TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Anu Käver 182912IAPM

# Extractive Question Answering for Estonian Language

Master's thesis

Supervisor: Tanel Alumäe
PhD

Tallinn 2021

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Anu Käver 182912IAPM

# Eestikeelse küsimus-vastus-süsteemi arendamine

Magistritöö

Juhendaja: Tanel Alumäe
PhD

Tallinn 2021

# Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Anu Käver

10.05.2021

# Abstract

Extractive question answering is a task in the field of natural language processing where the system answers a question about a pregiven text, posed by humans in their natural language, by extracting the smallest continuous span of text that is suitable for the answer. Developing an extractive question answering system requires large training resources – tens of thousands of annotated question-answer pairs. Such resources are not available in majority of languages, including Estonian.

The thesis compares different methods for composing an extractive question answering model in Estonian, taking benefit from transfer learning. Traditional methods for such scenarios are used (translate-train, translate-test and zero-shot) along with experiments to combine them. Different large pretrained contextual language models are used as the basis – two multilingual models, XLM-RoBERTa and multilingual BERT, and monolingual Estonian model EstBERT. Best training method is proposed, combining methods that are usually applied in isolation. Out of underlying language models, the best results are achieved with XLM-RoBERTa. For testing and fine-tuning purposes, new native Estonian QA dataset with 1115 questions is presented. The dataset is available for download through META-SHARE.[1]

This thesis is written in English and is 45 pages long, including 7 chapters, 9 figures and 11 tables.

[1] https://metashare.ut.ee/repository/browse/estqa-question-answering-dataset/dabdfdeaa74911eba6e4fa163e9d45471d05c5c43d8e46788aac3c1694c0e4ac/ (short URL: https://tinyurl.com/2dppnvb8)

# Annotatsioon

Eestikeelse küsimus-vastus-süsteemi arendamine

Ekstraheeriv küsimustele vastamine on loomuliku keele töötluse valdkonda kuuluv ülesanne. Selle käigus vastab süsteem inimeste poolt loomulikus keeles esitatud küsimustele etteantud teksti kohta, leides vastuseks lühima tekstilõigu, mis küsimusele vastab. Ekstraheeriva küsimus-vastus-süsteemi arendamiseks on vajalik suur treeningandmestik, mis ulatub kümnete tuhandete küsimus-vastus-paarideni. Enamiku keelte, nende seas eesti keele jaoks selline andmestik puudub.

Käesolev magistritöö võrdleb erinevaid meetodeid, kuidas arendada eesti keele jaoks ekstraheerivat küsimus-vastus-süsteemi, kasutades teadmiste ülekannet (*transfer learning*). Inglise keeles eksisteerivat suurt treeningandmestikku kasutatakse eestikeelse süsteemi arendamiseks. Teadmiste ülekande meetoditest kasutatakse nii selliseid, mis on teemakohases kirjanduses levinud, kui ka kombineeritakse neid, jõudes uudsete treeningmeetoditeni. Levinud meetoditest on kasutuses treeningandmestiku tõlkimine (*translate-train*), testandmete tõlkimine (*translate-test*) ja nii-öelda null-lasu meetod (*zero-shot*), kus aluseks olevat mitmekeelset keelemudelit treenitakse inglise keeles ja testitakse koheselt eesti keeles. Tõlkemeetodite puhul kasutatakse magistritöös masintõlget.

Arendatavate mudelite alusena kasutatakse suuri kontekstipõhiseid, eeltreenitud keelemudeleid – kaht mitmekeelset mudelit, XLM-RoBERTa-t ja mitmekeelset BERT-i, ning eestikeelset mudelit EstBERT.

Mudelite testimiseks, aga ka täiendavaks peenhäälestamiseks koostatakse uus eestikeelne 1115 küsimusega andmestik, mis põhineb eestikeelse Vikipeedia artiklitel. Nii andmestiku koostamise metoodika kui ka struktuur ja küsimuste-vastuste tüübid jälgivad levinud ingliskeelse andmestiku SQuAD eeskuju. Andmestik on avalikustatud

keeleressursside registris META-SHARE.[1] Ingliskeelseks andmestikuks, mida kasutatakse treeningul nii originaalkujul kui tõlgituna, on samuti SQuAD.

Magistritöös pakutakse välja parim treeningmetoodika, mille abil arendada väheste treeningandmetega keele jaoks küsimus-vastus-süsteemi. Selleks on kombineeritud meetod, kus mudelit treenitakse alguses suure ingliskeelse andmestikuga, seejärel sama andmestiku tõlgitud versiooniga ning viimaks peenhäälestatakse osaga uuest eestikeelsest andmestikust.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 45 leheküljel, 7 peatükki, 9 joonist, 11 tabelit.

[1] https://metashare.ut.ee/repository/browse/estqa-question-answering-dataset/dabdfdeaa74911eba6e4fa163e9d45471d05c5c43d8e46788aac3c1694c0e4ac/ (lühike aadress: https://tinyurl.com/2dppnvb8)

# List of abbreviations and terms

| | |
|---|---|
| BERT | A language representation model (Bidirectional Encoder Representations from Transformers). Entails deep bidirectional representations of language, having been conditioned both on left and right context during the training. Model was trained with masked language modelling and next sentence prediction objectives. |
| Deep learning | A subset of machine learning algorithms based on multi-layered artificial neural networks, with the goal to progressively extract higher-level features from raw input. |
| EM | Exact Match – measure of performance of a Question Answering system. Value for each question-answer pair is 1 if proposed answer matches exactly one of possible ground-truth answers, and 0 otherwise. For a dataset, the arithmetic average is calculated across all question-answer pairs. |
| EstBERT | Language model based on BERT, trained on Estonian cased corpus. |
| EstQA | Estonian Question Answering dataset with 1115 questions, developed for the current master's thesis. |
| Extractive QA | Also known as answer extraction. Type of tasks under selective QA where system answers questions by finding the suitable phrase from a given context. It differs for example from answer selection, where system finds the whole sentence that contains answer. |
| F1 score | Measure of performance of a Question Answering system. Value for each question-answer pair is between 0 and 1. Measures the proportion of words in the proposed answer with all possible ground-truth answers. Combines precision and recall. Score for the best possible match is used. For a dataset, the arithmetic average is calculated across all question-answer pairs. |
| Golden answer | Synonym for ground truth answer in the field of QA; answer that is marked as correct in the training/test dataset. |
| mBERT | Multilingual version of language model BERT. |
| MLM | Masked Language Modelling. Technique where during the training of a language model, part of input is masked, and training goal is to predict the masked input. |

| | |
|---|---|
| NLP | Natural Language Processing. |
| QA | Questions Answering, a subdomain of natural language processing where the goal of the system is to answer a question formulated in natural language. |
| RoBERTa | Language model based on BERT, with modified hyperparameters, longer training time, training with larger mini-batches and learning rates, also without the next-sentence prediction objective. |
| Selective QA | Form of QA tasks where system does not generate the answer to the question but instead selects it from pre-existing texts. |
| SQuAD | Stanford Question Answering Dataset; designed for extractive question answering tasks, based on English Wikipedia articles. Version 1 contains ca 100 000 question-answer pairs, version 2 has in addition ca 50 000 questions for which answer cannot be found from the context. |
| Transfer learning | A set of machine learning techniques where resources or systems developed for one task are used as the starting point of performing another, different task. |
| XLM-RoBERTa | Multilingual version of language model RoBERTa. |

# Table of contents

# List of figures

# List of tables

# 1 Introduction

Natural Language Processing (NLP) plays an integral role in the development of artificial intelligence systems. The goal of NLP is to produce systems that are capable of handling human languages on the level that is comparable or higher to that of human beings. This involves text and speech comprehension and generation, analysis, classification, translation, etc [1]. In the category of language comprehension, an important set of tasks are those of Question Answering (QA) which seek to find information from texts in response to questions proposed by humans in their natural language. The application areas of QA include search engines, chatbots and dialogue systems. Given the large quantities of electronically available data, the finding and extraction of specific information is a task of high relevance.

This master's thesis will develop a QA system in Estonian language. Due to lack of training resources required for such a system, there is no Estonian QA system yet available. The thesis will use different strategies to overcome the lack of resources, proposing also an original new resource, a dataset of 1115 questions/answers in Estonian (EstQA) that can be used for developing and evaluating such QA systems. Both the developed model and the EstQA dataset are publicly available, along with a demo environment for investigating its capabilities[1].

The exact task at hand is extractive question answering. It is described by Table 1 which depicts a sample from the new Estonian dataset. The new system must be capable, based on a given passage of text (context), to find an answer to a naturally phrased question. The answer is the shortest suitable continuous span in the passage. The system is in

[1] **Model**: https://huggingface.co/anukaver/xlm-roberta-est-qa
**Dataset**: https://huggingface.co/datasets/anukaver/EstQA and https://metashare.ut.ee/repository/browse/estqa-question-answering-dataset/dabdfdeaa74911eba6e4fa163e9d45471d05c5c43d8e46788aac3c1694c0e4ac/ (short URL: https://tinyurl.com/2dppnvb8)
**Demo application**: https://qa.akaver.com

essence predicting the start and end indexes of the answer. Extractive QA is compared to some other QA tasks in chapter 3.

Table 1. Context, question, and answer in new EstQA dataset.

| | **Original in Estonian** | **Translation to English** |
|---|---|---|
| **Context** | Üldist teenistuskohustust asuti riikides kodanikele kehtestama peamiselt **19. sajandil**. Toonane taristu ja tehnika (sh raudtee, laevandus, side, raskerelvastus; hiljem juba ka autod ja lennundus) kiire areng võimaldas koondada sõja pidamiseks suuri sõjaväelaste ja vahendite hulki ning paisata neid kiirelt ja ootamatult pikkade vahemaade taha. | General service obligations were introduced to citizens in the countries mainly **in the 19th century**. The rapid development of infrastructure and technology at that time (including railways, shipping, communications, heavy weapons; later also cars and aviation) made it possible to mobilize large numbers of troops and equipment for war and to deploy them quickly and unexpectedly over long distances. |
| **Question** | Millal sai ajateenistus alguse? | When did conscription begin? |
| **Answer** | 19. sajandil | in the 19th century |

Current state-of-the-art results in QA are achieved via deep learning [2], using in general two components, both with their own requirements for resources.

First, large contextual[1] language models are used as the basis of the QA system. Such universal models, e.g., BERT [3], give a mathematical representation of a given natural language with its syntax, grammar, and internal relations. They entail the vocabulary of the language as vectors in a high-dimensional vector space. The dimensionality (hundreds of dimensions) allows for the representation of complex relations between elements of the language.

Second, such models can be used for subsequent fine-tuning on a variety of downstream NLP tasks, including Question Answering. This requires large QA-specific datasets, meaning a big quantity of questions and answers, along with the context paragraphs where answers must be found from.

[1] While training the model, emphasis is put on the context of the words in the source texts.

Both components of successful QA systems – large pretrained language models and datasets for fine-tuning, are naturally bound by language context. Knowledge that a machine learning system has obtained about English language, cannot be directly used to answer questions for example in Estonian. This constitutes a problem for training QA systems in so-called low-resource languages where large datasets are not available (arguably, every language besides English and Chinese [4]). Among those is also Estonian.

There exist several large language models that entail knowledge about Estonian and can be used as basis for QA systems. First, there are models trained on multiple languages simultaneously – for example, multilingual BERT (mBERT) and XLM-RoBERTa [5] which are trained on 104 and 100 languages respectively, including Estonian. Second, there is also a model based purely on Estonian language, EstBERT, which was published in November 2020 [6]. The training and architecture of EstBERT followed the example of BERT.

As to the datasets suitable for fine-tuning language models for Question Answering task, then there are none available in Estonian. The problem is the required quantity of data. The most widely used dataset in English, the Stanford Question Answering Dataset (SQuAD) [7] in its initial version consists of 100 000 pairs of questions and answers. SQuAD is based on Wikipedia articles. The second version of SQuAD adds another 50 000 samples, those being unanswerable questions – based on the context, it is not possible to answer the questions [8]. Another popular dataset called NewsQA [9] that relies on CNN articles, has 120 000 pairs of questions and answers. The effort of collecting datasets of comparable size is considerable.

There have been several strategies employed by researchers to develop QA systems in low-resource languages where large question-answer datasets or dedicated large language-models are not available. The idea is to use resources that are available in high-resource languages in the benefit of low-resource target languages (transfer learning).

Strategies for the transfer learning in QA are mainly machine translation (translating training set to target language or test set to source language), and so-called zero-shot strategy. Zero-shot means that a multilingual language model is trained on existing data in high-resource language, and directly used on the desired target language. The analogue

is that of a polyglot who is taught his new work assignment in English but must perform the task in another language that he knows. Translation and zero-shot strategies are compared for example in [10] and [11].

There is also the possibility to not use transfer learning but despite the effort, compose the needed dataset in the low-resource target language. This was done for example for French language by project PIAF described in [4], relying on the process used in composing SQuAD, with focus on crowdsourcing. PIAF ended up containing 3800 question/answer pairs.

All those strategies give a good starting point to develop Question Answering system also for Estonian as a low-resource language.

However, majority of papers use each of those strategies separately, comparing the results. They are not combined to, possibly, enforce each other. This leaves room for experiments. For example, a multilingual language model can be fine-tuned not using just an English dataset (for zero-shot strategy), or the same set machine-translated to Estonian – but both. This can be done either training with the two datasets in sequence, or by shuffling them and obtaining a dataset twice the size. Also, to develop a QA model at all, there must be obtained at least a small dataset in the target language, as is also done in the current thesis. Otherwise, it is impossible to test the performance. But it may also be beneficial to split this small dataset and use one part of it also in training phase, despite of the small quantity of data. This has been suggested also in [12].

