# NLP Report question-answers

Quentin Le Helloco, Etienne Sharpin, Ségolène Denjoy

November 24, 2021

## 1 Introduction

For this project, we have coded a question-answering engine. The engine is composed of the following parts.

- A search engine based on semantic similarity embedding
- A nearest neighbour approximation index
- An extracting QA model

Link to the subject: here

Figure 1: Image credit: msmefinanceta.eu

# Contents

# 2 The Datasets

## 2.1 SQuAD v2 dataset

Let's look at the SQuAD v2 dataset:

### 2.1.1 First look

Train set: 20 000 elements
Validation set: 5 000 elements

One element looks like that :

```
{'id': '56be85543aeaaa14008c9063',
 'title': 'Beyonce',
 'context': 'Beyonce Giselle Knowles-Carter (/...,
 'question': 'When did Beyonce start becoming popular?',
 'answers': {'text': ['in the late 1990s'],
 'answer_start': [269]}}
```

Minimum length of a context: 151 characters
Maximum length of a context: 3076 characters
Mean length of contexts: 598 characters
The lengths are clearly unbalanced.

### 2.1.2 Categories Distribution

Here's the number of elements per category. We can see that the dataset is not well balanced : some categories have many items (American Idol with 753) and some are under-represented (Warsaw Pact with 146 items).

```
Counter({'2008_Sichuan_earthquake': 521,
         '2008_Summer_Olympics_torch_relay': 500,
         'Adult_contemporary_music': 216,
         'Alexander_Graham_Bell': 325,
         'American_Idol': 790,
         ...
         'Warsaw_Pact': 146,
         'Wayback_Machine': 208,
         'Web_browser': 192})
```

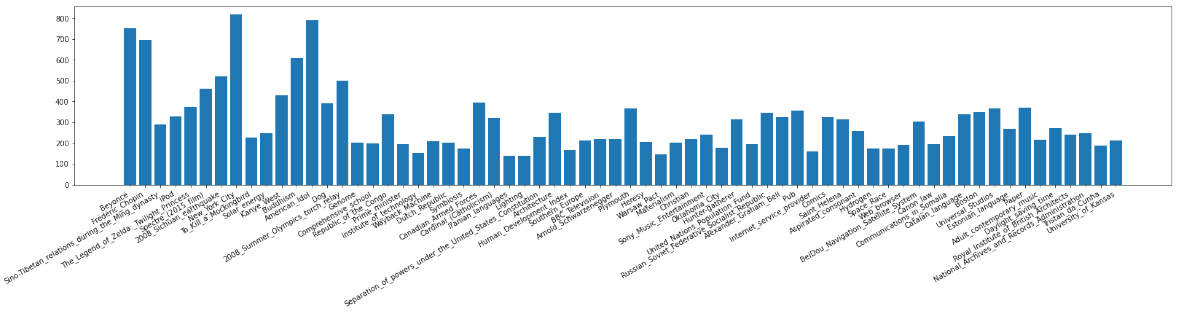

Figure 2: SQuAD v2 categories

Min value : 'Iranian languages' with 140 elements
Max value : 'New York City' with 817 elements
Mean = 303 elements

When looking at the train dataset categories, we can think that when using it, there will be a biais towards american (or english-speaking) culture as many categories are about it and they have most of the elements.

## 2.2 DBPedia entity dataset

We will take some of the titles and their text (context) to add to the SQuAD v2 dataset to add more noise for the MRR calculus. It will also help to have more possible context from which we can pick an answer for a general and unknown question.

Dataset size : 4635922 elements

It's composed of 3 parts: corpus, queries, qrels. As we would only use the text, only the 'corpus' part was used.

One element looks like that:

```
('<dbpedia:Animalia_(book)>',
 {'text': "Animalia is an illustrated children's book by Graeme Base. ...",
  'title': 'Animalia (book)'})
```

Minimum length of a context: 1 character (".", for 8 elements)
Maximum length of a context: 42891 characters (for Agrilus, which is a insect, as the wikipedia page lists all the kinds of Agrilus)
Mean length of contexts: 289 characters
For this dataset, the lengths are even more unbalanced.

Figure 3: https://en.wikipedia.org/wiki/Agrilus

# 3 Train a question-answering model

First we will fine-tune a basic model to see the basic performance we can expect from a model which has not been created for this task, then we will take the model all trained for the task by the teacher.

## 3.1 Fine tune the model

The first model we will take is a basic model named **distilbert-base-uncased**.

