# Cross Lingual Open-Retrieval Question Answering

*Project to be submitted in partial fulfillment of the*

*requirements for the degree*

*Of*

**Bachelor of Technology**

*In*

**Computer Science and Engineering**

*By*

**Aditi Singhal - 1910110030**

**Ashwin Nair- 1910110101**

**Rahul Madan- 1910110300**

Under the guidance of

**Dr. Sonia Khetarpaul**


**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**OCTOBER 2022**

# Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

# Acknowledgement

First and foremost, we would like to express our sincere gratitude to our mentor, **Dr. Sonia Khetarpaul**, for her continuous support, patience, motivation, enthusiasm, and immense knowledge. We are extremely indebted to her for all the valuable guidance and dedicated involvement in every step of the project. She taught us the methodology to carry out research and to present the research works as clearly as possible. It was from her that we learnt the importance and value of studying topics in-depth and not being satisfied with surface-level knowledge. Understanding the very sophisticated field of Information Retrieval from scratch seemed very daunting to us at first, but her vision and support throughout made the task easy. We could not have asked for a better advisor and mentor. Completing the project required more than academic support and we have our family, friends and professors to thank for listening to and, at times, tolerating us. We cannot begin to express our gratitude and appreciation for their support.

# Abstract

*Dealing with multilingual questions and answers usually assumes that the answers are in the same language as the question. In practice, however, many languages face both information deficit (few reference articles in the language) and information asymmetry (cultural bias). Confidently making progress on multilingual modeling requires challenging, trust- worthy evaluations.*

*Open-domain question addressing (OpenQA) intends to respond to inquiries through text recovery and reading comprehension. A bunch of neural network-based models have been proposed and accomplished promising outcomes in OpenQA. Nonetheless, the outcome of these models depends on an enormous volume of training data which is mostly in English, which isn't available in many other languages. Because of this reason, researching multi-lingual OpenQA is fundamental. The conclusion obtained is that the performance of cross-lingual OpenQA is connected with not just how similar the desired language and English are, yet in addition how difficult the questions in the desired target language are.*

*This report extends upon initial attempts to open retrieval question answering to a cross-language setting, wherein firstly, a basic TF-IDF model is applied directly on russian news data which takes query in the target language only (russian here) and then another TF-IDF model on translated data (russian to english).*

# Contents



# List Of Figures

# Chapter 1

# Introduction and Literature Review

## 1.1 Introduction

Information-seeking questions are questions from people who are actually looking for an answer which has been increasingly studied in question answering (QA) research. Fulfilling these information needs has caused the research community to look further for answers: beyond paragraphs and articles in just a single language and towards performing information retrieval on large-scale document collections. In this paper, we bring together information seeking questions, open-retrieval and multilingual retrieval.

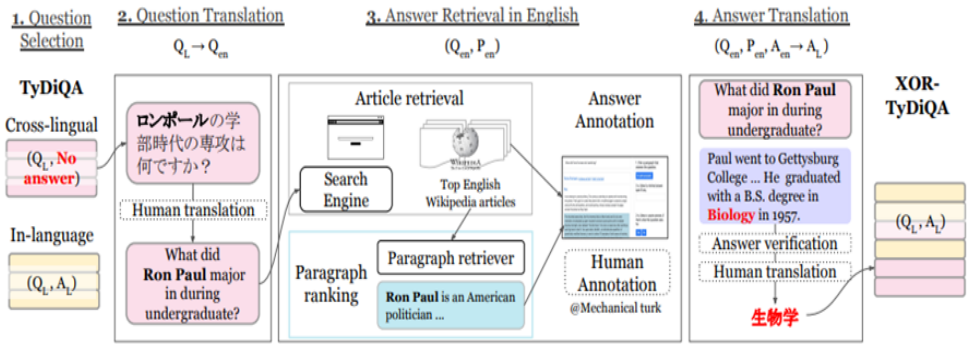
Multilingual Information Retrieval systems would benefit the many speakers of non-English languages. Hence, we introduce the task of multilingual Information Retrieval which aims at answering multilingual questions from non-English native speakers given multilingual resources.

For this purpose we have chosen the Russian News Dataset which contains news articles from popular Russian media Lenta.Ru, performed EDA to find out properties of various features of the dataset, to find out how to transform our original data to perform various operations on it and then created a TF-IDF model after preprocessing the data and evaluated the performances on two different cases, (1)Translating the corpus to an anchor language and then training a unified model/index. (2)Building independent models/indexes for each language and translating the query for model and result for user.

## 1.2 Literature Review

### 1.3.1 XOR QA: Cross-lingual Open-Retrieval Question Answering

[1] This work stretches out open-retrieval question answering to a cross-lingual setting enabling questions from one language to be answered through answer content from another language. For this, a task framework called Cross-lingual OpenRetrieval Question Answering (XOR QA), comprises three subtasks involving cross lingual document retrieval from multilingual and english resources. We lay out baselines with state-of-the-art machine translation systems and cross-lingual pretrained models. Exploratory outcomes propose that XOR QA is a testing task that will work with the improvement of novel strategies for multilingual question answering.

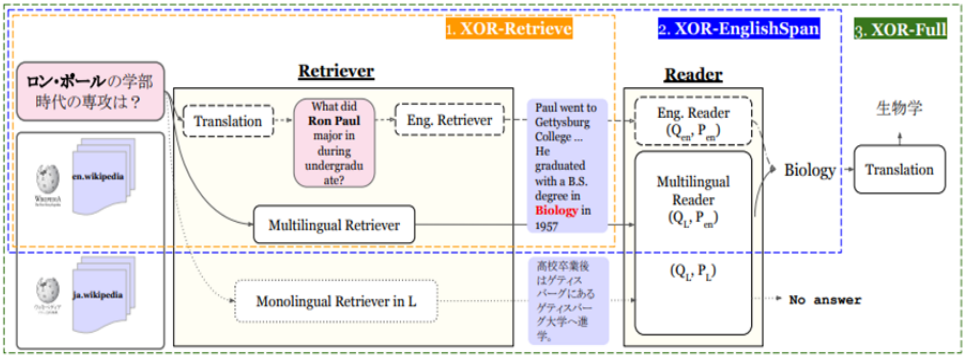



*Note.* Overview of the annotation process for XOR-TYDI QA. [1]

XOR QA Tasks and Baselines

We introduce three new tasks : XOR RETRIEVE, XOR-ENGLISH SPAN, and XOR-FULL with our newly collected XOR-TYDI QA dataset and construct strong baselines for each task. XORFULL defines our goal of building a multilingual open-retrieval QA system that uses

both crosslingual and in-language questions from XOR-TYDI QA.




*Note.* Overview of the tasks and baselines. Each dotted rectangle represents one of the three tasks and surrounds used pipeline modules. [1]

**XOR Retrieve** - Given a question in Li and English Wikipedia Weng, the task is to retrieve English paragraphs for the question.

**XOR English Span** - Given a question in Li and English Wikipedia Weng, a system retrieves paragraphs from Weng and extracts an answer.

**XOR-FULL** - Given a question in target language Li and Wikipedia in both English and Li (Weng and Wi ), a system is required to generate an answer in Li.

**Evaluation** – The recall is measured by computing the fraction of the questions for which the minimal answer is contained in the top n tokens selected.

**NOTE** : XOR-FULL evaluation data includes both cross-lingual and in-language data, while XOR-RETRIEVE and XOR-ENGLISH SPAN only use cross-lingual data during evaluation.

## Challenges faced

- Finding evidence paragraphs from large-scale document collections like Wikipedia is a challenging task, especially when a query and documents are in different languages and systems cannot perform lexical matching.

- In addition to fundamental problems of information scarcity and asymmetry in multilingual QA, questions can be labeled as unanswerable simply because of annotation errors.

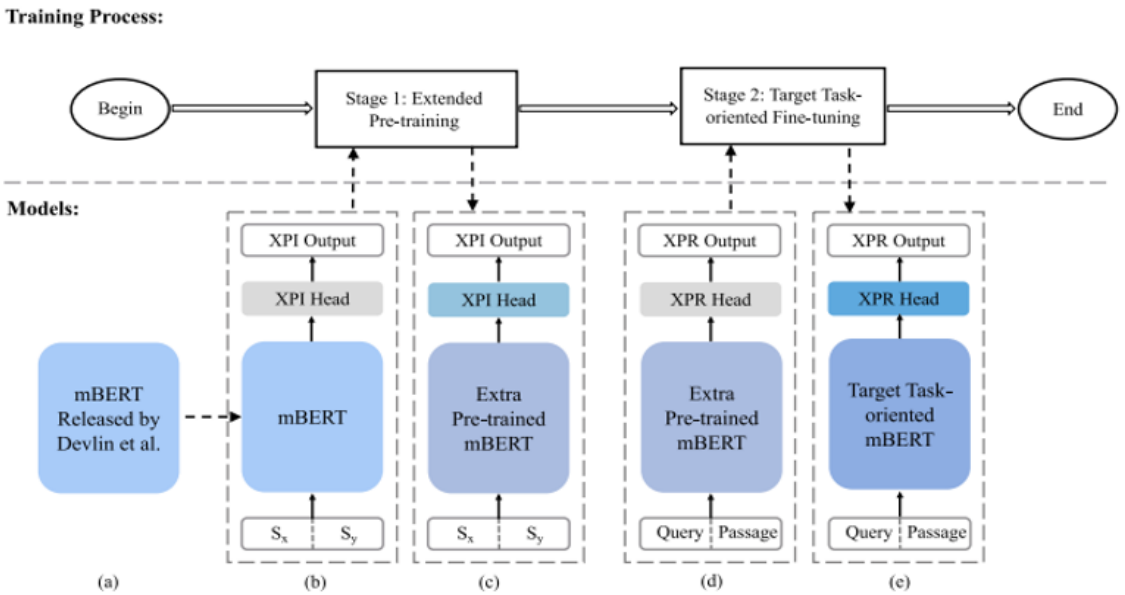
## Cross-Lingual Passage re-ranking with multi-lingual BERT

[9] Cross-lingual Passage Re-ranking (XPR) is a task which is designed to rank a list of candidate passages in multiple languages. Then a Cross-lingual Paraphrase Identification (XPI) as an extra pre-training task is proposed that aims to further augment the cross-lingual alignment.

The task of Cross-lingual Passage Re-ranking (XPR) aims to rank a list of candidate passages with multiple languages in a query, it commonly faces two main problems: (1) query and the passages to be ranked are often written in different languages, so strong cross-language alignments are required, and (2) Lack of annotated data for model training and evaluation. The task of cross-lingual paraphrase identification(XPI) is used as additional pre-training to improve alignment using a large unattended parallel corpus. This task aims to determine if two sentences, presumably in different languages, have the same meaning. Experimental results show that extended pretraining contributes significantly to this XPR task.

To alleviate the scarcity of annotated data for XPR, a new dataset is created by combining a monolingual one with its already available translation. Besides, we present three effective strategies for model training. Extensive experiments have been conducted on in-domain as well as out-domain sets.