Those combined strategies introduce the same problem to the model from different angles, thus possibly giving it more generalizing power.

This master's thesis is comparing the use of those strategies separately and in combinations. Combining the strategies results in clear improvement of the performance. As the underlying language model, multilingual models (multilingual BERT and XLM-RoBERTa), as well as monolingual EstBERT are used in comparison. The best method for training a multilingual QA model is proposed based on the experiments and comparisons.

The models developed for the thesis and the 1115-question new EstQA dataset can be used in follow-up Estonian language NLP tasks, and concrete applications, e.g., search engines or other language comprehension tasks.

All work done for this thesis is individual effort, except for building the hosting and continuous integration pipeline for the demo application. Outside help was used for that.

# 2 Related work

As mentioned in the introduction, the datasets needed to develop a Question Answering system reach more than 100 000 question-answer pairs. Smaller quantities can be used, but with losses in performance.

In recent years, many efforts have been made to tackle the issue of training QA models for low-resource languages. The goal for many research papers has been to avoid the path of collecting the necessary large resources, and instead use the resources from high-resource languages via different methods of transfer learning. The most common of those methods are introduced in Table 2.

Table 2. Common strategies of transfer learning for QA.

| **Strategy** | **Explanation** |
|---|---|
| Translate-train | The training dataset is translated from high-resource source language to target language. It is used to train a multilingual model, or a model in target language, if it exists. |
| Translate-test | The test dataset is translated into high-resource language. Training also takes place on the high-resource language. With this setup, the input to the model must always be translated into the high-resource language and answer translated back to the low-resource language, thus adding some overhead. |
| Zero-shot | A multilingual model is trained on the dataset in high-resource language. In testing phase, the model is directly applied on the low-resource target language. If some data in low-resource language is also available for training, the strategy may also be referred to as few-shot. |

The strategies are most often all used comparatively, although there are also research papers concentrating for example only on translation-based methods. In following overview, the cases where common training methods have been combined in some form, have been emphasized.

## 2.1 Comparison of strategies in two-step Question Answering

Research done by Liu et al [10] compares the three most common methods for Question Answering (Table 2) in eight lower-resource languages. The work focuses on two-step Question Answering where first step is the selection of relevant document from among many documents, second is the extraction of the question from the document (the task in this thesis focuses only on the second). The dataset was gathered automatically from Wikipedia's daily "Did you know?" box, and later made publicly available by the name XQA. Training set existed only in English. Development and test sets for other languages ranged from ca 350 to 3000 samples.

The results showed that zero-shot model based on multilingual BERT performed best across almost all languages. For the translation-based methods, translate-train was used only for Chinese and German where it mostly performed better than translate-test. The authors pointed out that translation-based methods performed worse, because they depend heavily on the quality of machine translation. For different languages, the quality varied, and authors presented some obvious translation errors where even named entities were translated incorrectly.

The results of the work were not spectacular: the F1 score (see more in chapter 5) that measures the proportion of matching words between predicted answer and possible ground truth answers, varied between 13 and 40 %. This goes for the combination of two steps in the pipeline: the selection of documents plus answer extraction. Authors still admitted as one of their conclusions that cross-lingual QA is in fact a difficult task.

## 2.2 Comparison of strategies on answer extraction

All the three methods were also used by Lewis et al [11] for seven languages, concentrating on extracting the answer from a pre-given document.

As the dataset, they used SQuAD for training, and for each language the test set was chosen from identical articles from Wikipedia. The questions were first crowd-sourced on English version, then human-translated to specific target languages. The dataset was made publicly available under the name MLQA. Authors pointed out one questionable feature regarding their source material quality – identical articles from Wikipedia had

achieved their identicalness usually through being initially machine-translated, which is a strongly discouraged practice in Wikipedia community.

Given that the task for this research was containing only one step, the results were vastly better with F1 score varying between 54 and 68 %. Here also, zero-shot showed the best results in most of the languages. Two underlying multilingual models were used, XLM and multilingual BERT, whereas XLM (a work extending on BERT, see more from chapter 4.1.4) always performed better than multilingual BERT. From translation-based methods, translate-train here also outperformed translate-test.

Good results in extracting answers from pre-given texts have been achieved for Chinese language. The size of available training data for Chinese is relatively high, compared to many other languages. For example, both Jing and Xiong [13] and Cui et al [14] have been working on two datasets with 10 000 and 27 000 samples respectively. This allows them to use techniques unavailable for lower-resource languages.

Cui et al compared a translate-test system with a new model, a dual mBERT model which they trained simultaneously on Chinese QA data and the same data translated to English. This is possible only in the existence of sufficient native training data. The dual model helped them to achieve F1 scores of 90.2 to 91.6 on two Chinese test sets.

Jing and Xiong made an experiment on combining different strategies, similar to one of the experiments in current thesis. They compared zero-shot transfer and translate-train with a method where translated and non-translated data was shuffled (as is done in chapter 4.2.7). However, they did not use pretrained contextual language models. The F1 scores of their models were inferior to BERT-models by at least 5 percentage points, but still reached as high as 81%. Shuffling the data produced the best results. Translate-train and zero-shot results were similar, both lagging by 1-2 percentage points. This research supports the idea of combining common training strategies.

Jing and Xiong also built an ensemble of their different models – the results from several models were compared and aggregated. The results however were not stable and remained inferior to separate models.

## 2.3 Enhancement of translation-based methods

Lee and Lee [15] took a closer look at knowledge transfer between English and Chinese languages, using machine translation. They focused on situation where sentence-level machine translation is unachievable and worked on a training method that builds upon word-level translation, projecting the sentences from the two languages to a common space. This they compared to traditional sentence-level machine translation. As the dataset, SQuAD and NewsQA were used, in combination with a pre-existing Chinese dataset of 27 000 question-answer pairs. This research also concentrated on answer extraction from an already given document.

The research showed that best results were acquired via combination of all available techniques – using sentence-level machine-translation in combination with the new training method that used word-level translation. As training data, the combination of translated SQuAD and original Chinese data yielded the best results. Highest F1 score achieved was 87.26%.

These results also suggest that combination of different techniques is a promising training strategy.

## 2.4 Effects of small-scale fine-tuning on zero-shot strategy

Lauscher et al [12] studied ways to improve the performance of zero-shot strategy for different NLP tasks. They discovered that a surprisingly small number of extra training-samples in target language can significantly boost the performance. They made a separate fine-tuning round on a zero-shot QA model with questions and answers from 2-10 articles. The F1 score of mBERT model increased by 2.5-4.57 percentage points. Results for XLM-RoBERTa were more modest, increasing by up to 2.1 percentage points.

The authors pointed out that this is a very cost-effective way of improving the performance.

## 2.5 Cross-domain transfer learning

Transfer learning in QA is also used in cross-domain, not only cross-language situations. For example, Kratzwald and Feuerriegel [2] worked on building a QA system for a

domain-specific field – financial news. They tackled the task in two layers – first, the selection of relevant documents, followed by the extraction of specific answer.

Their dataset consisted of less than 400 domain-specific samples which they merged with SQuAD. Since their data amounted to only 0.6% of the resulting dataset, they used a technique called fuse-and-oversample to make the specific knowledge more relevant. They oversampled the specific data with ratio of 1:3. On the other hand, the authors did not try to use the two datasets consecutively (first the open-domain dataset, then the specific domain dataset).

The results of Kratzwald and Feuerriegel, using the document selection step before the extraction, showed performance increase as result of fuse-and-oversample. They reached the highest F1 score of 63.7%.

## 2.6 Transfer learning for NLP tasks in Estonian

For Estonian language, there is no knowledge of a specific QA system. However, cross-lingual transfer learning has been used for other Estonian NLP tasks [16] by partly the same the group of scientists who later published language model EstBERT [6]. This research used cross-lingual learning on following tasks:

1. Universal part-of-speech (UPOS)
2. Language-specific part-of-speech (XPOS)
3. Morphological tagging
4. Rubric classification
5. Sentiment classification
6. Named entity recognition

These downstream tasks were each executed on four different multilingual language models: mBERT, DistilmBERT, XLM-100 and XLM-RoBERTa. Across all the tasks, XLM-RoBERTa performed the best.

Later when EstBERT was published, the authors conducted measurements on identical tasks across three models: EstBERT, mBERT and XLM-RoBERTa. It turned out that although EstBERT proved its superiority in 5 tasks out of 6, then XLM-RoBERTa outperformed it on the 6$^{th}$, and was close behind also on the other tasks. Multilingual

BERT constantly held the last position. The authors, who had developed EstBERT on the example of BERT, concluded that it makes sense to also train a RoBERTa model for Estonian.

# 3 Original dataset for Question Answering in Estonian

Question Answering task in natural language processing can take different forms with different real-life applications. The tasks where the answer is not generated by the system, but selected from pre-existing texts, are called selection-based QA. Jurczyk et al [17] divide this further to three subcategories:

1. **Answer extraction** – selecting the suitable answer phrase from given context.
2. **Answer selection** – selecting whole sentence that contains the answer from given context.
3. **Answer triggering** – also selecting the whole sentence, whereas answer context may or may not be present in given document.

For this master's thesis I concentrate on answer extraction. Compared to answer selection, it requires the system to be capable of more precise reading comprehension. Answer triggering on the other hand adds a next layer of complexity – decision if answer can be found, and this is not in the scope of current thesis.

To train a system on answer extraction, the fine-tuning of the model and testing its performance must take place on a dataset that is composed for this category of QA. All answers must be annotated as continuous phrases in pre-given context.

Attempts to find some data source in Estonian that would be suitable for this task with no or little adjustments, failed. Whereas Liu et al [10] used the section "Did you know?" from national Wikipedias for similar task, it did not work for Estonian. Instead of factual questions from editors with links to answers, the Estonian Wikipedia has in this section a list of one-sentence descriptions of historical events on nearby dates. The topic is thus limited and there are no naturally phrased questions available.

In the lack of pre-existing data, it was decided to compose and annotate an original dataset. This had to be similar to any large English dataset that would be also included in training for all the transfer learning scenarios. This way they can be more easily used on consecutive training and one can be used in testing results from training the other.

## 3.1 Choice of training dataset in high-resource language

The most popular English extractive QA dataset is currently SQuAD with ca 100 000 question-answer pairs ([7] and [8]). It stands out among other similar, Wikipedia-based datasets with its size. Some other Wikipedia datasets contain only 1000-15 000 questions [17]. Also, compared to for example WikiQA, it has better question quality. WikiQA relies on search engine queries which are often worded differently from natural language, whereas SQuAD is crowdsourced.

There are also large QA datasets available that use other source data than Wikipedia, e.g., CNN/Daily Mail [18] corpus or NewsQA [9] which also relies on CNN. For those, some are also not using crowd-sourced questions. In CNN/Daily Mail corpus, the questions are generated synthetically, and answers are very short. NewsQA used crowdsourcing to produce its 120 000 questions. But authors also admit that the answer extraction models trained on their dataset perform worse than those trained on SQuAD.

SQuAD that relies on Wikipedia also serves as good example for composing original Estonian dataset. There is sufficient quantity of Estonian Wikipedia articles which can easily be retrieved through Wikipedia's storage of data dumps [19]. Given the focus of current thesis, it is reasonable to use the first version of SQuAD which consists of 100 000 question-answer pairs based on ca 536 Wikipedia articles. The second version of SQuAD added questions which were unanswerable based on given context and this form of task is out of current scope.

SQuAD v1.1 consists of triples with following structure (see example in Table 3):

- Context – a paragraph from a Wikipedia article.
- Question – any question about the paragraph, proposed by a real person in a freely phrased manner.
- Answer – a continuous span from the context, along with the start index.

Table 3. Sample of SQuAD v1.1 training set – a paragraph of Wikipedia as context, along with a question and an answer. For the answer, both the content and start index are supplied.

| | | |
|---|---|---|
| **Context** | The Rev. Theodore Hesburgh, C.S.C., (**1917–2015**) served as president for 35 years (1952–87) of dramatic transformations. In that time the annual operating budget rose by a factor of 18 from $9.7 million to $176.6 million, and the endowment by a factor of 40 from $9 million to $350 million, and research funding by a factor of 20 from $735,000 to $15 million. Enrollment nearly doubled from 4,979 to 9,600, faculty more than doubled 389 to 950, and degrees awarded annually doubled from 1,212 to 2,500. | |
| **Question** | What was the lifespan of Theodore Hesburgh? | |
| **Answer** | Text | 1917–2015 |
| | Answer start index | 37 |

## 3.2 Composing Estonian Question Answering dataset EstQA

The goal with original Estonian dataset was to produce enough data to use it both in testing and as an extra fine-tuning step in training. Target was 1000 questions with the answers. The target was based on examples from related work. E.g., the cross-domain QA research that was described in 2.5, had only 400 samples from target domain. In the XQA dataset referred to in 2.1, the size of target language datasets began from 350. So, a dataset with 1000 samples can still be divided to training and test sets, while remaining in bounds that have proved to be sufficient in related research.

Another goal was to ensure compatibility between the new dataset and SQuAD, which was chosen as main training dataset. Therefore, the process used in producing SQuAD [7] was taken as example, along with French dataset PIAF [4] which also followed SQuAD to produce a compatible Wikipedia-based dataset in another language.

### 3.2.1 The choice of Wikipedia articles

As for both example datasets, SQuAD and PIAF, a large quantity of most relevant Wikipedia articles was retrieved in the target language, Estonian. The relevance was determined in the terms of Wikipedia's internal page rank, the ranking of the article in the web of cross-references and links between articles. As in both examples, Project Nayuki's PageRank algorithm [20] was used. The highest-ranking articles are presented in Table 4.

