ANALYSIS AND EXTENSION OF BERTSERINI

Open-Domain Question Answering framework

IntelligenzeNaturali

Armenia Sara 279590
Lacriola Francesco 292129
Tomasello Noemi 272600

1859
Politecnico di Torino

Question Answering

Question Anwering attempts to find out the **correct answer** to a **question posed in natural language by humans**.

QA can be addressed using **Information Retrieval systems** that take a natural language question, process it, retrieve and rank the most relevant passages.

The returned answer is in the form of short texts rather than a list of relevant documents.

Closed-domain QA vs open-domain QA

In **closed-domain QA** questions and answers relate to specific fields and exploit **domain-specific knowledge.**

Open-domain question answering deals with questions about nearly anything, and can only rely on **world knowledge**. These systems usually have much more data available to extract the answer from.

Open-domain Retrieval QA and Reading comprehension

Open-retrieval QA focuses on the most general setting in which we need to **retrieve relevant documents from a large corpus.** Then they are processed to identify the most relevant answer with machine reading comprehension systems.

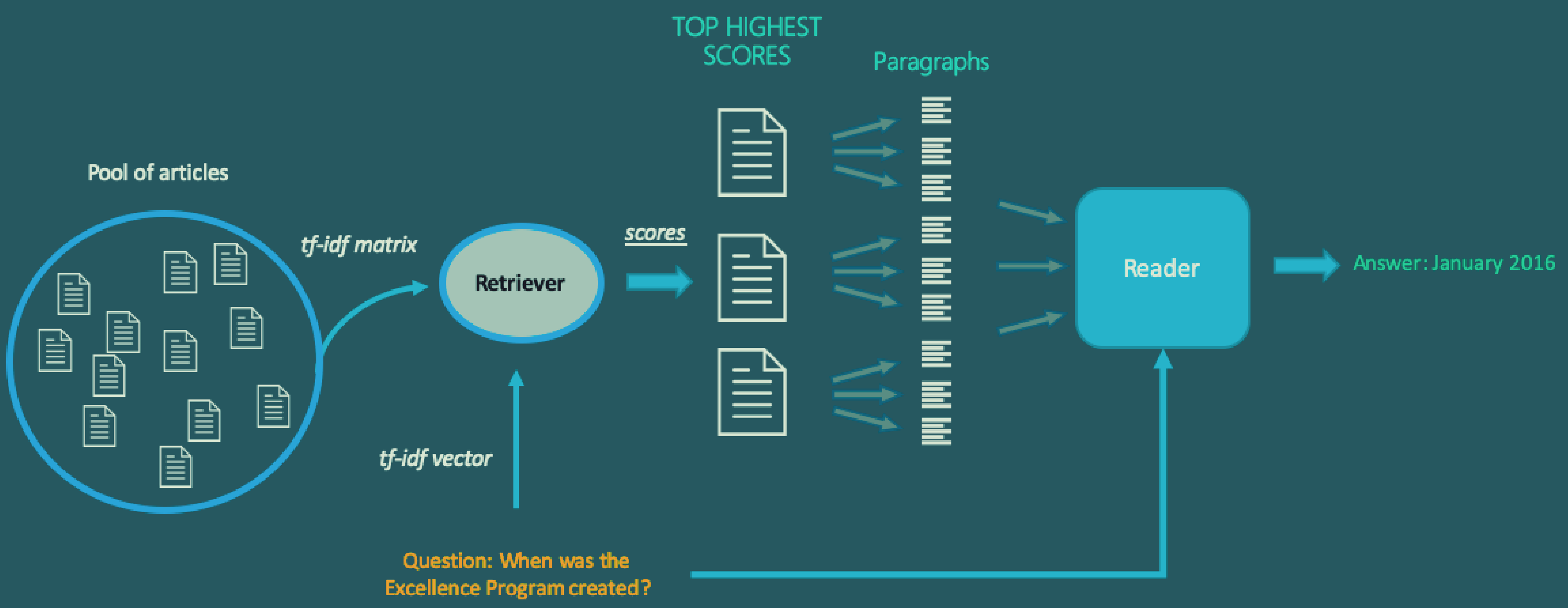
Reading comprehension is the ability to **read a piece of text and then answer questions** about it. Machine reading comprehension assumes that we have access to the ”gold” paragraph that contains the answer. The task is to **discover the shortest fragment of text in the document that answers the user's query.**