## Methodology




*Note.* The framework of cross-lingual passage re-ranking. Different colors indicate different model parameters, e.g., gray color denotes random initialization. [9]

## Proposed Model

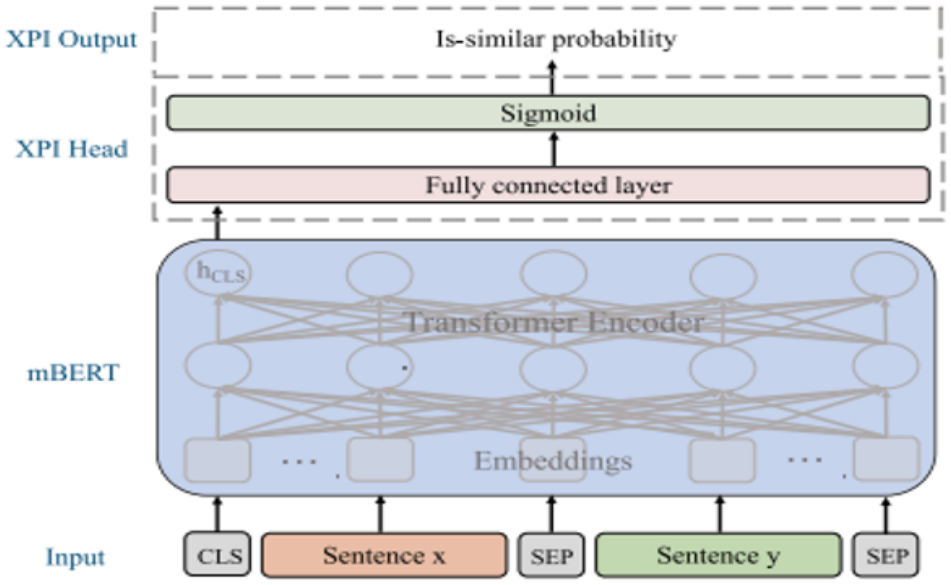


Fig. mBERT for cross-lingual paraphrase identification. [9]

## Evaluation metrics

**Top-k accuracy** - measures the percentage of the queries with at least one relevant passage in the list, Pr, of top-k ranked passages.

**Mean Average Precision (MAP)** - the mean of the average precision scores for each query across all queries

**Mean Reciprocal Rank (MRR)** - obtained using the binary relevance judgments and is defined as the reciprocal rank of the first relevant passage averaged across all queries.

## Observations

Comparisons with pre-trained model Mbert – performs very well on reading comprehension, document classification, etc.

Baseline – fine-tuning Mbert (GELU activation function).

For extended pretraining XPI to see if they have a similar meaning, BERT used.

| Dataset | Method | acc@1(%) | acc@10(%) | MRR(%) | MAP(%) |
|---|---|---|---|---|---|
| BiPaR | mBERT+Mixed Training | 15.13 | 39.50 | 23.41 | 23.41 |
| | mBERT+XPI+Mixed Training | **17.07** | **42.73** | **25.95** | **25.95** |
| MLQA | mBERT+Mixed Training | 29.96 | 58.93 | 39.48 | 39.48 |
| | mBERT+XPI+Mixed Training | **33.33** | **64.68** | **43.04** | **43.04** |
| XQuAD | mBERT+Mixed Training | 47.9 | 83.19 | 59.81 | 59.81 |
| | mBERT+XPI+Mixed Training | **56.64** | **88.40** | **67.80** | **67.80** |

Comparison of models [9]

It is observed that Pre-training XPI improves the results and continuing pre-training the model toward a specific task provides significant benefits.

Alignment between languages (sentence level), is essential for XPR.

Combination of XPI and Mixed Training proves the best, better performance is achieved when the domains of training data are similar.

## One Question Answering Model for Many Languages with Cross-lingual Dense Passage Retrieval

[5] Cross-lingual Open-Retrieval Answer Generation (CORA) is a many-to-many unified QA model that uses dense passage retrieval algorithm to retrieve documents across languages for any question

CORA substantially outperforms the previous state of the art on multilingual open QA benchmarks across 26 languages, 9 of which are unseen during training.

CORA extends the retrieve-then-generate approach of English open QA (Lewis et al., 2020; Izacard and Grave, 2021b) with a single cross-lingual retriever and a generator that do not rely on language-specific retrievers or machine translation modules.

Several recent work introduces single multilingual models for many languages using pre-trained multilingual models such as mBERT or mT5 in many NLP tasks.

Answering multilingual questions requires retrieving evidence from knowledge sources of other languages than the original question since many languages have limited

reference documents or the question sometimes inquires about concepts from other cultures.

## Methodology

**mDPR:** Multilingual Dense Passage Retriever
Used to create dense embeddings for question and passage.

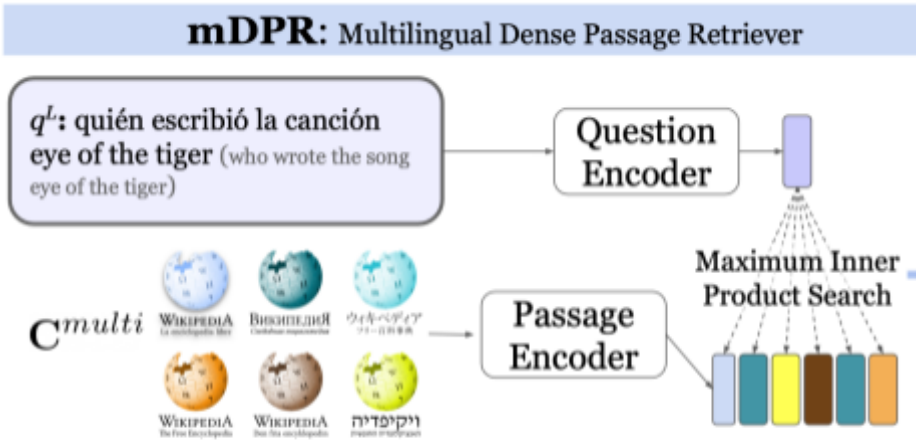



Fig. Overview of CORA (mDPR). [5]

**mGEN**: The generation module (mGEN) is trained to output an answer in the target language conditioned on the retrieved multilingual passages

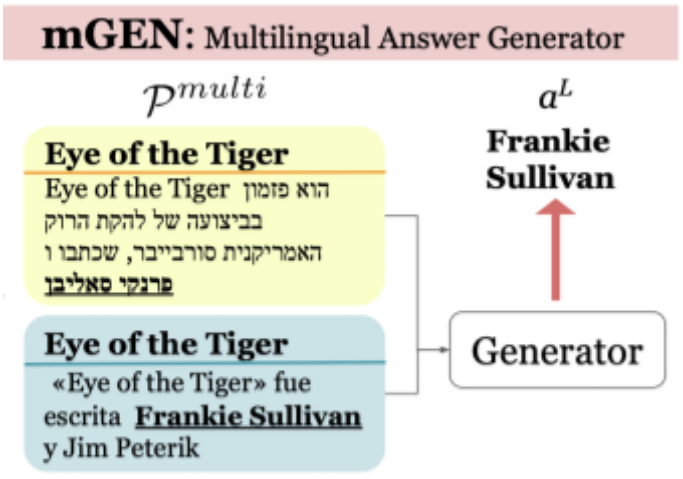


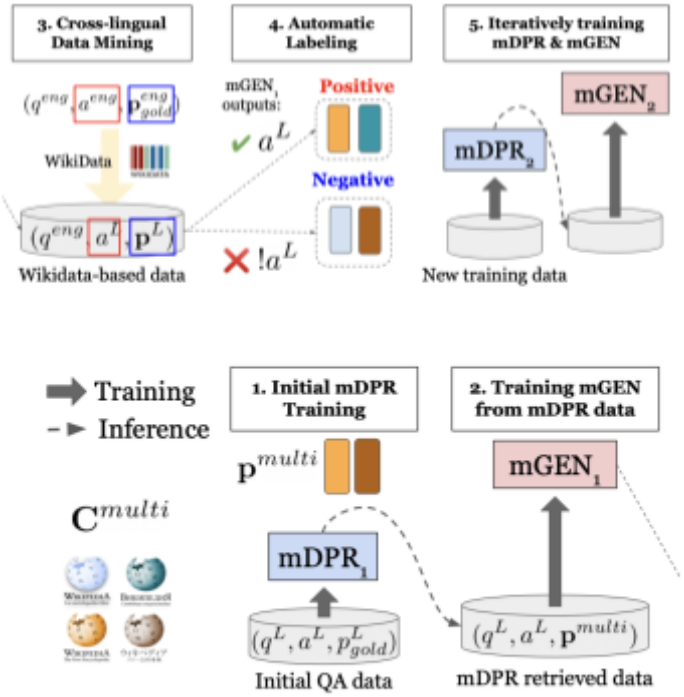

Fig. Overview of CORA (mGEN). [5]

Training and data mining

Initial training data is a combination of multilingual QA datasets: XOR- TYDI QA and TYDI QA and NQ, an English open QA dataset: Natural Questions, Kwiatkowski, 2019)

CORA is evaluated on two multilingual open QA datasets across 28 typologically diverse languages.
CORA **achieves state-of-the-art performance** across 26 languages, and greatly outperforms previous approaches that use language-specific components such as question or answer translation.

CORA uses an **iterative training method** that encourages cross-lingual retrieval and answer generation conditioned on multilingual passages. Each iteration proceeds over two stages: parameter updates where mDPR and mGEN are trained on the current training data and cross-lingual data




Note. Overview of CORA iterative training and data mining. [5]

mining where training data are automatically expanded by Wikipedia language links and model predictions.

## Benchmark used

**(1)XOR-TYDI QA** (Asai et al., 2021) and (2) **MKQA** (Longpre et al., 2020)

XOR-TYDI is a multilingual open QA dataset consisting of **7 typologically diverse languages**, where questions are originally from TYDI QA (Clark et al., 2020) and posed by information-seeking native speakers. The answers are annotated by extracting spans from Wikipedia in the same language as the question (**in-language data**) or by **translating English** spans extracted from English Wikipedia to the target language **(cross-lingual data**). XOR-TYDI QA offers both training and evaluation data.

MKQA (Longpre et al., 2020) is an evaluation dataset created by translating 10k Natural Questions Dataset to 25 target languages.

XOR-TYDI QA and MKQA have five languages in common. The **parallel data** enabled them to compare the models' performance across typologically diverse languages, in contrast to XOR-TYDI QA. MKQA has evaluation data only.

## Observation

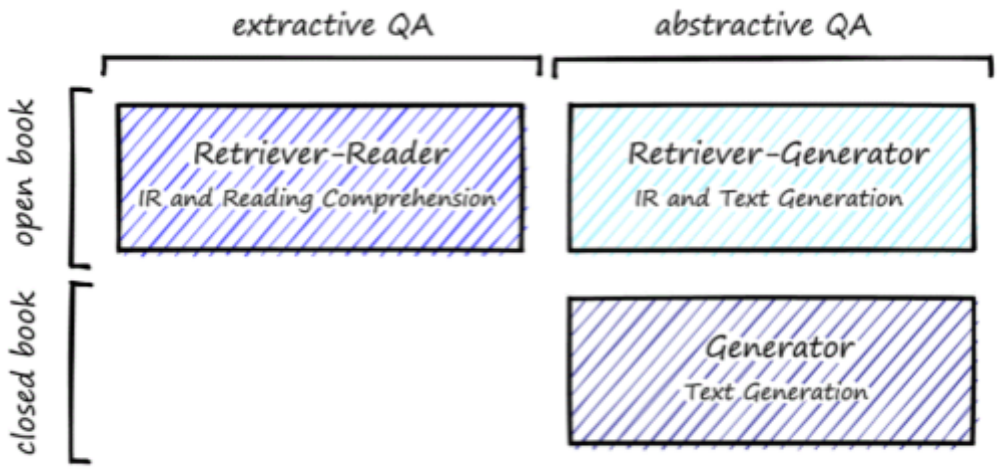
It was observed that through cross-lingual retrieval, CORA can find answers to 20% of the multilingual questions that are valid but are originally annotated as unanswerable by humans due to the lack of evidence in the English knowledge sources.

## Previous work

Previous work in multilingual open QA (Ture and Boschee, 2016; Asai et al., 2021) translates questions into English, applies an English open QA system to answer in English, and then translates answers back to the target language. Those pipeline approaches suffer from error propagation of the

machine translation component into the downstream QA, especially for low-resource languages. Moreover, they are not able to answer questions whose answers can be found in resources written in languages other than English or the target languages.