Table 4. The highest-ranking Wikipedia articles in Estonian.

| | **Article title** |
|---|---|
| 1. | "Eesti" |
| 2. | "Ameerika Ühendriigid" |
| 3. | "Tallinn" |
| 4. | "Venemaa" |
| 5. | "Saksamaa" |
| 6. | "Ladina keel" |
| 7. | "Inglise keel" |
| 8. | "Keeletoimetamine" |
| 9. | "Tartu" |
| 10. | "Taksonoomia" |

As was experienced by authors of the French PIAF, the Estonian high-ranking articles were often those that described events in a certain period, e.g., "1991", "18. sajand" (18th century), "17. jaanuar" (January 17), etc. Those articles were essentially bullet points, listing historical events, not containing coherent textual paragraphs. Following the French example, I excluded those articles. Out of 10 000 initial articles, 8779 remained.

Next, too short paragraphs and too short articles were filtered out. As for PIAF, I chose articles that contained at least 5 paragraphs of sufficient length (discarding all shorter paragraphs). Two possible length limits were experimented with – 500 characters as for SQuAD and PIAF, and 400, given the potentially smaller amount of Wikipedia data in Estonian. The results showed that whereas there were 1168 articles with enough 400-character articles, there were also 746 articles with 500-character articles. The total number of articles used for the whole SQuAD was 536, so higher character limit produced more than sufficient number of articles.

Finally, a random choice was made among the 746 articles to produce a set of 20 articles (see Table 6) containing 226 paragraphs. The quantity was sufficient, because for my example datasets, the crowd-workers had to propose preferably five questions per paragraph[1]. Following the same example would fill the target of 1000 questions.

[1] Final average number of questions per paragraph was 4.3 in SQuAD and 5.0 in PIAF.

### 3.2.2 Question-answer collection

Both SQuAD and PIAF used crowdsourcing to produce the questions and annotate the answers in their articles. The process used in current thesis followed the setup and recommendations brought out by both sets of authors, but instead of crowdsourcing, the annotation process was a one-person effort.

For producing a dataset that is structurally identical to SQuAD, cdQA-annotator tool was used [21]. Screenshot of the annotation in the tool is available in Figure 1.

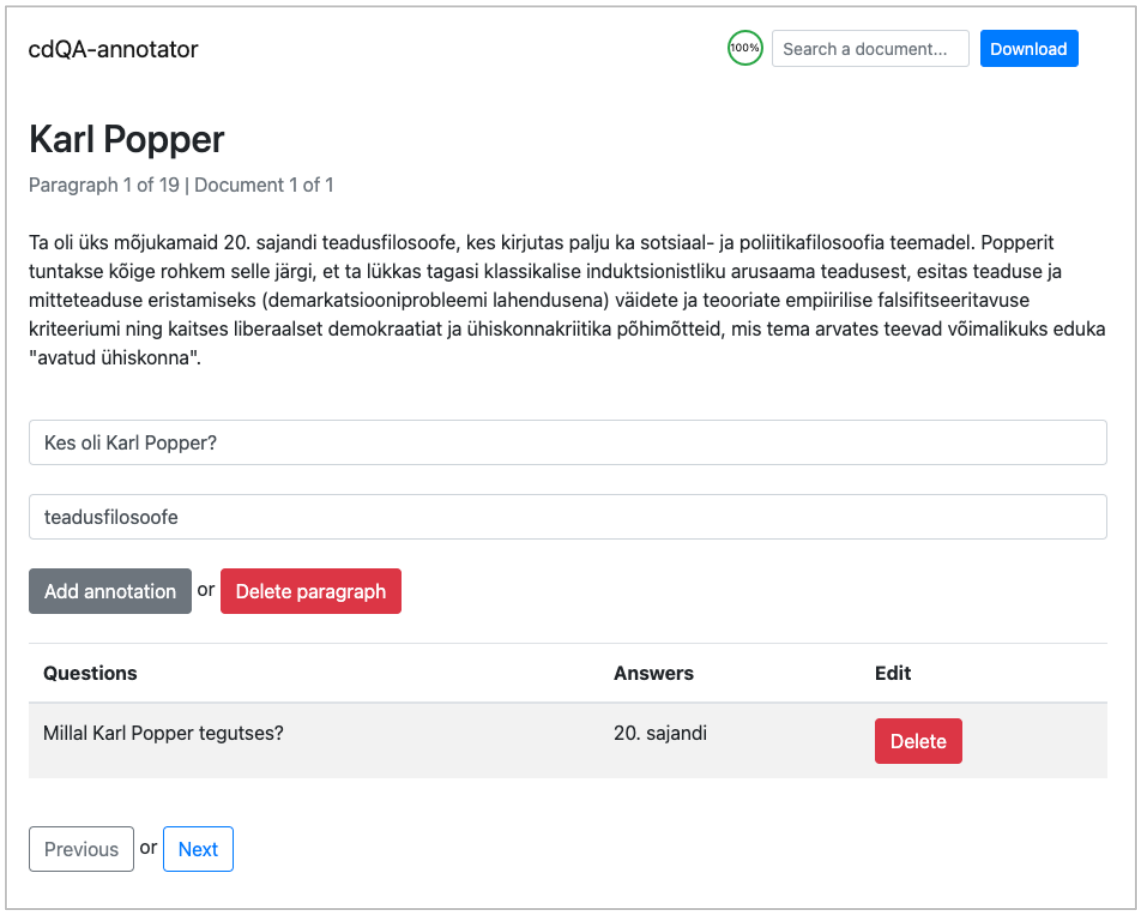



Figure 1. Screenshot of the cdQA-annotator tool that was used for creating the questions and answers. The question was freely written, the answer was selected via highlighting text with cursor.

The rules for proposing the questions followed those of SQuAD:

1. Goal was to ask five questions per each paragraph. It was possible in vast majority of the cases, excluding altogether three paragraphs where the text was incoherent, and no questions were possible to ask.

2. The questions were phrased in natural language.

3. Goal was to pose “difficult questions”, using different words in the question than in the answer span, also, if possible, trying to pose questions where answer can be deduced by using information from multiple sentences.

For the answer, the shortest span containing the answer to the question was annotated. One question could also have multiple answers. This counts for cases where different spans can be considered as adequate answer to the question. E.g., answer to a question about occurrence of an event can be equally well “in the 19th century” or “19th century”. In the training phase only one answer per question can be used at a time, thus asking the same question multiple times. But in the test phase, both F1 and EM metrics (see chapter 5) are well usable for multiple ground truth answers in parallel, choosing the best result. An example of a question with multiple options for answer is given in Appendix 2.

An example of different questions in the new dataset are visible in Table 5. First question can be considered more difficult as it requires information from across the whole paragraph to identify the answer. Second question is easier, relying on information only from a single sentence, but it is still using different wording in the question than in the answer.

Table 5. Example of questions with different difficulty level in the new dataset.

| | **Estonian version** | **Translation to English** |
|---|---|---|
| **Context** | USA algatas mitmed Kuuga seotud kosmoseprogrammid, millest kuulsaim on Apollo programm. JPL-i Rangeri programm tegi Kuu pinnast esimesed lähifotod, Lunar Orbiteri programm kaardistas Kuu pinna ja Surveyori programmi raames maandusid Kuul esimesed USA kosmoseaparaadid. Mehitatud lennud Apollo programmi raames said võimalikuks peamiselt arvutite, tarkvara ja kuumuskilpide suure arengu tõttu 1960. aastatel. Lisaks oli programmi juhtkond väga kompetentne juhtima hiiglaslikku projekti. Programmi raames saadeti 1968. aastal Kuu orbiidile | The United States has launched several space programs related to the Moon, the most famous of which is the Apollo program. The JPL Ranger program took the first close-ups of the lunar surface, the Lunar Orbiter program mapped the lunar surface, and the first U.S. spacecraft landed on the moon as part of the Surveyor's program. Manned flights under the Apollo program became possible mainly due to the great development of computers, software and hotplates in the 1960s. In addition, the program management was very competent to manage a giant project. The program sent |

| | Estonian version | Translation to English |
|---|---|---|
| | esimene mehitatud missioon, Apollo 8, ja 1969. aastal toimunud mehitatud maandumist Kuule peavad paljud kosmosevõidujooksu kulminatsiooniks. | the first manned mission to the Moon's orbit in 1968, Apollo 8, and many consider the manned landing on the Moon in 1969 to be the culmination of the space race. |
| **Question 1** | Mis riik saatis 1960. aastatel inimesed Kuule? | Which country sent people to the moon in the 1960s? |
| **Answer 1** | USA | The United States |
| **Question 2** | Millal jõudsid esimesed kosmonaudid Kuu orbiidile? | When did the first astronauts reach lunar orbit? |
| **Answer 2** | 1968. aastal | in 1968 |

Altogether, dataset containing 1115 questions and 1668 answers was composed. This gives average answers-per-question ratio of 1.5, which is similar to that of SQuAD development set where it is 1.7. The time spent to compose the dataset was approximately ten full working days.

The dataset was further divided into two separate parts, one to be included in training cycle, the other solely for testing. Articles represented in the sets are listed in Table 6:

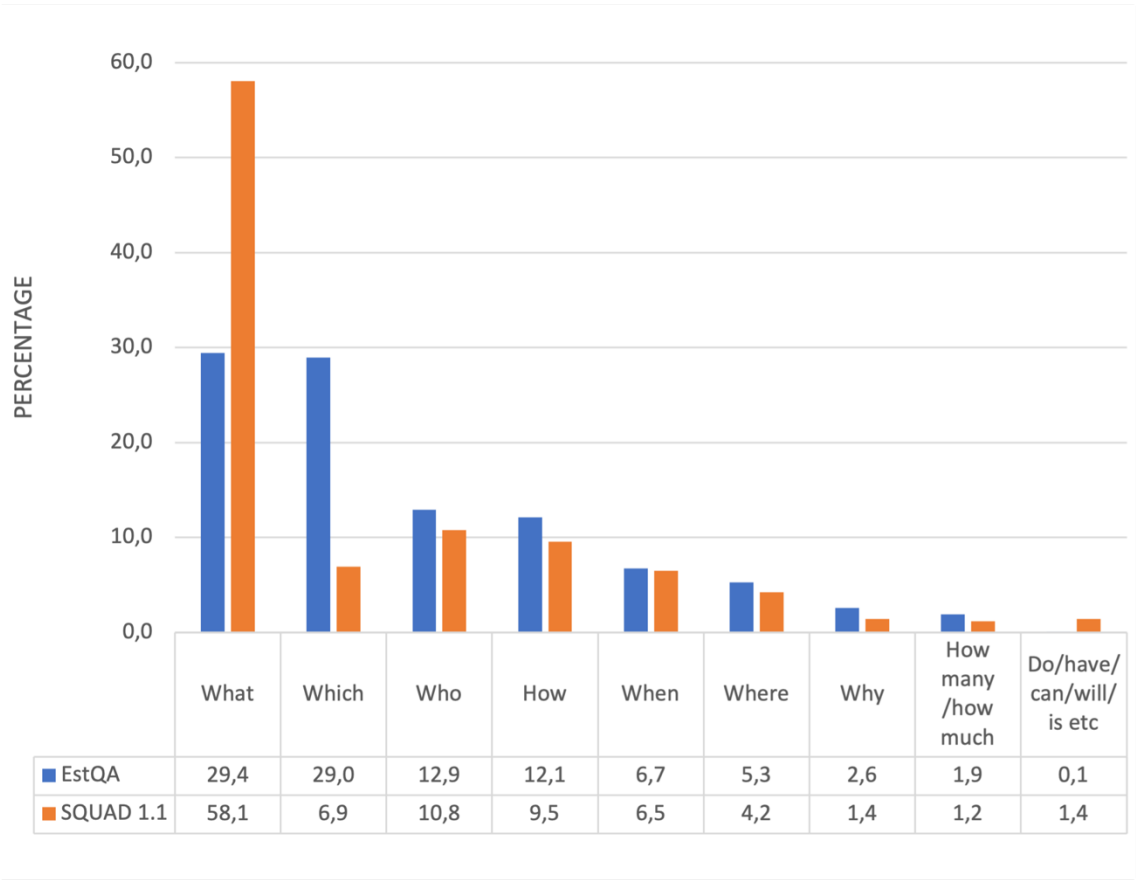
Table 6. Articles in the EstQA training set and test set.

| Articles in the training set | Articles in the test set |
|---|---|
| "Ajateenistus" | "Charles Sanders Peirce" |
| "Apollon" | "Eestimaa kubermang" |
| "Burundi" | "Kaitseseisukord" |
| "Estonia puiestee" | "Karl Popper" |
| "Gröönimaa" | "Kõrgem Kunstikool Pallas (1919-1940)" |
| "Johannese evangeelium" | "Kuu" |
| "Metsandus" | "Liivimaa ordu" |
| "Novgorodi vabariik" | "Loomaaed" |
| "Sardiinia kuningriik" | "Saaremaa" |
| "Veenus" | "Tuberkuloos" |

The number of units (context-question-answer triplets) was 776 in the training data, all possible answers separated. The test data consisted of 603 units, with several ground truth answers possible. Altogether the test set included 892 answers.

