BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

MULTILINGUAL OPEN-DOMAIN QUESTION ANSWERING

VÍCEJAZYČNÝ SYSTÉM PRO ODPOVÍDÁNÍ NA OTÁZKY NAD OTEVŘENOU DOMÉNOU

MASTER'S THESIS
DIPLOMOVÁ PRÁCE

AUTHOR **Bc. MICHAL SLÁVKA**
AUTOR PRÁCE

SUPERVISOR **Ing. MARTIN FAJČÍK**
VEDOUCÍ PRÁCE

BRNO 2020

Brno University of Technology
Faculty of Information Technology

Department of Computer Graphics and Multimedia (DCGM) Academic year 2020/2021

Master's Thesis Specification

23369

Student:	**Slávka Michal, Bc.**
Programme:	Information Technology
Field of study:	Machine Learning
Title:	**Multilingual Open-Domain Question Answering**
Category:	Speech and Natural Language Processing

Assignment:

1. Describe current state-of-the-art approaches to open-domain question answering and machine translation.
2. Choose suitable open-domain QA datasets in multiple languages.
3. Design a multilingual open-domain QA system.
4. Implement and evaluate your design.
5. Create a poster that presents your work.

Recommended literature:

- Karpukhin, V., Ouz, B., Min, S., Wu, L., Edunov, S., Chen, D. and Yih, W.T., 2020. Dense Passage Retrieval for Open-Domain Question Answering. *arXiv preprint arXiv:2004.04906*.
- Guu, K., Lee, K., Tung, Z., Pasupat, P. and Chang, M.W., 2020. Realm: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*.
- Lee, K., Chang, M.W. and Toutanova, K., 2019. Latent retrieval for weakly supervised open domain question answering. *arXiv preprint arXiv:1906.00300*.
- Yang, W., Xie, Y., Lin, A., Li, X., Tan, L., Xiong, K., Li, M. and Lin, J., 2019. End-to-end open-domain question answering with bertserini. *arXiv preprint arXiv:1902.01718*.

Requirements for the semestral defence:

- Items 1 to 3.

Detailed formal requirements can be found at https://www.fit.vut.cz/study/theses/

Supervisor:	**Fajčík Martin, Ing.**
Head of Department:	Černocký Jan, doc. Dr. Ing.
Beginning of work:	November 1, 2020
Submission deadline:	May 19, 2021
Approval date:	May 11, 2021

Abstract

This thesis explores automatic Multilingual Open-Domain Question Answering. In this work are proposed approaches to this less explored research area. More precisely, this work examines if: (i) utilization of an English system is sufficient, (ii) multilingual models can benefit from a translated question into other languages (iii) or avoiding translation is a better choice. English system based on the T5 model that uses a machine translation is compared to natively multilingual systems based on the multilingual MT5 model. The English system with machine translation only slightly outperforms its monolingual counterparts in multiple tasks. Compared to multilingual models, the English system was trained on a much larger dataset, but the results were comparable. This shows that the use of natively multilingual systems is a promising approach for future research. I also present a method of retrieving multilingual evidence using the BM25 ranking algorithm and compare it with English retrieval. The use of multilingual evidence seems to be beneficial and improves the performance of the systems.

Abstrakt

Táto práca sa zaoberá automatickým viacjazyčným zodpovedaním na otázky v otvorenej doméne. V tejto práci sú navrhnuté prístupy k tejto málo prebádanej doméne. Konkrétne skúma, či: (i) použitie prekladu z angličtiny je dostačujúce, (ii) multilinguálne systémy vedia využiť preklad otázky do iných jazykov (iii) alebo je výhodnejšie nepoužívať žiaden preklad. Porovnávam použitie anglického systému založeného na modeli T5, ktorý využíva strojový preklad s natívne viacjazyčnými systémami založenými na viacjazyčnom modeli MT5. Anglický systém so strojovým prekladom mierne prekonáva svoje jednojazyčné náprotivky vo viacerých úlohách. Napriek tomu, že tento model bol natrénovaný na väčšom množstve dát zlepšenie nie je dostatočne signifikantné. To ukazuje, že použitie natívne viacjazyčných systémov je sľubným prístupom pre budúci výskum. Tiež prezentujem metódu získavania dokumentov v rôznych jazykoch pomocou algoritmu BM25 a porovnávam ju s anglickým retrievalom. Používanie viacjazyčných dôkazov sa javí ako prospešné a zlepšuje výkonnosť systému systémov.

Keywords

Natural Language Processing, Question Answering, Information Retrieval, Multilingual, BM25, Transformers

Kľúčové slová

Spracovanie Prirodzeného jazyka, Automatické Odpovedanie na Otázky, Získavanie Informácií, BM25, Transformers

Reference

SLÁVKA, Michal. *Multilingual Open-Domain Question Answering.* Brno, 2020. Master's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Ing. Martin Fajčík

Rozšírený abstrakt

Systémy schopné automaticky odpovedať na otázky sú užitočné v poskytovaní priamych odpovedí na užívateľove otázky. Získavanie informácie zvyčajne vyžaduje zdĺhavý proces hľadania zdrojov obsahujúcich chcenú informáciu a ich následné preštudovanie až kým nenájdeme cieľovú informáciu. Už od nepamäti sa získavanie informácií neprestajne zjednodušuje a odpoveďou na tento trend by mohol byť práve automatické odpovedanie otázok (Question Answering (QA)).

Automatické odpovedanie otázok (QA) je výskumná oblasť čerpajúca zo spracovania prirodzeného jazyka (Natural Language Processing) a vyhľadávania informácie (Information Retrieval (IR)). Úlohou QA je automaticky nájsť odpoveď na otázku položenú človekom.

Systém odpovedajúci na otázky zvyčajne podmieňuje svoje odpovede dokumentom alebo súborom dokumentov obsahujúcich relevantné informácie požadované na zodpovedanie otázky. Dokument môže byť buď zaobstaraný systémom alebo vybratý z väčšej zbierky dokumentov. Vyberanie takého dokumentu je hlavnou oblasťou záujmu Informational Retrieval. Ak systému nebol poskytnutý dokument s relevantnou informáciou, tak túto úlohu nazývame Open-Domain Question Answering.

Vyvodzovanie odpovede z dokumentu alebo dokumentov vyžaduje prirodzené jazykové porozumenie, ktoré je výskumnou oblasťou v prirodzenom spracovaní jazyka. Táto téza skúma systémy schopné odpovedania otázok položených vo viacerých jazykoch. Viacjazyčné odpovedanie otázok (Multilingual Question Anwering) je nová výskumná oblasť, ktorá ešte nie je veľmi preskúmaná a obsahuje zaujímavé výskumné výzvy. Viacjazyčné QA systémy sú zaujímavé z niekoľkých dôvodov. Viacjazyčné systémy sa stávajú bežnejšími, ale ich presnosť stále zaostáva za jednojazyčnými systémami.

Jazyky nepredstavujú iba komunikačnú bariéru, ale aj kultúrnu, ktorá sa odzrkadľuje v asymetrii informácií. Viacjazyčné systémy majú veľký potenciál na prekonanie jednojazyčných systémov vďaka väčšiemu množstvu informácií dostupných v iných jazykoch. Napríklad, ak by chcel niekto, kto rozpráva po francúzsky, zistiť informáciu o kultúre v Brne v Českej republike a bol by odkázaný iba na francúzske zdroje, tak je tu vysoká pravdepodobnosť, že ju nebude schopný nájsť. Avšak táto informácia je takmer určite dostupná v češtine alebo v inom jazyku, ktorý sa viac spája s mestom. Používanie viacjazyčného korpusu môže dodatočne poskytnúť rozličné popisy rovnakej informácie, ktorá by mohla byť využitá na agregáciu informácií pre zodpovedanie otázky.

Pôvodne bol QA výskum anglicko-centrický, avšak väčšina sveta nehovorí po anglicky. Aj keď sa môže považovať za *lingua franca*, iba okolo 16% svetovej populácie dosahuje nejakú úroveň plynulosti v angličtine a iba okolo 5% sú rodení hovorcovia. Viacjazyčné systémy by si preto vychutnali oveľa väčšie publikum a umožnili by ľahší prístup k informáciám pre veľké časti obyvateľstva.

Táto práca sa zaoberá automatickým viacjazyčným zodpovedaním na otázky v otvorenej doméne. V tejto práci sú navrhnuté prístupy k tejto málo prebádanej doméne. Konkrétne skúma, či: (i) použitie prekladu z angličtiny je dostačujúce, (ii) multilinguálne systémy vedia využiť preklad otázky do iných jazykov (iii) alebo je výhodnejšie nepoužívať žiaden preklad.

Porovnávam použitie anglického systému založeného na modeli T5, ktorý využíva strojový preklad s natívne viacjazyčnými systémami založenými na viacjazyčnom modeli MT5. Anglický systém so strojovým prekladom mierne prekonáva svoje jednojazyčné náprotivky vo viacerých úlohách. Napriek tomu, že tento model bol natrénovaný na väčšom množstve dát zlepšenie nie je dostatočne signifikantné. To ukazuje, že použitie natívne viacjazyčných systémov je sľubným prístupom pre budúci výskum.

Tiež prezentujem metódu získavania dokumentov v rôznych jazykoch pomocou algoritmu BM25 a porovnávam ju s anglickým retrievalom. Používanie viacjazyčných dôkazov sa javí ako prospešné a zlepšuje výkonnosť systému systémov.

Vyhodnotil som modely na troch samostatných úlohách, aby som izoloval efekt prekladu na retrieval a aj samotný model. Najlepší výkon v nastaveniach reálneho sveta, dosiahol anglický model v spojení s prekladateľskými strojmi, ktoré v angličtine dosiahli skóre EM 32% a 11% v priemere pre všetky jazyky. Zdá sa, že viacjazyčné vyhľadávanie prekonáva jednojazyčné vyhľadávanie vo všetkých jazykoch.

Multilingual Open-Domain Question Answering

Declaration

I hereby declare that this Bachelor's thesis was prepared as an original work by the author under the supervision of Martin Fajčík. I have listed all the literary sources, publications, and other sources, which were used during the preparation of this thesis.

. . . .
Michal Slávka
May 18, 2021

Contents

1 Introduction 3

2 Datasets 6
2.1 Multilingual Knowledge Questions and Answers 7
2.2 Multilingual Question Answering 8
2.3 TyDi QA and XOR- TyDi QA 9
2.4 Datasets Overview 10

3 Pre-trained Neural Models 12
3.1 Transformers 12
3.1.1 Multi-Head Attention layer 14
3.2 Text-to-Text Transfer Transformer 14
3.3 Multilingual Text-to-Text Transfer Transformer 15
3.4 BERT 16

4 Information Retrieval 17
4.1 Okapi BM25 17
4.2 Neural Information Retrieval 18
4.2.1 Open-Retrieval Question Answering 18
4.2.2 Dense Passage Retriever 20
4.3 Multilingual Retrieval 20
4.3.1 Pivot Through English 21
4.3.2 XOR- TyDi QA 21
4.4 Concurrent Work 21

5 Proposed Multilingual Systems 23
5.1 Readers 23
5.2 Retrieval 24
5.2.1 Monolingual Retrieval 24
5.2.2 Multilingual Retrieval 25
5.3 Proposed Models 25
5.3.1 Model with Pivot Language 26
5.3.2 Multilingual Model with Tag 26
5.3.3 Original Question with Multilingual Retrieval 27
5.3.4 Original Question with English Retrieval 27

6 Experiments 29
6.1 Experimental Setup 29

6.1.1 Language Selection . . . 29
6.1.2 Wikipedia Pre-Processing . . . 29
6.1.3 BM25 . . . 31
6.1.4 Machine Translation . . . 31
6.1.5 Evaluation . . . 32
6.1.6 Training . . . 32
6.2 Results . . . 33
6.2.1 Human Translations . . . 33
6.2.2 Translated Questions . . . 34
6.2.3 Real-World Performance . . . 34
6.2.4 Multilingual BM25 . . . 35

7 Conclusion 40

Bibliography 41

Chapter 1

Introduction

Systems capable of automatically answering questions are useful in providing direct answer to the user's question. Typically obtaining an information requires lengthy process of finding resources containing wanted information and reading trough them until the information is found. Ease of acquiring information have been growing steadily since the begging of time. And Question Answering might be the next step in this trend.

Question Answering (QA) is a research area drawing from Natural Language Processing and Information Retrieval (IR). The task of Question Answering is to automatically find an answer to a question asked by a human.

A system answering a question usually conditions its answer on a document or a set of documents containing relevant information required to answer the question. The document can be either provided to the system or selected from a bigger collection of documents. Selecting such document is a domain of interest of Information Retrieval. When the system is not provided with a document with relevant information the task is called Open-Domain Question Answering. Summarized question answering tasks can be seen in figure 1.1.

Inferring an answer, from document or documents, requires Natural Language Understanding which is a research are in Natural Language Processing. The task is to determine which of these documents contain an answer and where it is located within the document. This is called Extractive Question Answering because the model is trained to identify substring containing an answer. Extracting an answer from a document is done by a sequence to sequence system, which converts input sentence into a sequence of probabilities expressing how likely is a certain word start or end of the answer.

For example given a question *"Since when has there been women's professional soccer championships?"* and a document *"FIFA Women's World Cup has been held every four years since 1991. Under the tournament's current format, national teams vie for 23 slots in a three-year qualification phase."* the system assigns a probability to each span of minimal text units the system is working with. A minimal text unit can be a character, a word or most commonly a learned sequence of characters represented by a token. The system is than expected to assign the highest probability to a span containing *"1991"*, which is the correct answer.

A different approach would be to a generative system, which instead of assigning span probabilities is asked to generate an answer token by token, given a question and optionally a relevant document. Therefore, the approach can even generate answers not present in the document.

