Architecture

BERT model architecture is based on deep neural networks, and it consists of two parts. The first part is used for pretraining of the language model, and the second part is used for finetuning to the particular task. Apart from output layers, the same architectures are used in both the pretraining and finetuning phase. These architectures consist of several Transformer blocks and then one fully connected layer that predicts the output for the given input, see Figure 4.2.

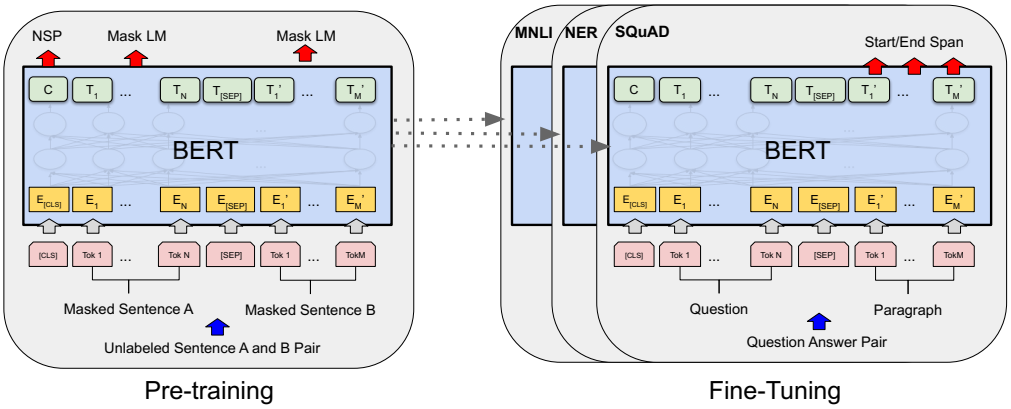

Figure 4.2: Pretraining and finetuning process for BERT. Source [Devlin et al., 2018]

Transformer

The Transformer [Wolf et al., 2019] is a deep learning model designed to process and handle text sequential data. It is widely used for NLP tasks such as question answering or language translation. Before Transformers, recurrent neural network approaches were used for text processing. The difference from traditional RNN is that the Transformers do not necessarily process the data in the order they obtain them, in other words from the beginning to the end of the sequence. They rather identify the context that provides meaning to each word in the sentence. Transformers allow better parallelization and reduce training time therefore, they replaced classical RNNs in text processing tasks.

The Transformer architecture consists of several repeating blocks of multi-head-attention, normalization and feedforward layer, see Figure 4.3.

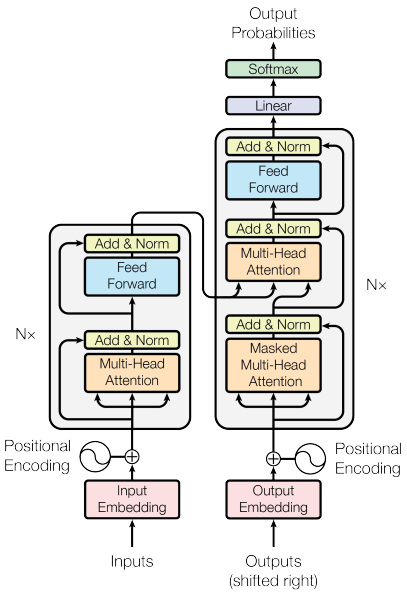

Figure 4.3: Overall Transformer architecture. Source [Vaswani et al., 2017]

Attention

Transformers are based on attention mechanism [Vaswani et al., 2017], which strongly improves the model's ability of generalisation. This mechanism allows systems to concentrate on the target area in the context of a query and use the most relevant parts of the input sequence for predicting the output. It is achieved by a weighted combination of all of the encoded input vectors, where the most relevant vectors are attributed by the highest weights.

Bidirectional attention is an attention mechanism with two directions of processing vectors. The first direction is text to attention query, which marks words in the question which are most relevant for each word in the content text. The second direction is a query to attention that marks, which words from the text are the most relevant for each word in the question and which of them are also important for the answer. A matrix saying how much the words from the query fit the words from the text is created. A vector saying what from the question matches the words in the text the most is obtained from this matrix. For each word in the text, a vector saying what from the question matches this word the most is obtained as well. These two direction vectors allow encoding the context better and create more accurate models, see Figure 4.4.

The multilayer Transformer in combination with a multi-head bidirectional attention mechanism creates powerful architecture to train a language model for a variety of natural text processing tasks.

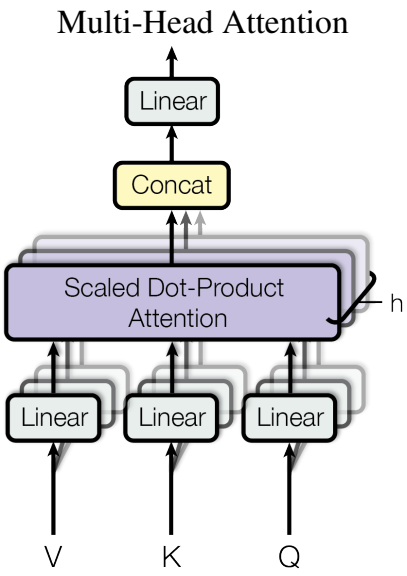

Figure 4.4: Multi-head attention architecture. Source [Vaswani et al., 2017]

4.2.2 Training Procedure

Pretraining

The training procedure has 2 steps. The goal of the first step is to train and create a high capacity language model on an unlabeled corpus. Unlabeled corpus for such training is arbitrary texts split into sentences and single words where each word is called a token. The input for the language model during pretraining is a sequence of tokens, that is first embedded into vectors and then processed by the neural network-based model. The output is a sequence of vectors of a particular size while a vector at a particular position corresponds to an input token with the same position. The training process consists of two parts: Masked Language Models and Next Sentence Prediction, see Figure 4.5.

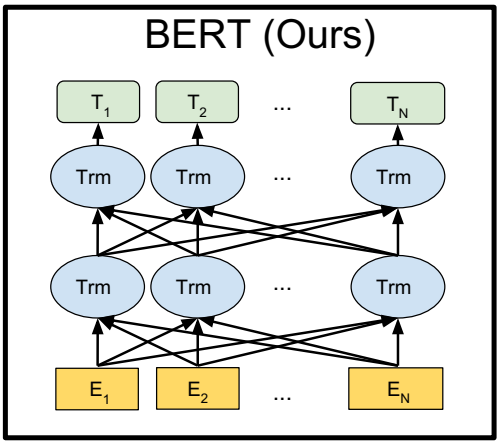

Figure 4.5: Illustration of pretraining BERT model architecture. Source [Devlin et al., 2018]

1. **Masked Language Models**

   The challenge is to create suitable vectors for the input tokens so that the relations among words and sentences are preserved. Therefore, in BERT model 15% words in each sequence are replaced with a [MASK] token and the model then attempts to predict the original value of the masked token based on the context of other non-masked words. Many models predict the next word in a sequence by a unidirectional approach, which predicts the masked words based on previous words only. This approach strongly limits context learning. Therefore, BERT uses a bidirectional approach which predicts the masked words based on both previous and following words and it enables the preservation of more relations among words and text context. The model is trained to predict the original token with cross-entropy loss based on the prediction of the masked values. Devlin et al. [2018].