### 3.2.3 Question type comparison with SQuAD

To compare the questions in the new EstQA dataset and SQuAD, I categorized them by the interrogative word used in the question. Results in Figure 2 show that the datasets share clear similarities.




| | What | Which | Who | How | When | Where | Why | How many /how much | Do/have/ can/will/ is etc |
|---|---|---|---|---|---|---|---|---|---|
| EstQA | 29,4 | 29,0 | 12,9 | 12,1 | 6,7 | 5,3 | 2,6 | 1,9 | 0,1 |
| SQUAD 1.1 | 58,1 | 6,9 | 10,8 | 9,5 | 6,5 | 4,2 | 1,4 | 1,2 | 1,4 |

Figure 2. Question categories in EstQA and SQuAD v1.1 datasets.

The majority of questions are of type *What/Which*. In Estonian both are equally represented, in English SQuAD there is significantly more *What* questions, but both interrogatives are semantically similar. The questions from type *Who*, *How*, *When*, *Where* and *Why* follow in the same order in both datasets.

Noticeable difference is visible in questions of type *Do/Does/Is/Can/Will/Would* etc, which in Estonian is covered by the interrogative *Kas*. In my dataset there is only a single instance of this type, as I considered it a questionable question type – in a natural context it is usually an easy question, but it implies a yes/no answer which is currently not possible, as the answer must be a span of the context. On the other hand, the span to prove either "yes" or "no" can be relatively long. In SQuAD this question type amounts to 1.4%

of the questions which reflects how the crowd-workers, with their own internal preferences, tended to pose the questions.

Lack of crowdsourcing is the biggest difference about the new dataset, compared to SQuAD and PIAF. Also, the authors of MLQA dataset referenced in 2.2 used crowdsourcing for annotating their data. Proposing questions and defining answer spans is not exact science and is open to subjectivity. Using multiple contributors who often also cross-validate each other's work, makes the results more universal. This means that when the QA system is put to practice, then training on crowd-sourced and cross-validated data is more probable to correspond to how any random user interacts with the system. Also, multiple contributors may help to diversify the data.

On the other hand, the annotation process was currently under control of a single person. This helped to guarantee the integrity of the data – from the papers of SQuAD and PIAF it is visible that some annotated answers were in fact not answering the posed question. Also, currently there were no cases of technical failure such as experienced by PIAF's team, where incomplete words were marked as answers. This benefit, however, can be also achieved in a multi-worker environment. It requires putting more effort into the validation of the process and the results. Also, even in the controlled one-person environment, there were two cases of wrongly annotated answers spans that were not discovered before empirical work on developing the QA models, and were included in train and test sets.

The same topic becomes relevant again in the analysis of the results of the QA models in 5.6. It will be visible that on hindsight, some answers predicted by the system and marked as incorrect could have been considered ground truth by a different annotator. Therefore, crowdsourcing is one of the paths to go if this dataset is to be enhanced.

### 3.2.4 Answer type comparison with SQuAD

The analysis of the diversity of the answers in the dataset was conducted identically to SQuAD v1.1. The authors of SQuAD distinguished two kinds of numerical answers (dates and other numerical values), three kinds of answers that consisted of an entity (people, locations and other), common nouns, and finally, different other parts of a sentence – adjectives, verb phrases, clauses, and other. Examples of the types of answers encountered in both datasets are visible in Table 7.

Table 7. Categorisation of answers in EstQA and SQuAD v1.1 datasets.

| Category | Example from SQuAD | Example from EstQA | Example from EstQA (translated to English) |
|---|---|---|---|
| **Date** | 19 October 1952 | 31. märtsil 1933 | March 31, 1933 |
| **Other numeric** | 10.9% | 23-kraadise | 23 degrees |
| **Person** | Thomas Coke | Johannes Vares | Johannes Vares |
| **Location** | Germany | Austria | Austria |
| **Other entity** | ABC Sports | Kristlik-Demokraatlik Partei | Christian Democratic Party |
| **Common noun phrase** | property damage | rahutused | unrest |
| **Adjective phrase** | second-largest | viletsa | lousy |
| **Verb phrase** | returned to Earth | osa hoonest hävis | part of the building was destroyed |
| **Clause** | to avoid trivialization | et õpinguteks raha teenida | to earn money for studies |
| **Other** | quietly | Jh 14:16, Jh 14:26, Jh 15:26, Jh 16:79 | Jh 14:16, Jh 14:26, Jh 15:26, Jh 16:79 |

Comparative analysis in Figure 3 shows that results for the new dataset and SQuAD are well comparable. Most notable exception is in the category of locations where Estonian dataset has three times more occurrences. On the other hand, the categories of people and other entities have higher representation in SQuAD.

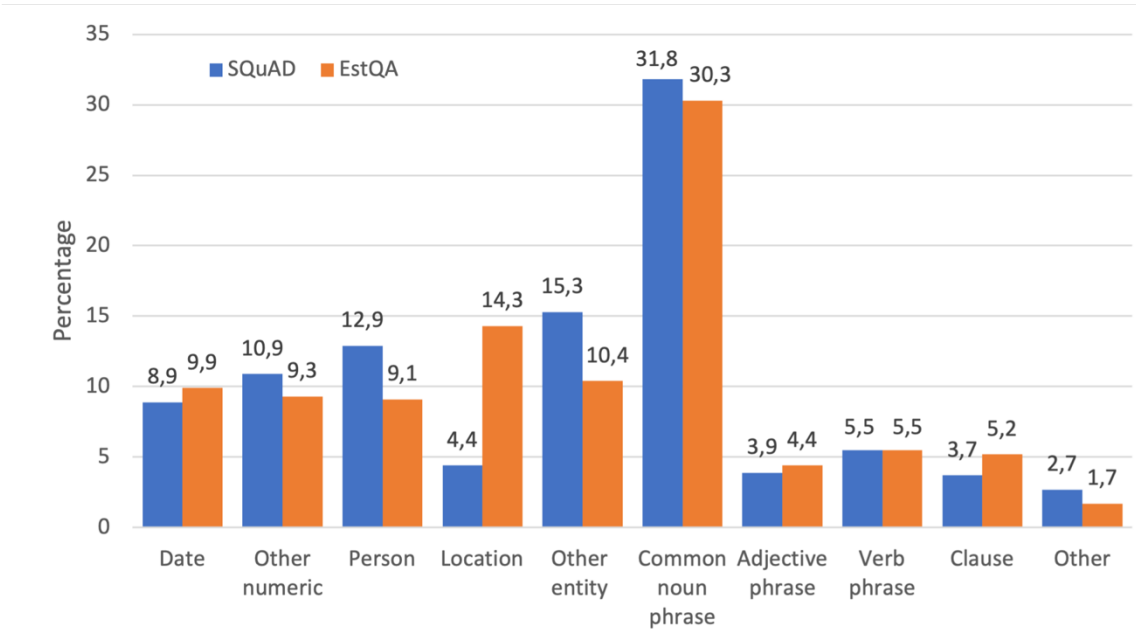



Figure 3. Detailed distribution of answer categories in EstQA and SQuAD v1.1 datasets.

When looking at the data in the broader categories mentioned above (numerical/entity/other), the similarities between the two datasets are nearly a complete match as is visible on Figure 4.

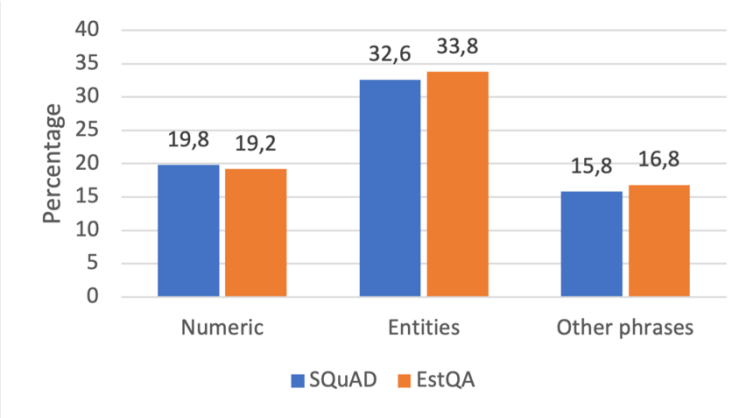



Figure 4. Broad distribution of answer categories in EstQA and SQuAD v1.1 datasets.

In conclusion, as for the questions, so for the answers, the diversity and categorization in the two datasets are very similar. This means that they are a good match to be used together in the training and testing of a Question Answering model.

# 4 Experiments

## 4.1 Choice of pretrained language models

The specific Question Answering datasets will be used to fine-tune a large contextual pre-trained language model. Three language models are used in comparison. This chapter introduces the models and explains the characteristics why they were chosen.

### 4.1.1 Multilingual BERT

BERT [3] was in 2019 one of the first pretrained language models using the Transformers architecture that was introduced in 2017 [22]. The Transformers architecture was a significant improvement in sequence-to-sequence NLP models, replacing sequential recurrent neural networks or convolutional networks with only attention mechanisms, including self-attention. This meant that the model could represent each word in the input text in the context of the full input. So, it could handle well the long-range dependencies in the texts.

BERT (Bidirectional Encoder Representations from Transformers) differed from many previous models which handled the texts in a unidirectional manner. Unidirectional means that at a time, text is handled either from left to right or vice versa. BERT was conditioning simultaneously both on the left and right context during the training. This improved the quality of language modelling. BERT was also trained for capturing relationship between consecutive sentences, as one of the tasks that was used in its training was next sentence prediction. The other and main task was masked token prediction – some tokens in the input were randomly masked and the goal was to predict these. The technique is labelled as Masked Language Modelling (MLM).

BERT is ready to be fine-tuned on a wide variety of downstream NLP tasks, without a need for task-specific architecture. All inputs that consist of a single text sequence or two (such as question and context in QA) can be simply used as input for the pretrained model. Model is initialized with the pre-trained parameters and those are fine-tuned based on the data from the downstream tasks. As the authors put it: “For each task, we simply plug in the task-specific inputs and outputs into BERT and fine-tune all the parameters end-to-end.”

When Google released BERT in 2019, it was tested for Question Answering among other NLP tasks. It climbed to the top of the SQuAD leader board both for v1.1 and v2.0 with a significant margin. The F1 score for version 1.1 was 93.2% and for version 2.0 it was 83.1% [3].

The multilingual version of BERT (mBERT) was trained in parallel with monolingual Wikipedia corpora in 104 languages. In essence, it was trained as a model for single language, where this only "language" contained all the 104 natural languages, as stated by Pires et al [23]. They noted that mBERT is well capable of transferring the knowledge that it gained from fine-tuning the model on one language to another language. E.g., mBERT that was fine-tuned for part-of-speech task in English, got 87% accuracy in Bulgarian.

Many researchers, e.g., [23], [24] and [12] have investigated the cross-lingual performance of mBERT. They have noted that cross-lingual transfer is the best when the two languages share linguistic and structural similarities, notably the word order. In this regard, both English and Estonian languages have the subject-verb-object type of word order, as opposed to the type of subject-object-verb (e.g., Japanese, Korean, partly German).

The multilingual BERT is probably the most often used multilingual model for cross-lingual zero-shot QA tasks. It was also used for evaluating different NLP tasks in Estonian (referenced in 2.6). Also, it was the example by which the Estonian model EstBERT was trained.

### 4.1.2 XLM RoBERTa

XLM RoBERTa [5] is the multilingual version of RoBERTa language model [25], developed by Facebook AI. RoBERTa itself, short for Robustly Optimized BERT Pretraining Approach, improved on the training method of BERT, paying more attention to different hyperparameters and abandoning the next sentence prediction objective. Authors also found BERT to be significantly undertrained and improved on that. RoBERTa was also tested on SQuAD by its authors. It outperformed BERT by achieving F1 score of 94.6 on SQuAD v1.1 and 89.4 on v2.0.

The multilingual XLM RoBERTa was trained in close comparison with the architecture and process of mBERT and another multilingual language model XLM [26]. The authors used CommonCrawl training data which was significantly larger and more diverse than Wikipedia dumps. Among other differences they also avoided using language-embeddings to gain better code-switching in cross-lingual tasks.

The resulting multilingual model outperformed mBERT and XLM (see 4.1.4). For question-answering, they used results from Lewis et al [11] as benchmark for seven languages. Average achieved F1 score was 70.7%, compared to the benchmark of 61.6%. The model also displayed strong monolingual capacities, outperforming English-language BERT on the English language.

XLM-RoBERTa is often used in comparison with mBERT, as was the case for many NLP tasks in Estonian, in the recent works [16] and [6].

In this thesis, the smaller, base version of XLM-RoBERTa is used to be more comparable with mBERT. Both models use 12 layers, number of hidden states in the models is 768 and 12 attention heads are used.

### 4.1.3 EstBERT

EstBERT [6], the only Estonian large contextual language model, has been trained based on the Estonian National Corpus 2017 which includes several sub corpora, including data from Wikipedia, other web sources (blogs, news etc), dissertations, and fiction.

As the authors of EstBERT state then based on existing studies, a language specific BERT model is expected to outperform multilingual ones. The results confirmed that for EstBERT. However, the tasks that were compared, did not include Question Answering. So, in this thesis we gain new knowledge about EstBERT's performance in QA, also in comparison with multilingual models.

### 4.1.4 Discarded options – XLM, DistilmBERT

Not all existing multilingual language models were included in my comparison, to contain and manage the scope of the task.

XLM multilingual language model [26], trained on Wikipedia, was discarded from my choice of underlying models because of its performance, as reported for example in [5]

and [16]. It is steadily outperformed by XLM-RoBERTa. On the other hand it outperforms mBERT, so it is not expected to extend the result boundaries.

Romano [27] compares mBERT, XLM and XLM-RoBERTa for multilingual tasks, and suggests using the latter as the model with best performance.