This thesis explores systems capable of answering questions given in multiple languages. Multilingual Question Answering a new research area which is not very well explored and

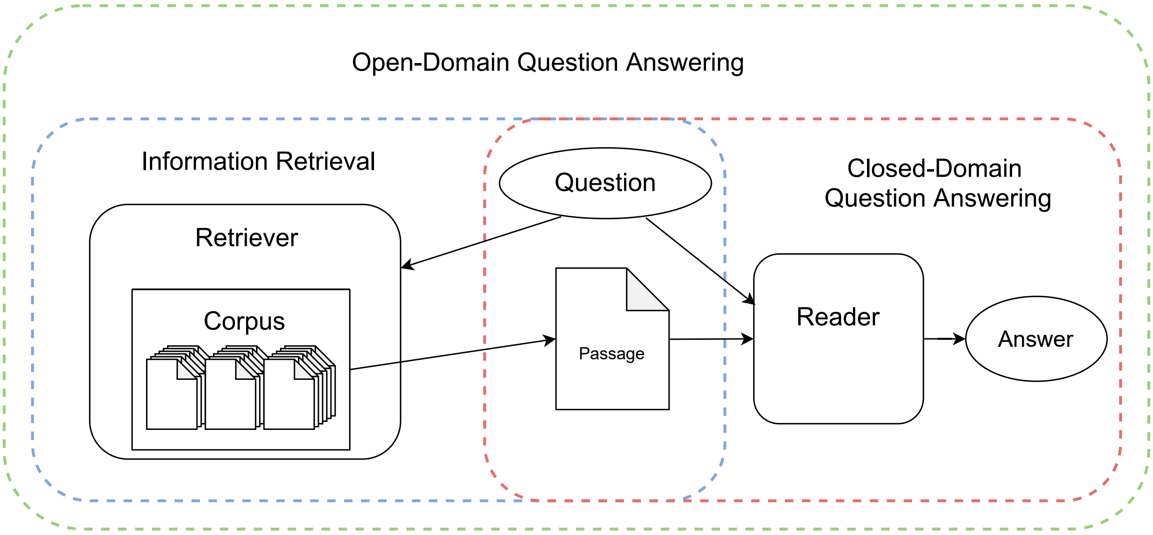

Figure 1.1: Question Answering tasks.

poses an interesting research challenges. Multilingual QA systems are interesting for couple of reasons. Multilingual systems are becoming more common but their accuracy still lacks behind monolingual systems.

Languages do not pose only communicational barrier but also cultural barrier which is reflected in information asymmetry. Multilingual systems have great potential to outperform monolingual systems, because of larger pool of information available in different languages. For example if someone speaking French would like to find information about city Brno in Czech Republic and would be constrained only to French resources, there is a significant chance that they won't be able to find it. However this information is almost certainly accessible in Czech or other language which has more ties to the city. Using multilingual corpus could additionally provide are different description of the same information, which could be utilized for evidence aggregation.

Traditionally Question Answering research was English-centric. However most of the world does not speak English. Although it can be considered a lingua franca only around 16% of the worlds population achieves some degree of fluency in English and only around 5% are native speakers [9]. Therefore multilingual systems would enjoy much larger audience and would make access to information easier for large proportion of population.

Another motivation for creating multilingual QA systems is rising popularity of virtual assistants, where question answering is one of the most commonly used feature [1]. Virtual assistants are available in wide variety of smart devices and their use is predicted to be growing, however their language support is limited supporting up to 30 languages [2].

In this work I propose two approaches to multilingual question answering. The first is using question translations and language tags to identify what language the answer should be in. The second does not use any translations at all.

I propose a method of multilingual retrieval based on the BM25 algorithm. In comparison to English retrieval multilingual retrieval achieves slightly higher accuracy and seems to work better with multilingual models.

[1] https://blog.adobe.com/en/publish/2018/09/06/adobe-2018-consumer-voice-survey.html.

[2] Siri from Apple https://www.apple.com/ios/feature-availability., Google assistant https://en.wikipedia.org/wiki/Google_Assistant.

Lastly, I compare multilingual models with English model combined with machine translation. The English system with machine translation works slightly better compared to multilingual models. However, this difference is only marginal and for some tasks multilingual models performs better.

In the following chapter 2 is an overview of multilingual datasets. The chapter 3 contains description of neural models that can be used for generating or extracting an answer. In the chapter 4 are described methods of paragraph retrieval. In the chapter 5 is description of proposed approaches and description of models. The chapter 6 contains experimental setup, implementation details and experimental results.

Chapter 2

Datasets

The main driver of machine learning research are datasets. Statistical models used in natural language processing are getting larger and larger, with hundreds of millions or even billions of trainable parameters, to increase their performance. Even more modest models have hundreds of millions of trainable parameters. Large models require a large collection of examples to be able to learn their parameters to approximate probability distribution over data.

Training a QA model usually consists of pre-training and fine-tuning. Pre-training a model done on an unsupervised task, such as language modeling, where the model is trying to predict a word which was was removed from the input sequence. Examples of language modeling tasks are next word prediction, where the model is trying to predict a word following a sequence of words, or masked language model, where some words in a sequence are masked and the model is predicting what the masked out words should be. Pre-training is the most computationally expensive part of training and models are usually published with pre-trained parameters. The goal of pre-training is to reduce the amount of annotated data required for training for a specific task.

To fine-tune a model for question answering some form of supervision is required. Question answering datasets contain a collection of questions and annotations in form of an answer string or answer span within a context containing the answer.

A multilingual QA dataset can be in form of multiple monolingual datasets in different languages, but more often than not authors of multilingual dataset pose more constraints on the data it contains to fit a certain purpose. These constraints can influence the choice of languages and data selection. For example, if human translators are required then the availability of translators for a language has to be considered, or if the aim is to have data with diverse languages, then selected languages should not be mutually intelligible and should belong to different language families. Although monolingual datasets exist in different languages, simply joining them makes cross-lingual analyses more challenging because of different annotation setups [20].

Datasets are usually split into training, development and test sets, each of which has a different purpose during the training process.
Training set contains data for training. Parameters of a model are optimized on training data and usually train set is larger than the development and test set.
Development set, or *Validation set*, is used during training to indicate when optimization should stop to prevent over-fitting of a model on Train set and for hyper parameter optimization.

Test set is used exclusively to evaluate a model on different data than those used during training, to get an unbiased assessment of performance.

2.1 Multilingual Knowledge Questions and Answers

Multilingual Knowledge Questions and Answers (MKQA) [21] is an open-domain question answering dataset, created for evaluation purposes. It contains 10,000 question-answer pairs randomly sampled from English Natural Questions [18]. Each pair was translated by a human translator into additional 24 languages.

The languages were selected to maximize both typological diversity and the number of speakers in the world. The selected languages reach is allegedly 90% of the world's population.

Along with answer translation annotations also contain answer type of answer, binary (yes/no), short, number with the unit, the number without unit, numeric, date, unanswerable, long, entity. Answer type long has only an answer in English and was not translated. Long answers and unanswerable types form 32.5% of the dataset, which means that the actual number of samples used during training was 6758 out of the original 10000 per language.

An example from MKQA dataset:

```
{
  "query": "what is the lowest point of the earth",
  "answers": {
    "en": [
      {
        "type": "entity",
        "entity": "Q23883",
        "text": "Dead Sea",
        "aliases": ["Salt Sea", ... ]
      },
    ],
    "de": [...],
    ...,
    "fi": [...]
  },
  "queries": {
    "en": "what is the lowest point of the earth",
    "de": "Welches ist der tiefste Punkt der Erde",
    ...,
    "fi": "mika on maapallon matalin piste"
  },
  "example_id": -7841128731892324000
}
```

The number of usable examples in this dataset is relatively small, however, compared to MLQA it is still larger and includes more languages. For this reason, MKQA was selected as a training dataset.

As this dataset was created for evaluation purposes, it is not divided into training, validation, and test split.

Natural Questions

Natural Questions (NQ) [18] is a large scale dataset in English for training QA models.

Instances contain *question, Wikipedia page, long answer, short answer.*

Annotations are in form of long and short answer spans and were created by showing an annotator a question and a Wikipedia article. Questions were selected from google queries, which means they were asked by people seeking information. In other more common datasets, annotators were asked to ask a question that is answered in a provided article and then select an answer. This makes the NQ dataset a unique resource, with naturally asked questions. Questions are paired with the whole Wikipedia page and answer span within the page.

The open domain version of NQ [19] consists only of questions with short answers and discards any answers longer than 5 tokens.

NQ contains 307,373 training examples and 7,830 examples annotated by 5 annotators for development data. 49% of the examples have long answer and 36% have a short answer or yes/no answer.

Selected answers are mostly paragraphs (73%) and tables (19%), other answers contain table rows, lists, or list items.

2.2 Multilingual Question Answering

Multilingual Question Answering (MLQA) [20] is a multilingual dataset. It was created for the evaluation of multilingual models. Compared to MKQA dataset is considerably smaller and was created for evaluation purposes. It contains 7 languages:

English, Arabic, German, Spanish, Hindi, Vietnamese and Simplified Chinese.

MLQA is parallel across languages. This was achieved by extracting contexts in multiple languages. Questions were crowd-sourced in English and then translated to other languages by professional translators and answers annotated in aligned non-English contexts. If the answer was not present in a parallel context then it was discarded.

Although not all contexts are parallel in all languages, the majority is 4-way aligned. The number of parallel instances between languages can be seen in the table 2.1.

Formally for each each English question q_{en} and English context p_{en} there is at least one translation of question q_{en} in language l, q_l with context in language l that contains the answer.

Instances consists of question q_{lq} in language lq, context d_{lc} in language lp, *answer span* marking the correct answer in context d_{lc} and an id to match parallel question in other language.

	English	German	Spanish	Arabic	Chinese	Vietnamese	Hindu
English	12738						
German	5029	5029					
Spanish	5753	1972	5753				
Arabic	5852	1856	2139	5852			
Chinese	5641	1811	2108	2100	5641		
Vietnamese	6006	1857	2207	2210	2127	6006	
Hindu	5425	1593	1910	2017	2124	2124	5425

Table 2.1: Number of parallel instances between languages in MLQA dataset.

2.3 TyDi QA and XOR- TyDi QA

Typologically Diverse Question Answering Dataset [7] (TyDi QA) is a dataset created to be a multilingual counterpart of the NQ datasets but has a slightly different data collection process. Languages in this dataset were chosen to be very typologically diverse. The dataset contains 10 languages:

English, Arabic, Bengali, Finnish, Indonesian, Japanese, Kiswahili, Korean, Russian, Telugu, Thai.

All of them are from distinct language families and have a different linguistic structure such as word order, case markings, plurality systems, writing systems, and more. An overview of the size of the dataset can be seen in the table 2.2

Questions were created by human annotators. They were shown a short passage and were asked to write a question that is *not* answered by the passage and that they would like to be answered. The questions were created independently for each language therefore they can concern topics specific to native speakers of the language.

The questions were then paired with an Wikipedia article in corresponding language. Annotators were asked to select a passage in the article containing an answer if the article contained the answer. Than they were asked to select minimal answer. Most often the answer is couple of words but it can be even a whole sentence.

The TyDi QA dataset consists of 204 thousands instances out of which 37 thousands are 3-way annotated. The 3-way annotated instances are in dev and test sets. Rest of the dataset is only one-way annotated and is in train set.

Language	Train	Dev	Test
English	9,211	1031	1046
Arabic	23,092	1380	1421
Bengali	10,768	328	334
Finnish	15,285	2082	2065
Indonesian	14,952	1805	1809
Japanese	16,288	1709	1706
Kiswahili	17,613	2288	2278
Korean	10,981	1698	1722
Russian	12,803	1625	1637
Telugu	24,558	2479	2530
Thai	11,365	2245	2203

Table 2.2: TyDi QA statistics.

XOR- TyDi QA

XOR- TyDi QA [2] adapted TyDi QA for cross-lingual open-domain question answering. TyDi QA consists only of native questions and native contexts, but many questions do not have an answer in the native language, caused by smaller information availability and information asymmetry. Such questions are in TyDi QA labeled as unanswerable although they can be answered using for example more resourceful English Wikipedia.

The XOR- TyDi QA dataset contains question-answer pairs with an answer and in addition it has unanswered questions from TyDi QA answered in English.

The number of languages was reduced to 7, compared to 10 from TyDi QA, selected with regards to cost and availability of translators. From each language 5000 questions with no answer were randomly sampled and were translated into English. Annotators than selected contexts containing in English Wikipedia articles and selected minimal answer spans.

Languages selected for this dataset were:
English, Arabic, Bengali, Finnish, Japanese, Korean, Russian, Telugu.

2.4 Datasets Overview

The problematic aspect of working with multilingual systems is finding the right dataset, containing the required set of languages. In some cases, the most viable option could only be creating a new dataset, which is expensive and time demanding compared to using an existing one.

With the addition of extra languages, the number of ways how to define a task grows significantly and each dataset was created with a different purpose. The task considered in this work is a multilingual, open-retrieval, and cross-lingual system. Out of considered datasets, TyDi QA [7], XOR- TyDi QA [2], MLQA [20] and MKQA [21], the best suited was MKQA.

The models presented in this work were trained on a subset of the MKQA dataset. The limiting factors were support from Lucene library used for retrieval of passages and availability of pre-trained translation models. Overview of languages in datasets, availability of translation models, and Lucene retrieval library support can be seen in table 2.3.