2. **Next Sentence Prediction**

   In the second step of the pretraining process of BERT, the model receives pairs of sentences as input and learns to predict if the second sentence is following the first sentence in the original document. In 50% pairs, the second sentence is the actual subsequent sentence in the original document, and in the remaining 50% pairs, a random sentence from the corpus is taken as the second sentence. The model is trained with the accuracy of the next sentence prediction Devlin et al. [2018].

During the pretraining phase, the loss function of the BERT model is computed

as the combination of the loss functions of these two strategies. The target of the pretraining process is to minimize this overall loss function.

Finetuning

In the second step, the language model can be finetuned and adapted on the labelled dataset specific to the target task. BERT can be used for a wide variety of language tasks by adding only one output layer to the model and training its parameters.

In question answering, the model receives a context and the question and it is required to mark the answer in the context. The input for the model is a pair of contexts and questions related to it. The output is a starting index and length of the answer to the question in the context. BERT model is trained by using the same architecture as in the pretraining phase extended by two extra vectors that can mark the beginning and the end of the answer in the context Devlin et al. [2018], see Figure 4.6.

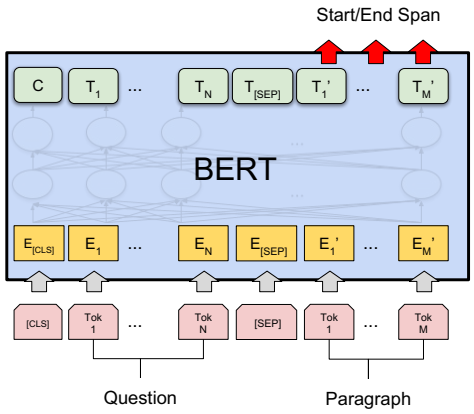

Figure 4.6: Illustration of finetuning BERT model on question answering task. Source [Devlin et al., 2018]

4.2.3 Results

The model was trained on the unlabeled dataset and it was finetuned for the QA task on SQuAD 1.1 dataset [Devlin et al., 2018]. Two versions of models were used: $\text{BERT}_{\text{BASE}}$ and $\text{BERT}_{\text{LARGE}}$. They differ in the number of layers, the number of self-attention heads and hidden layer size. The $\text{BERT}_{\text{LARGE}}$ has outperformed the $\text{BERT}_{\text{BASE}}$ model and has reached an F1 score of 93.2% on the testing dataset and 92.2% on the development dataset, see Table 4.2.

System	Dev(EM)	Dev(F1)	Test(EM)	Test(F1)
$\mathbf{BERT_{based}}$	84.2%	91.1%	85.1%	91.8%
$\mathbf{BERT_{large}}$	86.1 %	92.2%	87.4%	93.2%

Table 4.2: Comparison of results of BERTs trained on English. Source [Devlin et al., 2018].

4.3 Other BERT-based Approaches

There is a huge family of models based on BERT architecture that are extending and improving BERT-based architecture and results.

4.3.1 Multilingual BERT

Multilingual BERT (mBERT) [Devlin et al., 2018], released by [Devlin et al., 2018], is a single language model pretrained on monolingual corpora in 104 languages including Czech. Cross-lingual transfer capability of mBERT has been mentioned in 2019 by many authors [Kondratyuk, 2019], [Hsu et al., 2019] for morphosyntactic analysis or for reading comprehension.

4.3.2 XLM-RoBERTa

XLM-RoBERTa (XLM-R) [Liu et al., 2019] is another BERT-based model pretrained on 100 languages and is available in both base and large sizes. It has 355M parameters (which is 3 times more than the BERT model) which enabled the training of more powerful and accurate models for question answering tasks. It has also modified its structure to improve overall results. The model was trained for a longer time on longer sequences with bigger batches of an extended dataset. The next sentence prediction objective was removed and the dynamic changing of the masking pattern applied to the training data was added. These changes increased the EM to 94.6% and the F1 score to 89.4% on SQuAD 1.1 dataset which overcame the BERT results.

4.4 Conclusion

Overall model results and accuracies prove that the highest score for the question answering task is achieved by BERT-based models. The best global results among the BERT-based models were obtained from XLM-RoBERTa. The overall accuracies and cross-lingual transfer capabilities of mBERT and XLM-RoBERTa

indicate that these models are the most suitable for QA tasks in multiple languages. Therefore, these two models were selected for training the Czech question answering model in this thesis.

5. Constructing Czech Question Answering Dataset

There are plenty of datasets for question answering in English. However, QA in Czech does not have such a boom, although Czech is one of the best processable languages in NLP and one of the languages with the greatest coverage of corpora and other language data. In this thesis, we try to change this and develop Czech QA systems.

The SQuAD datasets [Rajpurkar et al., 2016] and BERT-based models Devlin et al. [2018] were selected as the most suitable for the training of the Czech question answering model. The selection process of the dataset is more described in Chapter 3 and the selection process of the model is more described in Chapter 4. However, SQuAD dataset is only in the English language. To train similar models for QA in the Czech language, the dataset has to be translated. As every translation brings noise into the dataset, the ideal model training would not require the necessity of translating any data. This chapter shows how previously described datasets and models can be reused for reaching this goal.

5.1 Dataset

The source English datasets SQuAD 1.1 and SQuAD 2.0 were downloaded from `https://rajpurkar.github.io/SQuAD-explorer/` and they were used for training, evaluation and creation of the Czech dataset.

The structure of the datasets is as follows. There are two JSON files. The first one *train-v1.1.json* contains data for training. It consists of context paragraphs with several questions and each question has one correct answer. The second one *dev-v1.1.json* is used for evaluation. The structure of this file is almost the same with the only difference. As it was annotated manually by several crowd workers, there can be several answers to one question. While the evaluation process, the most matching answer was always chosen to be compared with the predicted one to reach the highest accuracy. This also allows a little deviation in answering, which can be useful as the predicted answer is not always identical to the original one and still can be correct. The size of the training dataset is 87,599 questions and the development set is 10,570 questions for SQuAD 1.1 and 130,319 questions of the training set and 11,873 questions in the development set. Both SQuAD datasets are almost the same as SQuAD 2.0 is only SQuAD 1.1 extended by

43,498 unanswerable questions [Rajpurkar et al., 2016].