## 1.3 Open Domain Question Answering

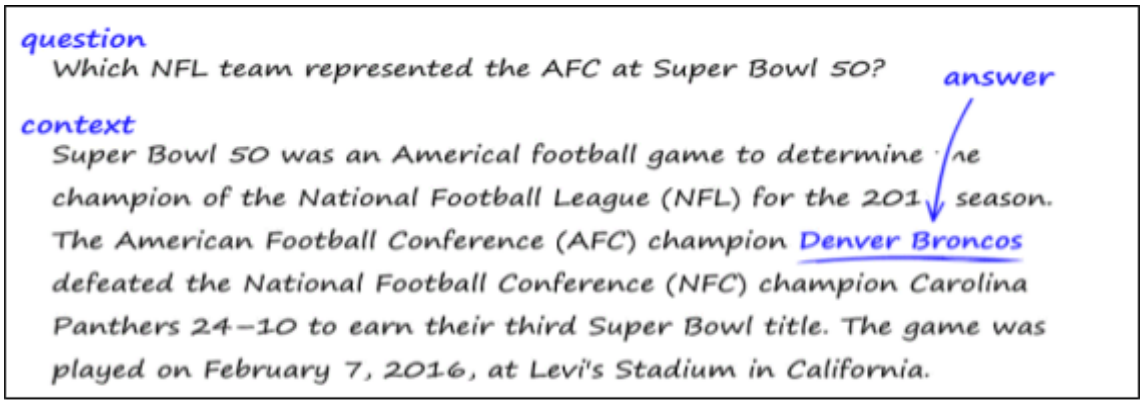


Various approaches to question answering(QA) [9]

Open-book extractive QA is the most popular type of QA (top-left above). Here, we combine a reading comprehension (RC) stage with an information retrieval (IR) step.

Any open-book QA procedure must include an IR phase to obtain pertinent data from the "open-book." The model can make reference to an outside source of information, such like with open-book exams where students can consult their own books for information while taking the test. This information can come from any source that isn't the model itself, including internal company documentation, Wikipedia, Reddit, and others.

The RC (reader) phase receives the pertinent documents that were retrieved by the IR step. RC involves taking a concise response from a sentence or paragraph, often known as the document or context.




Example of question, relevant context and answer[9]

Instead of retrieving answers, the other two QA approaches depend on generating them. The GPT models from OpenAI are well-known examples of generative transformer models.

### 1.4.1 Extractive QA

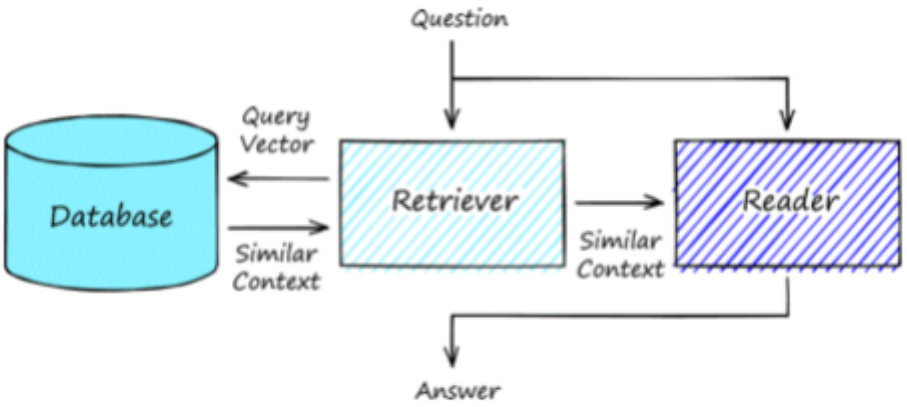
One of the most extensively used type of question-answering is extractive QA. It enables us to ask a question from a brief sentence before extracting a response. For instance, the following phrase (or context):

“Super Bowl 50 was an American football game to determine the champion of the National Football League (NFL) for the 2015 season. The American Football Conference (AFC) champion Denver Broncos defeated the National Football Conference (NFC) champion Carolina Panthers 24–10 to earn their third Super Bowl title. The game was played on February 7, 2016, at Levi's Stadium in the San Francisco Bay Area at Santa Clara, California.”

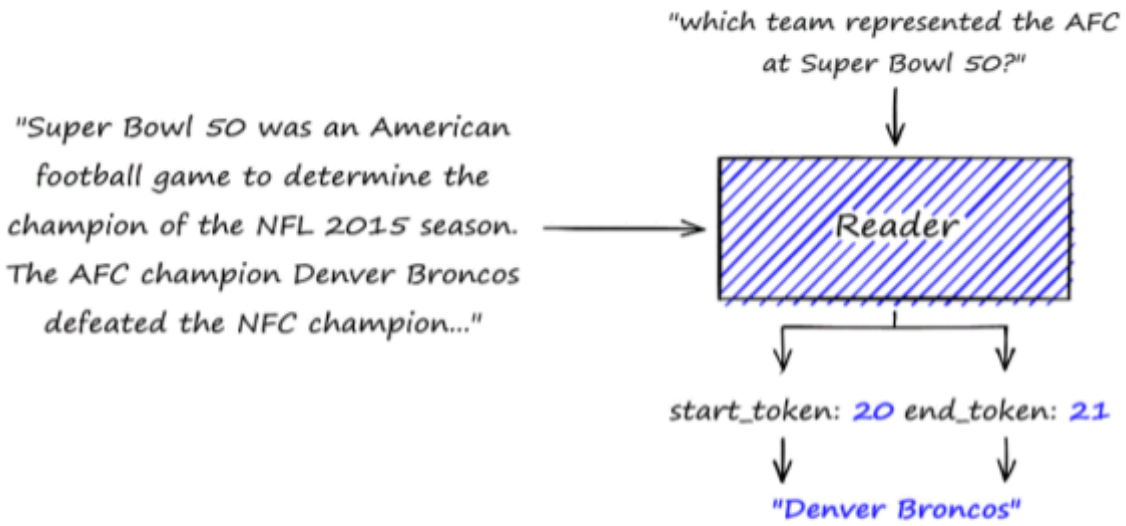
To which we could ask the question, "which team represented the AFC at Super Bowl 50?" and we should expect to return "Denver Broncos".

Reading comprehension is an illustration of how we might provide a single context and draw conclusions from it (RC). This isn't very helpful by itself, but when combined with another data source, we can search across multiple contexts as opposed to just one. Our term for this is "open-book extractive QA." Often referred to as simply extractive QA. It is actually made up of three parts rather than being a single model:

- Indexed data (document store/vector database)
- Retriever model
- Reader model




Open-book extractive QA stack, retriever and reader model [9]



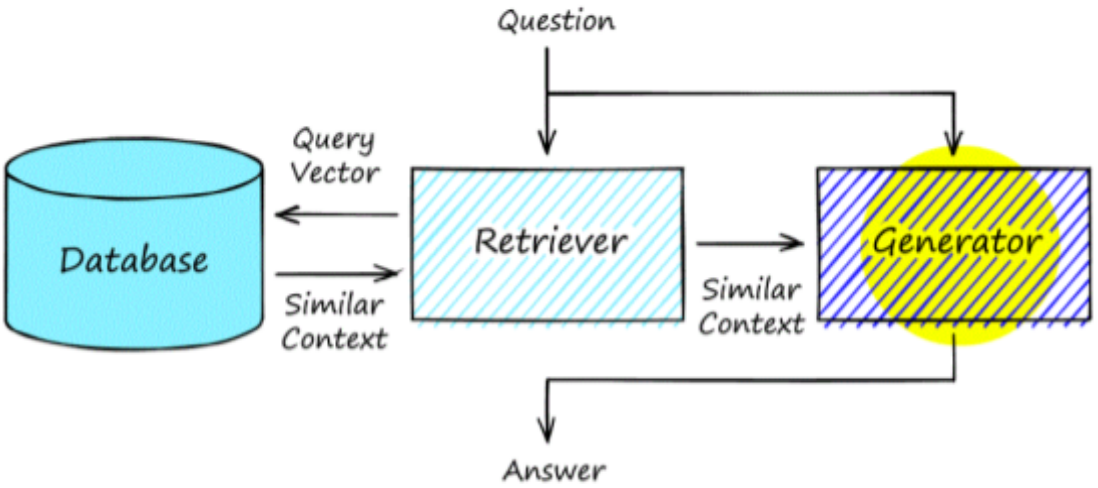

Reader model predicts start and end positions of answer [9]

### 1.4.2 Abstractive QA

Abstractive QA can be split into two types: open-book and *closed*-book.

### 1.4.3 Open Book

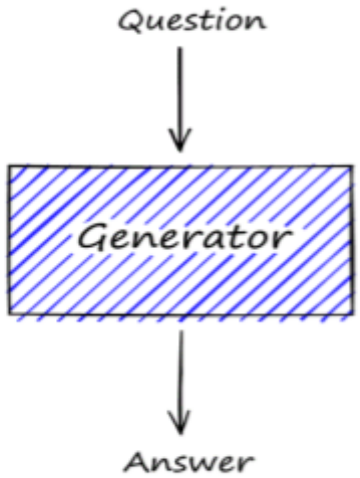
Similar to extractive QA, the initial IR step in open-book abstractive QA involves retrieving pertinent contexts from a third party. The text generation model (like GPT) is given these settings, and it uses them to generate—not extract—an answer. Contexts are used as input (together with the question) to a generative sequence-to-sequence (seq2seq) model rather than being used to extract replies. The query and context are used by the model to produce an answer.



Open book abstractive QA pipeline [9]

### 1.4.4 Closed Book

An alternative option is closed-book abstractive QA. There is no IR step present here; merely a text generation model. The generator model will produce a response using its own internal, learned model of the outside environment. The term "closed book" refers to the fact that it cannot refer to any outside sources of data. Closed-book abstractive QA, In actuality, this is nothing more than a generative model that uses only its own internal knowledge to answer a query. There is no stage for retrieval.



Closed book architecture (simper than generator model) [9]

# Chapter 2

# Dataset and Preprocessing

## 2.1 Dataset

The dataset that we are using for our project is **Russian News Dataset** which is based on the continuous historical archive of notable events in the Russian Subcontinent from Sep 1999 to Dec 2019.

[7] ***Link to dataset:-***

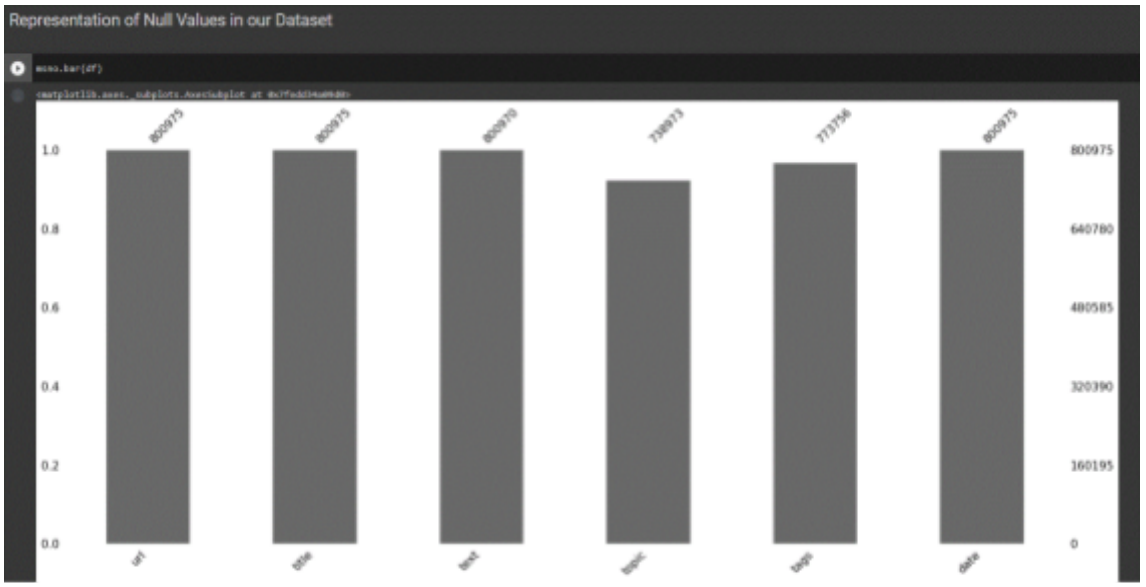
*https://www.kaggle.com/datasets/yutkin/corpus-of-russian-news-articles-from-lenta [7]*

- *Size: 2 Gb*
- *News articles: 800K+*
- *Dates: Sept. 1999 - Dec 2019*
- *Python script[19] used for Downloading News*

## 2.2 Exploratory Data Analysis

EDA is applied to investigate the data and summarize the key insights. It will give you the basic understanding of your data, its distribution, null values and much more. Therefore we checked the same for various important features in our Russian dataset with several results:

- Our dataset contains 800,975 rows and 6 columns.
- The amount of missing values in our dataset is either or negligible and can be replaced by “”.