Open-domain retrieval QA

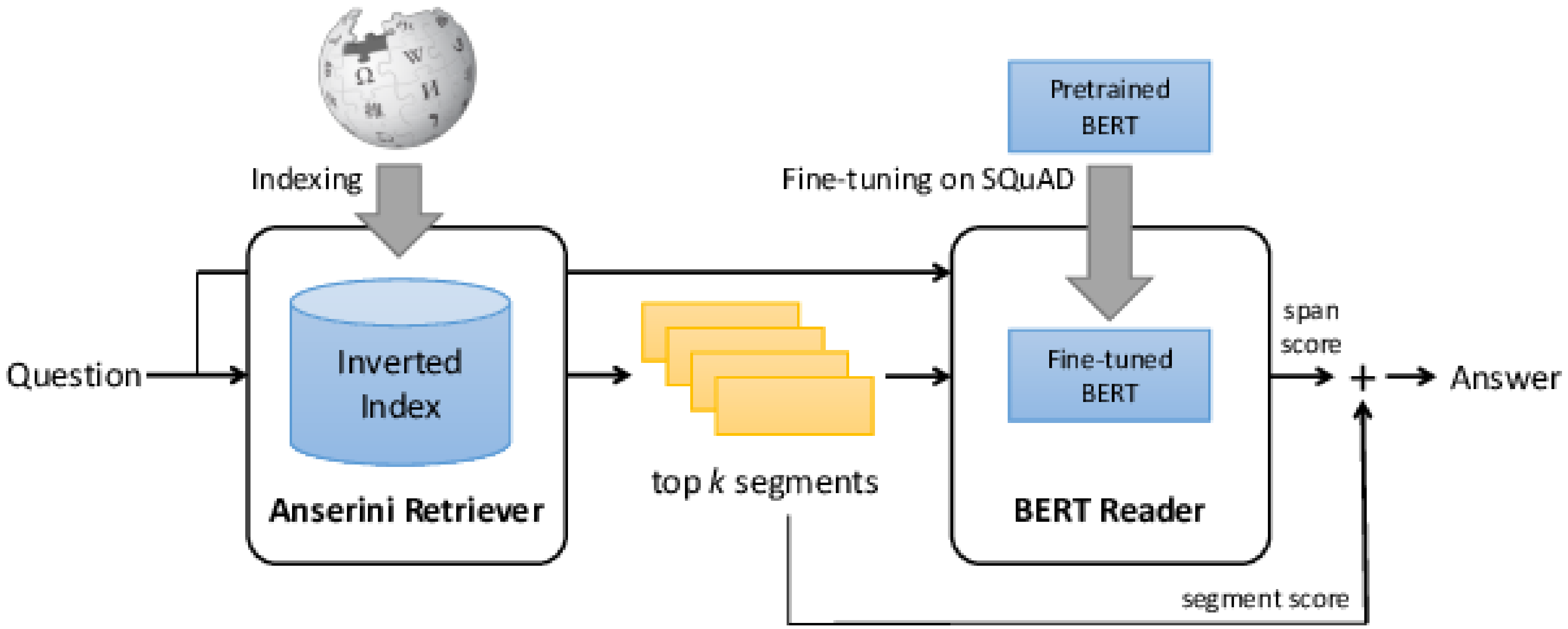
To implement the retriever we can use the **classic information retrieval systems** or **embedding vectors by neural networks**

Before BERT, the reader was mainly implemented with **Bi-directional LSTM**

The retriever and the reader components can be **trained independently** or **end-to-end**



BERTserini



Wei Yang et al: End-to-End Open-Domain Question Answering with BERTserini

BERTserini system is made up of two main components: the open-source Anserini information retrieval and BERT for machine reading comprehension

It is a single-stage retriever that identifies segments of text from Wikipedia to pass to the reader.

Information retrieval searcher is built on top of Lucene.

BM25 is used as ranking function.

Reader: BERT

Pre-training is done taking into account two main objectives

We use the pre-trained model **bert-base-uncased.** It is pre-trained on BookCorpus in a self supervised fashion.
Then, Bert is finetuned on SQuAD dataset.

Masked langauge modeling: taking a sentence, the model randomly masks 15% of the words, run the masked sentence through the model and has to predict them.