The structure of both data files looks as follows. There is a tag *data* containing a list of all articles. Inside this tag, there is always a title of the article in *title* tag having a list of single paragraphs containing the context related to the title. They are called *paragraphs* tags. Each paragraph has its list with answers and questions in *qas* tag, which furthermore consists of three tags. The first one is the *question* tag, which contains the text of the question. The second one is the *id* tag, as each question has its id for easier identification. Last one is *answers* tag containing the text of the answer in *text* tag, and also, starting index of the answer in the text represented in the *answer_start* tag. See Listing 5.1

Listing 5.1: An example of the structure of the dataset.

```
{ Data[{
        title
        paragraphs {[
                context
                qas [{
                        answers[
                                text
                                answer_start
                        ]
                }]
                question
                id
        ]}
}]
version }
```

5.1.1 Translation of the Data

We have used several data translations of the SQuAD dataset to Czech and possibly back to the English language. We have used CUBBITT Translator [Popel et al., 2020], which is the best translator between Czech and English developed at the Faculty of Mathematics and Physics at Charles University by the Institute of Formal and Applied Linguistics. Translation of all texts, questions and answers from SQuAD 2.0 took 3 days and from SQuAD 1.1 similarly.

In English dataset, the answer in the *text* tag in the *answers* tag is exactly the same as a part of the text in the *context* tag. Unfortunately, the translation

of the dataset brings a noise into it and the part of the text containing the correct answer and the answer is *text* in *answers* tag may differ. It is caused by different grammatical rules and word lengths in both of these languages. The Czech language has a much richer inflectional morphology which can cause problems during translation. Moreover, the sentence in the context of the paragraph can be translated in a different way than the answer itself. Therefore, the *answer_start* tag value must be recomputed as the order and the length of the words may have changed after translation.

5.1.2 Index Recomputation

Because the answers are subsequences of the given text in SQuAD, we needed to locate the translated answers in the text. We considered several alternatives.

1. **Attention mechanism for estimation of the alignment**

   Estimate the alignment of the source and target tokens using the attention of the machine translation system and then choose the words aligned to the source answer. Unfortunately, we could not reliably extract alignment from the attention heads of a Transformer-based machine translation system.

2. **Marking the answer before translation**

   Mark the answer in the text before the translation, using for example quotation marks. Such an approach would however result in a dataset with every question linked to a custom text, which would deviate from the SQuAD structure.

3. **Locate the answer in the given text after the translation independently**

   Locate the answer in the given text after the translation, without relying on assistance from the machine translation system.

We chose the third alternative and located the translated answers in the texts independently of the translation process. The problem is that we cannot use an exact match as the answers may not fit the text exactly. At first, we have considered going word by word and finding the longest common substring by starting with the whole text and computing a match between it and the translated answer. Meanwhile, we would systematically delete the first word from the remaining words until we have an empty string and we would measure which

Table 5.1: Size of the translated Czech variant of SQuAD 1.1 and SQuAD 2.0.

Dataset		**English Questions**	**Czech Questions**	**Percentage Kept**
SQuAD 1.1	Train	87,599	64,164	73.2%
	Development	10,570	8,739	82.7%
SQuAD 2.0	Train	130,319	107,088	82.2%
	Development	11,873	10,845	91.3%

of the resultant common substrings is the longest one. We found out that we would unnecessarily lose a lot of answers because of the Czech grammar and declination and conjugation and therefore, we have used a more complex algorithm.

1. We lemmatized the translated text and answer using MorphoDiTa.
2. We replaced the lemmas with roots of their word-formation relation trees according to the DeriNet 2.0 lexicon.
3. We found all continuous subsequences of the text with the same DeriNet roots as the answer, but with any word order.
4. Finally, if several occurrences were located, we chose the one with the relative position in the text that is the most similar to the relative position of the original answer in the original text.

We believe the proposed algorithm has high enough precision after manually verifying many of the located answers. From the SQuAD 1.1 training dataset, we kept 64,164 questions (73.2% of the questions from the original English training dataset) and in the development dataset, we kept 8,739 questions (82.7% of the questions from the original English development dataset). In the SQuAD 2.0 training dataset, we kept 107,088 questions (82.2% of the questions from the original English training dataset). In the development dataset, we kept 10,845 questions (91.3% of the questions from the original English development dataset). See Table 5.1.

To facilitate our work with translated data, we have modified the final JSON file. Two new tags into the *answers* tag were added. The first one is *answer_end*, which is computed during the recomputation of the starting index. It is pointing to the end of the last word of the answer in the text and it was added because of the easier visualization of the answer in the context paragraph. This tag is also useful while selecting the answer from the text as in *text* tag, we have translated

the answer and not exactly the text of the answer from the text. The other one is *answer_match* and it is the value of the score of the match. See Listing 5.2.

Listing 5.2: An example of the updated structure of the dataset.

```
{ Data[{
        title
        paragraphs [{
                context
                qas [{
                        answers[
                                text
                                answer_start
                                answer_end
                                answer_match
                        ]
                }]
                question
                id
        }]
}]
version }
```

5.1.3 Machine Translation Problems

Every machine translation causes mismatches between the original and the translated text. The most common ones are listed here.

1. **Word order**

   During the translation, word order is not preserved as every language has its own grammatical rules. It is confusing the system while recomputing the start index of the answer in the text. See Figure 5.1.

2. **Synonyms**

   Some words in the original language can have several different translations in the target language. The translator may choose two different Czech words in the question and answer for one word in English. See Figure 5.2.

3. **Declination and conjugation**

   Each language has different grammar rules for word creation and declination or conjugation. Czech words are declined, and English ones are not. See Figure 5.3

4. **Numbers**

   Numbers can be written as words and after the translation written as numbers which are also confusing for the index recomputing index process. See Figure 5.4. The same problem is with the names. See Figure 5.5.

Pes

CONTEXT	Pes domácí (Canis lupus familiaris neboli Canis familiaris) je domestikovaný kanid, který byl po tisíciletí selektivně chován pro různé způsoby chování, smyslové schopnosti a fyzické vlastnosti.
QUESTION	Co je Canis familiaris?
ANSWER	domácí pes

Figure 5.1: Example of the selected answer by the algorithm with changed word order between text and answer.