Language	Lucene	Translator	Used	MKQA	MLQA	TyDi	XOR- TyDi
Arabic	yes	yes	yes	6758	5852	17866	3062
Bengali	yes	yes	no			3764	3022
Chinese (Hong kong)	no	yes	no	6758			
Chinese (Simplified)	yes	yes	no	6758	5641		
Chinese (Traditional)	no	yes	no	6758			
Danish	yes	yes	yes	6758			
Dutch	yes	yes	yes	6758			
English	yes	yes	yes	6758	12738	4741	
Finnish	yes	yes	yes	6758		7967	2980
French	yes	yes	yes	6758			
German	yes	yes	yes	6758	5029		
Hebrew	no	yes	no	6758			
Hindu	yes	yes	no		5425		
Hungarian	yes	yes	yes	6758			
Indonesian	yes	yes	no			6312	
Italian	yes	yes	yes	6758			
Japanese	yes	yes	yes	6758		6305	3033
Khmer	no	yes	no	6758			
Korean	yes	no	no	6758		3168	3415
Malay	no	no	no	6758			
Norwegian	yes	no	no	6758			
Polish	yes	yes	yes	6758			
Portuguese	yes	yes	yes	6758			
Russian	yes	yes	yes	6758		8193	2431
Spanish	yes	yes	yes	6758	5753		
Swahili	no	no	no			4879	
Swedish	yes	yes	yes	6758			
Telugu	no	yes	no			7983	1921
Thai	yes	yes	yes	6758		6800	
Turkish	yes	yes	yes	6758			
Vietnamese	no	yes	no	6758	6006		

Table 2.3: Dataset overview with the number of question-answer pairs, omitting pairs without minimal answer. The Lucene column indicates whether the implementation of an analyzer is present in the Lucene library. The Translator column is yes for languages that can be translated from or to using models trained by the University of Helsinki. For simplification, two multilingual models were chosen, translating from and to English, which was used as a pivot language. Models *opus-mt-mul-en* and *opus-mt-en-mul* can be found here: https://huggingface.co/Helsinki-NLP. The Used column states if language was used in the system described in this thesis.

Chapter 3

Pre-trained Neural Models

The task of machine reading comprehension in Question Answering is to take a question with a document containing an answer and generate an answer or select a span within the document, depending on the task definition. Some systems do not use a document and generate an answer directly given only a question [26].

The input text needs to be converted to a sequence of numbers for a model to understand it. Each number, called a token, represents a part of the input text. A token can represent a character, word, or sub-word. Using sub-word tokenization is the most common. The sub-word tokenization can be trained to create better-suited word splitting using for example Byte Pair Encoding [30] or unigram language model [17] algorithms.

Systems selecting a span are also called extractive systems. The task for extractive systems is to assign a probability to every possible span within a document and select the most likely one to contain the answer. This probability is conditioned on a question and a document.

Systems generating answers are called generative. Generative systems instead of assigning a probability to each span assign a probability to every token in its vocabulary conditioned on a question and optionally a document, but also on the previously generated token. The newly generated token is from the probability distribution over the vocabulary. The generation starts with a start token and ends when an end token is generated. For answer generation document is not required and can be omitted, however, experiments show that generation conditioned on document poses lower requirements on model size because information does not have to be encoded in model parameters [13].

Open-domain readers usually process more than one document because the document is not guaranteed to contain the information required to answer a question. By including multiple documents, the probability, that one of them contains the sought-after information rises.

3.1 Transformers

Neural networks with fixed input and output size are not very well suited for variable-length sequences. One solution to this problem is recurrent neural networks (RNN), such as *Long Short-Term Memory* [12] (LSTM) and *Gated Recurrent Unit* [6] (GRU). These networks take a sequence as an input and are able to output a sequence of the same length in such a way that each following item in the output sequence is conditioned on all previous inputs. This is achieved by processing each item from an input sequence separately and

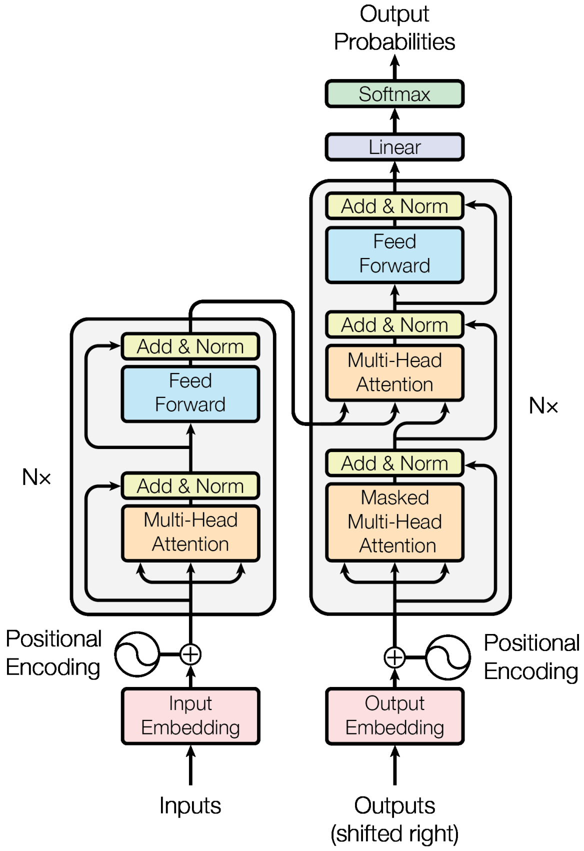

Figure 3.1: Encoder-Decoder structure of the tranformer architecture

along generating an output the network keeps a hidden state, which represents previously processed inputs. However, this means that the processing of a sequence is inherently sequential and therefore it cannot be parallelized as well as another type of neural network is called transformers, which makes processing longer sequences more time demanding. Another major problem is the numerical stability of a gradient, which made them trickier to train. Also, transfer learning does not work with RNNs as well as with transformer. RNNs however perform better when trained on smaller datasets.

Transformers [35] replaced RNNs in many areas and became new state-of-the-art architecture for language modeling. This model architecture introduced an attention mechanism that makes it possible to parallelize sequence processing while keeping output dependent on the whole input sequence. Transformers were initially created for machine translation so they use an encoder-decoder structure (figure 3.1). Encoder transforms input sequence into a representation of the input sequence and decoder given the representation and previous output tokens generates new output token.

Encoder block consists of the attention layer, residual connections, [11], layer normalization [3] and a fully connected linear layer. The input is passed to a self-attention and the output of self-attention is added to the input and normalized. The normalized output of self-attention is then passed to a fully connected linear layer. The output of the encoder block is the output of a fully connected layer added to the input of a fully connected linear layer, which is then normalized.

The encoder consists of multiple encoder blocks stacked one after another. Input is passed through the first encoder block and its output is the input of the following encoder block.

The decoder has a similar structure to the encoder but adds a cross attention layer after self-attention and before the linear layer. Self-attention is allowed to attend only to past outputs. Cross attention attends to the output of the encoder. Schema of encoder-decoder structure can be seen in figure 3.1.

3.1.1 Multi-Head Attention layer

Multi-Head attention is a variant of attention that processes a sequence by replacing each element by a weighted average of the rest of the sequence. It has multiple attention heads and the outputs are concatenated.

In order to keep output dependent on variable length input each head computes three vector representations of each token x_i called query q_i, key k_i and value v_i of length d, where i is an index of a token in an input sequence. Each representation is linearly transformed token embedding x_i by a corresponding transformation matrices:

$$\begin{aligned} q_i &= x_i^T \cdot W_Q \\ k_i &= x_i^T \cdot W_K \\ v_i &= x_i^T \cdot W_V, \end{aligned} \tag{3.1}$$

where W_Q, W_K and W_V are transformation matrices for query, key and value respectively, which are trained separately for each head.

The dot product between query q_i with all keys k_j^T produces scaling factors, which adjusts a corresponding value v_j in respect to the query q_i. Values are then adjusted by normalized scaling factors and summed. The result of this operation is a new representation z_i of input x_i.

This can be computed using matrix multiplication for each input token at the same time:

$$Z = \text{softmax}(\frac{Q \cdot K^T}{\sqrt{d_k}}) \cdot V \tag{3.2}$$

where $Q = (q_1, q_2, ..., q_N)$, $K = (q_1, q_2, ..., q_N)$ and $V = (v_1, v_2, ..., v_N)$ and $Z = (z_1, z_2, ..., z_d)$ is a vector of new embeddings and constant d is the dimension of the key vector. Softmax normalization ensures that the new representation is not growing every time it passes through attention and division by the square root of the dimension of the vectors stabilizes gradients.

Attention as described is positionally invariant order-independent. The order of inputs in an input sequence is passed to the attention via positional encoding. A position is encoded by a vector which is added to corresponding input embedding.

3.2 Text-to-Text Transfer Transformer

Text-to-Text Transfer Transformer [26] (T5) is a text-to-text transformer model trained on *Colossal Clean Crawled Corpus* (C4) consisting of hundreds of gigabytes of English text scraped from the web. The text-to-text framework allows the adaptation of the model to various tasks with the same architecture, objective, and training procedure.

The T5 model keeps the original Transformer architecture, with some small changes in layer normalization, positional embedding and placement of layer normalization.

Each model was pre-trained on the C4 dataset for 524,288 steps, with a maximum sequence length of 512 and a batch size of 128. The objective for pre-training was the masked language model [8] (MLM). In MLM objective the model is trained to predict missing or otherwise corrupted tokens in the input.

The Colossal Clean Crawled Corpus [1] (C4) was created from Common Crawl data [2] by authors of the model. Common Crawl is a publicly available web archive containing text from scraped web pages without markup and other non-text content. However the majority of the content of Common Crawl is not natural language but menus, error messages duplicated text, or generic text. To extract natural language data from Common Crawl following heuristics were used:

- Only retained lines were the ones that ended with either period, exclamation mark, question, or end quotation mark.
- Pages with fewer than 5 sentences and lines with fewer than 3 words were discarded.
- All pages containing „bad language“ were discarded.
- All pages with placeholder „lorem ipsum“ text were discarded.
- Pages with curly brackets were discarded, to remove pages with code.
- Discarding all duplicate three-sentence spans.

The final dataset contains around 750GB of reasonably clean natural English text.

3.3 Multilingual Text-to-Text Transfer Transformer

Multilingual Text-to-Text Transfer Transformer [37] (mT5) is a multilingual version of the T5 model. It was pre-trained on Common Crawl based dataset containing 101 languages (mC4).

The mT5 model is based on T5.1.1 architecture which improves original T5, with some minor changes in architecture and pre-training procedure. To avoid over-fitting or under-fitting during pre-training the languages were sampled according to probability $p(L) \propto |L|^{\alpha}$, where $p(L)$ is the probability of sampling text from a given language during pre-training, $|L|$ is the number of examples in the language and α is a hyperparameter set to $\alpha = 0.3$ for this model.

The mC4 dataset is was created similarly to the C4. Only one filtering heuristics was changed, instead of checking if the line ends with a punctuation mark, line length was used. Filtered pages were the group by language and all languages with 10,000 pages or more were included in the corpus.

The model was evaluated on multilingual QA datasets, however, the evaluation task was not open domain but a question was presented along with evidence containing an answer to the question. The model was fine-tuned on English data and translations in all target languages which helped increase performance slightly.

[1] The C4 dataset is available at https://www.tensorflow.org/datasets/catalog/c4.

[2] commoncrawl.org

The smallest version of the model mT5-Small ($\approx$ 300M parameters) achieved a 38.8% exact match on the MLQA dataset and 34.0% on TyDi dataset. While the largest model mT5-XXL ($\approx$ 13B parameters) achieved 58.2% and 67.8% on MLQA and TyDi datasets respectively.

3.4 BERT

Bidirectional Encoder Representations from Transformers (BERT) [8] is a model pre-trained on unsupervised tasks using a large amount of data. BERT can be fine-tuned for a certain task with minimal changes to the original architecture. It consists only from *encoder* architecture from *Transformers* but with a different number of layers. Using only the encoder part from the Transformers allows the model to transform the input sequence into a different vector space, for example in the masked language model each token is transformed into a space of probability distribution over vocabulary.

There are two versions, $BERT_{BASE}$ and $BERT_{LARGE}$ with 12 and 24 encoder layers respectively.

Pre-training of the model was done on two unsupervised tasks.

- Masked Language Model

  Language models are trained to maximized the joint probability of a sequence of words $P(w_1, w_2, ..., w_i)$, where $w_1, w_2, ..., w_i$ are words. Standard conditional language models can only be trained in one direction, which means they factorize the joint probability:

  $$P(w_1, w_2, ..., w_i) = P(w_i|w_1, w_2, ..., w_{i-1})P(w_{i-1}|w_1, w_2, ..., w_{i-2})...P(w_2|w_1)P(w_1). \tag{3.3}$$

  Bidirectional models are not able to use such factorization because of bidirectional dependencies, they have access to previous and next words at the same time. Instead of autoregressive factorization, a masked language model is used, sometimes referred to as the Cloze task [33]. From input sequence 15% of tokens are masked and the model tries to predict what was the word which was masked out.

- Next Sentence Prediction

  The goal of the next sentence prediction is to train a model to understand the relationship between two consecutive sentences, which could be helpful for tasks such as Question Answering [31]. The model was trained on pairs of sentences and was predicting whether the next sentence was following the first in the original text or not.

  The two sentences are separated with $[SEP]$ token and also learned segmentation embedding is added to token embedding. A similar setup is then used for Question Answering.

For question answering BERT is fine-tuned to make a prediction of how likely each input token corresponds to the beginning and end of an answer span. This probability distributions are computed using a linear transformation of last hidden states of the encoder into a vector of logits and normalizing it using softmax.

Chapter 4

Information Retrieval

Reading comprehension solves the problem of obtaining an answer, given evidence. But the task in Open-Domain QA is formulated in such a way that the evidence (the golden document) containing information to answer a question is not known beforehand. Here comes Information Retrieval into play. For a given question the golden document has to be found in a large collection of documents for a reader to be able to extract the answer. Retriever performs ranking of the documents based on their relevance to a given question and there is no guarantee that the best-ranked document is the golden document. To counteract that, multiple documents can be retrieved and all of which are then evaluated by the reader and possibly re-ranked.

There are multiple approaches for retrieving relevant information. The first approach discussed in this thesis is based on a probabilistic relevance framework. Specifically, the text retrieval algorithm Okapi BM25, which is commonly used for question answering as a benchmark retriever. The second approach utilizes neural networks to transform documents and questions to a lower-dimensional space where their relevance can be expressed as a distance measure. The neural approach was developed specifically for question answering.

An example of a different approach is a graph-based Path Retriever [1], which sequentially retrieves each evidence document, given the history of previously retrieved documents to form several reasoning paths in a graph of entities.