### 3.1.1 Evaluation

Here are the metrics to evaluate a model :

- 'exact': Exact match (the normalized answer exactly match the gold answer)
- 'f1': The F-score of predicted tokens versus the gold answer
- 'total': Number of score considered
- 'HasAns_exact': Exact match (the normalized answer exactly match the gold answer)
- 'HasAns_f1': The F-score of predicted tokens versus the gold answer
- 'HasAns_total': Number of score considered
- 'NoAns_exact': Exact match (the normalized answer exactly match the gold answer)
- 'NoAns_f1': The F-score of predicted tokens versus the gold answer
- 'NoAns_total': Number of score considered
- 'best_exact': Best exact match (with varying threshold)
- 'best_exact_thresh': No-answer probability threshold associated to the best exact match
- 'best_f1': Best F1 (with varying threshold)
- 'best_f1_thresh': No-answer probability threshold associated to the best F1

Evaluation of the fine-tuned model :

| | |
|---:|:---|
| exact | 45.72 |
| f1 | 49.77346504131027 |
| total | 5000 |
| HasAns_exact | 51.21555915721232 |
| HasAns_f1 | 59.427603406220214 |
| HasAns_total | 2468 |
| NoAns_exact | 40.363349131121645 |
| NoAns_f1 | 40.363349131121645 |
| NoAns_total | 2532 |
| best_exact | 50.9 |
| best_exact_thresh | 0.0 |
| best_f1 | 53.1008787674189 |
| best_f1_thresh | 0.0 |

## 3.2 Pre-trained model

### 3.2.1 Evaluation

The second model we will take is a distilbert model fine-tuned by the teacher named **distilbert-base-uncased-finetuned-squad2**.

Evaluation of the Pre-trained model :

| | |
|---|---|
| exact | 65.3 |
| f1 | 68.988572537942 |
| total | 5000 |
| HasAns_exact | 67.09886547811993 |
| HasAns_f1 | 74.5716623540156 |
| HasAns_total | 2468 |
| NoAns_exact | 63.54660347551343 |
| NoAns_f1 | 63.54660347551343 |
| NoAns_total | 2532 |
| best_exact | 65.36 |
| best_exact_thresh | 0.0 |
| best_f1 | 69.00857253794186 |
| best_f1_thresh | 0.0 |

We can see that the f1-score of the pre-trained model is better than the first model. We will use this model for the rest of the project.

## 3.3 Badly classified examples

To see where the model struggles, we can fetch some wrong answer and search for pattern which could explain why the predictions were bad.

### 3.3.1 Basic prediction

Let's look at wrongly predicted answer for our fined-tuned basic model.

#### Wrong answer number #1

```
{'id': '5ad3ae14604f3c001a3fec3a', 'prediction_text': 'pagii', 'no_answer_probability': 0.0}
```

```
{'id': '5ad3ae14604f3c001a3fec3a', 'answers': {'text': [], 'answer_start': []}}
```

```
{'id': '5ad3ae14604f3c001a3fec3a', 'title': 'Normans',
'context': 'Before Rollo\'s arrival, its populations did not differ from Picardy ...',
'question': 'What Viking groups were conquered by Rollo?',
'answers': {'text': [], 'answer_start': []}}
```

For the first false prediction here, for the question What Viking groups were conquered by Rollo?, the basic model predicted that the answer was pagii when none was include in the text. We can understand its mistake because the waited answer was about an ethnicity and the predicted answer is a lost word for country in latin.

**Wrong answer number #2**

```
{'id': '56dde27d9a695914005b9651', 'prediction_text': '', 'no_answer_probability': 0.0}

{'id': '56dde27d9a695914005b9651', 'answers': {'text': ['Catholicism', 'Catholicism',
                               'Catholicism'], 'answer_start': [121, 121, 121]}}

{'id': '56dde27d9a695914005b9651', 'title': 'Normans',
'context': 'The descendants of Rollo\'s Vikings and their Frankish wives ...',
'question': 'What was the Norman religion?',
'answers': {'text': ['Catholicism', 'Catholicism', 'Catholicism'],
                                              'answer_start': [121, 121, 121]}}
```

For the second prediction, the question was What was the Norman religion? and the model did not find an answer in the text but Catholicism could be find. It is certainly due to a lack of training, because as we will see below, the best model found something (even if not right) for this question.

### 3.3.2 Teacher prediction

Let's look at wrongly predicted answer for the teacher pre-trained model.

**Wrong answer number #1**

```
{'id': '56dde27d9a695914005b9651', 'prediction_text': 'Norse', 'no_answer_probability': 0.0}

{'id': '56dde27d9a695914005b9651', 'answers': {'text': ['Catholicism', 'Catholicism',
                               'Catholicism'], 'answer_start': [121, 121, 121]}}

{'id': '56dde27d9a695914005b9651', 'title': 'Normans',
'context': 'The descendants of Rollo\'s Vikings and their Frankish wives ...',
'question': 'What was the Norman religion?',
'answers': {'text': ['Catholicism', 'Catholicism', 'Catholicism'],
                                              'answer_start': [121, 121, 121]}}
```

For the first false prediction, the question was What was the Norman religion?, the teacher model predicted that the answer was Norse which refers to the replaced religion and Catholicism was the right answer. We can see that the model grasped the sense of religion for those words, but is mistaken on which word so it may lack training for getting the context right.