Frédéric_Chopin

CONTEXT	Všechny Chopinovy skladby zahrnují klavír. Většina je určena pro sólový klavír, i když napsal také dva klavírní koncerty, několik komorních skladeb a některé písně k polským textům. Jeho klávesový styl je vysoce individuální a často technicky náročný; jeho vlastní výkony byly proslulé svými nuancemi a citlivostí. Chopin vymyslel koncept instrumentální balady. K jeho významným klavírním dílům patří také mazurky, valčíky, nokturnovky, polonézy, études, impromptus, scherzos, předehry a sonáty, z nichž některé vyšly až po jeho smrti. K vlivům na jeho kompoziční styl patří polská lidová hudba, klasická tradice J. S. Bacha, Mozarta a Schuberta, hudba všech, které obdivoval, a také pařížské salony, kde byl častým hostem. Jeho inovace ve stylu, hudební formě a harmonii a spojení hudby s nacionalismem měly vliv po celé pozdní romantické období i po něm.
QUESTION	Jaký nástroj obsahovaly všechny Frédéricovy skladby?
ANSWER	piano

Figure 5.2: Example of the selected answer by the algorithm with synonyms in text and answer.

To_Kill_a_Mockingbird

CONTEXT	Jako jižanský gotický román a Bildungsroman obsahují hlavní témata filmu Zabít ptáčka rasovou nespravedlnost a zničení nevinnosti. Učenci zaznamenali, že Lee se v americkém hlubokém jihu zabývá také otázkami třídních, odvahy, soucitu a genderových rolí. Kniha je široce vyučována ve školách ve Spojených státech s lekcemi, které zdůrazňují toleranci a dehonestující předsudky. Navzdory svým tématům je "Zabít ptáčka" předmětem kampaní za odstranění z veřejných tříd, které jsou často napadány za používání rasových nadávek.
QUESTION	O tom, jak zabít ptáčka, se hodně čte ve školách ve kterých zemích?
ANSWER	Spojené státy

Figure 5.3: Example of the selected answer by the algorithm with different declination in text and answer.

Saint_Barths

CONTEXT	Jeden senátor zastupuje ostrov ve francouzském Senátu. První volby se konaly 21. září 2008 a poslední v září 2014. Svatý Bartoloměj se dne 1. ledna 2012 stal zámořským územím Evropské unie, ale obyvatelé ostrova zůstávají francouzskými občany se statusem EU, kteří jsou držiteli pasů EU. Francie je odpovědná za obranu ostrova a jako taková na ostrově rozmístila bezpečnostní síly, které tvoří šest policistů a třináct četníků (vysláni na dvouleté období).
QUESTION	Kolik senátorů zastupuje St. Barts ve Francii?
ANSWER	Jedna

Figure 5.4: Example of the selected answer by the algorithm with non-translated numbers in text and answer.

Atlantic_City,_New_Jersey

CONTEXT	Díky své poloze v Jižním Jersey, obklopujícím Atlantský oceán mezi bažinami a ostrovy, bylo Atlantic City vnímáno developery jako prvotřídní nemovitost a potenciální rekreační město. V roce 1853 byl postaven první komerční hotel The Belloe House, který se nacházel u Massachusetts a Atlantic Avenue.
QUESTION	Jak se jmenuje první komerční hotel postavený v Atlantic City?
ANSWER	Dům Belloe

Figure 5.5: Example of the selected answer by the algorithm with partially translated names in text and answer.

5.1.4 Translated Data Analysis

After data translation and start and end indices recomputation, we measured newly created data sizes to ensure their quality. See Table 5.1. Note that we have obtained a bit different results for both sets. The number of preserved answers in the training dataset is lower than in the development dataset, which is probably caused by the character of answers as the development dataset contains more answers for one question – the question is preserved when at least one of the answers is with the required match. Also, note that the ratio of the kept data in SQuAD 1.1 is lower because unanswerable questions of SQuAD 2.0 are always preserved.

5.2 Evaluation Metrics

For evaluation, we have used the same metrics as in [Rajpurkar et al., 2016]. *Exact match score* compares translated answers with the answer in the text and returns a point if they are equal. *F1 score* is based on precision and recall. See more detailed description in Chapter 3.

The words in the Czech language are declined or conjugated which causes differences in word morphology in predicted and original answers. Afterwards, the evaluation of translated Czech dataset is not accurate. Therefore, we have lemmatized all predicted answers and the original answers in the development dataset by MorphoDiTa before comparing them during the evaluation. To obtain the model accuracy, we have evaluated these lemmatized answers to achieve more relevant results.

6. Constructing Czech Question Answering Model

We have translated SQuAD 1.1 and 2.0 datasets [Rajpurkar et al., 2016], [Rajpurkar et al., 2018] to Czech using CUBBITT [Popel et al., 2020] translator to create Czech question answering datasets and we have trained several BERT-based models: BERT [Devlin et al., 2018], Multilingual BERT [Devlin et al., 2018] and XLM-RoBERTa [Liu et al., 2019] to create Czech question answering models. The training took from 3 to 20 hours on GPU depending on the model and size of the datasets.

During the training process, the English embeddings used for the English dataset had to be changed to Czech embeddings. We have used Czech embeddings created on 4 billion Czech words using the word2vec model, keeping embeddings for the most frequent 15 million words.

After translation, starting indices of answers in newly created Czech datasets had to be recomputed. We lemmatized the translated texts and answers, we replaced the lemmas with roots of their word-formation relation trees, we found all continuous subsequences of the text with any word order, and if several occurrences were located, we chose the one with the most similar position in the text to the position of the original answer in the original text. In the dataset, we kept 73.2% of the original training part and 82.7% of the testing part in SQuAD 1.1 and we kept 82.2% of the original training part and 91.3% of the testing part in SQuAD 2.0. We have manually verified, that the dataset has high quality and still is large enough to be used for further training of question answering models.

6.1 BERT Models Selection

The current best SQuAD models are all BERT based and therefore, we trained a BERT-based architecture. We downloaded the BERT from `https://github.com/google-research/bert` and we finetuned it on the SQuAD dataset to create Czech question answering models. Our main goal is Czech reading comprehension, and therefore, we considered also multilingual BERT-based models which already included Czech in their pretraining procedure.

As a reference, we also include the English BERT [Devlin et al., 2018] base in two versions: cased and uncased. BERT cased contains all paragraphs, ques-

tions and answers with diacritics and both cases of the letters. BERT uncased is the lowercased and diacritics-stripped version of BERT cased.

Subsequently, we trained the Multilingual BERT (mBERT) [Devlin et al., 2018] in both versions: cased and uncased. It was an extension of the BERT model to more languages. It was pretrained on the top 104 languages with the largest Wikipedia using a masked language modelling objective. It extends BERT also for QA in other languages.