4.1 Okapi BM25

Okapi BM25 [28] is an *TF-IDF*-like (Term Frequency - Inverse Document Frequency) algorithm developed by Robertson et al. in 1994. It was developed for ranking documents based on their lexical overlap and does not consider word order.

There multiple implementations of BM25 and they slightly vary in terms of how documents are scored [14]. The discussed implementation is from Lucene indexing and searching framework [25].

Lucene implementation uses analyzers before scoring documents, which lematises the words and removes stop words. For example question “Who invented the first airplane that didn’t fly?” is converted to terms „who“, „invent“, „first“, „airplane“, „didn’t“, „fly“.

Document score for a question is computed as:

$$score(D,Q) = \sum_{q \in Q} IDF(q) \cdot \frac{f(q,D) \cdot (k_1 + 1)}{f(q,D) + k_1 \cdot (1 - b + b \cdot \frac{|D|}{\frac{1}{N}\sum_i^N |D_i|})}, \tag{4.1}$$

where q is a term from a question Q, D is a document, $f(q, D)$ is a frequency of term q in document D, k_1 and b are hyper parameters. $IDF(q)$ is inverse document frequency (IDF) of the term q is

$$IDF(q) = ln\left(\frac{N - n(q) + 0.5}{n(q) + 0.5} + 1\right), \tag{4.2}$$

where N is number of documents $n(q)$ is total number of documents containing term q. IDF is putting more importance on terms that are rare in corpus and penalizes the most frequently occurring terms, which tend to be less informative.

The k_1 hyperparameter alters how frequent a term has to be in a document to saturate. The larger the k_1 parameter is, the more occurrences of a term have to be in a document for a term to contribute to the final score. In general, for longer documents, for example, books, larger k_1 tends to perform better and vice versa for shorter documents smaller k_1 is more suitable. Usually, the range of values the k_1 is set to be $k_1 \in< 0, 3 >$, which was experimentally shown to perform well[1], but technically it can be any positive number.

The b parameter in contrast to the k_1, which parameter sets how many times a term has to be in a document without taking into account its length, can be used to penalize documents that are too long and prefers shorter documents. The b parameter has to be in the range $b \in< 0, 1 >$ and it was shown that the optimal value is in $b \in (0.3, 0.9)$ [32].

4.2 Neural Information Retrieval

This is a second approach to Information Retrieval for QA. The basic idea is to use two neural networks to process documents and questions independently. The networks encode a text to fixed-length vector representation and are trained to minimize a similarity between the vector representation of a question and a golden document. The similarity between a vector representation of a question and documents can be then interpreted as a relevance score and the closer the document is to the question, the more relevant it is.

If we have two neural networks with parameters Θ_Q and Θ_P

$$\begin{aligned} embed_q &= network_{\Theta_Q}(q) \\ embed_d &= network_{\Theta_P}(d) \end{aligned} \tag{4.3}$$

where $embed_q$ and $embed_d$ are embeddings of a question and a document respectively, which are vectors of a fixed dimension. The distance of these vectors is computed using a similarity function:

$$score = sim(embed_q, embed_d) \tag{4.4}$$

where $score$ is relevance of a document d to a question q and sim is similarity measure.

For similarity measure, the inner product is usually used because of its simplicity, but other measures like euclidean distance could be used.

4.2.1 Open-Retrieval Question Answering

Open-Retrieval Question Answering [19] (ORQA) is a joint retriever and reader model. The retriever is using BERT-based encoders to transform questions and documents to

[1] https://www.elastic.co/blog/practical-bm25-part-3-considerations-for-picking-b-and-k1-in-elasticsearch

fixed-length embeddings. The model used for reading comprehension in this paper is also the BERT model [8] for extractive question answering. However only the retriever part is described in more detail with pre-training procedure and subsequent fine-tuning, omitting the details on the reader.

This was the first model shown to outperform BM25 in information retrieval for question answering.

The ORQA retriever processes input sequence with the BERT model. The model creates a representation for each input token, which is the last hidden state of the model. The first token in an input sequence is always a special $[CLS]$ token regardless of the task. The ORQA retriever utilizes only the last hidden state corresponding to the $[CLS]$ token to compute a vector representation of a document or a question as shown in equation 4.5.

$$\begin{aligned} h_q^T &= W_q \cdot \mathrm{BERT_Q}(q)\,[\mathrm{CLS}] \\ h_d^T &= W_d \cdot \mathrm{BERT_D}(d)\,[\mathrm{CLS}] \end{aligned} \tag{4.5}$$

where $\mathrm{BERT}(x)\,[\mathrm{CLS}] \in d_h$ denotes the last hidden state corresponding to the [CLS] token and $W \in \mathbb{R}^{d \times d_h}$ is a transformation matrix. $h_q \in \mathbb{R}^d$ and $h_p \in \mathbb{R}^d$ are dense vector representations of a question q and a document d respectively. Similarity between a question and a document then is

$$\mathrm{sim}(q, d) = h_q^T \cdot h_p \tag{4.6}$$

Retriever was pre-trained using Inverse Cloze Task (ICT), which is an unsupervised pre-training procedure. Standard Cloze Procedure [33] is predicting masked-out word based on the surrounding context. It was developed as a psychological tool for measuring the effectiveness of communication. For example "Chickens cackle and ...... quack.", if the word "ducks" was guessed correctly a cloze unit is scored for closing the gap in the language pattern.

In Inverse Cloze Task the goal is to predict the context of a sentence given the sentence. The idea is that there is extra information contained in an answer compared to a question.

$$P_{ICT}(d|q) = \frac{\exp(\mathrm{sim}(d, q))}{\sum_{d' \in \mathrm{BATCH}} \exp(\mathrm{sim}(d', q))} \tag{4.7}$$

where q is a random sentence, d is a surrounding text and BATCH is a set containing a positive document and multiple negative documents. The sentence is not removed from the context 10% of the times in order for the model to be able also to match words.

After pre-training on ICT the retriever is fine-tuned jointly with the reader. Only question encoder and reader are being trained as zero-shot evidence retrieval performance is expected to be sufficient. Fine-tuning objective comprises of two losses. The first is the log-likelihood of spans containing an answer in the top k documents marginalized over the documents in the equation 4.8.

$$L_{full}(q, a) = -\log \sum_{d \in \mathrm{TOP}(k)} \sum_{s \in \mathrm{GT}(d,a)} P(d, s|q) \tag{4.8}$$

where a is answer string, d is a document from set of k documents with highest score $\mathrm{TOP}(k)$ and $s \in \mathrm{GT}(d)$ is span within a document d that exactly matches the answer a.

The probability of a span being an answer span is

$$P(d, s|q) = \frac{\exp(S(d, s, q))}{\sum_{d' \in \text{TOP}(k)} \sum_{s' \in d'} \exp(S(d', s', q))} \tag{4.9}$$

where d is a document, s is span, q is question string and $S(d, s, q) = \text{sim}(d, q) + S_{\text{read}}(d, s, q)$ is sum of retriever and reader scores.

The second part of the objective L_{early} in the equation 4.10 is similar to the pre-training loss. It differs in the set of documents, during fine-tuning the set of documents BATCH contains top c documents instead of positive and negative documents, where c is larger than k, because it is relatively cheap to compute.

$$L_{early}(q, a) = -\log \sum_{d \in \text{TOP}(c), \text{GT}(d,a)} P_{early}(d|q) \tag{4.10}$$

where the probability P_{early} is:

$$L_{early}(q, a) = \frac{\exp(S_{retr}(d, q))}{\sum_{d' \in \text{TOP}(c)} \exp(S_{retr}(d, q))} \tag{4.11}$$

where c is the number of selected documents, it is larger than k in the equation 4.9, because this loss is relatively inexpensive to compute.

The final objective is a sum of the two losses

$$L(q, a) = L_{early}(q, a) + L_{full}(q, a). \tag{4.12}$$

4.2.2 Dense Passage Retriever

Dense Passage Retriever [15] (DPR) is based on the ideas from ORQA and improves it by showing that computationally expensive ICT pre-training is unnecessary and that fine-tuning paragraph encoder could also be beneficial.

DPR uses a similar setup to ORQA for encoders. Two BERT encoders are used, one for questions and one for documents and score is computed from last hidden state representation of the [CLS] token.

Both encoders are fine-tuned directly to maximize similarity between a representation of a question and a representation of a positive paragraph, by minimizing loss function:

$$L(q, d^+, d_1^-, ..., d_n^-) = -\log \frac{\exp(\text{sim}(q, d^+)}{\exp(\text{sim}(q, d^+) + \sum_{i=1}^{n} \exp(sim(q, d_i^-))} \tag{4.13}$$

where d^+ is a positive paragraph, the paragraph containing an answer, for a question q and $d_i^-, i = 1, ..., n$ are negative paragraphs, which do not contain an answer. Selecting negative paragraphs is not as trivial, because of a large pool of negative paragraphs to choose from. The best performance achieved the combination of positive paragraphs from other questions in the same mini-batch, to increase performance, and one negative paragraph retrieved by BM25.

4.3 Multilingual Retrieval

Not many attempts were made in multilingual retrieval.

4.3.1 Pivot Through English

Pivot Through English [22] utilizes a different approach to multilingual question answering. It uses high resource language (English) to answer questions in low resource languages using cross-lingual pivots.

This approach is based on finding a most similar question in high resource language which is paired to an answer in the high resource language and then is translated to the low resource language of the original question. The method of finding the closest question in another language was called Reranked Multilingual Maximal Inner Product Search (RM-MIPS). The low resource question is embedded into latent vector space. Maximal inner product search is used to find the approximate closest question in a high resource language. Then cross encoder is used to rerank high resource queries.

The selected high resource query is then matched with the corresponding answer from a database and translated back to the low resource langue using either machine translation or WikiData knowledge graph.

4.3.2 XOR- TyDi QA

In the paper introducing XOR- TyDi QA dataset [2] two baseline systems for multilingual retrieval were presented.

The first system was using English as a pivot language. Questions are translated into English, and used to retrieve passages only from English Wikipedia. The authors compared results of different retrieval approaches, namely BM25, BM25 with neural reranker (Path Retriever [1]) and end-to-end neural retriever (DPR described in section 4.2.2) and different translation methods, Human translations, Google Machine Translation and a machine translation model trained by the authors.

The second system was using DPR, which is adjusted to the multilingual setting by using multilingual BERT encoders. Multilingual DPR is also used to retrieve only from English Wikipedia, but without translating the question.

Comparison of the retrieval methods and use of translation methods of question to English can be seen in table 4.1. The best performance achieved DPR with human translations. Using machine translation under-performed human translation, in the case of the model trained by the authors the difference in performance was significant. Also, BM25 retriever has achieved much poorer results compared to the other two methods. Multilingual DPR is comparable to BM25, however, it can be suspected this could be due to lack of training data.

4.4 Concurrent Work

Concurrently with this work, another work is created on the topic of Open-Domain Question Answering in Czech language by student Jonáš Sasín [29] also utilizing BM25 algorithm for information retrieval. The corpus used for passage retrieval is Czech Wikipedia. The reading comprehension is done by an extractive model, with a focus on the comparison of the English model with machine translation and a multilingual model fine-tuned for the Czech language.

	Human			GMT		MT		
	DPR	PATH	BM25	DPR	PATH	DPR	PATH	Multi DPR
Ar	69.1	**70**	41.6	65.8	63.3	51.6	51.6	45
Bn	**82.8**	82	57	83.2	78.9	58.4	64.8	49.2
Fi	**72.8**	70.2	43.7	65.8	64.1	64.1	59.5	47
Ja	**66.2**	63	38.8	57.9	52.3	48.9	41.7	32.4
Ko	**69.7**	63.6	43.8	63.8	54	46.7	37.6	39.4
Ru	61.8	**63.7**	35.2	58.9	56.5	46.8	38.1	40
Te	**69.5**	64.1	44.6	63.6	62.5	22.7	18.1	47.4
Avg.	**70.3**	68.1	43.5	65.6	61.7	48.5	44.5	42.4

Table 4.1: Comparison of methods on XOR-QA retrieval task measured by recall (fraction of the questions for which the minimal answer is contained in the top 5000 selected tokens). Human, GMT, and MT are translation methods, standing for human translations, translations from Google's online machine translation service, and translations from the machine translation model trained by the authors respectively. Multi DPR denotes multilingual DPR and PATH denotes Path Retriever.

Chapter 5

Proposed Multilingual Systems

In this chapter, are proposed multiple multilingual systems. Two main approaches are suggested, a monolingual English system in conjunction with translator and multiple variants of a native multilingual system.

In this work, it is assumed that a language of an answer is always the same as the language of the question The reason being is that when someone asks a question, they expect an answer to be in the language they asked the question in. However, the same does not necessarily apply to evidence providing needed information to answer the question. The evidence for multilingual models can be in an arbitrary language.

5.1 Readers

In this chapter, multiple multilingual open-domain question answering systems are presented. All of the proposed systems are based on generative T5 model [26]. Advantages of generative models compared to extractive models, such as multilingual BERT [8], in the multilingual setting is that an answer span annotation is not required. Annotation of an answer string is sufficient, which is lacking in some multilingual datasets. The disadvantage of generative models is that in general, they tend to have a larger number of parameters and subsequently require more data to fine-tune. The size difference is due to the decoder, which is not present in extractive models.

All the systems presented are using the Fusion-in-Decoder method [13] (FiD). The T5 model relies on packing the information required to answer a question in its parameters. The Fusion-in-Decoder is a method of utilizing a retriever with a generative model to improve performance.

The FiD reader takes as input the question with title and passage separated by special tokens. Each retrieved passage with its title and the question is processed by the encoder independently and the representations are fused in the decoder. The input of encoder for a question Q and passage and a passage $P_i, i \in 1, ..., k$ and title of the passage T_i from the top-k retrieved passages is constructed as:

$< \text{question} > Q < \text{title} > T_i < \text{passage} > P_i$,

where $< \text{question} >$, $< \text{title} >$ and $< \text{passage} >$ are special tokens. The fusion is done by concatenating the question and passage representations from the encoder and passing the concatenated hidden states into the decoder as in figure 5.1.