Representation of NULL values in dataset collected

- The two most important columns in our dataset are title and text which contains the title and the body of the news article respectively and hence we needed to perform several operations on these two features.
- We created a new column which contains the total number of characters in each title and did the same for the number of words.

A new Column containing Character Count of all the titles

```
[ ] df['title_char_count'] = df['title'].str.len()
```

```
[ ] df
```

| | url | title | text | topic | tags | date | title_char_count |
|---|---|---|---|---|---|---|---|
| 0 | https://lenta.ru/news/1914/09/16/hungarnn/ | 1914. Русские войска вступили в пределы Венгрии | Бои у Сольдина и Дурсунки закончились отступ... | Библиотека | Первая мировая | 1914/09/16 | 49 |
| 1 | https://lenta.ru/news/1914/09/16/lermontov/ | 1914. Празднование столетия М.Ю. Лермонтова от... | Министерство народного просвещения, в виду про... | Библиотека | Первая мировая | 1914/09/16 | 53 |
| 2 | https://lenta.ru/news/1914/09/17/nesterof/ | 1914. Das ist Nesteroff! | Штабс-капитан П. Н. Нестеров на днях удивил в... | Библиотека | Первая мировая | 1914/09/17 | 24 |
| 3 | https://lenta.ru/news/1914/09/17/bulldogv/ | 1914. Бульдог-конец под Львовом | Фотограф-корреспондент Daily Mirror рассказыва... | Библиотека | Первая мировая | 1914/09/17 | 31 |
| 4 | https://lenta.ru/news/1914/09/18/zver/ | 1914. Под Люблином пойман шавбский зверь | Лица, приехавшие в Варшаву из Люблина, передаю... | Библиотека | Первая мировая | 1914/09/18 | 40 |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 800970 | https://lenta.ru/news/2019/12/14/shnur/ | Шнуров раскритиковал Гагарину на «Голосе» | Певец Сергей Шнуров раскритиковал слова коллеги... | NaN | ТВ и радио | 2019/12/14 | 41 |
| 800971 | https://lenta.ru/news/2019/12/14/doby/ | В России предложили изменить правила въезжани... | Министерство юстиции России предложило изменит... | NaN | Все | 2019/12/14 | 53 |
| 800972 | https://lenta.ru/news/2019/12/14/dark_euro/ | В России назвали «черную дату» для Европы | Испытание США ранее запрещенной Договором о ли... | NaN | Политика | 2019/12/14 | 41 |
| 800973 | https://lenta.ru/news/2019/12/14/meteo/ | Россиянам пообещали аномально теплую погоду | В ближайшие дни в европейской части России пог... | NaN | Общество | 2019/12/14 | 43 |
| 800974 | https://lenta.ru/news/2019/12/14/olimp/ | В конкурсе прогнозов на АПЛ разыграют 100 тыс... | Ведущие футбольные чемпионаты ушли на зимние к... | NaN | Английский футбол | 2019/12/14 | 79 |

800975 rows × 7 columns

Character count of News titles

A new Column containing word counts of all the titles

```
[ ] df['title_word_count'] = df['title'].apply(lambda x: len(x.split()))
```

```
[ ] df
```

| | url | title | text | topic | tags | date | title_char_count | title_word_count |
|---|---|---|---|---|---|---|---|---|
| 0 | https://lenta.ru/news/1914/09/16/hungarnn/ | 1914. Русские войска вступили в пределы Венгрии | Бои у Сольдина и Дурсунки закончились отступ... | Библиотека | Первая мировая | 1914/09/16 | 49 | 7 |
| 1 | https://lenta.ru/news/1914/09/16/lermontov/ | 1914. Празднование столетия М.Ю. Лермонтова от... | Министерство народного просвещения, в виду про... | Библиотека | Первая мировая | 1914/09/16 | 53 | 6 |
| 2 | https://lenta.ru/news/1914/09/17/nesterof/ | 1914. Das ist Nesteroff! | Штабс-капитан П. Н. Нестеров на днях удивил в... | Библиотека | Первая мировая | 1914/09/17 | 24 | 4 |
| 3 | https://lenta.ru/news/1914/09/17/bulldogv/ | 1914. Бульдог-конец под Львовом | Фотограф-корреспондент Daily Mirror рассказыва... | Библиотека | Первая мировая | 1914/09/17 | 31 | 4 |
| 4 | https://lenta.ru/news/1914/09/18/zver/ | 1914. Под Люблином пойман шавбский зверь | Лица, приехавшие в Варшаву из Люблина, передаю... | Библиотека | Первая мировая | 1914/09/18 | 40 | 6 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 800970 | https://lenta.ru/news/2019/12/14/shnur/ | Шнуров раскритиковал Гагарину на «Голосе» | Певец Сергей Шнуров раскритиковал слова коллеги... | NaN | ТВ и радио | 2019/12/14 | 41 | 5 |
| 800971 | https://lenta.ru/news/2019/12/14/doby/ | В России предложили изменить правила въезжани... | Министерство юстиции России предложило изменит... | NaN | Все | 2019/12/14 | 53 | 7 |
| 800972 | https://lenta.ru/news/2019/12/14/dark_euro/ | В России назвали «черную дату» для Европы | Испытание США ранее запрещенной Договором о ли... | NaN | Политика | 2019/12/14 | 41 | 7 |
| 800973 | https://lenta.ru/news/2019/12/14/meteo/ | Россиянам пообещали аномально теплую погоду | В ближайшие дни в европейской части России пог... | NaN | Общество | 2019/12/14 | 43 | 5 |
| 800974 | https://lenta.ru/news/2019/12/14/olimp/ | В конкурсе прогнозов на АПЛ разыграют 100 тыс... | Ведущие футбольные чемпионаты ушли на зимние к... | NaN | Английский футбол | 2019/12/14 | 79 | 14 |

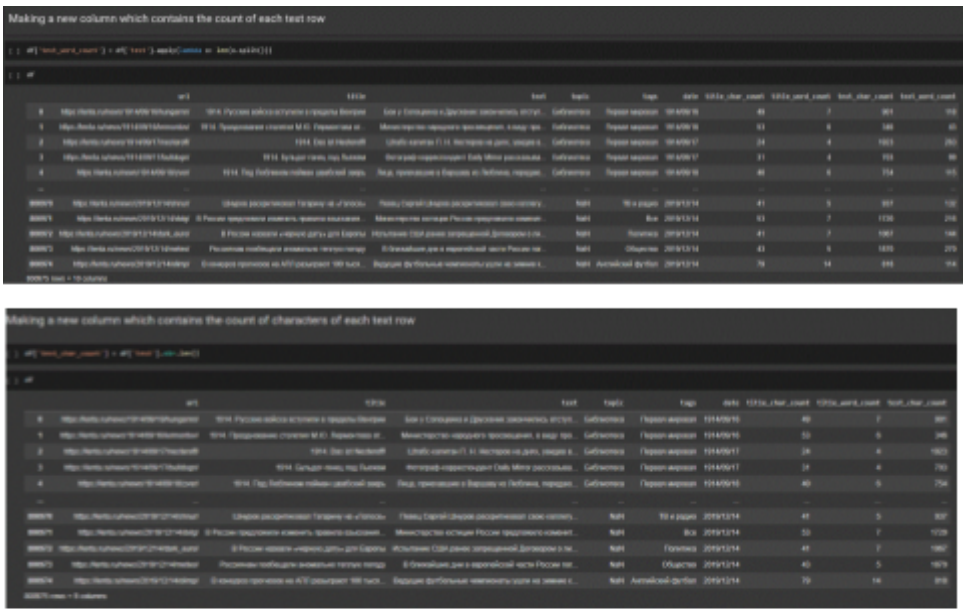
800975 rows × 8 columns

Word count of News Titles

- After finding out the maximum and minimum number of characters and words in our title, we found out that there is no need to change the title and it can be directly fed to our transformer for translation.

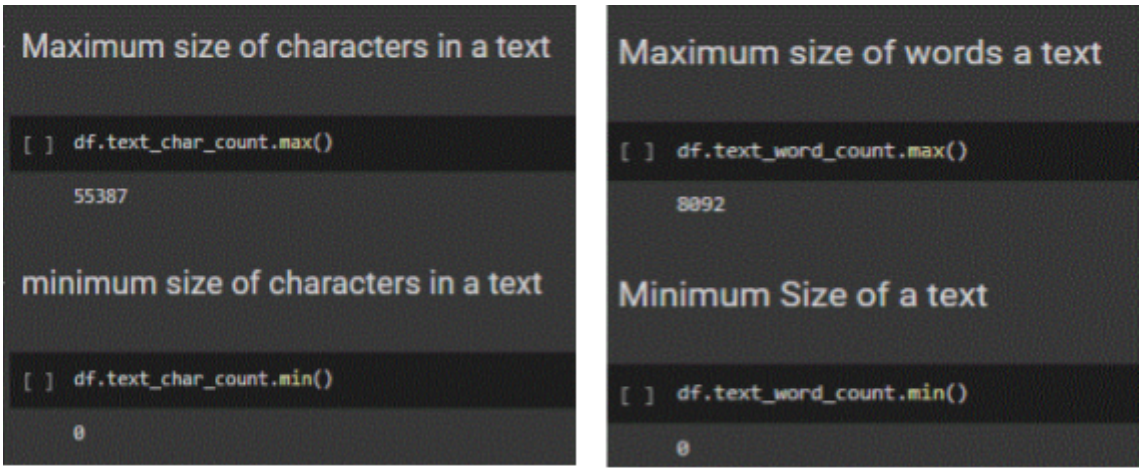

Word and character min max count of news Titles

- We performed the same activities for our text columns which on an average is much bigger than its respective titles.




Word and character count of News body

- After creating the new columns which contain the number of characters and words in the title column, we found that text columns had to be segregated before feeding into the transformer since our transformer at max can take 512 characters at a single point of time.




Min and max count of characters in News Text

- Finally at the end we created a new column by taking the ceiling of the number of characters/512 which tells us how many various segments we need for each and every text row.

While experimenting on artificially easy datasets may result in overly optimistic conclusions that lead the re- search community to abandon potentially fruitful lines of work.

To address the aforementioned issue, we have used **Google's TyDiQA** Dataset and **MKQA dataset.**

## TyDiQA dataset

[4] **TyDiQA** is a question answering dataset covering 11 typologically diverse languages with **204K question-answer pairs**. It is the first public large- scale multilingual corpus of

information-seeking question-answer pairs using a simple-yet-novel data collection procedure that is **model-free** and **translation-free**.

The dataset is made translation free because the process of translation, including human translation, tends to introduce problematic artifacts to the output language such as preserving source-language word order.

Models performing well on this dataset can be generalized to work well across a large number of the world's languages.