Next sentence prediction: the model concatenates two masked sentences as input during pre-training. It has to predict whether two sentences are following each other or not.

Experiments

- **Model variation**
- Data Enrichment
- Machine Translation

DistilBERT: a distilled version of BERT

Knowledge distillation: compression technique in which a compact model, the student, is trained to reproduce the behaviour of a larger model, the teacher.

Distillation loss: the model is trained to return the same probabiltiies as the BERT model

Masked language modeling: this is part of the original training loss of the BERT model

Cosine embedding loss: the model is trained to generate hidden states as close as possible as the BERT base model

RoBERTa: a robustly optimized BERT pretraining approach

The model is **trained longer**, with **larger batches**, considering a **bigger dataset**

Use of **dynamic masking** instead of static masking

The training is made on **longer sequences without nex sentence prediction** objective

Use of **larger byte-level Byte-Pair Encoding**

RESULTS: Finetuning on SQuAD

Model type	EM	F1-score	Training Time
BERT	74,78	86.68	4h 50m
DistilBERT	72.59	81.52	3h 20m
RoBERTa	85.40	91.63	5h 05m

Experiments

- Model variation
- **Data Enrichment**
- Machine Translation

Data Enrichment

1

Trivia QA

Reading comprehension dataset with over 650K question-answer-evidence triplets. The evidence documents are taken from Wikipedia and general Web sources.

2

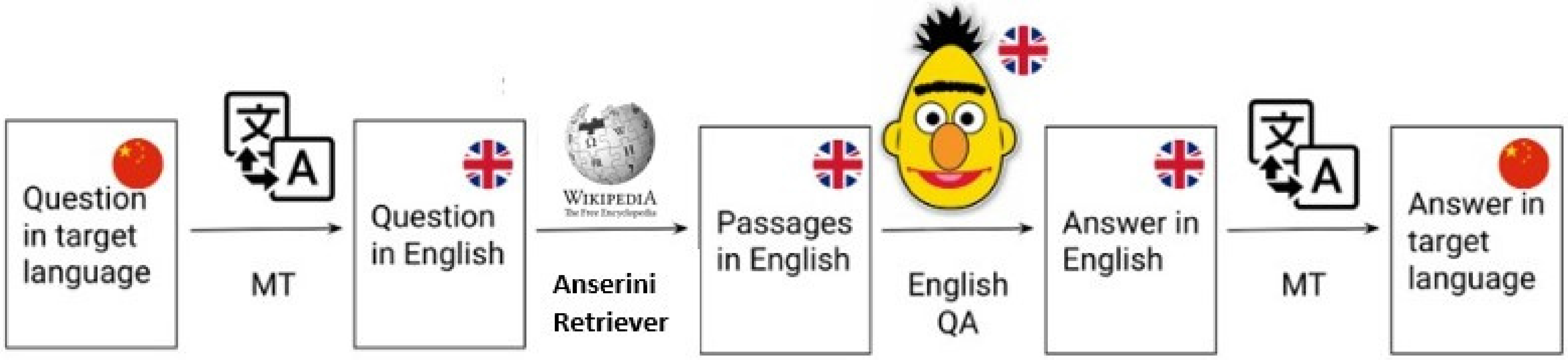
SQuAD

Leading dataset to perform QA. It is made of question-answer pairs posed by crowd-workers on Wikipedia articles.

Experiments

- Model variation
- Data Enrichment
- **Machine Translation**

Machine Translation



Metrics

Exact match: percentage of the predictions that strictly match the ground truth answer. It is a binary function, EM=1 if the answer is correct, EM = 0 otherwise.

(Macro-averaged) F1 score: computed over individual words of the prediction against those in the ground truth. It is based on Precision and Recall metrics.

RESULTS: BERTserini

	EM	F1-score
BERTserini (BERT on SQuaD)	25.0	29.76
BERTserini (DistilBERT on SQuAD)	24.3	28.19
BERTserini (RoBERTa on SQuAD)	29.81	32.33
BERTserini (BERT on TriviaQA + SQuAD)	25.0	33.09

Conclusions

The goal of our project was to create a valid replication of BERTserini. Despite the hardware constraints provided by Colab, we can appreciate the results reached with the model variations and the surprising data enrichment experiment.

Thank you