Finally, we trained XLMRoBERTa (XLM-R) [Liu et al., 2019] in two versions: base and large. The model was also pretrained on 2.5TB of filtered CommonCrawl data containing 100 languages. The base and large versions only differ in the size of the parameter, see Table 6.1.

Model	**Layers**	**Hidden**	**Heads**	**Parameters**
BERT	12	768	12	110 M
mBERT	12	768	12	110 M
XLM-R-base	12	768	12	125 M
XLM-R-large	24	1024	16	355 M

Table 6.1: Comparison of details of the models containing the number of layers, number of hidden layers, number of heads in the attention mechanism and number of parameters. Source `https://huggingface.co/transformers/v2.4.0/pretrained_models.html`

6.2 Selected Models Finetunning

We finetuned all models using the transformers library [Wolf et al., 2019]. For all base models, we used two training epochs, learning rate 2e-5 with a linear warm-up of 256 steps and batch size 16. For XLM-RoBERTa we increased batch size to 32 and for XLM-RoBERTa large we decreased the learning rate to 1.5e-5 and increased warm-up to 500 [Macková and Straka, 2020].

6.2.1 Epoch Number Selection

We trained BERT on the original English SQuAD 1.1 dataset with several numbers of epochs. We obtained the best results with 2 epochs. Therefore, the other models were trained with 2 epochs, see Table 6.2.

BERT	EM [%]	F1 [%]
1 epoch	79.2	87.35
2 epochs	80.81	88.27
3 epochs	80.03	87.8

Table 6.2: Comparison of results of BERT trained and evaluated on English using different numbers of epochs.

6.2.2 Finetuning Steps

- **English**

  For reference, we trained and evaluated all the above models on English SQuAD 1.1 and SQuAD 2.0 datasets. Our results are similar to the published result with slightly lower accuracy, see Table 6.3 and Table 6.4.

Model	Ref EM [%]	Ref F1 [%]	Our EM [%]	Our F1 [%]
SQuAD 1.1	84.2	91.1	81.43	88.88
SQuAD 2.0	78.8	81.9	72.85	76.03

Table 6.3: Comparison of our results of training and evaluation BERT models on English on the SQuAD 1.1 and 2.0 datasets. Source [Devlin et al., 2018]

Model	SQuAD 1.1 Ref [%]	SQuAD 2.0 Ref [%]	SQuAD 1.1 Our [%]	SQuAD 2.0 Our [%]
BERT	91.1	81.9	88.88	76.03
XLM-R	94.6	89.4	93.24	86.23

Table 6.4: Comparison of F1 scores of training and evaluation BERT models on English on the SQuAD 2.0 dataset. Source [Liu et al., 2019]

- **Czech Training and Czech Evaluation**

  Our first baseline model is trained directly on the Czech training dataset and evaluated directly on the Czech development dataset. The relative performance of the BERT variants is very similar to English, but the absolute performance is considerably lower. Several facts could contribute to the performance decrease – a smaller training set, noise introduced by the translation system and morphological richness of the Czech language [Macková and Straka, 2020], see Table 6.5.

- **English Models and Czech Evaluation via Machine Translation**

  Our second baseline system (denoted C-E-C in the results) reuses English models to perform Czech reading comprehension – the Czech development set is first translated to English, and the answers are then generated using English models, and finally translated back to Czech. The translation-based approach has slightly higher performance for base models, which may be caused by the smaller size of the Czech training data. However, for the large model, the direct approach seems more beneficial [Macková and Straka, 2020], see Table 6.6.

- **Cross-lingual Transfer Models**

  The most interesting experiment is the cross-lingual transfer of the English models, evaluated directly on Czech (without using any Czech data for training). Astonishingly, the results are very competitive with the other models evaluated on Czech, especially for XLM-R large, where there are within 1.6 percentage points in F1 score and 2.75 percentage points in an exact match of the best Czech model [Macková and Straka, 2020], see Table 6.7.

Table 6.5: Development performance of models trained and evaluated in Czech on translated Czech SQuAD 1.1 and 2.0 datasets.

Model	**SQuAD 1.1**		**SQuAD 2.0**	
	EM [%]	**F1** [%]	**EM** [%]	**F1** [%]
mBERT cased	59.49	70.62	66.60	69.61
mBERT uncased	62.11	73.94	64.96	68.14
XLM-R base	69.18	78.71	64.98	68.15
XLM-R large	76.39	85.62	75.57	79.19

Table 6.6: Development performance of models trained on English and evaluated with translations from Czech to English and then back to Czech on SQuAD 1.1 and 2.0 datasets.

Model	**SQuAD 1.1**		**SQuAD 2.0**	
	EM [%]	**F1** [%]	**EM** [%]	**F1** [%]
BERT cased	64.06	76.78	64.35	69.11
BERT uncased	63.57	76.61	65.26	69.86
mBERT cased	65.09	77.47	67.40	71.96
mBERT uncased	65.00	77.38	66.20	70.72
XLM-R base	64.52	76.91	65.62	70.00
XLM-R large	69.04	81.33	72.82	78.04

Table 6.7: Development performance of models trained on English and evaluated directly on Czech without any translation using SQuAD 1.1 and 2.0 datasets.

Model	**SQuAD 1.1**		**SQuAD 2.0**	
	EM [%]	**F1** [%]	**EM** [%]	**F1** [%]
BERT cased	9.53	21.62	53.48	53.84
BERT uncased	6.16	21.75	54.78	54.83
mBERT cased	59.49	70.62	58.28	62.76
mBERT uncased	62.09	73.89	59.59	63.89
XLM-R base	64.63	75.85	62.09	65.93
XLM-R large	73.64	84.07	73.50	77.58

6.3 Overall Results

All our results are presented in Table 6.8 and graphically in Figure 6.1.

Table 6.8: Development performance of English and Czech models on SQuAD 1.1 and 2.0 datasets. Source [Macková and Straka, 2020].