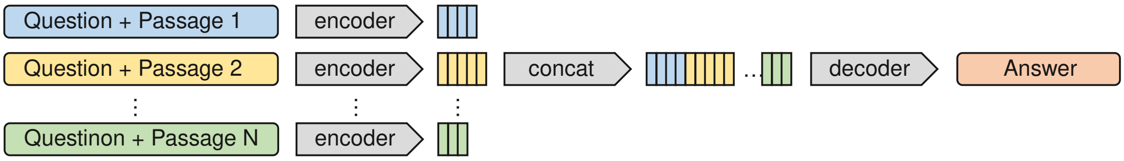

Figure 5.1: Encoding and decoding in Fusion-in-Decoder.

5.2 Retrieval

The most common source of evidence for Open-Domain QA systems is Wikipedia. Wikipedia is a free, multilingual encyclopedia maintained by volunteer contributors. It is the largest encyclopedia, which makes it a viable source of information. Availability of Wikipedia in multiple languages is also very useful for obtaining multilingual evidence, however, different language Wikipedias are being extended independently from each other. This means that each Wikipedia has a different size. There are more than 300 Wikipedias in different languages, the largest being English Wikipedia containing over 6 million articles on different topics, but more than half of Wikipedias contain less than 10 000 articles.

Multilingual systems are able to utilize multilingual corpus. For this reason, there are multiple strategies for selecting evidence.

Using only monolingual English corpus. English is the default choice in research for publishing and subsequently for developing NLP systems. The amount of information in English corpora is for many tasks sufficient but is restricted to an English-centric knowledge base.

Using evidence in the language of the question. Using evidence in the language of the question is beneficial for utilizing the knowledge that already exists in a given language. This approach is better suited for high resource languages and for question on topics unique for the language.

Using multilingual evidence. Using multilingual corpus and searching for evidence in multiple languages is potentially the most promising as it mitigates the problems of the previous approaches. However it also substantially increases the size of the corpus and therefore also the number of irrelevant documents.

The proposed systems use either English monolingual retrieval or multilingual retrieval. The size of some monolingual corpora, such as Thai Wikipedia with only approximately 140 thousands articles, make them less feasible to be used as a single source of information for a monolingual system. But using them as an additional source of information in the multilingual corpus can be useful, because it is created independently from other Wikipedias and can contain articles on different topics or different additional information to the same topic.

5.2.1 Monolingual Retrieval

The first strategy is a English monolingual retriever in combination with a multilingual model. The passages from English Wikipedia are scored by the BM25 using translated questions from original language to English and top-k passages are retrieved.

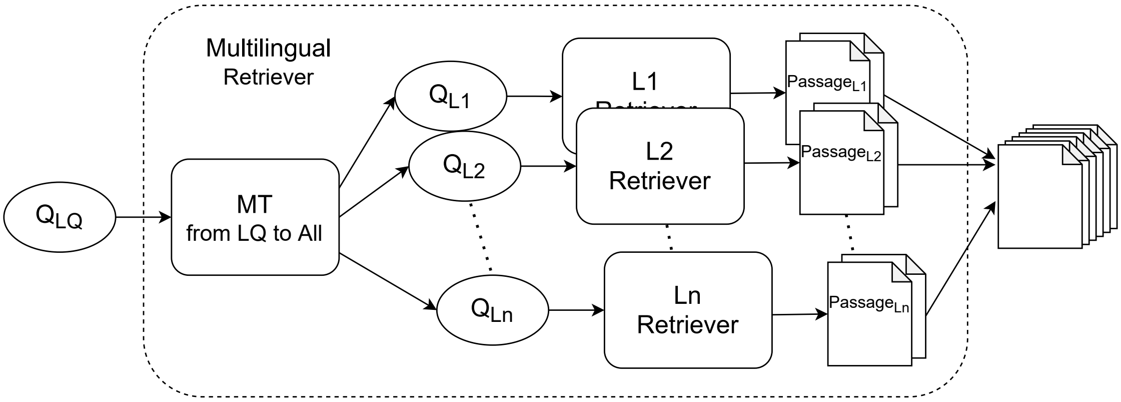

Figure 5.2: Multilingual passage retrieval, retrieving passages for languages $\mathrm{L1, L2 \ldots Ln}$, given question $\mathrm{Q_{LQ}}$ in language LQ.

This strategy is used by system with pivot language described in section 5.3.1 and OQER system described in section 5.3.4. Many state-of-the-art monolingual systems rely on English Wikipedia as the sole source of evidence. As discussed in the next chapter 6.1.2 English Wikipedia is significantly larger compared to other Wikipedias.

5.2.2 Multilingual Retrieval

The passage selection strategy employed in systems with multilingual retrieval is retrieving passages separately for each language using translated questions. Afterward for each language top-k are selected from each language based on BM25 retriever score. In practice, the number of passages k is smaller compared to monolingual retrieval because the total number of passages is multiplied by the number of languages. Figure 5.2 illustrates multilingual passage selection.

The disadvantage of the proposed method of using the BM25 algorithm in the multilingual setting is that scores are not comparable between languages. This makes it is harder to balance the exploration of low resource languages and-exploitation languages. For example, if one document is selected from each language when there is only one document in one of high resource language containing an answer, this document has to be ranked first among all documents from that language to be selected. But if only the documents from the language containing the answer were considered, it is much more likely that this document will appear in the selected set of documents. A possible solution could to this problem would be a use of reranker, but this is left for further research.

5.3 Proposed Models

In this thesis, multiple systems are proposed with different approaches to passage retrieval and to indicating what is the language of the answer. The base for multilingual models is a multilingual T5 transformer described in section 3.3 and the BM25 passage retrieval. The passages are passed to the reader using the FiD method. Note that the multilingual model can rely only on a monolingual corpus or can take advantage of a multilingual corpus.

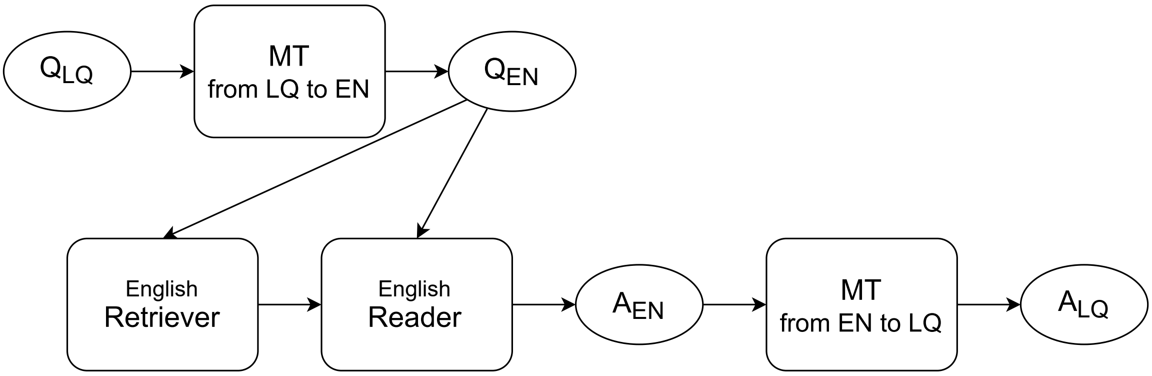

Figure 5.3: Schema of model with pivot language. The input question Q_{LQ} is translated to English Q_{EN}. The reader generates the answer A_{EN}, which is translated to the language LQ and answer A_{LQ} is returned.

5.3.1 Model with Pivot Language

This approach is utilizing a monolingual English language model, which is able to exploit a much larger English training dataset and does not require a multilingual training dataset. However, for multilingual inference, machine translation needed to translate its inputs to English and outputs from English to the target language.

During inference, the original question Q_{LQ} in language LQ is translated to the pivot language. The translated question Q_{EN} is then used for scoring passages in English corpus with the BM25 algorithm. Afterward, top-k English passages are selected and are passed to the model. The model generates an answer in English A_{EN} which is translated back to language LQ and the translated answer A_{LQ} is returned. The schema of the model with pivot language can be seen in figure 5.3.

5.3.2 Multilingual Model with Tag

When using multilingual BM25 retrieval a question has to be translated into all other languages. This makes it reasonable to provide the translated questions to the model paired with the same language passage. However, the system does not have the information on which question was the original to decide what should be the target language.

To solve this problem I propose a solution using a special string indicating the target and add it to each question. The original input question Q_{LQ} in language LQ is translated into each language $Li, i \in 1, ..., n$ from the n supported languages. Each translated question is pre-pended special string Tag_{LQ} indicating that the original question was in language LQ. The input sequence for a question in language Li is constructed as

$<$ question $>$ $\text{Tag}_{LQ}\, Q_{Li}$ $<$ title $>$ T_{Li} $< passage >$ P_{Li},

where P_{Li} is a passage in language Li and T_{Li} is the title of the passage.

For example if a question “Welches ist der tiefste Punkt der Erde?” in German and its translation to English “What is the lowest point of the earth?” the input sequences would be constructed as followed:

<question> »de« Welches ist der tiefste Punkt der Erde? <title> Totes Meer <passage> Das Tote Meer ist in einen nördlichen und einen südlichen Teil getrennt. Seine Wasseroberfläche...

<question> »de« What is the lowest point of the earth? <title> Extreme points of Earth <passage> This is a list of extreme points of Earth, the geographical locations that are higher or lower in elevation than...

The proposed solution to this problem is inspired by the solution employed in machine translation for a similar problem. Machine translation models trained by Helsinki-NLP capable of translating to multiple languages are using special string prefix to indicate the target language. A similar technique is employed to indicate the target language of the answer. The schema of system with tag can be seen in figure 5.4.

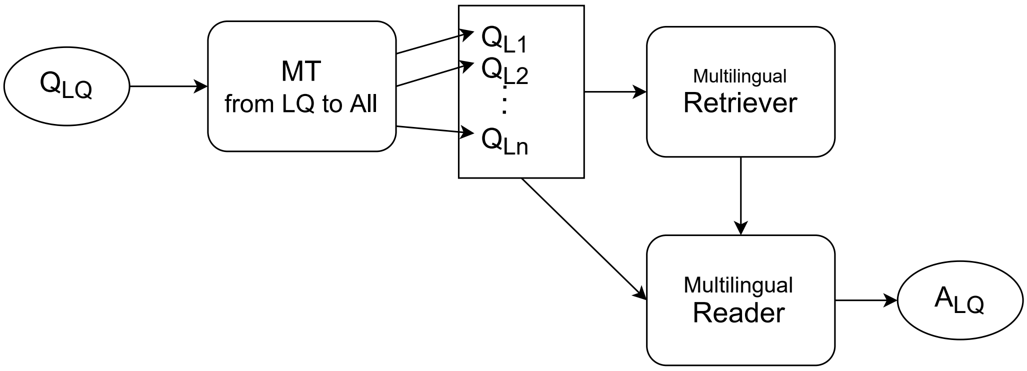

Figure 5.4: Schema of system with multilingual retrieval and translated questions with language tags. The input Q_{LQ} in language LQ is translated to all languages $L1, L2, ..., Ln$. The translated question $Q_{L1}, Q_{L2}, ..., Q_{Ln}$ are passed to the reader, which generates an answer A_{QL}.

5.3.3 Original Question with Multilingual Retrieval

Original Question with Multilingual Retrieval (OQMR) uses the same retrieval method as the previously described model. However, the reader does not utilizes translated questions. The input sequences consist of multiple passages in different languages paired with the original question. The schema of the system can be seen in figure 5.5. Not using question translations and using only original questions removes additional noise introduced by a translator, but the encoder encodes two different languages at the same time.

For the question "Welches ist der tiefste Punkt der Erde?" the input sequences for English and German passages would be constructed as:

<question> Welches ist der tiefste Punkt der Erde? <title> Totes Meer <passage> Das Tote Meer ist in einen nördlichen und einen südlichen Teil getrennt. Seine Wasseroberfläche...

<question> Welches ist der tiefste Punkt der Erde? <title> Extreme points of Earth <passage> This is a list of extreme points of Earth, the geographical locations that are higher or lower in elevation than...

5.3.4 Original Question with English Retrieval

Original Question with English Retrieval (OQER) combines monolingual retrieval with the multilingual reader. As discussed in section 5.2.2, the method of multilingual retrieval can be disadvantageous, when only a subset of monolingual corpora contains the required information.

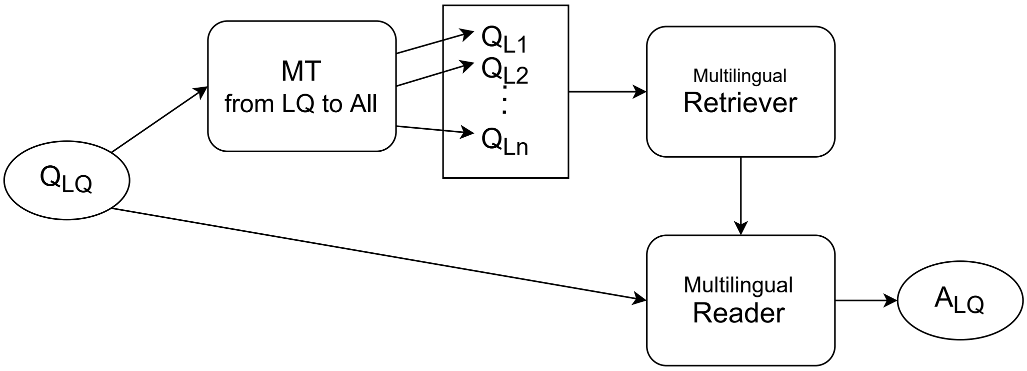

Figure 5.5: Schema of system with multilingual retrieval and original question on input. The input question Q_{LQ} is passed directly to the reader with multilingual passages. The generated answer A_{LQ} is in the languages of the question LQ.

This approach utilizes only the English corpus, which is the biggest monolingual corpus. The input sequences consist of top-k English passages paired with the original question and the system is trained to generate an answer in the languages of the question. This approach is illustrated in figure 5.6.