There are 2 main tasks that TyDiQA would like researchers to focus on by using the dataset:

1. Passage Selection Task: returns the matching passage which contains the answer or NULL if no such passage exists.
2. Minimal Answer Span Task: Given a selected passage which has the answer, return the answer in minimum letters. YES and NO for closed questions(which only have 2 possible answers)

**Data Collection**

**204K question-answer pairs** are collected in 11 typologically diverse languages**.** To provide a realistic information-seeking task and avoid priming effects, questions are written by people who want to know the answer, but don't know the answer yet, and the data is collected directly in each language without the use of translation.

Process:-

1. **Question elicitation:** Human annotators are given short prompts consisting of the first 100 characters of Wikipedia articles and asked to write questions that they are *actually interested in* knowing the answer to. They were also shown inspiration prompts to generate questions on a wide variety of topics. It was ensured that the questions were not answered in the prompts.

2. **Article retrieval:** A Wikipedia article is then paired with each question by performing a Google search on the question text

3. **Answer labeling:** Finally, annotators are presented with the question/article pair and asked first to select the best passage answer. Annotators were asked to select, if possible, a minimal answer: (Eg. Yes or No).

**Quality control**

To validate the quality of questions, we sampled questions from each annotator and verified with native speakers that the text was fluent.10 We also verified that annotators were not asking questions answered by the prompts. We provided minimal guidance about acceptable questions, discourag- ing only categories such as opinions (e.g. *What is the best kind of gum?*) and conversational questions (e.g. *Who is your favorite football player?*).

## MKQA Dataset

[6] Multilingual Knowledge Questions and Answers (MKQA), an open-domain question answering evaluation set comprising 10k question-answer pairs across 26 diverse languages (260k question-answer pairs in total) where answers are based on heavily curated, language-independent data making results comparable across languages and independent. MKQA selects 10k realistic English queries from the Natural Questions database and human translates them into 25 additional languages and dialects.

**Data Collection**

MKQA aims for certain properties of our evaluation set: (i) realistic questions, (ii) reliable annotations (e.g. via inter-annotator agreement), enabling fair comparison between any approach. The steps involving Dataset collections are:

1. Sample 10k queries, discarding passage-embedded annotations.
2. Annotators research queries to manually generate or select answers.
3. Resolve answers by linking to Wikidata KB and normalizing formats.
4. Answers + Wikidata aliases are verified when ≥ 2 resolved annotations agree, or a domain expert resolves disagreement.
5. Answers are localized to 25 (other) languages by Wikidata QIDs or bilingual translators.
6. Queries are localized to 25 (other) languages by bilingual translators.

## 2.3 Preprocessing the data

### Translation:

We used an open source Russian to English translation model from Hugging face https://huggingface.co/Helsinki-NLP/opus-mt-ru-en[20] and ensured not using black box APIs like google translate to allow the community to verify and reproduce the same results.

### Data Preprocessing:

Data preprocessing is one of the most significant steps in text analytics. The purpose is to remove any unwanted words or characters which are written for human readability, but won't contribute to topic modeling in any way.

- Cleaning the data.
- Case Folding
- Word Tokenization
- Stop words removal
- Lemmatization

## Searching Algorithms with formulae:

- Calculating the ranking by cosine similarity ( tf-idf scoring):

TF => term frequency

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,j}}$$

IDF => Inverse Document Frequency

$$idf(w) = log(\frac{N}{df_t})$$

- Calculating the weight of word in given document by multiplying tf-idf

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

**On creating both the models, we observed:**

1. TF-IDF would give very similar results in any language which has white space token separators(eg. English, Russian, Arabic, Korean etc.)
2. The F1 score of TF-IDF models can be improved if we reduce the number of total unique words in the corpus.

3. Preprocessing steps such as case lowering, stop word removal, stemming and lemmatization are required to achieve the goal of reducing unique words in corpus without reducing/changing the context of the text.

4. Number of unique words in the Russian corpus is 10% more than the number of unique words in the English corpus.

5. Recall of the TF-IDF models could not be calculated as the total number of relevant documents to any query was unknown in the dataset.

| Number of unique words | Before preprocessing | After Preprocessing |
|---|---|---|
| English | 45559 | 21946 |
| Russian | 73974 | 24133 |

# Language 1: Russian

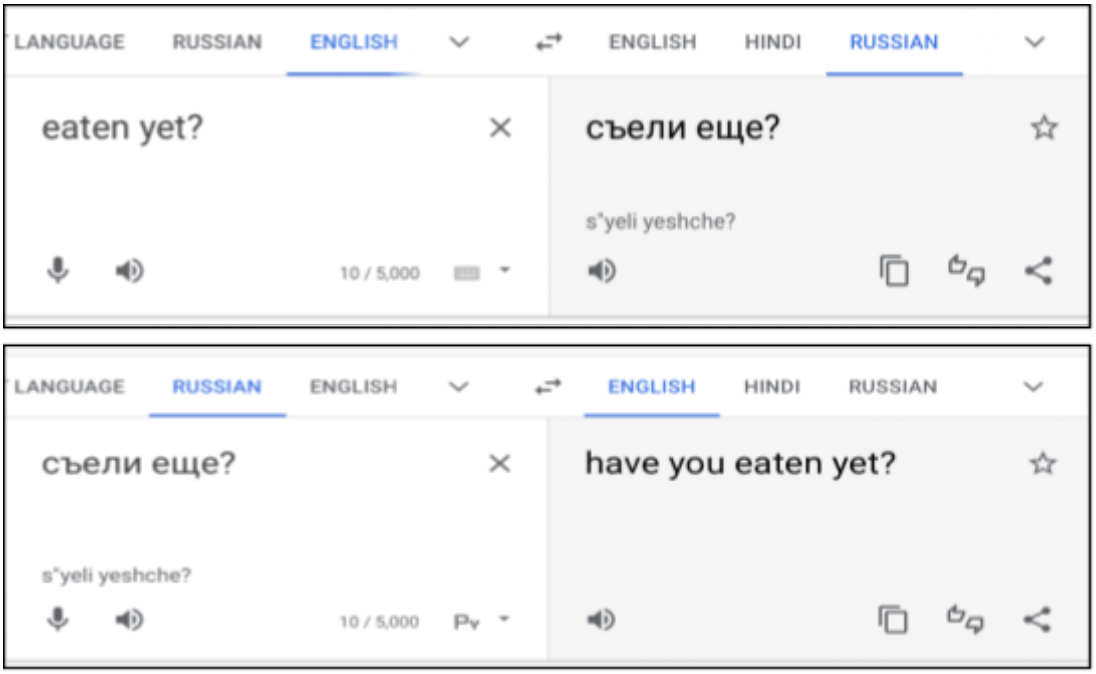
Russian is an Eastern Slavic language using the Cyrillic alphabet. An inflected language, it relies on case marking and agreement to represent grammatical roles. Russian uses singular, paucal,17 and plural numbers. Substantial fusional morphology is used along with three grammatical genders, extensive prodrop, and flexible word order.

The **Paucal number** represents a few instances—between singular and plural. In Russian, paucal is used for quantities of 2, 3, 4, and many numerals ending in these digits.

**Fusional morphology** expresses several grammatical categories in one unsegmentable element. Eg. The components '3rd person possessive' and 'plural' are fused together in the English word their

The three **grammatical genders** used in the Russian language are: Masculine, feminine and neuter. Compared to four grammatical genders in the English language which are: masculine, feminine, common, and neuter.

- Masculine nouns refer to words for a male figure or male member of a species (i.e. man, boy, actor, horse, etc.)
- Feminine nouns refer to female figures or female members of a species (i.e. woman, girl, actress, mare, etc.)
- Common nouns refer to members of a species and don't specify the gender (i.e. parent, friend, client, student, etc.)
- Neuter nouns refer to things that have no gender (i.e. rock, table, pencil, etc.)




Example of extensive prodrop in Russian Language [12]

**Extensive prodrop** in the Russian language refers to proposition drop in the language.

Examples of prodrop in sentences(Fig xx).

~~Are you~~ going to the store?

~~Have you~~ eaten yet?

~~Do you~~ see him often?

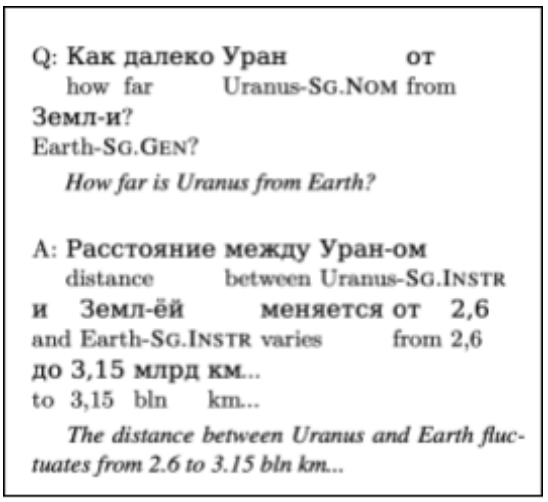



Fig: Example of **morphological variation across question-answer pairs[4]**

Figure: Due to the difference in syntactic context: the entities are identical but have different representation, making simple string matching more difficult. The names of the planets are in the subject and object of the preposition context in the question. The relevant passage with the answer has the names of the planets in a coordinating phrase that is an object of a prepositionBecause the syntactic contexts are different, the names of the planets have different case markings.

## Script switching in Russian

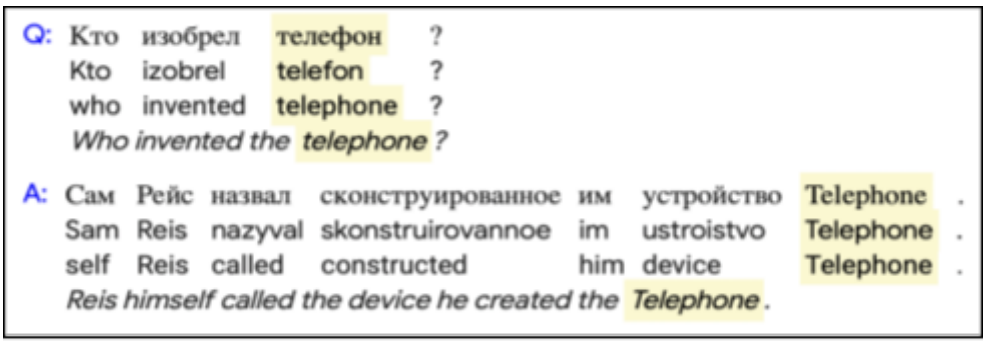



Fig: Example of Script Switching in Russian[17]

[17] This Russian example demonstrates how some entities of non-native origin may maintain the original Latin-script spelling, especially when the term has been directly borrowed into Russian and is phonologically similar to the original. Here the question about the inventor of the telephone contains the more common Cyrillic rendition of the term, 'телефон'. However, the answer passage has it in the original English spelling as 'Telephone'.

## Vowel diacritization in Russian

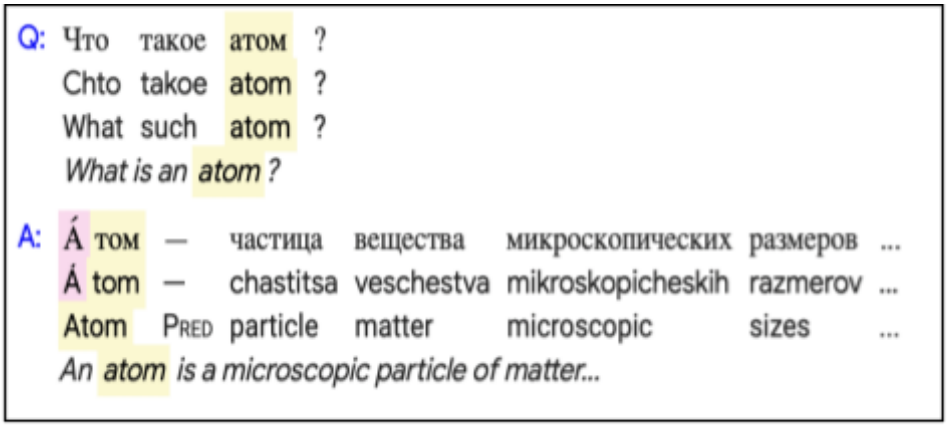



Fig: Example of Vowel Diacritization in Russian Language [17]

[17] This Russian example demonstrates variation in stress marking on Russian vowels. Russian encyclopedia entries typically include stress marks on vowels of the entity name to emphasize correct pronunciation. Here, 'Атом' (atom)

receives an acute diacritic on the first vowel to indicate the word initial stress: 'Átom'. The use of stress marks in questions, however, is very uncommon. This disparity in spelling between the question and the answer poses a challenge for establishing context matching: models may be mislead by the special character and discard the head entity 'Átom' in the answer as a match candidate for 'Atom' in the question – assuming normalized capitalization.