Model	**Train**	**Dev**	**SQuAD 1.1**		**SQuAD 2.0**	
			EM [%]	**F1** [%]	**EM** [%]	**F1** [%]
BERT cased	EN	EN	81.43	88.88	72.85	76.03
BERT uncased	EN	EN	80.92	88.59	73.35	76.59
mBERT cased	EN	EN	81.99	89.10	75.79	78.76
mBERT uncased	EN	EN	81.98	89.27	74.88	77.98
XLM-R base	EN	EN	80.91	88.11	74.07	76.97
XLM-R large	EN	EN	87.27	93.24	83.21	86.23
BERT cased	EN	CZ	9.53	21.62	53.48	53.84
BERT uncased	EN	CZ	6.16	21.75	54.78	54.83
mBERT cased	EN	CZ	59.49	70.62	58.28	62.76
mBERT uncased	EN	CZ	62.09	73.89	59.59	63.89
XLM-R base	EN	CZ	64.63	75.85	62.09	65.93
XLM-R large	EN	CZ	73.64	84.07	73.50	77.58
BERT cased	EN	C-E-C	64.06	76.78	64.35	69.11
BERT uncased	EN	C-E-C	63.57	76.61	65.26	69.86
mBERT cased	EN	C-E-C	65.09	77.47	67.40	71.96
mBERT uncased	EN	C-E-C	65.00	77.38	66.20	70.72
XLM-R base	EN	C-E-C	64.52	76.91	65.62	70.00
XLM-R large	EN	C-E-C	69.04	81.33	72.82	78.04
mBERT cased	CZ	CZ	59.49	70.62	66.60	69.61
mBERT uncased	CZ	CZ	62.11	73.94	64.96	68.14
XLM-R base	CZ	CZ	69.18	78.71	64.98	68.15
XLM-R large	CZ	CZ	76.39	85.62	75.57	79.19

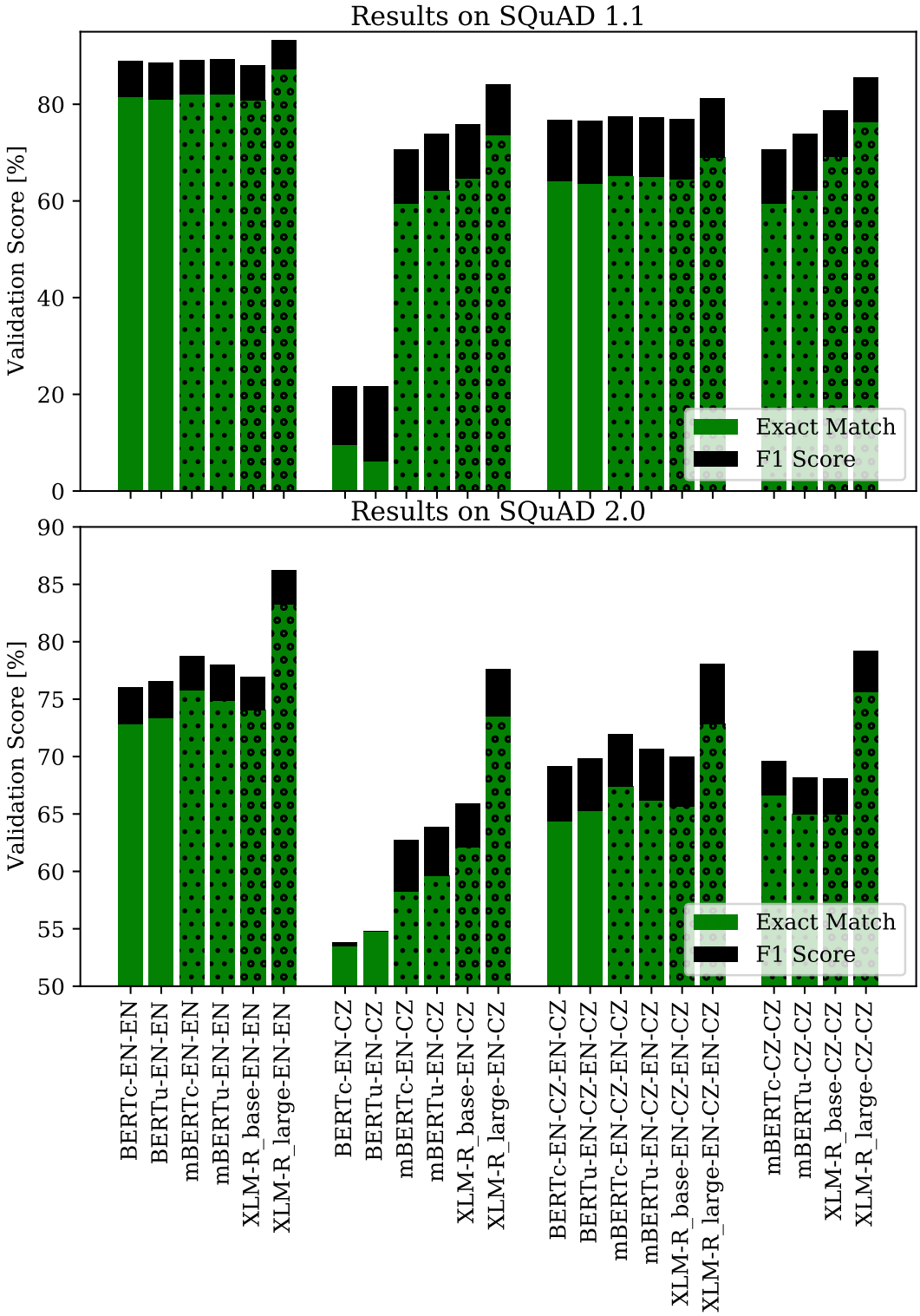

Figure 6.1: Development set performance of all models for English and Czech SQuAD 1.1 and SQuAD 2.0 datasets. Source [Macková and Straka, 2020].

6.4 Main Findings

- **Why Does Cross-lingual Transfer Work**

  The performance of the cross-lingual transfer model is surprising. This model never saw any Czech reading comprehension data or any parallel Czech-English data before. Despite this, it reaches nearly the best results among all evaluated models. This strong performance is an indication that mBERT and XLM-R represent different languages in the same shared space, without getting an explicit training signal in form of parallel data. Instead, we hypothesise that if there is a large enough similarity among languages the model exploits by reusing the same part of the network to handle this phenomenon across multiple languages. This in turn saves the capacity of the model and allows reaching a higher likelihood, improving the quality of the model. Furthermore, word embeddings for different languages demonstrate a remarkable amount of similarity even after a simple linear transformation, as demonstrated for example by [Artetxe et al., 2018]. Such similarities are exploitable (and as indicated by the results also exploited) by BERT-like models to achieve shared representation of multiple languages [Macková and Straka, 2020].

- **Pre-training on Czech is Required**

  The strong performance of cross-lingual models does not necessarily mean the models can "understand" Czech – the named entities could be similar enough in Czech and English, and the model could be capable of answering without understanding the question. Therefore, we also considered an English reading comprehension model based on English BERT, which did not encounter any other language but English during pretraining. Evaluating such a model directly on Czech delivers surprisingly good performance on SQuAD 2.0 – the model is unexpectedly good in recognizing unanswerable questions. However, the performance of such a model on SQuAD 1.1 is rudimentary – 9.53% exact match and 21.62% F1 score, compared to 62.90% exact match and 73.89% F1 score of an mBERT uncased model [Macková and Straka, 2020].