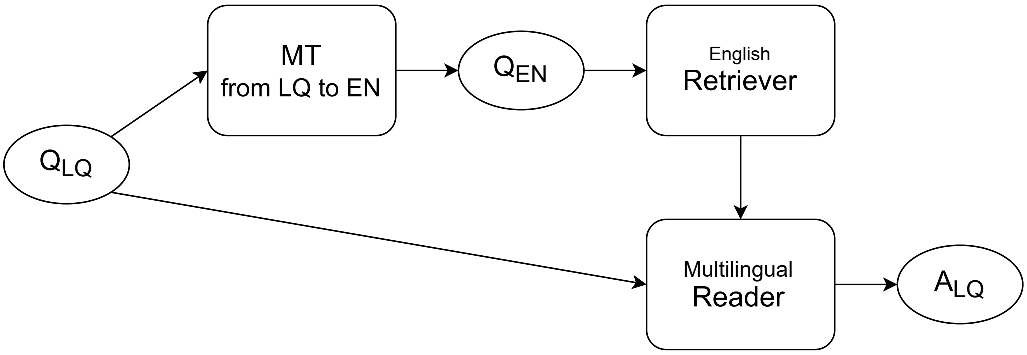

Figure 5.6: Multilingual system with English retrieval. The input question Q_{LQ} is passed directly to the reader with English passages. The generated answer A_{LQ} is in the languages of the question LQ.

For example for the question "Welches ist der tiefste Punkt der Erde?" the input sequences with only English passages would be look like:

<question> Welches ist der tiefste Punkt der Erde? <title> Extreme points of Earth <passage> This is a list of extreme points of Earth, the geographical locations that are higher or lower in elevation than...

<question> Welches ist der tiefste Punkt der Erde? <title> Dead Sea <passage> The lake's surface is 430.5 metres (1,412 ft) below sea level, making its shores the lowest land-based elevation on Earth...

Chapter 6

Experiments

This chapter contains experimental setup and evaluation of proposed systems on different tasks.

6.1 Experimental Setup

I selected Python programming language [1], as the main implementation language because it's popular in the machine learning community and many machine learning libraries and frameworks contain bindings for this language.

Since the trend in NLP is to use transfer learning and pre-trained models, I used the Transformers python library [36] with a large collection of pre-trained models running on Pytorch [24] backend.

The implementation of the reader is based on the implementation of R2-D2 model [10]. The R2-D2 source code was released and is available here [2].

For the Information Retrieval algorithm BM25, I used Apache Lucene implementation.

6.1.1 Language Selection

The language selection was subordinate to three constraints:

1. Dataset availability.
2. Lucene analyzers support.
3. Availability of machine translation model.

A summarization of the constraints can be seen in table 2.3. In total 17 languages were meeting the constraints. For brevity, ISO-639-1 codes are used to refer to the respective language. Selected languages along with their classification, writing systems, and their ISO-639-1 codes are shown in table 6.1.

6.1.2 Wikipedia Pre-Processing

The retrieval was done over a multilingual corpus, created from Wikipedia dumps [3]. Each dump was processed, as described in Dense Passage Retrieval for Open-Domain Question

[1] https://www.python.org/doc/

[2] The R2-D2 source code:https://github.com/KNOT-FIT-BUT/scalingQA.

[3] Wikipedia dumps can be downloaded here: https://dumps.wikimedia.org/.

Language family	Language Group	Writing system	Language	ISO code
Indo-European	Germanic	Latin	Danish	da
		Latin	Dutch	nl
		Latin	English	en
		Latin	German	de
		Latin	Swedish	sv
	Romance	Latin	French	fr
		Latin	Italian	it
		Latin	Portuguese	pt
		Latin	Spanish	es
	Slavic	Latin	Polish	pl
		Cyrillic	Russian	ru
Uralic	Finno-Ugric	Latin	Finnish	fi
		Latin	Hungarian	hu
Semitic	Central Semitic	Arabic	Arabic	ar
Japonic	Japonic	Kana/Kanji	Japanese	ja
Kra-Dai	Tai	Thai	Thai	th
Turkic	Oghuz	Latin	Turkish	tr

Table 6.1: Selected languages with their classification, writing system, and ISO-639-1 language codes.

Answering [15], to create 100 word long passages. The passage splits from the DPR paper were made available by authors, so the same English Wikipedia data were used in this work.

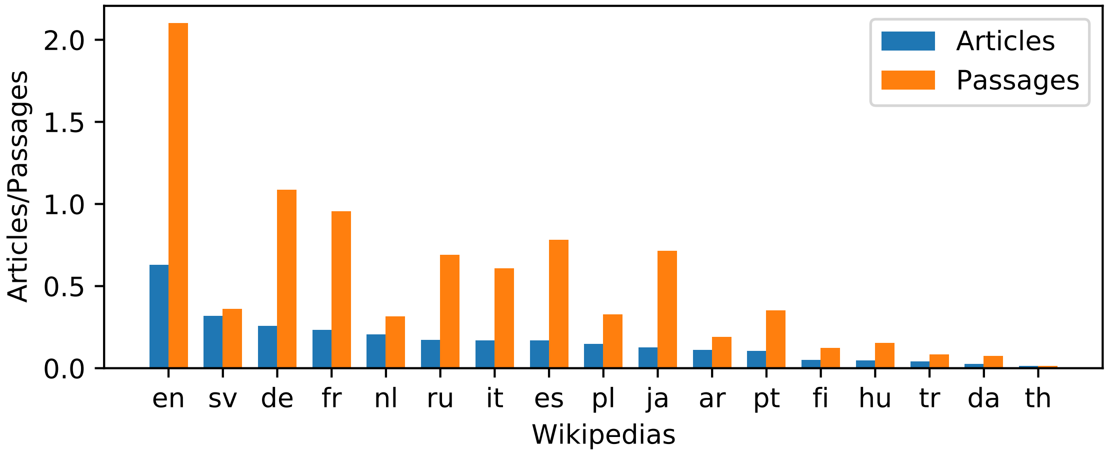

Figure 6.1: Number of Wikipedia articles and number of passages after processing for different languages.

The non-English Wikipedia dumps were from Jan. 3, 2021, and were processed analogously. From dumps clean text was extracted, removing semi-structured data such as tables, disambiguation pages, and lists, using pre-processing code release in DrQA [5]. The reminding clean text was then split into 100 word long passages. If the remainder of an article was shorter than 100 words, the reminder was filled in from the beginning of the article. The number of articles and final passages for each language is shown in figure 6.1 and table 6.2.

	Articles	Passages
English	6 294 702	21 015 324
Swedish	3 196 626	3 612 145
German	2 573 726	10 870 498
French	2 326 064	9 559 330
Dutch	2 054 338	3 154 149
Russian	1 721 794	6 898 313
Italian	1 691 467	6 072 981
Spanish	1 681 092	7 809 593
Polish	1 472 246	3 283 254
Japanese	1 267 288	7 141 013
Arabic	1 115 094	1 894 884
Portuguese	1 066 054	3 523 032
Finnish	508 881	1 225 855
Hungarian	487 415	1 542 793
Turkish	401 588	845 427
Danish	266 599	745 507
Thai	139 618	151 351

Table 6.2: Number of articles and passages in each Wikipedia, ordered by the number of articles.

to English		from English	
Language	BLEU	Language	BLEU
Italian	54.8	Swedish	42.7
Portuguese	51.4	Danish	42.5
Swedish	51.4	Dutch	37.6
Danish	50.5	Spanish	36.7
Spanish	47.9	Italian	33.9
Dutch	47.9	Portuguese	31.6
French	45.1	French	31.5
Russian	42.7	German	25.2
Polish	41.7	Turkish	24.1
Turkish	40.5	Polish	22.7
German	39.6	Russian	22.2
Thai	36.3	Finnish	18.5
Finnish	35.8	Hungarian	17.1
Hungarian	33.2	Japanese	7.8
Arabic	26.4	Arabic	6.3
Japanese	10.0	Thai	2.2

Table 6.3: BLEU scores the model translating to English and the model translating from English on Tatoeba corpus. Languages are ordered by their score.

6.1.3 BM25

Passages were retrieved using the BM25 [28] ranking algorithm described in section 4.1.

Lucene's implementation of the BM25 ranking algorithm was used. It's free parameters k_1 and b I set to 0.92 and 0.22 respectively. These values were the result of optimization with the hyperparameter optimization tool [4]. Compared to commonly used values $k_1 = 1.2$ and $b = 0.75$ after optimization the F1 score improved on the MLQA Test dataset by more than 1 point and Exact Match improved by 0.8 points.

6.1.4 Machine Translation

To reduce the need for a translator model for each language pair, two multilingual models were used. When not translating from or to English each translation is done in two steps:

- Translation to English from the source language.
- Translation from English to the target language.

Each of the steps was done by a separate model. When translating to or from English one of the two steps is redundant and is skipped, also when the source language and target language is the same both of the steps are skipped and the result is identical.

The translator models used were trained by the Helsinki University NLP team [34] and are available in Transformers library [4]. The models were trained on Opus open parallel corpus [5]. The trained models are relatively compact with less than 300Mb of parameters. The performance overview of the two models in different languages can be seen in table 6.3.

[4] Machine translation models: https://huggingface.co/Helsinki-NLP/opus-mt-en-mul, https://huggingface.co/Helsinki-NLP/opus-mt-mul-en.

[5] Opus parallel corpus: https://opus.nlpl.eu/.

6.1.5 Evaluation

Model Evaluation

Model predictions were evaluated using the standard exact match metric (EM) [27]. Exact match metric measures the ratio between correct answers and the total number of answers. A correct answer generated by a system is an answer that is the same as one of the annotated ground-truth answers. Predicted answers and the set of ground truth answers were normalized before comparison, using the same normalization as in ORQA [19].

Retrieval Evaluation

Retrieval was evaluated using top-k retrieval accuracy, which is the fraction of cases where k documents with the highest score contain an answer and the total number of cases. A document was determined to contain an answer using a heuristic sub-string match.

Multilingual Accuracy

The method of multilingual retrieval with BM25 is described in section 5.3.4. In experiments top 1 the passage is selected for each language, therefore in total 17 passages are selected. Accuracy at the 17 multilingual passages are referred to as Multilingual Accuracy.

Translator Evaluation

The evaluation metric used for the evaluation of translators is a BLEU score [23]. The BLEU score measures the similarity between machine-translated text and a set of reference translations. The higher the score, the better is the translation.

6.1.6 Training

The multilingual models were based on mT5-small model and were fine-tuned solely on MKQA dataset. The model with pivot language was based on T5-small. The models were trained for 15 000 optimization steps, using Adam optimizer [16] and batch size of 64 samples.

The MKQA dataset contains 6758 samples which I dived into training, validation, and test splits. Each MKQA sample contains the same question in multiple languages. For each sample in the training set, five languages were samples and for each language, a training example was created. Validation and test examples were created for each language, therefore there were 17 examples per sample. The total number of examples in each split roughly corresponds to 70%/15%/15% of examples per split.

The model with pivot language was fine-tuned on the NQ-open dataset [19]. MKQA dataset consists of a subset of NQ examples, for this reason, additional fine-tuning on the MKQA dataset was omitted. The number of samples in each split and the number of MKQA examples is summarized in table 6.4 for both datasets.

	Training	Validation	Test
MKQA samples	6 000	379	379
MKQA examples	30 000	6 443	6 443
NQ-open	79168	8757	3610

Table 6.4: Number of examples in dataset splits, for MKQA and NQ-open dataset. The row "MKQA samples" contains number of individual samples in the MKQA dataset in each split. The row "MKQA examples" contains the total number of examples.

6.2 Results

With each model, multiple experiments are performed to assess the effect of translation for each setup. To isolate the impact of machine translation on the performance of the models and retrieval three tasks are considered. First I consider a setup, where human translations for questions from MKQA dataset are available to the model. Models taking translated questions on input use questions translated by human translators and retrieval results are also based on questions translated by humans. In the second task, the reader does not have access to questions translated by a human, however, the retrieval is based on human-translated questions. The final setup is a real-world performance of the systems, where no additional information is provided and the system only has access to the original question in the respective language.

6.2.1 Human Translations

This setup assesses the performance of a model if it had access to a "perfect" translator. The questions were translated by a human translator from an original English question.

	Avg.	ar	da	de	es	en	fi	fr	hu	it	ja	nl	pl	pt	ru	sv	th	tr
Tag	**19.23**	8.97	24.54	24.01	21.90	22.43	20.05	20.05	21.11	21.90	12.14	21.64	19.79	21.37	12.93	24.27	10.29	19.53
OQMR	**18.22**	7.65	23.22	21.37	22.16	21.64	18.47	21.11	19.53	21.11	8.97	19.79	21.11	21.11	11.08	22.96	7.65	20.84
OQER	**9.62**	1.06	13.98	11.61	11.87	22.69	9.23	11.61	9.23	9.76	1.32	14.78	7.92	9.76	5.54	10.82	4.49	7.92
Pivot	**21.54**	10.55	27.70	28.76	23.48	31.93	23.48	27.44	24.27	25.59	6.60	25.07	21.90	23.48	7.92	26.91	6.07	25.07

Table 6.5: Comparison of EM score of systems with access to human translations, with score for each language.

Table 6.5 shows the result with human translations. The best performance achieved system with pivot language. From the systems based on the MT5 model, the best performance achieved the model that takes translated questions on the input and indicates the language of the answer with a tag which performs better for the majority of languages compared to OQMR. The OQMR is better only for French and Spanish.

In terms of how the models perform in different languages, all models tend to perform better on languages written in Latin script, and the countries where are the languages spoken are in geographical cultural proximity. The geographical and cultural proximity could cause that the languages to have partially shared vocabulary and which makes it easier for the systems to understand them. This hypothesis also explains why the Russian language does not follow this trend although it is culturally similar but uses a different script, therefore the vocabulary can not transliterate.