# Chapter 3

# Experimentation and Methodology

We devised 2 approaches to solve the challenges of developing multilingual open information retrieval systems.

### 1. Translation to anchor language

This approach is used by Google Search engine to retrieve documents from other language sources. In this approach, first all the documents from other languages are converted to an anchor language(mostly english) then models and indexes are developed and trained over the translated documents.

**Pros:**

1. Unified model for information retrieval task
2. Easier to improve and debug the
3. Faster results as documents are preprocessed
4. Not required to translate the query.

**Cons:**

1. Preprocessing of all documents are required before retrieval
2. Translation can change the context of text.
3. Translation tends to introduce problematic artifacts to the output language such as preserving source-language word order

### 2. Development of multiple models/indexes

In this approach, we create independent models and indexes for each target language. This approach will keep the documents in its original form and will translate the query to target language and input it to respective models to retrieve relevant documents.

Pros:

1. Pre translation of documents is not required.

2. Text is retaining context and source-language word order.

Cons:

1. Maintain multiple models and indexes.
2. Multiple models would make it difficult to improve and debug.
3. Slower output- real time translation of query and output document is required.

In order to compare the approaches, we decided to create TF-IDF models using both the approaches.

We used an open pretrained translation model. In order to make it an open IR system, we did not use black box API like google translate, so results can be verified and reproduced by others too.

1. TF-IDF model

## Pros of TF-IDF

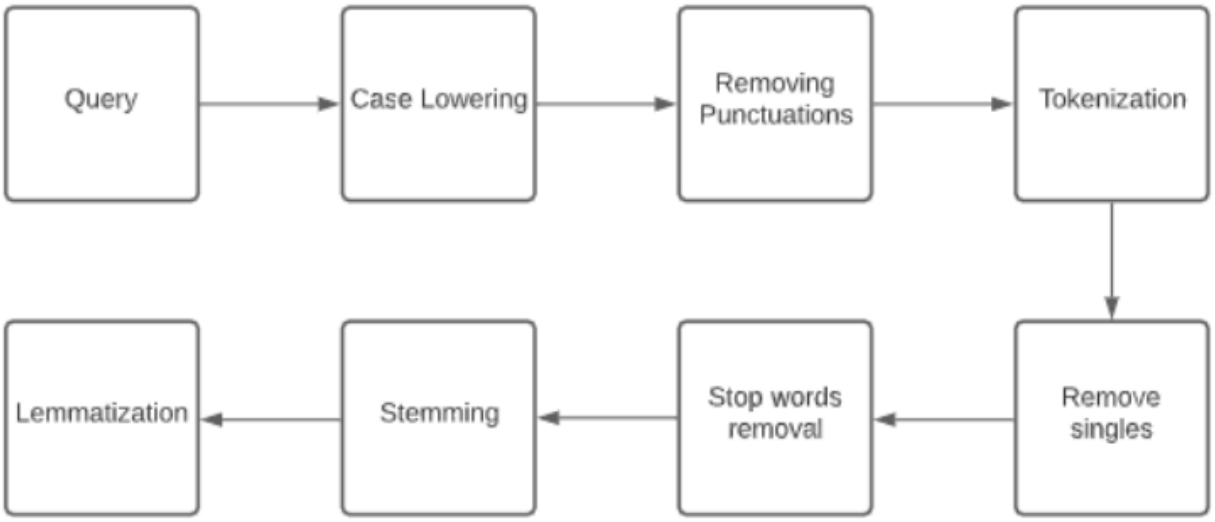
- You have some basic metric to extract the most descriptive terms in a document
- You can easily compute the similarity between 2 documents using it
- Easy to compute

- You have some basic metric to extract the most descriptive terms in a document
- Results are fast, as pre computations are done.

## Cons of TF-IDF

- TF-IDF is based on the bag-of-words (BoW) model, therefore it does not capture position in text, semantics, co-occurrences in different documents, etc.
- For this reason, TF-IDF is only useful as a lexical level feature.
- Cannot capture semantics (e.g. as compared to topic models, word embeddings )
- No support for GPU processing.
- eg. “Orange” fruit and colour would mean the same.

**Fig 1** Data




Preprocessing Phases

**Fig 2** TF-IDF working flowchart

# Libraries used:

**Table 1** Libraries and Packages used

| |
|---|
| Python 3.5+ |
| pip 19+ or pip3 |
| NLTK |
| TensorFlow-GPU |
| ANNOY |

# NumPy

NumPy is a Python library used for working with arrays. In Python we have lists that serve the purpose of arrays, but they are slow to process. NumPy aims to provide an array object that is up to 50x faster than traditional Python lists. NumPy arrays are stored at one continuous place in memory unlike lists, so processes can access and manipulate them very efficiently. This behavior is called locality of reference in computer science. This is the main reason why NumPy is faster than lists. Also it is optimized to work with the latest CPU architectures. NumPy is a Python library and is written partially in Python, but most of the parts that require fast computation are written in C or C++. Works on multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

Numpy was used to handle large corpus of data without running out of memory or wasting CPU resources for processing data.

# Sklearn

The most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistent interface in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistent interface in Python. It is probably the most useful library for machine learning in Python. The sklearn library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction. TfidfVectorizer from the sklearn library was used to find the cosine scores of query and corpus data more efficiently after preprocessing the data.

## Okapi BM25

TF-IDF just takes into account two factors.

1) Term frequency (TF) - Total words in a document divided by the total number of words in that document - TF(w(i),d(i)) (i)

2) Inverse Document Frequency - IDF(w(i),d(i)) = Log(Total Document Count / Documents containing the Word w(i)).

   Inverse document offers less weight to words that occur frequently and greater weight to words that occur less frequently.

But the length of the document is something that TF-IDF overlooks.

As an illustration, a paper with a million words has a greater chance of including all the words, which increases its likelihood of consistently ranking first for any given set of keywords.

This problem is resolved by the BM25 ranking by adding a parameter for document relative length.

$$\text{score}(D,Q) = \sum_{i=1}^{n} \text{IDF}(q_i) \cdot \frac{f(q_i,D) \cdot (k_1+1)}{f(q_i,D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\text{avgdl}}\right)},$$

BM25 Formula [21]

The sum of BM25Score(Machine) and BM25Score(Learning) would be the BM25 for the query "Machine Learning".

The term's IDF is the formula's first component.

The word's TF, which is normalised by its length, is the formula's second part.

The term frequency of the word q(i) in document D is f(q(i),D).

1) The ith query term is qi.

For instance, there is only 1 query term for "shane," thus q0 is "shane." In an English search for "shane connelly," Elasticsearch will detect the whitespace and tokenize the query as two terms: "shane" for q0 and "connelly" for q1. These query terms are added to the other components of the equation and the total is calculated.

2) The inverse document frequency of the ith search word is IDF(gi).
The IDF element of our approach calculates the frequency of a term across all texts and "penalises" frequent terms. For this portion, Lucene/BM25 actually employs the following formula:

$$ln\left(1 + \frac{(docCount - f(q_i) + 0.5)}{f(q_i) + 0.5}\right)$$

actual formula Lucene/BM25 [21]

Where f(qi) is the number of documents that include the ith query term and docCount is the total number of documents that have a value for the field.

3) It is seen that the length of the field is divided by the average field length as fieldLen/avgFieldLen.

This might be interpreted as a document's length in relation to the typical document length. The denominator increases when a document is longer than normal(decreasing the score) and decreases when a document is shorter than usual (increasing the score). Keep in mind that Elasticsearch bases its implementation of field length on the number of words (vs something else like character length). Although we do have an unique flag (discount overlaps) to handle synonyms particularly if you so choose, this is exactly how it was defined in the original BM25 document.

The way to think about this is that the more terms in the document — at least ones not matching the query — the lower the score for the document.

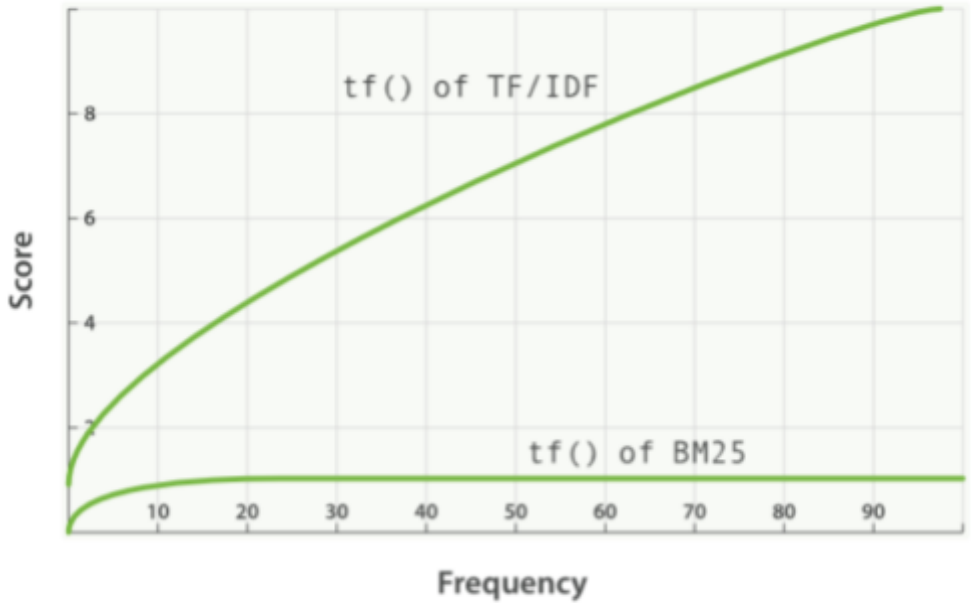
Once more, this is logical: if a paper is 300 pages lengthy and only includes my name once, it is less likely to be about me than a brief tweet that only mentions me once.

4) There is a variable named b that appears in the denominator and is multiplied by the previously described field length ratio. The impacts of the document's length relative to the average length are exacerbated if b is larger. To see this, consider what would happen if you set b to 0, the length ratio would have no effect at all and the score would not be affected by the length of the document. In Elasticsearch, b is set at 0.75 by default.

5) Last but not least, we observe two score factors, k1 and f, which appear in both the numerator and the denominator (qi,D).

The way to think about f(qi,D) is that a document will have a higher score when query term(s) appear in it more frequently. It stands to reason that a document with our name on it multiple times is more likely to be related to us than one with our name on it only once.

The variable k1 influences the features of term frequency saturation. In other words, it restricts how much a single search term can influence a document's score. It accomplishes this by getting close to an asymptote. The following shows how BM25 compares to TF/IDF:




Comparison of BM25 against TF/IDF [21]

A change in the slope of the "tf() of BM25" curve is indicated by a higher/lower k1 value. The way that "terms occuring additional times add extra score" is changed as a result. According to one interpretation of k1, for documents of average length, the word frequency value is what generates a score of 50% of the maximum score for the term under consideration.
The curve of the impact of tf on the score grows quickly when $tf() \leq k1$ and slower and slower when $tf() > k1$.