There exists also the distilled, smaller version of BERT, DistilBERT [28], which in turn has a multilingual version called DistilmBERT. It was included in language-transfer research for Estonian language [16] where it performed the worst. Thus, it does not make sense to include a distilled version of mBERT but only the full-scale one.

Unfortunately, there is no knowledge of a multilingual non-MLM language model. For example, there is the well-performing non-MLM English model XLNet [29], the authors of which label MLM as "corrupting the input with masks". The model outperformed both BERT and RoBERTa on English SQuAD. It would be interesting to compare the multilingual versions of them all for current Question Answering task.

## 4.2 Model architectures

The choice of strategies for the QA models took example from the related work in the field, using them both in isolation and combining them. Overview of different models is visible in Table 8 and further explained below.

Table 8. Models used in the training.

| **Strategy** | **EstBERT** | **mBERT** | **XLM-RoBERTa** |
|---|---|---|---|
| Only EstQA | ☑ | ☑ | ☑ |
| Translate-train | ☑ | ☑ | ☑ |
| Translate-train + EstQA | ☑ | ☑ | ☑ |
| Translate-test | | ☑ | ☑ |
| Translate-test + EstQA | | ☑ | ☑ |
| Zero-shot | | ☑ | ☑ |
| Zero-shot + EstQA | | ☑ | ☑ |
| Mix1 (sequential, see 4.2.6) | | ☑ | ☑ |
| Mix2 (combined, see 4.2.7) | | ☑ | ☑ |

The strategies where all or part of the training data was in English, could only be used to fine-tune multilingual language models, as EstBERT models only the Estonian language. That is the reason why some strategies could not be used with EstBERT.

### 4.2.1 EstQA

In the cases where EstQA dataset was used in training together with a bigger dataset, e.g., "Translate-train + EstQA", it was used there as a second round of fine-tuning. For example, model was trained for *n* iterations on SQuAD, and then further fine-tuned for *n* iterations with the small Estonian dataset. In the experimental phase of the research, this approach often proved to increase the performance. This however did not happen, if the two datasets were fused together. In case of fusing, the EstQA was seriously under-represented. In one of the related works [2], such situation was solved by oversampling the underrepresented data. I chose a different path, so here is an opportunity for further experiments.

Independent of the decision if EstQA was used in training or not, the test set was always the same for all the models. The allocation of data to training and test sets is described in 3.2.1 and Table 6.

For the strategy "Only EstQA", only the Estonian training set with its less than 800 units was used for training. Compared to the size of SQuAD training data with ca 88 000 units, it was a very small amount.

### 4.2.2 Translate-train

With translate-train approach, the training set was the SQuAD training data, translated into Estonian via Google machine translation. The answers were aligned from English context to Estonian context via tool called Awesome-Align[1] [30] which uses word embeddings from multilingual BERT. Using embedding alignments is an acclaimed method in the literature (see for example [13], also efforts by [14] to develop such a system by a separate machine learning module). It helps to overcome the ubiquitous problem that answer is translated differently in isolation than in the context. So, if we take

[1] https://github.com/neulab/awesome-align

the separately translated answer span and hope to find it in the translated context, to identify the start index, it is most often not there. The case of words has changed, a synonym is used, etc. Using alignments helps to see which word in the original context is aligned to which word in translated context. The words corresponding to the original answer are thus found.

In case answer could not be mapped back with the help of word alignments, or the mapping was not a continuous span, two other, more simple strategies were used as fallback.

First, a naïve alignment method where answer was separately translated. The translated answer text was searched for in the translated context and if it was found, then answer was mapped to this text and this start index. To handle possible multiple occurrences of the answer phrase, the number of occurrences was checked both in English and Estonian version and if it matched, the same occurrence was picked.

Second fallback was used via tool called fuzzysearch[1]. Here also, the answer was translated separately and its occurrence in the translated text was searched for with Levenshtein distance, which allowed for changes in the case, association-dissociation of words, different versions of words with the same stem, etc.

Altogether, out of 87599 training units in SQuAD v1.1, answer was successfully matched for 84400 units. Out of those, 71407 were matched with word embedding alignments and others with simple matching.

### 4.2.3 Translate-test

For translate-test method, the SQuAD v1.1 training set remained in English language. Estonian test set was also translated into English. In case EstQA was used for extra fine-tuning, then this was translated to English as well. For mapping the answer to the context, identical methods were used as for translate-train.

Out of extra fine-tuning set, 674 units out of 776 were successfully matched. For the test set, 580 units out of 603 were matched.

[1] https://pypi.org/project/fuzzysearch/

Translate-test scenario can also be used on top of a monolingual, in current case, English language model. This can be further experimented with.

### 4.2.4 Quality of machine translation

More than 80 000 translated paragraphs, questions and answers are not feasible to be manually checked for quality, at least in a one-person setup. Random validation was still used which showed that majority of translations were adequate, the meaning was not lost, and wording was understandable.

However, mistranslations occurred (see Table 9). It happened the least inside the context-part which had more info to interpret the meaning. Questions had less context and answers the least.

Table 9. Examples of wrong translations by Google Translate.

| **SAMPLE 1 – error in answer** | |
|---|---|
| Question from SQuAD | What was Napoleon Bonaparte's nationality? |
| Answer | French |
| Translated answer | prantsuse keel (*Instead of just "French" the translation interpreted it as "French language".)* |
| **SAMPLE 2 – error in question** | |
| Question from SQuAD | In what year were census respondents first able to select more than one race? |
| Translated question | Kui paljud ameeriklased teatasid 2000. aasta rahvaloendusel olevatest rohkem kui ühest võistlusest? (*Wrong translation of „race" was chosen – „contest", instead of ethnic concept. Also, singular/plural is misused.*) |

Looking at the second sample, we can see a weird mismatch between plural and singular in the word "olevatest" and looking strictly at the grammar, it could be argued that the sentence has no meaning. This kind of mismatches in the case of the word, singular/plural, etc, occurred consistently, less in the context paragraph, more in the question – thus depending on the amount of context available.

I made no fixes in the translations. One reason was to use the machine translation as such, without manual interference. Secondly, most of the errors occurred in the answers and vast majority of translated answers were never used. Instead, most of the answers could

successfully be mapped back to translated context via word embeddings. Also, if there was the need to use translated answers – then if the translation was inaccurate, it would not be matched back via naïve exact match or via Levenshtein distance either. Any match to the wrong phrase would be coincidental and unlikely to occur.

#### 4.2.5 Zero-shot

For this strategy, the multilingual models were trained with English SQuAD and tested on Estonian dataset.

In case Estonian data was also used for fine-tuning, then this remained in Estonian. This method could also be categorized as "few-shot". The technique was acclaimed as highly helpful in the referenced article by Lauscher et al [12].

#### 4.2.6 Mix 1 – sequential combination

Both Mix 1 and Mix 2 aim to combine the benefits that the models achieved from training either on one language or the other.

In Mix 1, the models were trained on different datasets sequentially, directed from most general to most task-specific dataset. First, the underlying language model was trained on the English SQuAD, then further fine-tuned on SQuAD translated into Estonian, and finally fine-tuned even more with the small EstQA training set.

#### 4.2.7 Mix 2 – fusing datasets

Differently from Mix 1, the English and Estonian SQuAD datasets were combined and shuffled, and model was trained on the resulting big dataset of 172 000 units. Subsequently, it was also fine-tuned on the Estonian dataset.

## 4.3 Training details

The models were trained using the HuggingFace NLP library example code for QA[1]. The code was edited and refactored from a tutorial to a concentrated script and result is available in Google Colab environment where all the training was conducted[2].

Most of the models were trained for 3 epochs. Experiments in the early phase of research showed that for both multilingual models, the performance started to deteriorate after the 3rd epoch. Some models were still trained longer – in case only the small Estonian dataset was used, all models were trained for 15 epochs. In this scenario, performance started to decrease only after that, and due to small size of data, the training times were feasible.

All models were tested with default hyperparameters in the HuggingFace example. Some experiments were conducted with learning rate, but they showed no significant impact. Thus, focus was on comparing the different architectures under the same circumstances, not further investigation into tuning the hyperparameters.

The hyperparameters used were learning rate of 2e-5 and weight decay of 0.01. Adam optimizer was used. Batch size was reduced from 16 to 8 compared to the original HuggingFace example, to reduce memory consumption.

---

[1] https://colab.research.google.com/github/huggingface/notebooks/blob/master/examples/question_answering.ipynb

[2] https://colab.research.google.com/drive/1fVDPnJkMGhfAppEx2sBhN1u0nQqod0dc?usp=sharing (short URL: https://tinyurl.com/rps5sfsz)

# 5 Results

This chapter gives overview of the performance of the different Question Answering models that were implemented. One result of the current thesis is also the EstQA dataset that was described and analysed in chapter 3. Here we concentrate on the performance.

We evaluate the results with two metrics that are common for the QA task evaluation:

1. **Exact Match** (EM) – the answer predicted by the system is compared to all answer variations marked as ground truth (golden answers). There can be several of those, as it is subjectively possible to answer the same question with different spans, including and excluding some parts of the sentence. In case of exact match with any golden answer, the score of the question will be 1, otherwise 0.
2. **F1 score** – the words in the predicted answer are compared to words in each ground truth answer. A score is calculated (see Equation 1) that combines:
   a) Precision - how many words in the predicted answer are "relevant", meaning that they are words that also exist in the golden answer.
   b) Recall – how many of the "relevant" words from the golden answer were represented in the predicted answer.

   The value of the F1 score ranges between 0 and 1. The best score across the golden truth answers is chosen as the score for this question.

$$\mathrm{F1} = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (1)$$

For both metrics, the score across the whole dataset is calculated as the arithmetic average of all questions and multiplied with 100 to convert it to percentage.

As baselines to compare with, the authors of SQuAD v1.1 presented results for random guess and human performance on their test dataset, visible in Table 10.

Table 10. Performance baselines on SQuAD v1.1.

| | **F1** | **EM** |
|---|---|---|
| **Random guess** | 4.3 | 1.3 |
| **Human performance** | 86.8 | 77.0 |

Up to nine different QA models were built for each underlying language model (see Figure 5). As some models can only be built upon multilingual models, those are not represented for EstBERT (see explanation in 4.2).

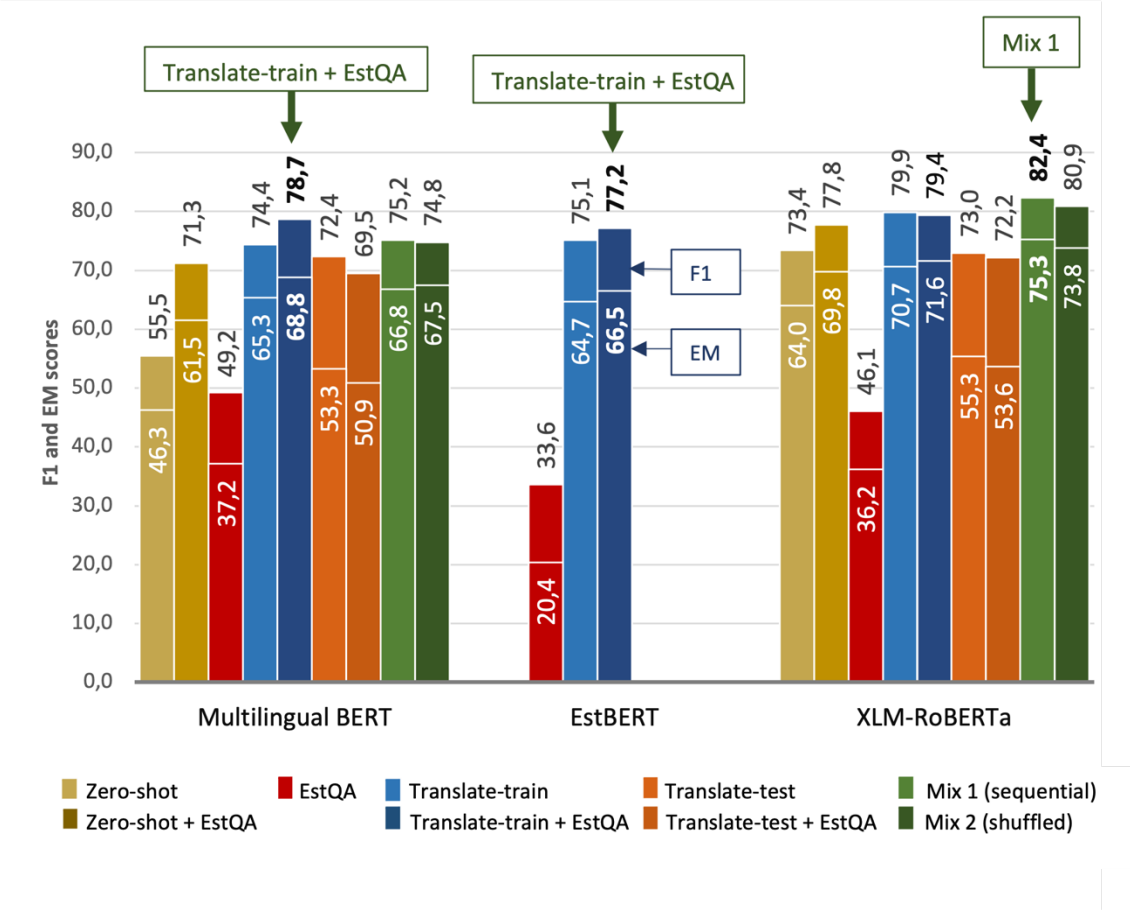



Figure 5. All results from Question Answering models.