Since human translations were used both for retrieval and for question translation to English, the difference in scores for each language for the pivot system is only caused by translation of the answer from English to other languages. From these result can be seen that the translation works well for languages that use Latin script, where the EM score is above 21% for all languages and fails for languages that use a different script where the score is below 11%.

The OQMR and OQER models differ only in retrieval. The OQMR uses multilingual retrieval and the OQER uses only English retrieval. Interestingly, although it is only slightly worse in accuracy as discussed in section 6.2.4, the OQMR achieves twice as good score as the OQER system. The models are only comparable when the question is in the English language. This would suggest that the imbalance of languages on the input of the OQER model, where the majority is in English, causes a loss of comprehension of other languages.

6.2.2 Translated Questions

Using translated questions in the reader but retaining retrieval performance with human translations showcases how additional noise from the translator affects the reader. Retrieved passages correspond to the previous setup, however, the reader is presented with machine-translated questions.

Compared to the previous task, where only human translations were used this affects only systems where the translated question is given on input. This is the system with pivot language described in section 5.3.1 and system with language tags described in section 5.3.2. The OQMR and OQER use machine translation only for retrieval so these systems are not affected in this setup. It should be noted that both the pivot system and the system with tag perform worse in this setup than the OQMR system but the drop in performance is not significant enough to drop below the OQER system.

In the previous task, the Pivot system got English questions, and translation was only used when the answer was translated to another language. In this setup, the Pivot system is presented with a question in an arbitrary language, which is then translated to English.

The system that uses a tag, compared to the previous setup translates each question to every language and pairs it with the respective passage. Both systems perform worse when presented with lower quality machine-translated questions as can be seen in table 6.6. The system with pivot language was affected more compared to the multilingual model, which was more robust.

The difference in EM scores when the translator is used and when system the system has access to questions translated by humans vary significantly among languages. However, there is no significant correlation between the BLEU score of the translator and the difference in EM score, as can be seen in figure 6.2.

6.2.3 Real-World Performance

In a setup where retrieval is based on translated questions and the reader also has access only to original questions and translations, the performance drops significantly. The figure 6.3 shows the effect of each setup on each model compared to real-world performance.

In this most realistic setup, the Pivot system performed the best, however compared to multilingual models the it did not perform significantly better. Out of the multilingual models, the OQMR was more affected compared to the previous setup but performs slightly

		Avg.	ar	da	de	es	en	fi	fr	hu	it	ja	nl	pl	pt	ru	sv	th	tr
	OQMR	**18.22**	7.65	23.22	21.37	22.16	21.64	18.47	21.11	19.53	21.11	8.97	19.78	21.11	21.11	11.08	22.96	7.65	20.84
	OQER	**9.62**	1.06	13.98	11.61	11.87	22.69	9.23	11.61	9.23	9.76	1.32	14.78	7.92	9.76	5.54	10.82	4.49	7.92
Tag	MT question	**15.48**	6.60	20.05	20.44	20.32	19.53	13.46	16.62	15.04	19.53	8.71	15.57	18.47	19.26	10.82	17.15	7.12	14.51
	Difference	**-3.75**	-2.37	-4.49	-3.57	-1.58	-2.90	-6.60	-3.43	-6.07	-2.37	-3.43	-6.07	-1.32	-2.11	-2.11	-7.12	-3.17	-5.01
Pivot	MT question	**14.90**	4.49	22.96	22.16	16.89	31.93	9.50	23.75	13.46	18.21	2.90	16.89	13.46	18.21	5.80	18.47	2.64	11.61
	Difference	**-6.64**	-6.07	-4.75	-6.60	-6.60	0.00	-13.98	-3.69	-10.82	-7.39	-3.69	-8.18	-8.44	-5.28	-2.11	-8.44	-3.43	-13.46

Table 6.6: EM score of Tag model and Pivot model when machine translated question is used with the difference when human translations are used.

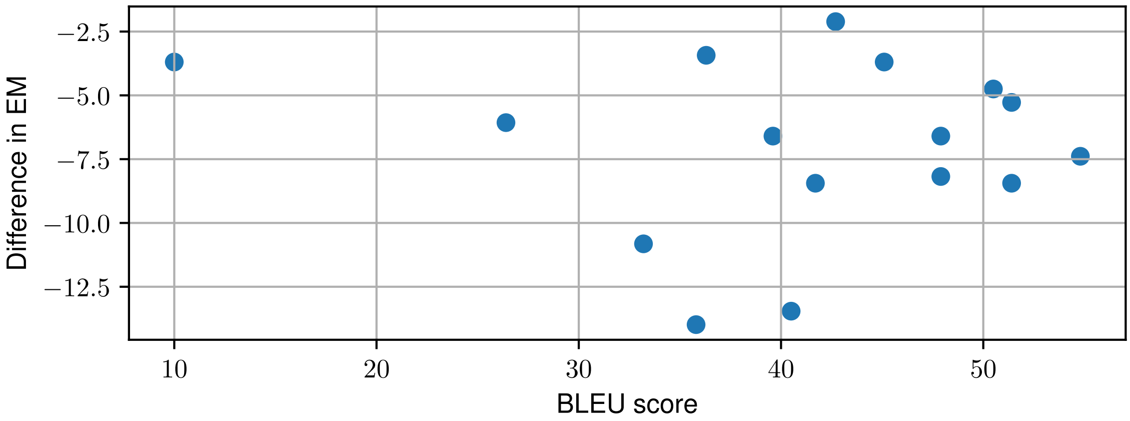

Figure 6.2: Difference in EM score for each language of Pivot system when using original English questions and machine-translated questions. On X-axes is the BLUE score of translator translating from multiple languages to English which is used for translating questions for the Pivot model. On Y-axes is a difference in EM score between using human translated questions and machine-translated questions.

better than the model with tag. The OQER system under-performed all other models in all of the tasks.

6.2.4 Multilingual BM25

Figure 6.4 shows the top-k accuracy of BM25 for different languages and "Multilingual Accuracy", using human translated questions as described in section 6.1.2. English has the best accuracy, compared to other languages, which can have multiple reasons.

- MKQA dataset is a subset of the English Natural Questions dataset [18] translated to other languages and answers were annotated in English Wikipedia.
- English corpus is twice as large as the second-largest corpus.
- BM25 works better with English than with other languages.

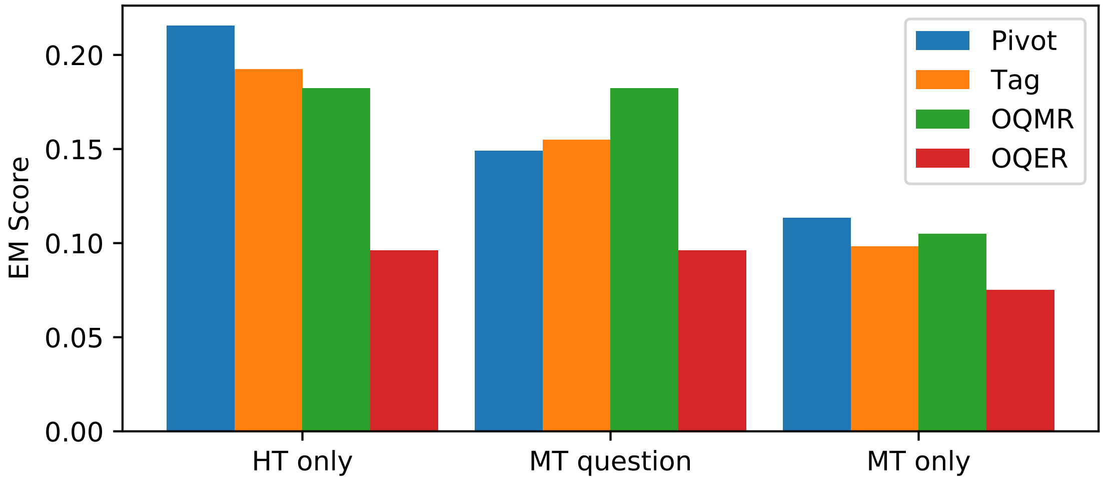

Figure 6.3: Comparison of models and performance in different setups. "HT only" corresponds to setup from section 6.2.1, where the system has access to human translations. "HT retrieval" corresponds to setup from the section 6.2.2, where the system has access to retrieval based on human translations, but the reader has access only to machine translated questions. "MT only" is when the system has no access to any additional information and the results are real-world performance.

As can be seen in figure 6.5 retrieval accuracy is correlated with the size of the corpus. However, accuracy depends even more on a language family, or a language group. This trend is most notable between Romance and Germanic languages, where the number of Wikipedia articles ranges from approximately one million up to more than three million articles, but the retrieval accuracy range is within 5% points. Only Danish and English are significantly misplaced in terms of accuracy, but both English and Danish have also significant differences in the number of Wikipedia articles. Danish Wikipedia contains one-fifth of the articles to the next smallest Wikipedia within the Germanic-Romance groups and English is having twice as many articles compared to the second largest.

Slavic languages also belong to the same language family as the aforementioned Germanic and Romance languages however, there is a notable drop in accuracy for languages in this group. Most likely this could be caused by richer morphology, for example, Polish distinguishes seven noun cases compared to Dutch which distinguishes only two, or Spanish with only one.

The other three outliers, with the worst accuracy, are Arabic, Japanese and Thai. Each of them belongs to a distinct language family and is using a unique writing system.

With respect to the size of the corpus and accuracy, the Japanese language underperformed all other languages. This could be because it uses two writing systems, Kana and Kanji, which are very different from other selected languages. Kanji is an adopted Chinese script, where each character represents one or more different words. In Kana, each character represents a syllable. The rest of the writing systems, Latin and Cyrillic, use characters to represent phonemes.

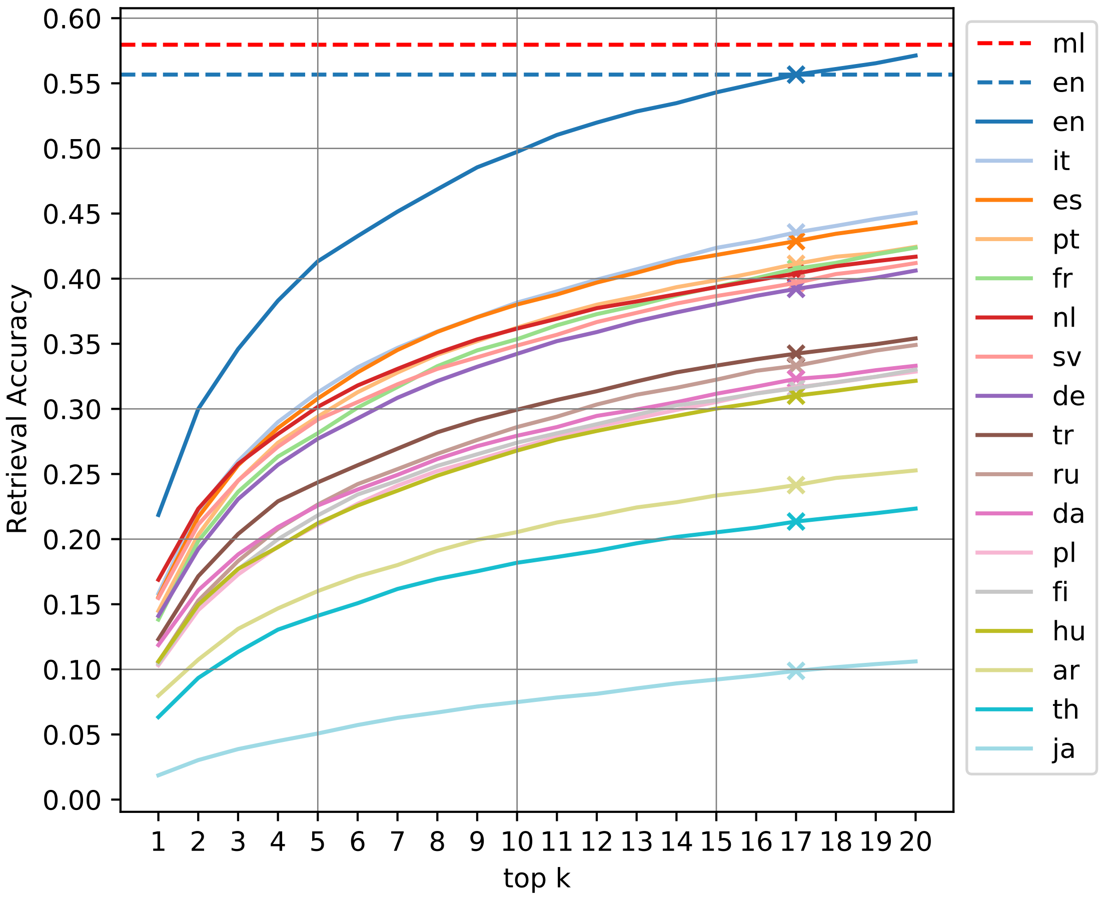

Figure 6.4: Comparison between languages of accuracy when using human translated question for retrieval. The red dashed line denotes Multilingual Accuracy (accuracy at 17 top 1 passages from each language) and the blue dashed line denotes the accuracy at top 17 for English. The crosses mark accuracy at the top 17 for each language. The legend is ordered according to accuracy at 17.