- **Cased versus Uncased**

  Consistently with intuition, cased models seem to perform generally better than uncased. However, in the context of cross-lingual transfer, we repeatedly observed uncased models surpassing the cased ones. We hypothesise

that this result could be caused by a larger intersection of Czech and English subwords of the uncased models (which discard not only casing information, but also diacritical marks) because a larger shared vocabulary could make the cross-lingual transfer easier [Macková and Straka, 2020].

The paper [Lewis et al., 2019] published in November 2019 was concerned with a similar problem. The authors also trained the BERT model for question answering in 6 different languages however, the Czech language was not used. They confirmed our hypothesis and results that Multilingual BERT is good even for other languages than English.

6.5 Summary

We have explored Czech reading comprehension without any manually annotated Czech training data using BERT-based models and SQuAD datasets. We trained several baseline BERT-based models using translated data. We also evaluated a cross-lingual transfer model trained on English and evaluated directly on Czech to avoid unnecessary translations. The performance of this model is exceptionally good, even though no Czech training data nor Czech translation system was needed to train it. This model achieved 73.64% EM and 84.07% F1 score on SQuAD 1.1 and 73.50% EM and 77.58% F1 score on SQuAD 2.0 datasets. These results are overpassing all the BERT, multilingual BERT and XLM-R base models trained on Czech or English and evaluated on Czech with or without translations. Its accuracy is comparable to other XLM-R large models with a huge advantage – it does not require any Czech data to train and the problem with the translation noise is eliminated.

7. Conclusion

In this thesis, we explored Czech reading comprehension and question answering without any manually annotated Czech training data. We translated the English datasets SQuAD 1.1 and SQuAD 2.0 to Czech to create Czech training and development datasets. We trained several baseline BERT-based models using original English data and also translated Czech data. We also evaluated a cross-lingual transfer model trained on English and then evaluated directly on Czech without the necessity of any translations. We compared all the models' results and selected the best one for the Czech QA task.

In particular, we trained and evaluated BERT and XLM-RoBERTa models on the Czech dataset. We also trained them on the original English dataset and we evaluated them on the Czech dataset translated to English and we translated the English answers from the model back to Czech. Finally, we trained Multilingual BERT in English and we evaluated on Czech dataset without any requirements for the data translation.

We compared the results and we observed that training in English gives better overall results than training in Czech. Moreover, XLM-RoBERTa has achieved much higher results in all QA tasks than BERT-based models in all compared approaches.

If we compare the results of the training of all models trained on Czech and evaluated on Czech, we can see that XLM-RoBERTa large is significantly better than the other models and the other models have lower but similar accuracy. The same result can be observed comparing models trained on English and evaluated on the data with Czech-English-Czech translations and also on models trained and evaluated only on English.

The most interesting comparisons can be observed on models trained on English and evaluated directly on Czech without the necessity of any translations. The basic BERT models have reached much worse results than the other models. It is caused by the fact that they were trained only in English and they have not seen any other language data before. Multilingual BERTs have similar results to the basic RoBERTA and overall they are slightly worse than the results with the translations.

We obtained the most surprising results with RoBERTa large which has reached similar accuracy as the other RoBERTa large models trained on Czech or on English with the translations. This result is extremely good, as the model has not seen any Czech data during training. We hypothesise that if there is

a large enough similarity among languages, the model exploits it by reusing the same part of the network to handle this phenomenon across multiple languages. This in turn saves the capacity of the model and allows reaching a higher likelihood, improving the quality of the model. This cross-lingual transfer approach is very flexible and provides reading comprehension in any language, for which we have enough monolingual raw texts, see Table 6.8.

Bibliography

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. *CoRR*, abs/1805.06297, 2018. URL http://arxiv.org/abs/1805.06297.

Michael A. Bauer and Daniel Berleant. Usability survey of biomedical question answering systems. *Human genomics*, 6(1):1–4, 2012. doi: 10.1186/1479-7364-6-17. URL https://link.springer.com/article/10.1186/1479-7364-6-17.

Danqi Chen, Jason Bolton, and Christopher D. Manning. A Thorough Examination of the CNN/Daily Mail Reading Comprehension Task. *CoRR*, abs/1606.02858, 2016. URL http://arxiv.org/abs/1606.02858.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading Wikipedia to Answer Open-Domain Questions. *CoRR*, abs/1704.00051, 2017. URL http://arxiv.org/abs/1704.00051.

David N. Chin. Knowledge Structures in UC, the UNIX Consultant. In *Proceedings of the 21st Annual Meeting on Association for Computational Linguistics*, ACL '83, page 159–163, USA, 1983. Association for Computational Linguistics. doi: 10.3115/981311.981342. URL https://doi.org/10.3115/981311.981342.

Philipp Cimiano, Christina Unger, and John McCrae. Ontology-Based Interpretation of Natural Language. *Synthesis Lectures on Human Language Technologies*, 7(2):1–178, 2014. URL https://www.morganclaypool.com/doi/abs/10.2200/S00561ED1V01Y201401HLT024.

Christopher Clark and Matt Gardner. Simple and Effective Multi-Paragraph Reading Comprehension. *CoRR*, abs/1710.10723, 2017. URL http://arxiv.org/abs/1710.10723.

Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Embodied question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–10, 2018. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Das_Embodied_Question_Answering_CVPR_2018_paper.html.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR*, abs/1810.04805, 2018. URL http://arxiv.org/abs/1810.04805.

Bert F. Green, Alice K. Wolf, Carol Chomsky, and Kenneth Laughery. Baseball: an automatic question-answerer. In *Papers presented at the May 9-11, 1961, western joint IRE-AIEE-ACM computer conference*, pages 219–224, 1961. URL https://dl.acm.org/doi/abs/10.1145/1460690.1460714.

Otthein Herzog and Claus-Rainer Rollinger. The LILOG inference engine. In *Text Understanding in LILOG: Integrating Computational Linguistics and Artificial Intelligence Final Report on the IBM Germany LILOG-Project*, pages 402–427. Springer Berlin Heidelberg, 1991. ISBN 978-3-540-38493-9. doi: 10.1007/3-540-54594-8_72. URL https://doi.org/10.1007/3-540-54594-8_72.

Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. He Goldilocks Principle: Reading Children's Books With Explicit Memory Representations. 2015. doi: 10.48550/ARXIV.1511.02301. URL https://arxiv.org/pdf/1511.02301.