**Wrong answer number #2**

```
{'id': '5ad3af11604f3c001a3fec63', 'prediction_text': 'Catholicism', 'no_answer_probability': 0.0}

{'id': '5ad3af11604f3c001a3fec63', 'answers': {'text': [], 'answer_start': []}}

{'id': '5ad3af11604f3c001a3fec63', 'title': 'Normans',
'context': 'The descendants of Rollo\'s Vikings and their Frankish wives ...',
'question': 'What was replace with the Norse religion?',
'answers': {'text': [], 'answer_start': []}}
```

For the second prediction, the question was What was replace with the Norse religion? and the model did find an answer Catholicism when none was in the text. It shows once again that the model gets the religion sense of those word but again, reverse the question context and predict that Norse is the new religion and Catholicism the replaced one.

# 4 Create a searchable index

The goal of this section is to create a searchable index to retrieve relevant document for a given question to maximize the chance of finding an answer to it in those contexts. To achieve that, we need to vectorize the question and all the contexts using a sentence2vec model and compare the cosine similarity of our vectorized question with all the contexts to find the most relevant ones.

## 4.1 Asymmetric similarity models

### 4.1.1 The models

We choose 2 asymmetric similarity models [3] in the list of the ones who can produce normalized vectors of length 1 and can be used with dot-product, cosine-similarity and Euclidean distance:

- msmarco-MiniLM-L12-cos-v5: here
- msmarco-distilbert-cos-v5: here

This page shows us on a table (replicated just bellow) that the distilbert model should be clearly slower but have better results.

| Model | MRR | Performance Semantic Search | Queries GPU / CPU per sec. |
|---|---|---|---|
| msmarco-MiniLM-L12-cos-v5 | 32.75 | 43.89 | 11,000 / 400 |
| msmarco-distilbert-cos-v5 | 33.79 | 44.98 | 7,000 / 350 |

### 4.1.2 The results

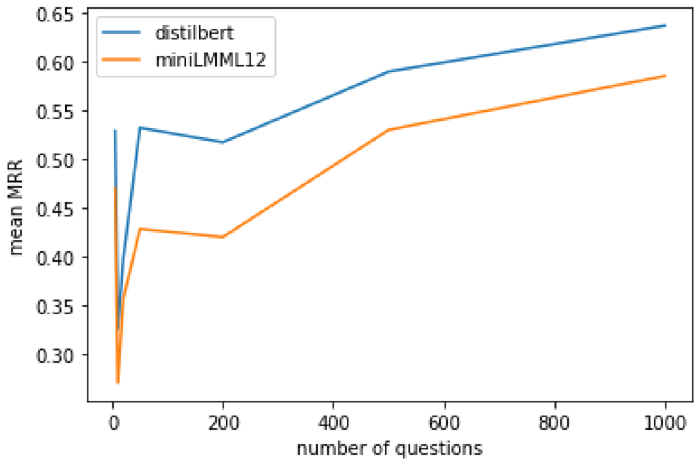

Figure 4: Plot MMR

The indexer gets a better overall MRR when using distilbert than when using MiniLM, so we will take this one for the rest of this project.

Indexing paragraph can be pretty slow when applied to a lot of questions, so we need to find a way to get our nearest neighbour faster.

## 4.2 Nearest neighbour approximation

Because searching for the nearest neighbour in all the dataset is very time consuming, we use a nearest neighbour approximation method to get relevant document faster.

Using the approximated method, we can see that the calculation time is greatly reduced. For 20 questions and finding the 500 nearest neighbour, it takes 45sec using the naive method and is nearly instantaneous when using the approximate one and the resulting context seems to be as relevant as with the naive version.

## 5 Putting everything together

The next part is about using the pre-trained model from the first part of our project and giving a new unseen question, retrieve answers from the 10 more relevant context in our now really big dataset (created in the second part of this project)

### 5.1 Results

When asking the question ”Who ruled France ?”, the model gave a few answers:

- Louis Philippe d’Orleans
- Claude Francois Chauveau-Lagarde
- Henry of Navarre
- Pierre de Langle
- French Crown
- grand audiencier of France

Looking at those answers, we can see that some are right (Henry of Navarre or Louis Philippe d’Orleans for example) and even wrong answer are pretty close to reality (French Crown can be a term used to describe a king). It means that the model grasped well the meaning of ”who”, ”ruled” and ”France” and got relevant text from the context indexer to find possible answers.

## 6 Conclusion

To conclude, many problems can occur during the merge between the datasets SQuAD v2 et DBPedia:

- DBPedia’s contexts can be a more rightful answer than the original one from SQuAD v2
- As we saw, some DBPedia’s contexts contains only a ”.” or are just a list of the same thing (an insect in the example)

Overall this project showed us a way of fine-tuning model using Huggingface library, vectorizing sentences to find similar ones and tricks like the approximate KNN to get faster relevant contexts.

## References

[1] Subject : here

[2] Nearest neighbour approximation : https://github.com/nmslib/hnswlib/

[3] Pretrained models : https://www.sbert.net/docs/pretrained_models.html