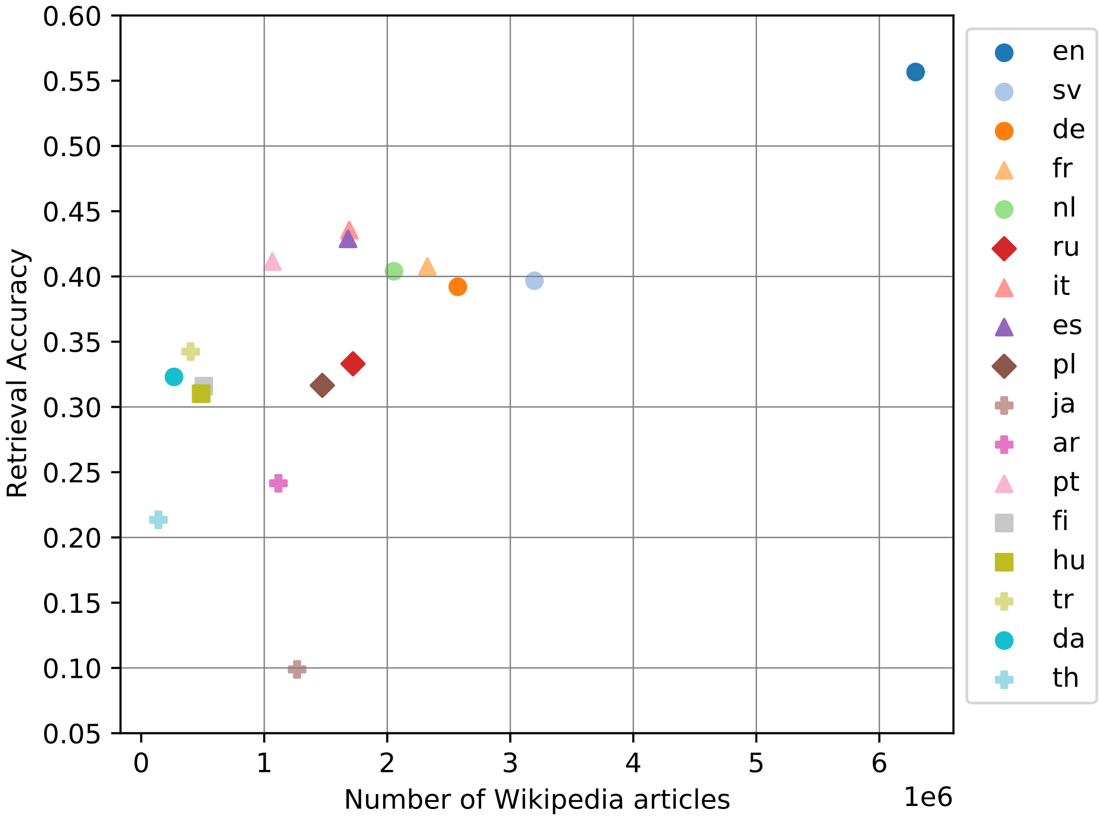

Figure 6.5: Correlation between the number of articles and BM25 retrieval accuracy. Legend is ordered by the accuracy for the top 17 passages. The shape indicates language family or language group. Circles are language from Germanic language group, triangles are Romance languages, diamond signs are Slavic languages and squares are Uralic languages. Plus signs are Turkish, Arabic, Thai, and Japanese which belong to distinct language families, Turkic, Semitic, Kra-Dai, and Japonic language families respectively. Germanic, Romance, and Slavic language groups belong to the Into-European language family. Kendall's tau correlation coefficient is 0.4 with statistical significance over 97%.

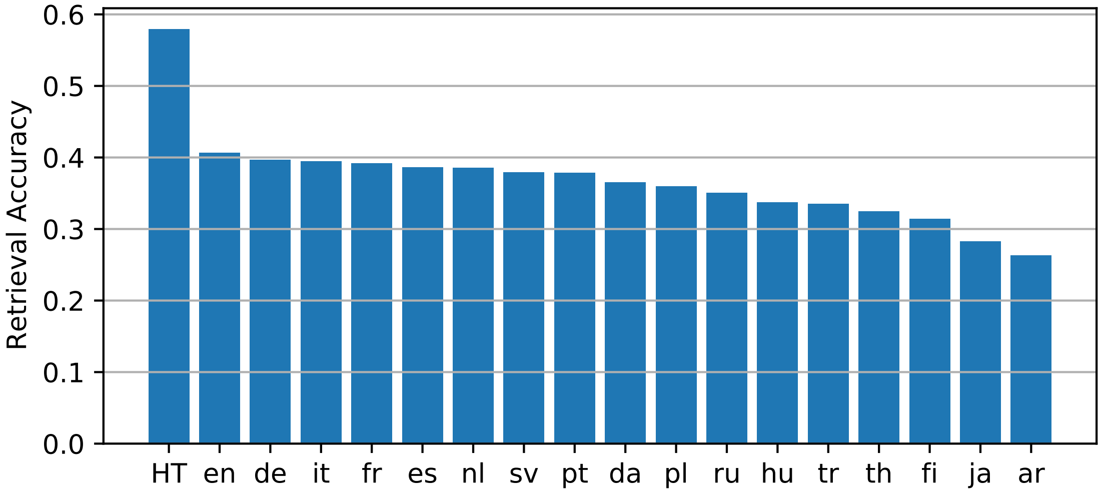

Figure 6.6: Influence of machine translation on retrieval accuracy. Each bar corresponds to accuracy when translating from a language. For example “ar” means that each question was translated from Arabic to each of the 17 languages and multilingual retrieval accuracy is computed. For each question translation a top 1 passage was selected and accuracy is a percentage of cases where either of the passages contained an answer.

Chapter 7

Conclusion

The objective of this work was to propose, develop and compare various approaches to Multilingual Open-Domain Question Answering. The proposed systems are based on generative transformer model T5. Two main approaches were tested. The first was adapting the English monolingual model to the multilingual task by using machine translation for translating both input and output. The second was the use of the natively multilingual model, which would take multilingual inputs and generate the answers directly in the target language. Multiple variants of the natively multilingual system were proposed, however, the best performance achieved the English monolingual system with machine translation.

This thesis also presents a method of multilingual retrieval with a probabilistic BM25 algorithm. This method seems to perform better in comparison to English retrieval. Using this method of retrieval is also much more effective compared to using only English retrieval in combination with multilingual models. The reason for this might be that if the majority of the input is only in one language the model tends to lower the performance in other languages. This loss of performance could be also observed with some languages that were more linguistically distant from the majority of languages. However, this method of retrieving multilingual passages relies on machine translation and the results are significantly influenced by the quality of machine translation. This drawback could be potentially solved by developing a more sophisticated method of multilingual retrieval, that is more robust to the quality of translations or does not use translations at all.

The use of multilingual models in question answering is a promising approach and potentially could outperform English-based models. The additional translation required with English-based models also transfers to inference time. Current multilingual datasets do not provide enough data to train a larger model that would achieve better performance. Another approach to dealing with this problem could be to a use more sophisticated method of training. For example machine translation could be also included as an additional task.

The results obtained in this thesis also suggest that creating a more specialized system that would be trained on a set of closely related languages, would be more effective. This could be especially beneficial for low resource languages, where creating a monolingual question answering system is not feasible.

Bibliography

[1] Asai, A., Hashimoto, K., Hajishirzi, H., Socher, R. and Xiong, C. Learning to Retrieve Reasoning Paths over Wikipedia Graph for Question Answering. In: *International Conference on Learning Representations*. 2020.

[2] Asai, A., Kasai, J., Clark, J. H., Lee, K., Choi, E. et al. XOR QA: Cross-lingual Open-Retrieval Question Answering. In: *NAACL-HLT*. 2021.

[3] Ba, J. L., Kiros, J. R. and Hinton, G. E. Layer normalization. *ArXiv preprint arXiv:1607.06450*. 2016.

[4] Bergstra, J., Komer, B., Eliasmith, C., Yamins, D. and Cox, D. D. Hyperopt: a Python library for model selection and hyperparameter optimization. *Computational Science & Discovery*. IOP Publishing. jul 2015, vol. 8, no. 1, p. 014008. DOI: 10.1088/1749-4699/8/1/014008. Available at: https://doi.org/10.1088/1749-4699/8/1/014008.

[5] Chen, D., Fisch, A., Weston, J. and Bordes, A. Reading Wikipedia to Answer Open-Domain Questions. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, July 2017, p. 1870–1879. DOI: 10.18653/v1/P17-1171. Available at: https://www.aclweb.org/anthology/P17-1171.

[6] Chung, J., Gulcehre, C., Cho, K. and Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *ArXiv preprint arXiv:1412.3555*. 2014.

[7] Clark, J. H., Choi, E., Collins, M., Garrette, D., Kwiatkowski, T. et al. TyDi QA: A Benchmark for Information-Seeking Question Answering in Ty pologically Di verse Languages. *Transactions of the Association for Computational Linguistics*. MIT Press. 2020, vol. 8, p. 454–470.

[8] Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, p. 4171–4186. DOI: 10.18653/v1/N19-1423. Available at: https://www.aclweb.org/anthology/N19-1423.

[9] Eberhard, G. F. S. and Fennig, C. D. *Ethnologue: Languages of the World. Twenty-fourth edition*. 2021. Available at: https://www.ethnologue.com/.

[10] Fajcik, M., Docekal, M., Ondrej, K. and Smrz, P. Pruning the Index Contents for Memory Efficient Open-Domain QA. *ArXiv preprint arXiv:2102.10697*. 2021.

[11] He, K., Zhang, X., Ren, S. and Sun, J. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, p. 770–778.

[12] Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*. MIT Press. 1997, vol. 9, no. 8, p. 1735–1780.

[13] Izacard, G. and Grave, E. Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering. In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online: Association for Computational Linguistics, April 2021, p. 874–880. Available at: https://www.aclweb.org/anthology/2021.eacl-main.74.

[14] Kamphuis, C., Vries, A. P. de, Boytsov, L. and Lin, J. Which BM25 Do You Mean? A Large-Scale Reproducibility Study of Scoring Variants. In: Jose, J. M., Yilmaz, E., Magalhães, J., Castells, P., Ferro, N. et al., ed. *Advances in Information Retrieval*. Cham: Springer International Publishing, 2020, p. 28–34. ISBN 978-3-030-45442-5.

[15] Karpukhin, V., Oğuz, B., Min, S., Wu, L., Edunov, S. et al. Dense Passage Retrieval for Open-Domain Question Answering. *ArXiv preprint arXiv:2004.04906*. 2020.

[16] Kingma, D. P. and Ba, J. *Adam: A Method for Stochastic Optimization*. 2017.

[17] Kudo, T. Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates. *CoRR*. 2018, abs/1804.10959. Available at: http://arxiv.org/abs/1804.10959.

[18] Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A. et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*. MIT Press. 2019, vol. 7, p. 453–466.

[19] Lee, K., Chang, M.-W. and Toutanova, K. Latent Retrieval for Weakly Supervised Open Domain Question Answering. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, p. 6086–6096. DOI: 10.18653/v1/P19-1612. Available at: https://www.aclweb.org/anthology/P19-1612.

[20] Lewis, P., Oğuz, B., Rinott, R., Riedel, S. and Schwenk, H. MLQA: Evaluating Cross-lingual Extractive Question Answering. 2020.

[21] Longpre, S., Lu, Y. and Daiber, J. MKQA: A linguistically diverse benchmark for multilingual open domain question answering. *ArXiv preprint arXiv:2007.15207*. 2020.

[22] Montero, I., Longpre, S., Lao, N., Frank, A. J. and DuBois, C. Pivot Through English: Reliably Answering Multilingual Questions without Document Retrieval. *CoRR*. 2020, abs/2012.14094. Available at: https://arxiv.org/abs/2012.14094.

[23] Papineni, K., Roukos, S., Ward, T. and Zhu, W.-J. Bleu: a method for automatic evaluation of machine translation. In: *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 2002, p. 311–318.

[24] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J. et al. Pytorch: An imperative style, high-performance deep learning library. In: *Advances in neural information processing systems*. 2019, p. 8026–8037.

[25] Pérez Iglesias, J., Pérez Agüera, J. R., Fresno, V. and Feinstein, Y. Z. Integrating the probabilistic models BM25/BM25F into Lucene. *ArXiv preprint arXiv:0911.5046*. 2009.

[26] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S. et al. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*. 2020, vol. 21, no. 140, p. 1–67. Available at: http://jmlr.org/papers/v21/20-074.html.

[27] Rajpurkar, P., Zhang, J., Lopyrev, K. and Liang, P. Squad: 100,000+ questions for machine comprehension of text. *ArXiv preprint arXiv:1606.05250*. 2016.

[28] Robertson, S. E., Walker, S., Jones, S., Hancock Beaulieu, M. M., Gatford, M. et al. Okapi at TREC-3. *Nist Special Publication Sp*. NATIONAL INSTIUTE OF STANDARDS & TECHNOLOGY. 1995, vol. 109, p. 109.

[29] Sasín, J. *Machine Learning for Natural Language Question Answering*. Brno, Czech Republic, 2021. Bachelor's thesis. Brno University of Technology.

[30] Sennrich, R., Haddow, B. and Birch, A. Neural Machine Translation of Rare Words with Subword Units. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, August 2016, p. 1715–1725. DOI: 10.18653/v1/P16-1162. Available at: https://www.aclweb.org/anthology/P16-1162.

[31] Shi, W. and Demberg, V. Next Sentence Prediction helps Implicit Discourse Relation Classification within and across Domains. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, November 2019, p. 5790–5796. DOI: 10.18653/v1/D19-1586. Available at: https://www.aclweb.org/anthology/D19-1586.

[32] Taylor, M., Zaragoza, H., Craswell, N., Robertson, S. and Burges, C. Optimisation Methods for Ranking Functions with Multiple Parameters. In: *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*. New York, NY, USA: Association for Computing Machinery, 2006, p. 585–593. CIKM '06. DOI: 10.1145/1183614.1183698. ISBN 1595934332. Available at: https://doi.org/10.1145/1183614.1183698.

[33] Taylor, W. L. "Cloze procedure": A new tool for measuring readability. *Journalism quarterly*. SAGE Publications Sage CA: Los Angeles, CA. 1953, vol. 30, no. 4, p. 415–433.

[34] Tiedemann, J. and Thottingal, S. OPUS-MT — Building open translation services for the World. In: *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*. Lisbon, Portugal: [b.n.], 2020.

[35] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L. et al. Attention Is All You Need. 2017.

[36] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C. et al. Transformers: State-of-the-Art Natural Language Processing. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, October 2020, p. 38–45. Available at: https://www.aclweb.org/anthology/2020.emnlp-demos.6.

[37] Xue, L., Constant, N., Roberts, A., Kale, M., Al Rfou, R. et al. *mT5: A massively multilingual pre-trained text-to-text transformer*. 2020.