Matthew B. Hoy. Wolfphram—Alpha: A Brief Introduction. *Medical Reference Services Quarterly*, 29(1):67–74, 2010. doi: 10.1080/02763860903485225. URL https://doi.org/10.1080/02763860903485225.

Tsung-Yuan Hsu, Chi-Liang Liu, and Hung-yi Lee. Zero-shot Reading Comprehension by Cross-lingual Transfer Learning with Multi-lingual Language Representation Model. *CoRR*, abs/1909.09587, 2019. URL http://arxiv.org/abs/1909.09587.

Daniel Kondratyuk. 75 Languages, 1 Model: Parsing Universal Dependencies Universally. *CoRR*, abs/1904.02099, 2019. URL http://arxiv.org/abs/1904.02099.

Patrick S. H. Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. MLQA: Evaluating Cross-lingual Extractive Question Answering. *CoRR*, abs/1910.07475, 2019. URL http://arxiv.org/abs/1910.07475.

Jimmy Lin. The Web as a Resource for Question Answering: Perspectives and Challenges. In *LREC*. Citeseer, 2002. URL https://aclanthology.org/L02-1085/.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*, abs/1907.11692, 2019. URL `http://arxiv.org/abs/1907.11692`.

Kateřina Macková. *Crosslingual Transfer in Question Answering*. PhD thesis, Charles University, 2020. URL `https://is.cuni.cz/studium/dipl_st/index.php?id=&tid=&do=main&doo=detail&did=221320`.

Kateřina Macková and Milan Straka. Reading Comprehension in Czech via Machine Translation and Cross-lingual Transfer. *CoRR*, abs/2007.01667, 2020. URL `https://arxiv.org/abs/2007.01667`.

Matthew E. Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. Dissecting Contextual Word Embeddings: Architecture and Representation. *CoRR*, abs/1808.08949, 2018. URL `http://arxiv.org/abs/1808.08949`.

Martin Popel, Markéta Tomková, Jakub Tomek, Łukasz Kaiser, Jakob Uszkoreit, Ondřej Bojar, and Zdeněk Žabokrtský. Transforming machine translation: a deep learning system reaches news translation quality comparable to human professionals. *Nature communications*, 11(1):1–15, 2020. URL `https://www.nature.com/articles/s41467-020-18073-9`.

John Prager, Eric Brown, Anni Coden, and Dragomir Radev. Question-Answering by Predictive Annotation. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, page 184–191. Association for Computing Machinery, 2000. ISBN 1581132263. doi: 10.1145/345508.345574. URL `https://doi.org/10.1145/345508.345574`.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100, 000+ Questions for Machine Comprehension of Text. *CoRR*, abs/1606.05250, 2016. URL `http://arxiv.org/abs/1606.05250`.

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know What You Don't Know: Unanswerable Questions for SQuAD. *CoRR*, abs/1806.03822, 2018. URL `http://arxiv.org/abs/1806.03822`.

Matthew Richardson, Christopher J.C.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. URL https://www.researchgate.net/publication/286965813_MCTest_A_challenge_dataset_for_the_open-domain_machine_comprehension_of_text.

Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional Attention Flow for Machine Comprehension. *CoRR*, abs/1611.01603, 2016. URL http://arxiv.org/abs/1611.01603.

Jana Straková, Milan Straka, and Jan Hajič. Open-Source Tools for Morphology, Lemmatization, POS Tagging and Named Entity Recognition. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 13–18. Association for Computational Linguistics, 2014. URL http://www.aclweb.org/anthology/P/P14/P14-5003.pdf.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. NewsQA: A Machine Comprehension Dataset. *CoRR*, abs/1611.09830, 2016. URL http://arxiv.org/abs/1611.09830.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. *CoRR*, abs/1706.03762, 2017. URL http://arxiv.org/abs/1706.03762.

Jonáš Vidra, Zdeněk Žabokrtský, Magda Ševčíková, and Lukáš Kyjánek. Derinet 2.0: towards an all-in-one word-formation resource. In *Proceedings of the Second International Workshop on Resources and Tools for Derivational Morphology*, pages 81–89, 2019. URL https://aclanthology.org/W19-8510.pdf.

Ellen M. Voorhees and Dawn M. Tice. Building a question answering test collection. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 200–207, 2000. URL https://trec.nist.gov/data/qa.html.

Joseph Weizenbaum. ELIZA—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45, 1966. URL https://dl.acm.org/doi/pdf/10.1145/365153.365168.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. HuggingFace's Transformers: State-of-the-art Natural

Language Processing. *CoRR*, abs/1910.03771, 2019. URL http://arxiv.org/abs/1910.03771.

William Woods. The lunar sciences natural language information system: final report. *BBN report*, 1972. URL https://cir.nii.ac.jp/crid/1574231874544133376.

Yi Yang, Wen tau Yih, and Christopher Meek. WikiQA: A Challenge Dataset for Open-Domain Question Answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018. Association for Computational Linguistics, 2015. doi: 10.18653/v1/D15-1237. URL https://aclanthology.org/D15-1237.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension. *CoRR*, abs/1804.09541, 2018. URL http://arxiv.org/abs/1804.09541.

Junbei Zhang, Xiao-Dan Zhu, Qian Chen, Li-Rong Dai, Si Wei, and Hui Jiang. Exploring Question Understanding and Adaptation in Neural-Network-Based Question Answering. *CoRR*, abs/1703.04617, 2017. URL http://arxiv.org/abs/1703.04617.

List of Figures

List of Tables

A. Overview of Electronic Attachments

- **Czech SQuAD**

  Translated SQuAD training and development datasets to Czech are too huge to be attached, but we have released them at `https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-3249`.

- **English SQuAD**

  Source SQuAD 1.1 and SQuAD 2.0 training and development datasets are too huge to be attached, but they are available `https://rajpurkar.github.io/SQuAD-explorer/`.

- **Article**

  We have published an article about reading comprehension in Czech via machine translation and cross-lingual transfer [Macková and Straka, 2020]. The arXiv preprint of the original article is attached.

- **Scripts**

  The file containing translation, preprocessing, lemmatizaton, evaluation and visualization scripts.

  - *compare_lcs_and_accord.py*
  - *create_html_visualization.py*
  - *evaluate-v1.1.py*
  - *lemmatize_dev.py*
  - *lemmatize_pred.py*
  - *select_data_above_threshold.py*
  - *translate_answers_EN-CZ.py*
  - *translate_dev_CZ-EN.py*
  - *translate_dev_EN-CZ.py*
  - *translate_predictions_to_cz.py*
  - *visualize_czech_several_epochs_epochs.py*
  - *vizualize_all_results_all_models.py*
  - *vizualize_data_sizes_after_translation.py*