All models were tested on the original EstQA test set. The results varied widely across the chosen architectures and underlying language models. I will first analyse the performance across the architectural choices, followed by analysis across underlying language models.

## 5.1 Using only EstQA dataset

As could be expected, the worst results were achieved when training only with the EstQA dataset that has less than 800 question-answer pairs. Given the amount of data, the results are even surprisingly good.

Multilingual BERT achieved F1 score close to 50% (with 37% of exact matches). While experimenting with smaller epoch numbers, it even reached 50.8%, but for this

comparison I used the same epoch count for all models. The chosen epoch count was 15, as the majority of the results peaked here. XLM-RoBERTa achieved F1 score of 46%, and EstBERT was further behind with the score of 33%.

Gaps in the performance are notable. Given the small quantity of training data, some randomness may theoretically be involved, but this is not the case. During experiments with different epoch counts, the performance of each model changed in consistent bounds, without any fluctuations.

## 5.2 Zero-shot

In the zero-shot scenario, the multilingual models were trained on English SQuAD training set and tested on Estonian test set. Alternatively, second round of fine-tuning was involved with the small Estonian training set.

Contrary to several research papers referenced in chapter 2, zero-shot strategy did not produce the best results. For multilingual BERT, it performed worse than either of the translation-based methods. For XLM-RoBERTa, it outperformed translate-test, but remained clearly behind translate-train. Inferiority to translate-train suggests that more task-specific, Estonian-language training data resulted in better performance. The reason why for multilingual BERT zero-shot strategy was bested also by translate-test, may be related to machine translation quality. This would imply that either Google Translate quality has improved in recent years or is better in Estonian than for other languages in related works. This hypothesis would need to be validated separately.

When comparing the results with and without additional fine-tuning round on Estonian data, then this extra round with less than 800 samples resulted in significant performance gain. This is consistent with the findings from Lauscher et al [12] that was referenced earlier.

The improvement was especially notable in the case of mBERT, where F1 score jumped from 55.5 to 71.3. It confirms that even a small number of samples that are relevant for the specific task can result in huge performance gain. The improvement was less significant but still noteworthy in the case of XLM-RoBERTa. The smaller effect of additional fine-tuning for XLM-RoBERTa compared to mBERT is also consistent with findings by Lauscher's team. The F1 score for XLM-RoBERTa improved from 73.4 to

77.8%. It still means that the number of incorrect answers decreased by ca 1/6 and confirms the importance of task-specific training even with a small amount of data.

## 5.3 Translation-based methods

In the translate-test scenario, the multilingual models were trained on English SQuAD. The Estonian datasets, both for extra fine-tuning and testing, were translated to English.

In the translate-train scenario, all underlying contextual language models could be used, because the training data was in Estonian. The machine-translated version of SQuAD was used. Extra fine-tuning (if used) and testing relied on the new EstQA dataset.

As was the case in the related works in chapter 2, translate-test performed worse than translate-train for all cases where both were used – that is, for the multilingual models.

To interpret this, one must consider that machine translation is not perfect (see 4.2.4). So, translate-train means that underlying language model, which presumably has pre-existing knowledge of "perfect", correct Estonian, is shown machine-translated data. This includes many examples in correct Estonian and many in not-perfect Estonian. Finally, it is asked to perform the test in correct Estonian. So, it has during the training phase seen many examples of the same type of language which it is tested against.

For the translate-test scenario, the language model has pre-existing knowledge about correct Estonian and correct English. Then it is introduced many samples in correct English and, if extra fine-tuning occurs, just a few in not-perfect English. Then it is asked to perform on many not-perfect English samples. So, during the training, it has seen less examples of the kind of data that it is tested against. It has had very little chance to actually "learn" the not-perfect language use and it is notable that this limited learning, fine-tuning with the non-perfect translated data actually decreases the performance.

For translate-train, the extra fine-tuning was of help for the mBERT and EstBERT. The F1 score raised from 74.4 to 78.7 and 75.1 to 77.2, respectively. Those results were for both underlying models the best results across the experiments. For the XLM-RoBERTa, the effect of extra fine-tuning was contrary, so F1 dropped a little from 79.9 to 79.4.

## 5.4 Combined methods

Finally, two experimental combinations of different scenarios were used.

First, a method where the underlying model was sequentially trained on different types of data. Initially, the model was trained on English SQuAD, then on machine-translated SQuAD, and finally on the original EstQA training set. The reason behind this sequencing was confirmed by results from other models. Initial idea was that the sequence should start from least task-specific data and move towards most specific one. Results from other models confirmed that the least specific data was indeed the least beneficial, and vice versa. The zero-shot and translate-test that had used the least specific English training data, had performed worse than translate-train which trained in Estonian. Also, the most specific data that was used the last, the small Estonian dataset, had brought proportionally the highest benefit (in the zero-shot scenario).

The second mix that was experimented with, combined the large datasets. English SQuAD and Estonian translation of it were combined and shuffled. The model was trained on it for 3 epochs and then fine-tuned on the Estonian dataset. Performance was measured after each epoch to discover if it peaks before the 3$^{rd}$ epoch. Reason was that previously (when training only with Estonian data) it became visible that the bigger the number of training samples, the less epochs were needed to reach peak performance. Now the number of training samples was bigger than in other experiments, which may translate to smaller number of epochs.

As part of the training data was in both cases in English, the combinations could only be used with the multilingual language models.

It is visible from Figure 5, that the first, sequential combination of scenarios produced the best result across the research. For XLM-RoBERTa, the previous best F1 score from translate-train (79.9) was surpassed and reached 82.4. The second, shuffled combination also bested translate-train and got the F1 score of 80.9. It can be concluded that combining different strategies can really increase the performance of a QA model. The increase of F1 score from 79.9 until 82.4 can be interpreted as the decrease of errors by a tenth which is a notable difference.

In the case of mBERT, the combined results with F1 scores of 75.2 and 74.8 remained behind the translate-train best result of 78.7.

As to the number of training epochs used in the second combination with the largest dataset, then it turned out that 3 epochs was too much for both multilingual models. Peak performance was reached for XLM-RoBERTa after 2 epochs and for mBERT after only 1 epoch. After that the F1 and EM scores started to deteriorate (see Figure 6).

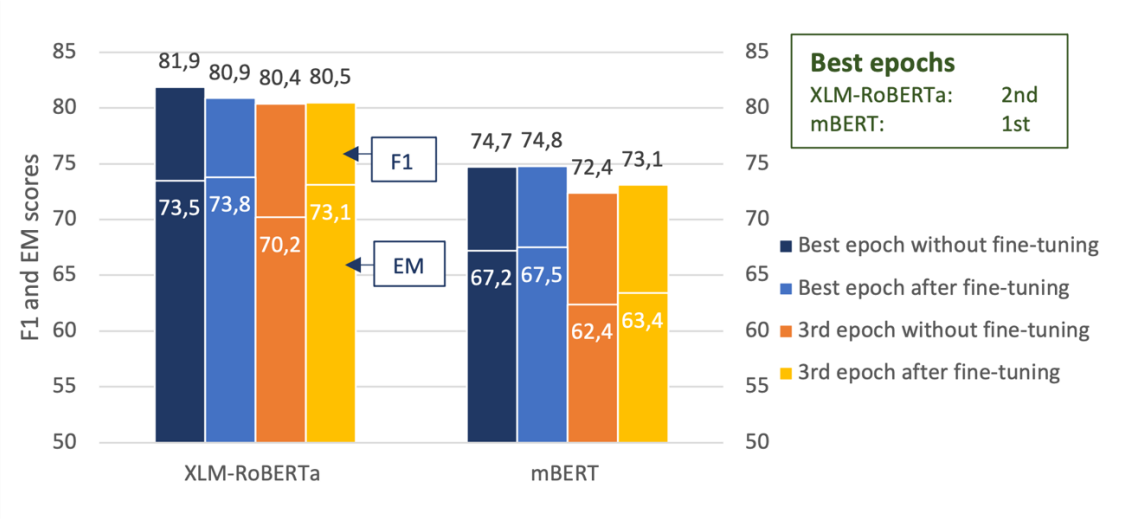


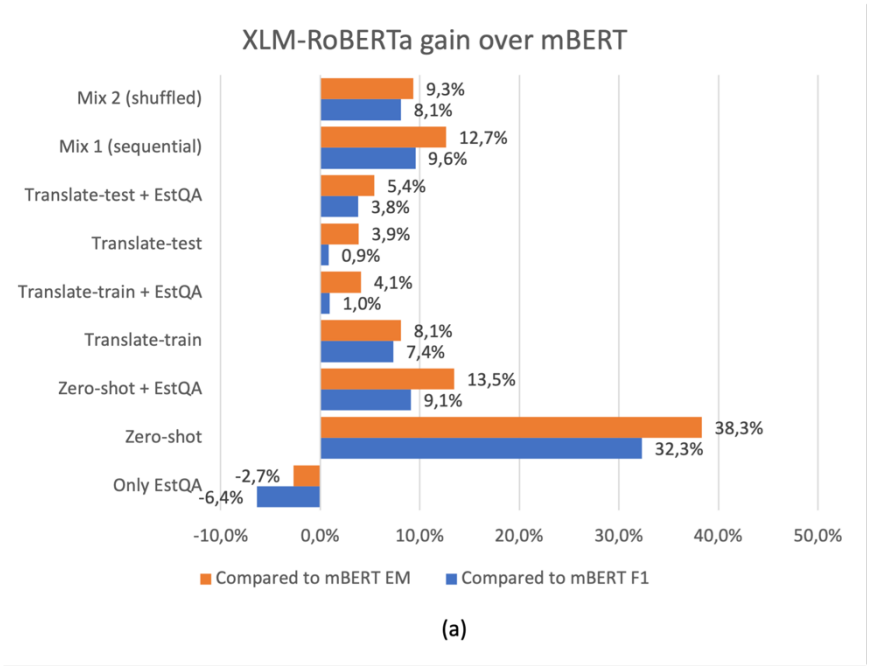

Figure 6. The F1 and EM scores from training with the extra-large dataset combined from English and translated SQuAD. Fine-tuning marks extra fine-tuning round with EstQA training set.

Also, for XLM-RoBERTa, the fine-tuning of the second mixed model with small Estonian dataset actually decreased the performance. This had happened also for translation-based methods. Before the fine-tuning, the F1 score reached 81.9 and decreased to 80.9 after fine-tuning. These results are very close to the results from the first, sequential mix (F1 score 82.4) which were the all-over best.

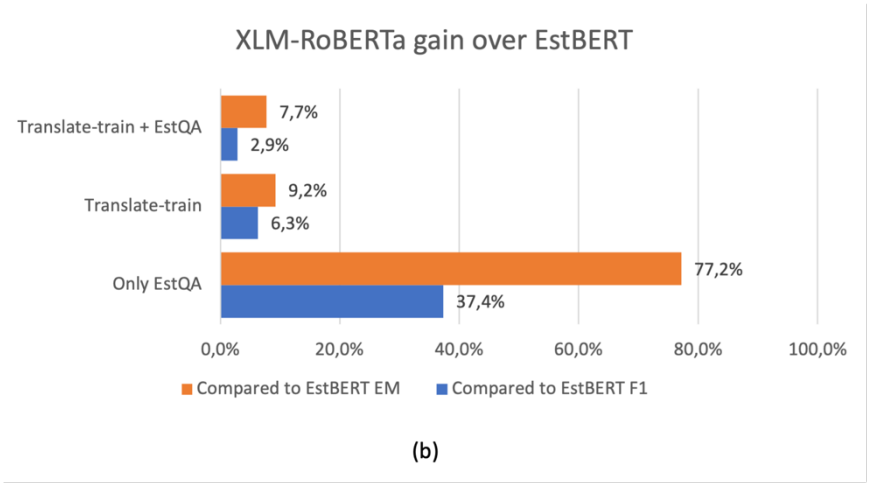
## 5.5 Comparison of underlying language models

As is visible from Figure 5, then the best-performing model across almost all experiments was XLM-RoBERTa. In translation-based methods the superiority was smaller, in zero-shot and combinatory methods larger. Only experiment where it lost to another model was with the small Estonian dataset. Here multilingual BERT was the best.

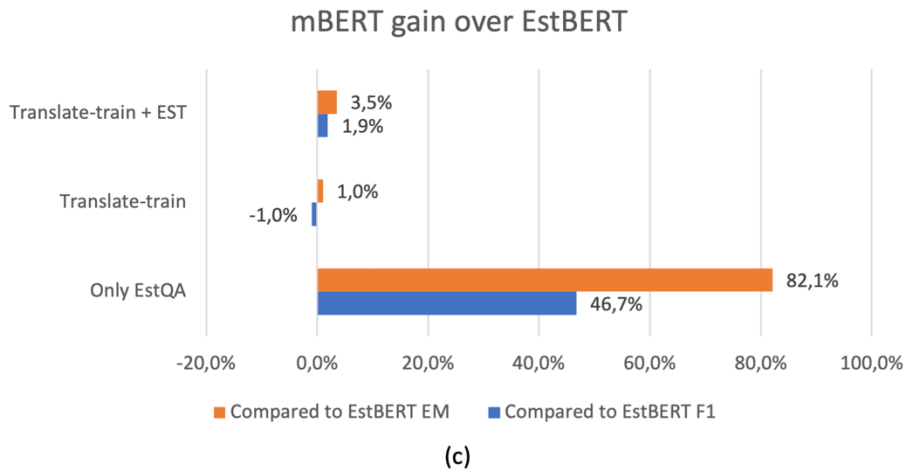
The measure of gain that XLM-RoBERTa achieved over other large contextual language models is visible in Figure 7.