## CLIR system using Okapi BM25 model-

Here we have tried to build and implement a CLIR system for a dataset where, given a query in German, searches text documents written in English and displaying the results in German.

**Data Used**

- bitext.(en,de): A sentence aligned, parallel German-English corpus, sourced from the Europarl corpus (which is a collection of debates held in the EU parliament over a number of years). We'll use this to develop word-alignment tools, and build a translation probability table.
- newstest.(en,de): A separate, smaller parallel corpus for evaulation of the translation system.
- devel.(docs,queries,qrel): A set of documents in English (sourced from Wikipedia), queries in German, and relevance judgement scores for each query-document pair.

The CLIR system will:

- **translate queries** from German into English (because our searchable corpus is in English), using word-based translation, a rather simplistic approach as opposed to the sophistication you might see in, say, *Google Translate*.

- **search over the document corpus** using the Okapi BM25 IR ranking model, a variation of the traditional TF-IDF model.
- **evaluate the quality** of ranked retrieval results using the query relevance judgements.

Steps involved -

Here's an overview of the tasks involved:

- Loading the data files, and tokenizing the input.
- Preprocessing the lexicon by stemming, removing stopwords.
- Calculating the TF/IDF representation for all documents in our wikipedia corpus.
- Storing an inverted index to efficiently documents, given a query term.
- Implementing querying with BM25.
- Test runs.

# Chapter 4

# Working Model

We downloaded Wikipedia data in 8 languages including english and using Bert the DPR project by facebook research team(https://github.com/facebookresearch/DPR)[24] Dense Passage Retrieval (DPR) - is a set of tools and models for state-of-the-art open-domain Q&A research, we created dense embeddings of all the data (8 hrs)
Paper of DPR by facebook research[25]

```
In [27]: data_to_dense_embeddings()
1/1 [==============================] - 0s 33ms/step                [ time left: ? ]
1/1 [==============================] - 0s 34ms/step          [ time left: 8:14:32 ]
1/1 [==============================] - 0s 33ms/step          [ time left: 8:11:01 ]
1/1 [==============================] - 0s 33ms/step          [ time left: 8:08:28 ]
1/1 [==============================] - 0s 33ms/step          [ time left: 8:08:17 ]
1/1 [==============================] - 0s 32ms/step          [ time left: 8:09:06 ]
1/1 [==============================] - 0s 34ms/step          [ time left: 8:08:33 ]
1/1 [==============================] - 0s 33ms/step          [ time left: 8:08:22 ]
```

Converting Wikipedia Data to Dense Embeddings

```
[[[4.73616354e-11 6.08903761e-11 6.71326467e-11 ... 7.35093444e-11
   6.83403473e-11 6.78980691e-11]
  [8.54597035e-12 8.95919363e-12 1.42811318e-11 ... 1.56710287e-11
   1.32579130e-11 1.05442755e-11]
  [1.20775785e-12 1.23653106e-12 1.26108781e-12 ... 1.69923017e-12
   1.41668708e-12 1.32877194e-12]
  ...
  [5.47548531e-08 4.21173034e-08 4.04112157e-08 ... 4.25296491e-08
   5.09019031e-08 4.29032916e-08]
  [1.63107501e-07 1.24722490e-07 1.13035455e-07 ... 1.28184212e-07
   1.44949794e-07 1.33982397e-07]
  [2.21718722e-07 1.64885094e-07 1.49764929e-07 ... 1.68960639e-07
   2.01739013e-07 1.77733270e-07]]]
```

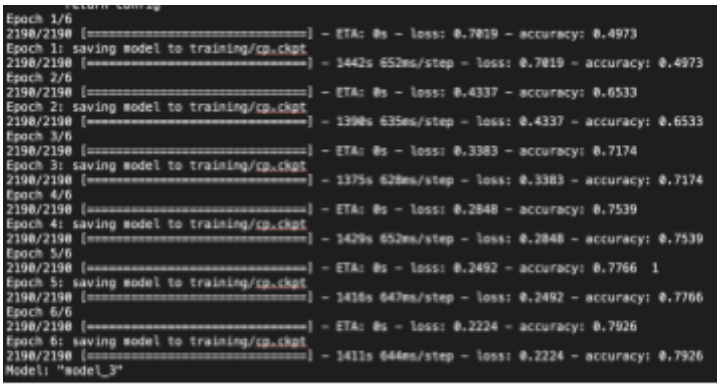
Example Embedding

**Features of Dense Passage Retrieval**
Dense retriever model is based on bi-encoder architecture.
Extractive Q&A reader&ranker joint model inspired by [22]

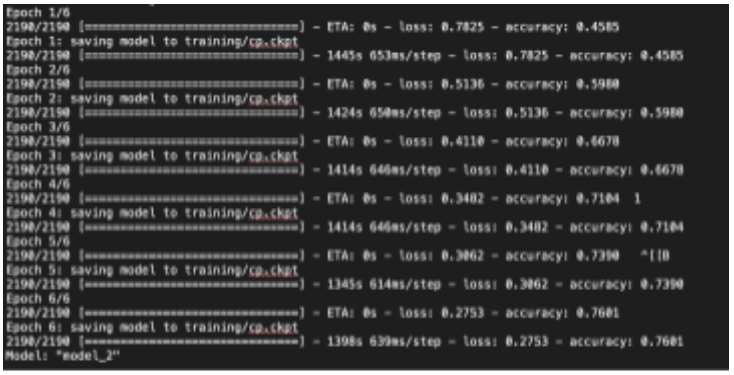
Related data pre- and post- processing tools.
Dense retriever component for inference time logic is based on the FAISS index.

We trained our multilingual QA model using SQuAD 2.0[23] Dataset. SQuAD is a question-answer dataset that poses a question with a context along with the identified answer. It's also possible to not have an answer. See the SQuAD dataset website[23].
Stanford Question Answering Dataset (SQuAD) is a reading comprehension dataset, consisting of questions posed by crowdworkers on a set of Wikipedia articles, where the answer to every question is a segment of text, or span, from the corresponding reading passage, or the question might be unanswerable[23]

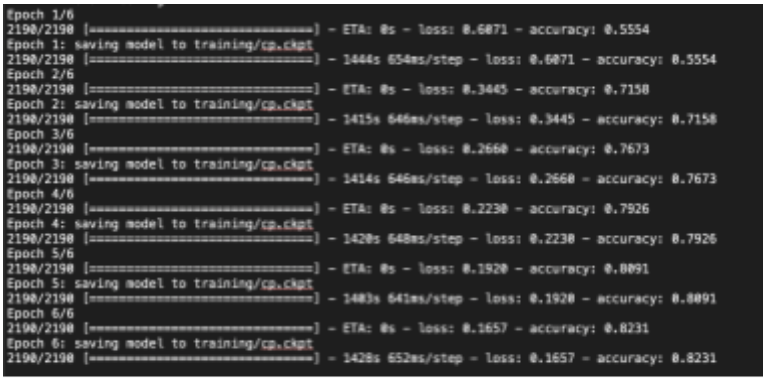
We attempted training the QA model using different learning rates and kept maximum epoch=6. We trained the model on AWS



Training QA model at learning rate = 0.00003



Training QA model at learning rate = 0.00002




Training QA model at learning rate = 0.00005

Training each model took 7 hrs (~1hr per epoch)
On following the configuration EC2 instance
AWS EC2 instance used:
Instance name: g4ad.8xlarge
GPU: 2 cores (Nvidia)
CPU: 32 cores
Memory: 128gb
Disk 250gb
Cost:$1.734/hr

## Model Architecture

```
Model: "model_4 architecture"
_________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
=================================================================================================
 input_word_ids (InputLayer)    [(None, 128)]        0           []

 input_mask (InputLayer)        [(None, 128)]        0           []

 segment_ids (InputLayer)       [(None, 128)]        0           []

 bert_layer_4 (BertLayer)       (None, 128, 768)     109482241   ['input_word_ids[0][0]',
                                                                  'input_mask[0][0]',
                                                                  'segment_ids[0][0]']

 tf.math.not_equal_4 (TFOpLambd  (None, 128)         0           ['input_mask[0][0]']
 a)

 time_distributed_4 (TimeDistri  (None, 128, 30522)  23471418    ['bert_layer_4[0][0]',
 buted)                                                           'tf.math.not_equal_4[0][0]']

=================================================================================================
Total params: 132,953,659
Trainable params: 44,735,034
Non-trainable params: 88,218,625
_________________________________________________________________________________________________
```

Input_word_ids are the indices that correspond to each token.

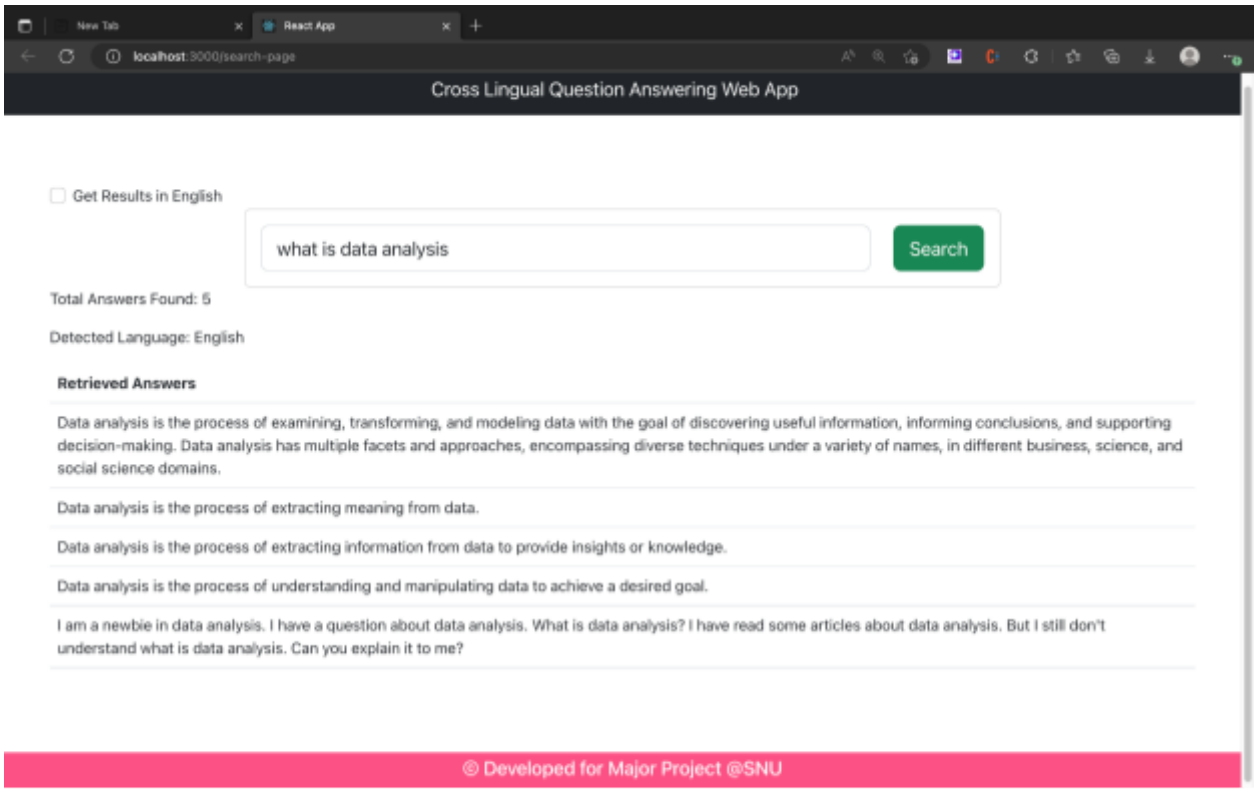
Mask IDs to indicate which elements in the sequence are tokens and which are padding elements.

Segment IDs used to distinguish different sentences.