(a)




(b)




(c)

Figure 7. The superiority of XLM-RoBERTa results compared to (a) mBERT and (b) EstBERT, and superiority of mBERT results compared to (c) EstBERT.

It is visible that the worst performer in almost all the categories was EstBERT. Only for translate-train without extra fine-tuning, was its F1 score 1% better than for multilingual BERT. This result is unexpected. As the authors of EstBERT noted [6], based on existing studies, a language-specific BERT model is expected to outperform a multilingual one. For majority of tasks that were used in validating EstBERT, it was true. In the research, EstBERT outperformed mBERT as well as XLM-RoBERTa.

However, the tasks that were used in the validation of EstBERT were mainly based on classification (part-of-speech and morphological tagging, named entity recognition, sentiment, and text classification). None of those dealt with sequence-to-sequence NLP tasks. Question Answering task is not generating new text, but the task is still more complicated. It is dealing with picking any start and end index from the context paragraph that would suit for the answer. Although the variety of possible outputs is finite, the number of choices is still vastly bigger than for a classification task.

Another reason why EstBERT was outperformed may be related to multilingual models as such. They may entail properties and generalization power that is not present in EstBERT and which was necessary for the current task, given that the Estonian data was not ideal. The EstQA fine-tuning set had relatively very little samples and the translated SQuAD suffered from discrepancies of the machine translation.

## 5.6 Error analysis

For the best-performing model, the sequential mixed model based on XLM-RoBERTa, the validation results were studied in detail. All predicted answers, where the EM score was not 1, were categorized.

Altogether there was 149 erroneous answers. The results of the categorization are visible in Figure 8.

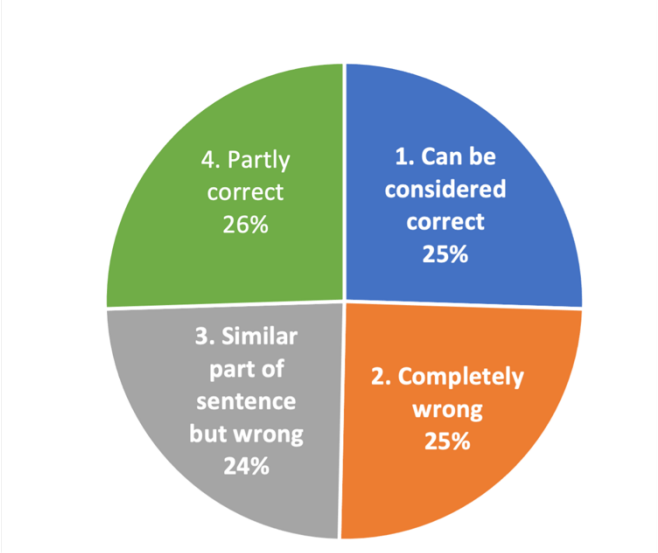



Figure 8. Wrongly predicted answers for the best-performing model.

Error type nr 1 represents the answers which could be considered correct (see examples of error types in Appendix 3). This is a drawback of composing the dataset in a one-person setup. Since annotating answers is not exact science, some possible answers were not annotated. Partly those cases were discovered in the process of training the first models, partly not.

Type nr 2 errors are completely wrong. However, in most of the cases there was still some logical connection with the question, the predicted answer was not random. For instance, the question was about entry fees to the London zoo in the 18$^{th}$ century, specifically about an alternative to monetary fee (you could bring a cat or dog for lion food). The answer was still about the sum of the monetary fee.

Type nr 3 represents errors that could be made by a reader who does not understand text or loses focus. For example, if the question was about a person, then the answer did indeed contain a person, but wrong person was chosen. E.g., there was a question about the philosophers who continued Charles Sanders Peirce's work on the theory of connections. The predicted answer, on the contrary, named a philosopher who preceded Peirce in this field. Often the distinction between type 2 and type 3 is subjective.

Type nr 4 includes answers which are partly correct. Three broad cases can be distinguished:

1. Most common (20 cases) were answers that contained too much information. For example, when in the article it was written that Aristotle was “Ancient Greek philosopher” and question asked where Aristotle was living, then the answer was full phrase “Ancient Greek philosopher”. It contained the correct info but was no longer the smallest span suitable for the answer.
2. Sometimes (14 cases) too little information was provided. For example, when context listed assignments of priests in Livonian Order, some were mentioned as main and some as frequent side assignments. Question asked about all the assignments, predicted answer referred only to the side assignments.
3. In a few instances (4 cases) the correctness of the answer was subjective. For example, the article talked about the oldest discovered settlements on Saaremaa island and named specific areas. Later it was mentioned that the first people on Saaremaa lived at the seashore. Question was about the location of first settlements, and predicted answer chose the seashore. Any discovered settlements were not mentioned regarding the seashore, but it was mentioned that in this place people lived the earliest. So, it may be an “intelligent” conclusion that they also created settlements there.

The types of errors all occurred in roughly similar proportions.

The type 1 errors, however, open new opportunities for performance estimation. If we consider all answers that were found in such cases as golden answers, then the performance of the most successful model rises even further, as visible from Table 11.

Table 11. Effect of adjusting answers according to model’s predictions.

| | **Original dataset** | **With corrected answers** |
|---|---|---|
| **F1** | 82.4 | 85.7 |
| **EM** | 75.3 | 81.6 |

This shows the further potential of critically working with the dataset and stresses the importance of involving multiple people, opening further opportunities to continue with the work started in this thesis.

# 6 Discussion

The experiments showed that combining different training strategies for a QA task with scarce resources can significantly raise the performance. Also, fine-tuning the models with even a small quantity of task-specific data from new native dataset improved the results in majority of the cases. However, for some models the results did not improve with those techniques, so it is dependent on the exact model and training scheme.

The best overall results were achieved with combining different training strategies. Compared to using only English or only translated SQuAD in training, the performance raised by several percentage points when both were used. This confirms that although combined strategies are not often used for QA models in low-resource languages, it can prove to be very beneficial.

Among the underlying contextual language models, best results were achieved with XLM-RoBERTa. Multilingual BERT gained the second-best performance, and the Estonian language specific EstBERT was outperformed by both multilingual models. This shows the strong generalization power of the multilingual models. As pointed out earlier in 5.5, the Estonian dataset that was used for training was not ideal. The amount of new data was very small, and the translated version suffered from shortcomings of less-than-perfect machine translation. EstBERT was the least capable of handling these discrepancies. Also, as pointed out earlier, the previously seen superiority of EstBERT over multilingual models was achieved with less difficult, classification-based tasks.

Among the multilingual models, XLM-RoBERTa outperformed mBERT by a significant margin. This trend has also been noted in related research, referred in chapter 2. The authors of the model have explained this by their training process which followed in BERTs footsteps but tuned the hyperparameters and increased the training time. Important difference is also in the underlying training set. Multilingual BERT relied on Wikipedia sources with their encyclopedic style and consistent grammar. XLM-RoBERTa used more diverse web crawl data from CommonCrawl. Probably this made it more adaptable to the non-perfect Estonian training data.

It is notable that the best QA model, the XLM-RoBERTa model being sequentially trained on original and translated SQuAD, has also very good multilingual QA-related

properties. Experiments have shown it to be well capable of cross-lingual tasks. For example, it can take a paragraph in Russian and find an answer to a question posed in Hindi, Finnish or German. None of those languages were included in the QA-specific fine-tuning. Investigating those multilingual properties more closely is out of the scope, however, those can be experimented with on the web site where I provided access to this model[1] (see Figure 9).

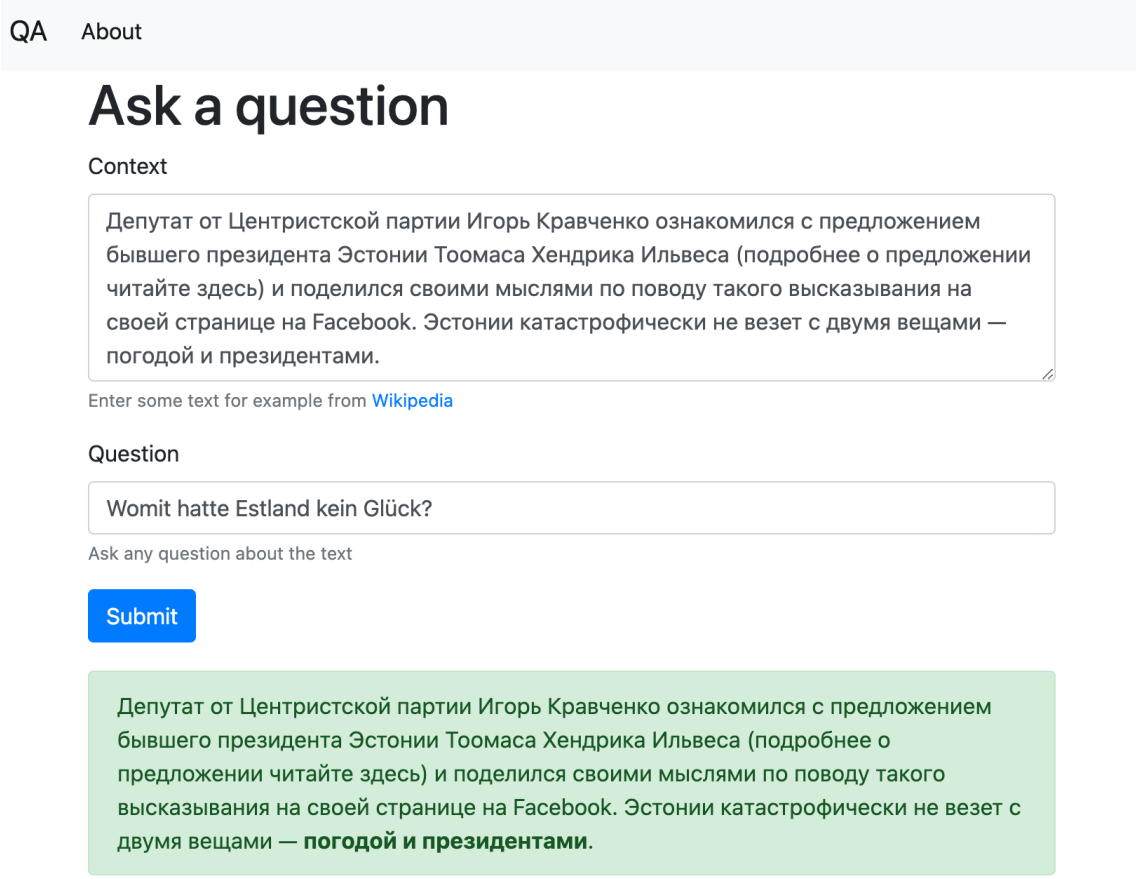



Figure 9. Screenshot from demo application, presenting the multilingual properties of the best model that was developed in the thesis. Answer to a question proposed in German is correctly found from a Russian paragraph.

The model performs especially well in English, which was best represented among XLM-RoBERTa pre-training and was also used in fine-tuning the current model. The F1 score of the model on English SQuAD v1.1 development set is 86.9 which is significantly better than the 82.4 for EstQA.

[1] https://qa.akaver.com

One of the ways to proceed with current thesis is to continue working on the dataset. First, the quantity should be increased to train the models more thoroughly with task-specific data – in correct Estonian language, without discrepancies from machine translation, also more representative of such topics that can come up in Estonian texts.

Creating the 1100 context-question-answer triplets took 10 days for one person, so if more people would be involved, the quantity can be increased manyfold with a feasible effort. Involving more people can also overcome the issues with subjective annotation, that occurred and was discovered only during validation of the results. Diversity would also increase. On the other hand, involving multiple people will bring the need to set up an infrastructure – processes, annotation tools, validation, and cross-validation. Examples of such infrastructure can be found from related research.

To improve performance, some enhancements to the models and training process could also be considered. Larger version of XLM-RoBERTa could be used to benefit from more precise language modelling. Also, ensemble learning could be used, where several instances of the same pretrained language model are created with different fine-tuning seeds. All are fine-tuned on the same data; results are compared and aggregated. This helps to avoid incidental effects from random initialization of the fine-tuning parameters. For example, authors of BERT have used it on SQuAD v1.1 with a clear benefit [3].

# 7 Summary

This master thesis compares several training methods for an extractive Question Answering system in the Estonian language where sufficient training resources are not available. Some methods well known from relevant literature are used and compared – translate-train, translate-test and zero-shot. In addition, experiments are made with combining those training methods. As the underlying large contextual language model, two multilingual models, mBERT and XLM-RoBERTa are used, along with BERT-based Estonian language model EstBERT.

For testing the models, as well as for extra fine-tuning, an original EstQA dataset is composed with 1115 questions based on Wikipedia articles. The structure of the dataset, along with the process of composing it, follow the example of the widely used English QA dataset SQuAD which is also used in the current thesis.

As a result of the experiments, the combined training methods outperform the isolated ones. The most successful model is based on XLM-RoBERTa, fine-tuning it first on English SQuAD dataset, then on the same dataset machine-translated into Estonian and finally on the small native EstQA dataset. The F1 score of this model is 82.4%.

Across underlying language models, XLM-RoBERTa proves superior over the others in eight experiments out of nine. Multilingual BERT follows and EstBERT is outperformed in all three experiments where the monolingual model was involved.

Following work should include improvement of the original dataset, both in quantity and by cross-validation of the content.

# References


[1] D. Lopez Yse, “Your Guide To Natural Language Processing,” 15 January 2019. [Online]. Available: https://towardsdatascience.com/your-guide-to-natural-language-processing-nlp-48ea2511f6e1. [Accessed 23 March 2021].