'trainable parameters' are those whose value is modified according to their gradient (the derivative of the error/loss/cost relative to the parameter), whereas 'non-trainable parameters' are those whose value is not optimized according to their gradient.

Output on example question:

We created an easy to use user interface for getting answers from our trained model.




User Interface of our trained model

Features:

1. It is a cross lingual QA platform, questions can be asked in any language.
2. Answers will be shown on the interface in the same language as question
3. Answers can be requested in English language by checking the checkbox “Get Results in English”
4. Laguage of the question will be auto detected and will be displayed with the answers retrieved
5. QA platform sports chaching feature to reduce the time taken by model to give output for any question that is repeated.

**Caching**

Search enigines can drastically improve their performance and reduce query processing time by using chahing. In our QA model, caching is done by translating the results to an anchor language to increase efficiency and are stored in a dictionary(hash map) as key-value pairs.

**Advantages of caching**

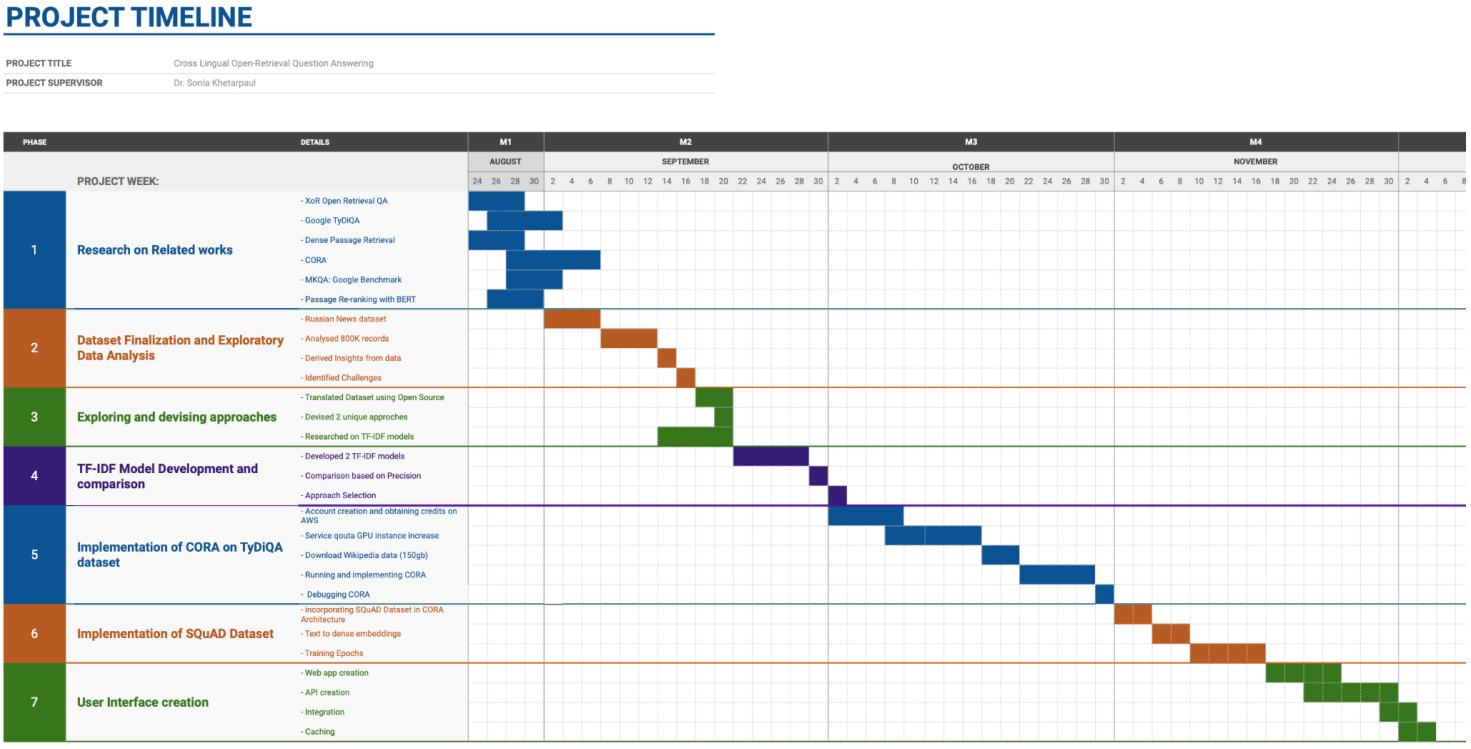
1. Faster search results
2. Less load on server
3. Improves application performance

# Chapter 5

# Summary and Conclusions

We offered an improved technique through our project, which can provide cross lingual answers to open domain questions. We got our dataset from kaggle.com for Russian news headlines and used the above-mentioned algorithm for our experiments. We will continue to work on improving the algorithm in the future.

# Project Timeline



# Future Work

To address the information needs of many non-English speakers, a QA system has to conduct crosslingual passage retrieval and answer generation. This many-to-many open QA model retrieves multilingual passages in many different languages and generates answers in target languages. It does not require language-specific translation or retrieval components and can even answer questions in unseen, new languages. We conduct extensive experiments on two multilingual open QA datasets and SQuad dataset across 8 languages, outperforming competitive models by up to 23 F1 points. Our extensive analysis and manual evaluation reveal that this model effectively retrieves semantically relevant passages beyond language boundaries, and can even find answers to the questions that were previously considered unanswerable due to lack of sufficient evidence in annotation languages (e.g., English). Nonetheless, our experimental results show that the retrieval component still struggles to find relevant passages for queries in some unseen languages. In future work, we aim to address these issues to further improve the performance and scale our framework to even more languages.

We also aim to improve and create a multilingual embedding space for our model by combining the idea of language knowledge transfer with a fresh cross-lingual consistency training method.

# References


1. Akari Asai, Jungo Kasai, Jonathan H. Clark (2021) 'XOR QA: Cross-lingual Open-Retrieval Question Answering', 3rd edition, 18, article number 10.48550, doi: https://doi.org/10.48550/arXiv.2010.11856
2. Stanford NLP (2020) 'PrimeQA' available at: https://primeqa.github.io/primeqa/ (Accessed: 21 Oct 2022)
3. Akari Asai (2020) 'XOR QA: Cross-lingual Open-Retrieve Question Answering' available at: https://github.com/AkariAsai/XORQA (Accessed: 21 Oct 2022)
4. Jonathan H. Clark, Eunsol Choi (2020) 'TyDi QA: A

Benchmark for Information-Seeking Question Answering in Typologically Diverse Languages', v1, 17, article number: 10.48550, doi: https://doi.org/10.48550/arXiv.2003.05002
5. Akari Asai, Xinyan Yu, Jungo Kasai, Hannaneh Hajishirzi (2021) 'One Question Answering Model for Many Languages with Cross-lingual Dense Passage Retrieval', v2, 23, article number: 10.48550, doi: https://doi.org/10.48550/arXiv.2107.11976
6. Shayne Longpre, Yi Lu, Joachim Daiber (2021) 'MKQA: A Linguistically Diverse Benchmark for Multilingual Open Domain Question Answering' v2, 17, article number: 10.48550 doi: https://doi.org/10.48550/arXiv.2007.15207
7. Lenta Ru (2019) 'News dataset from Lenta.Ru' available at: https://www.kaggle.com/datasets/yutkin/corpus-of-russian-news-articles-from-lenta (Accessed: 20 Oct 2022)
8. Google Central Search (2022) 'In-depth guide to how Google Search works' available at: https://developers.google.com/search/docs/fundamentals/how-search-works (Accessed: 20 Oct 2022)
9. Dongmei Chen, Sheng Zangh, Xin Zangh, Kaijing Yang (2020), 'Cross-lingual passage re-ranking with alignment augmented multilingual Bert. Available at: https://www.researchgate.net/publication/347292087_Cross-Lingual_Passage_Re-Ranking_With_Alignment_Augmented_Multilingual_BERT (Accessed: 20 Oct 2022), DOI:http://dx.doi.org/10.1109/ACCESS.2020.3041605
10. Nicolas Bertagnolli (2020) 'Translate Any Two Languages in 60 Lines of Python' available at: https://towardsdatascience.com/translate-any-two-languages-in-60-lines-of-python-b54dc4a9e739 (Accessed: 20 Oct 2022)
11. Neural Networks and Deep Learning lab (2018) 'TF-IDF Ranker' Available at: http://docs.deeppavlov.ai/en/master/features/models/tfidf_ranking.html (Accessed: 20 Oct 2022)
12. Google LLC. (2022) 'Google Translate' Available at: https://translate.google.com (Accessed: 20 Oct 2022)
13. Zhuolin Jiang, Amro El-Jaroudi, William Hartmann, Damianos Karakos, Lingjun Zhao (2020) 'Cross-lingual Information Retrieval with BERT', v1, 6, article number: 10.48550, doi: https://doi.org/10.48550/arXiv.2004.13005
14. Nishanth N (2020) 'Question Answering System with BERT' Available at:

https://medium.com/analytics-vidhya/question-answering-system-with-bert-ebe1130f8def [Accessed: 20 Oct 2022]
15. Ajit Rajput (2020) 'Semantic Search Engine using NLP' Available at: https://www.kaggle.com/code/ajitrajput/semantic-search-engine-using-nlp/notebook [Accessed: 20 Oct 2022]
16. John Snow Labs (2022) 'Russian Lemmatizer' Available at: https://nlp.johnsnowlabs.com/2020/03/12/lemma_ru.html#how-to-use (Accessed: 20 Oct 2022)
17. Google Research (2020) 'TyDi QATypologically Diverse Question Answering' Available at: https://ai.google.com/research/tydiqa [Accessed: 20 Oct 2022]
18. James Briggs, 'An Introduction to Open Domain Question-Answering' , Pinecone [Online]. Available at: https://www.pinecone.io/learn/question-answering/ [Accessed: 5-Dec-2022].
19. Dmitry Yutkin, 'Collect Russian News Dataset', Github (@yutkin) [Online]. Available at: https://github.com/yutkin/Lenta.Ru-News-Dataset/blob/master/download_lenta.py [Accessed: 5-Dec-2022].
20. Open Source Community, 'Language Translators', Hugging Face [Online]. Available at: https://huggingface.co/Helsinki-NLP/opus-mt-ru-en [Accessed: 5-Dec-2022].
21. Shane Connelly, 'Practical BM25 - Part 2: The BM25 Algorithm and its Variables', elastic (19 APRIL 2018) [Online]. Available at: https://www.elastic.co/blog/practical-bm25-part-2-the-bm25-algorithm-and-its-variables [Accessed: 5-Dec-2022].
22. Sewon Min, Danqi Chen, Luke Zettlemoyer, Hannaneh Hajishirzi, 'Knowledge Guided Text Retrieval and Reading for Open Domain Question Answering', Submitted on 10 Nov 2019 (v1), last revised 13 Apr 2020 (this version, v2) Available at: https://arxiv.org/abs/1911.03868 [Accessed: 5-Dec-2022].
23. Pranav Rajpurkar, 'SQuAD2.0 The Stanford Question Answering Dataset', Github (@rajpurkar), [Online], Available at: https://rajpurkar.github.io/SQuAD-explorer/ [Accessed: 5-Dec-2022].
24. Facebook Research Team, 'Dense Passage Retrieval', Github (2020 @facebookresearch) [Online], Available at: https://github.com/facebookresearch/DPR [Accessed: 5-Dec-2022]
25. Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, Wen-tau Yih,

‘Dense Passage Retrieval for Open-Domain Question Answering’, Submitted on 10 Apr 2020 (v1), last revised 30 Sep 2020 (this version, v3), Available at: https://arxiv.org/abs/2004.04906 [Accessed: 5-Dec-2022].

Signature of group Members

Signature of Project Advisor
(Dr. Sonia Khetarpaul)