[2] B. Kratzwald and S. Feuerriegel, “Putting Question-Answering Systems into Practice: Transfer Learning for Efficient Domain Customization,” *ACM Transactions on Management Information Systems,* February 2019.

[3] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics,* pp. 4171-4186, 2019.

[4] R. Keraron, G. Lancrenon, M. Bras, F. E. Allary, G. Moyse, T. Scialom, E.-P. Soriano-Morales and J. Staiano, “Project PIAF: Building a Native French Question-Answering Dataset,” *Proceedings of the 12th Conference on Language Resources and Evaluation,* pp. 5481-5490, 2020.

[5] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer and V. Stoyanov, “Unsupervised Cross-lingual Representation Learning at Scale,” *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics,* pp. 8440-8451, 2020.

[6] H. Tanvir, C. Kittask and K. Sirts, “EstBERT: A Pretrained Language-Specific BERT for Estonian,” *arXiv:2011.04784 [cs.CL],* 9 November 2020.

[7] P. Rajpurkar, J. Zhang, L. Konstantin and P. Liang, “SQuAD: 100,000+ Questions for Machine Comprehension of Text,” *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing,* pp. 2383-2392, November 2016.

[8] P. Rajpurkar, R. Jia and P. Liang, “Know What You Don't Know: Unanswerable Questions for SQuAD,” *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics,* pp. 784-789, 2018.

[9] A. Trischler, T. Wang, X. Yuan, J. Harris, A. Sordoni, P. Bachman and K. Suleman, “NewsQA: A Machine Comprehension Dataset,” *Proceedings of the 2nd Workshop on Representation Learning for NLP,* pp. 191-200, 2017.

[10] J. Liu, Y. Lin, Z. Liu and M. Sun, “XQA: A Cross-lingual Open-domain Question Answering Dataset,” *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics,* pp. 2358-2368, 2019.

[11] P. Lewis, B. Oguz, R. Rinott, S. Riedel and H. Schwenk, “MLQA: Evaluating Cross-lingual Extractive Question Answering,” *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics,* pp. 7315-7330, 2020.

[12] A. Lauscher, V. Ravishankar, I. Vulic and G. Glavaš, “From Zero to Hero: On the Limitations of Zero-Shot Cross-Lingual Transfer with Multilingual Transformers,” *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing,* pp. 4483-4499, 2020.

[13] Y. Jing and D. Xiong, “Effective Strategies for Low-Resource Reading Comprehension,” *International Conference on Asian Language Processing,* pp. 153-157, 2020.

[14] Y. Cui, W. Che, T. Liu, B. Qin, S. Wang and G. Hu, “Cross-Lingual Machine Reading Comprehension,” *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing,* pp. 1586-1595, 2019.

[15] C.-H. Lee and H.-Y. Lee, “Cross-Lingual Transfer Learning for Question Answering,” *arXiv:1907.06042 [cs.CL],* 13 July 2019.

[16] C. Kittask, K. Milintsevich and K. Sirts, “Evaluating multilingual BERT for Estonian,” *Human Language Technologies - The Baltic Perspective,* pp. 19-26, 2020.

[17] T. Jurczyk, A. Deshmane and J. D. Choi, “Analysis of Wikipedia-based Corpora for Question Answering,” *arXiv:1801.02073 [cs.CL],* 6 January 2018.

[18] K. M. Hermann, T. Kočiský, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman and P. Blunsom, “Teaching Machines to Read and Comprehend,” *Advances in Neural Information Processing Systems,* pp. 1693-1701, 2015.

[19] “Wikimedia Downloads,” [Online]. Available: https://dumps.wikimedia.org/backup-index.html. [Accessed 20 October 2020].

[20] Project Nayuki, “Computing Wikipedia’s internal PageRanks,” 19 February 2016. [Online]. Available: https://www.nayuki.io/page/computing-wikipedias-internal-pageranks. [Accessed 20 October 2020].

[21] F. Mikaelian, A. Farias, M. Amrouche, O. Sans and T. Nazon, “cdQA-annotator,” 2019. [Online]. Available: https://github.com/cdqa-suite/cdQA-annotator. [Accessed 20 October 2020].

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, “Attention Is All You Need,” *Advances in Neural Information Processing Systems,* pp. 6000-6010, 2017.

[23] T. Pires, E. Schlinger and D. Garette, “How multilingual is Multilingual BERT?,” *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics,* pp. 4996-5001, 2019.

[24] K. K, Z. Wang, S. Mayhew and D. Roth, “Cross-Lingual Ability of Multilingual BERT: An Empirical Study,” in *International Conference on Learning Representations*, 2020.

[25] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer and V. Stoyanov, “RoBERTa: A Robustly Optimized BERT Pretraining Approach,” *arXiv:1907.11692 [cs.CL],* 26 July 2019.

[26] G. Lample and A. Conneau, “Cross-lingual Language Model Pretraining,” *Advances in Neural Information Processing Systems,* pp. 7059-7069, 2019.

[27] S. Romano, “Multilingual Transformers: Why BERT is not the best choice for multilingual tasks,” Towards Data Science, 17 January 2020. [Online]. Available: https://towardsdatascience.com/multilingual-transformers-ae917b36034d. [Accessed 20 April 2021].

[28] V. Sanh, L. Debut, J. Chaumond and T. Wolf, “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter,” *arXiv:1910.01108 [cs.CL],* 2 October 2019.

[29] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov and Q. V. Le, "XLNet: Generalized Autoregressive Pretraining for Language Understanding," *Advances in Neural Information Processing Systems,* pp. 5754-5764, 2019.

[30] Z.-Y. Dou and G. Neubig, "Word Alignment by Fine-tuning Embeddings on Parallel Corpora," *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume,* pp. 2112-2128, April 2021.

# Appendix 1 – Non-exclusive licence for reproduction and publication of a graduation thesis[1]

I, Anu Käver

1. Grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis "Question Answering System for Estonian Language," supervised by Tanel Alumäe.
   1.1. to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;
   1.2. to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.
2. I am aware that the author also retains the rights specified in clause 1 of the non-exclusive licence.
3. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

10.05.2021

1 The non-exclusive licence is not valid during the validity of access restriction indicated in the student's application for restriction on access to the graduation thesis that has been signed by the school's dean, except in case of the university's right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.

# Appendix 2 – Example of question with answer variations in EstQA dataset

| | |
|---|---|
| **Context** | Ametlik õppeaeg Pallases oli kuus aastat. Keskmiseks õpingute kestuseks Pallases kujunes siiski 7 ⅓ aastat. Paljude õppeaeg venis kümne aasta piiridesse. Meesõpilastel langes õpingute aega ka sundaja teenimine kaitseväes. Mitmed õpilased, näiteks Karl Pärsimägi, Rudolf Sepp, Eduard Kutsar jt, pidid ajutiselt katkestama, et õpinguteks raha teenida. Kuid oli ka neid, kes lõpetasid Pallase ettenähtud kuue aastaga, andekamad – näiteks Elmar Kits, Richard Kaljo, Erik Haamer jt – vähemagagi. Oli neidki, kellel jäi koolitee pooleli. Sageli oli põhjuseks vähene andekus või siis majanduslikud raskused. Ateljeedes ei kiirustatud lõpetamisega, samuti meisterateljees. Õpetajate nõukogu kaalus iga meisterateljee õpilast, enne kui laskis kooli lõpetada. Lõpetajat püüti välja saata suutelisena elus läbi lüüa. |
| | *The official study time at Pallas was six years. However, the average duration of studies at Pallas was 7 ⅓ years. The study period of many lasted for ten years. For male students, the compulsory military service in the Defense Forces also took place during the studies. Several students, such as Karl Pärsimägi, Rudolf Sepp, Eduard Kutsar and others, had to temporarily pause in order to earn money for their studies. But there were also those who graduated from Pallas in six years, more talented - for example, Elmar Kits, Richard Kaljo, Erik Haamer and others. There were also those who quit school. This was often due to a lack of talent or financial hardship. The studios were in no hurry to push students to graduate, as well as in the master's studio. The teachers' council weighed each student in master's studio before allowing them to graduate. An attempt was made to have the graduate finish school with the ability to succeed in life.* |
| **Question** | **Mis pikendas meeste õppeaega Pallases? / *What prolonged the study time for male students?*** |
| **Answer 1** | "sundaja teenimine kaitseväes" / *"the compulsory service in the Defense Forces"* |
| **Answer 2** | "teenimine kaitseväes" / *"service in the Defense Forces"* |
| **Answer 3** | "sundaja teenimine" / *"the compulsory service"* |
| **Answer 4** | "langes õpingute aega ka sundaja teenimine kaitseväes" / *"the compulsory service in the Defense Forces also took place during the studies"* |

| | |
|---|---|
| **Answer 5**[1] | "langes õpingute aega ka sundaja teenimine" / *"the compulsory service also took place during the studies"* |

[1] The task was to annotate the answer as the smallest continuous span that is suitable for the answer. However, in many cases the decision about the smallest span is subjective. As visible from the table, it is a matter of interpretation if the factor stopping students from finishing the school was the military service as such or the fact that the time of the military service overlapped with the time of studies, plus combinations with other nuances in the sentence. All those options can be considered correct, so all were included in the possible ground truth answers (golden answers).

# Appendix 3 – Types of errors in the predictions

| | |
|---|---|
| **Error type 1 – answer could be correct** | |
| Context | 1880ndatel kasvas Peirce'i ükskõiksus oma **geodeesiaameti** töö bürokraatlike üksikasjade vastu, tema töö kvaliteet langes ja töö **geodeesiateenistuses** ei edenenud enam endise kiirusega. |
| | *In the 1880s, Peirce's indifference to the bureaucratic details of his work at the **geodesy agency** grew, the quality of his work declined, and his work in the **geodesy service** no longer progressed at its former pace.* |
| Question | Kus Peirce 1880ndatel töötas? / *Where did Peirce work in the 1880s?* |
| Predicted answer | geodeesiaameti / *geodesy agency* |
| Golden answers | geodeesiateenistuses / *geodesy service* |
| **Error type 2 – answer is definitely incorrect** | |
| Context | Popper sündis Austria-Ungari pealinnas Viinis jõukas ja haritud juudi keskklassi perekonnas. Tema vanemad olid **advokaat** Simon Siegmund Carl Popper (Viini liberaalse linnapea Raimund Grübli lähedane kaastööline) ja Jenny Popper (sündinud Schiff). […] Ema süstis pojasse nii suure muusikahuvi, et ta vahepeal kaalus **elukutseliseks muusikuks** hakkamist. |
| | *Popper was born in Vienna, the capital of Austria-Hungary, into a wealthy and educated Jewish middle-class family. His parents were **lawyer** Simon Siegmund Carl Popper (close associate of the liberal mayor of Vienna Raimund Grübli) and Jenny Popper (born Schiff). […] The mother injected so much interest in music into her son that in the meantime he was considering becoming a **professional musician**.* |
| Question | Mis ametit pidas Popperi isa? / *What job did Popper's father hold?* |
| Predicted answer | elukutseliseks muusikuks / *professional musician* |
| Golden answers | advokaat / *laywer* |
| **Error type 3 – answer is similar but incorrect** | |
| Context | Siiski pakkusid uued ruumid vaid ajutist leevendust ruumikitsikusele, pealegi tuli maja jagada **Tartu linna tervishoiuosakonna ja elukorteritega**. […] 1923. aasta augustis koliti uude majja ja Pallase ruumidesse Kalamehe tänaval asus **Naisühingu Käsitöökool**. |

| | |
|---|---|
| | *However, the new premises offered only temporary relief from space constraints, and the house had to be shared with the **Tartu City Health Department and residential apartments**. […] In August 1923, they moved to a new house and the **Women's Association Handicraft School** moved to the premises of Pallas on Kalamehe Street.* |
| Question | Mis asus Kalamehe tänava hoones lisaks Pallasele? / *What was located in the building on Kalamehe Street in addition to Pallas?* |
| Predicted answer | Naisühingu Käsitöökool / *Women's Association Handicraft School* |
| Golden answers | Tartu linna tervishoiuosakonna ja elukorteritega / *Tartu City Health Department and residential apartments* |
| **Error type 4 – answer is partly correct** | |
| Context | Ordumeistril aitasid ordut hiljemalt 15. sajandi keskpaigast juhtida 5–6 käsknikust koosnev sisemine ehk kitsam nõukogu. Kõige olulisemateks käsknikeks olid **maamarssal, Viljandi, Tallinna, Aluliina ja Kuldīga komtuurid ning Järva foogt**. Mõnikord kuulusid siseringi ka teised käsknikud, kõige tihemini Dünaburgi või Pärnu komtuur. |
| | *By the middle of the 15th century at the latest, the Master of the Order was helped to lead the order by the inner or narrower council of 5–6 commanders. The most important commanders were **the Marshal, the commanders of Viljandi, Tallinn, Aluliina and Kuldīga, and the bailiff of Järva**. Sometimes other commanders also belonged to the inner circle, most often the Dünaburg or Pärnu commander.* |
| Question | Millised käsknikud kuulusid kõige sagedamini ordu sisemisse nõukokku? / *Which commanders most often belonged to the inner council?* |
| Predicted answer | maamarssal, Viljandi, Tallinna, Aluliina ja Kuldīga / *the Marshal, the commanders of Viljandi, Tallinn, Aluliina and Kuldīga* |
| Golden answers | maamarssal, Viljandi, Tallinna, Aluliina ja Kuldīga komtuurid ning Järva foogt / *the Marshal, the commanders of Viljandi, Tallinn, Aluliina and Kuldīga, and the bailiff of Järva* |