Question Answering on SQuAD

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Natural Language Processing
Gayed Simone, Zhang Hanying, Zhou Jia Liang, Rossi Alex

Contents

1 Executive Summary

Question Answering is the task of answering questions (typically reading comprehension questions).The main method we rely on in this project is Transformers. Specifically, we take the DistilBERT model[7], pre-trained on Masked LM and Next Sentence Prediction, add new layers in the end, and train the new model for our question-answering task.

We started by trying a BiDAF model[8] but the reason we ended up using transformers instead of building a specific deep learning model (LSTM, CNN, etc.) which is suitable for question answering tasks is that we could do a quicker development and we could get better results by using fewer data. We believe that the same transfer learning shift as the one that took place in CV field several years ago would happen to NLP. Rather than training a new network from scratch each time, the lower layers of a trained network with generalized features (the backbone) could be copied and transferred for use in another network with a different task. The DistilBERT-based model is pre-trained on tasks other than question answering and fine-tuned on the SQuAD dataset[6], while the recurrent-based model, BiDAF, is trained from scratch. To come up with the best model we have experimented with different question heads which differ in the number of layers, activation function, and overall structure.

Our experimental results show that, as expected, DistilBERT-based models achieved better results with respect to BiDAF. Moreover, we implented a new question-answering head on DistilBERT, thanks to which we improved the HuggingFace[4] implementation.

2 Background

2.1 Datasets

Stanford Question Answering Dataset (SQuAD)[6] 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. SQuAD dataset is big, challenging and requires reasoning; these characteristics made it a famous dataset for the evaluation of models on question answering. Other existing dataset for reading comprehension and question answering are: Reading comprehension, a data-driven approach that goes back to Hirschman et al. (1999) who curated a dataset of 600 real 3rd– 6th grade reading comprehension questions, Open-domain question answering whose goal is to answer a question from a large collection of documents and Cloze datasets in which the goal is to predict the missing word (often a named entity) in a passage.

2.2 Metrics

The main objective of our project is to create an NLP system that, given a paragraph and a question regarding it, provides a single answer, which is obtained by selecting a span of text from the paragraph. For the model

evaluation two different metrics are used. Both metrics ignore punctuations and articles. The first metric is exact match, which measures the percentage of predictions that match any one of the ground truth answers exactly while the second one is (Macro-averaged) F1 score which measures the average overlap between the prediction and ground truth answer. Prediction and ground truth are treated as bags of tokens and compute their F1. The maximum F1 over all of the ground truth answers for a given question is taken, and then average over all of the questions. The resulting human performance score on the test set is 77.0% for the exact match metric, and 86.8% for F1. Mismatch occurs mostly due to the inclusion/exclusion of non-essential phrases rather than fundamental disagreements about the answer.

2.3 State-Of-The-Art Models

The best-performing methods for this task in the literature make use of transformers (Vaswani et al., 2017)[10] (Dehghani et al., 2019)[3], powerful neural network architectures that rely on a trainable attention mechanism that identifies complex dependencies between the elements of each input sequence. Some of these are LUKE[11], which uses deep contextualized Entity Representations with entity-aware Self-attention, XLNet[12], which integrates ideas from Transformer-XL[2], the state-of-the-art autoregressive model and the well-known transformer BERT.

3 System Description

For this project, we started with a model based on the BiDAF architecture (Seo et. al 2016)[9], a hierarchical multi-stage architecture for modelling the representations of the context paragraph at different levels of granularity, which includes character-level, word-level, and contextual embeddings, and uses bi-directional attention flow to obtain a query-aware context representation. To reach better performances, we built an architecture around the backbone of the DistilBert transformer that will be explained in detail in this section.

3.1 BiDAF

3.1.1 Embedding

Firstly, character embeddings are computed with a CNN layer for each input word, working as features extractor on single words, just like filters on images to extract different features. We use N=100 filters. We convolve these filters on top of the word embeddings, obtained through an embedding layer. In the end, to obtain a fixed size vector for each word as the concatenation of max pooling over the dimension of the output from the CNN layer, for each filter.

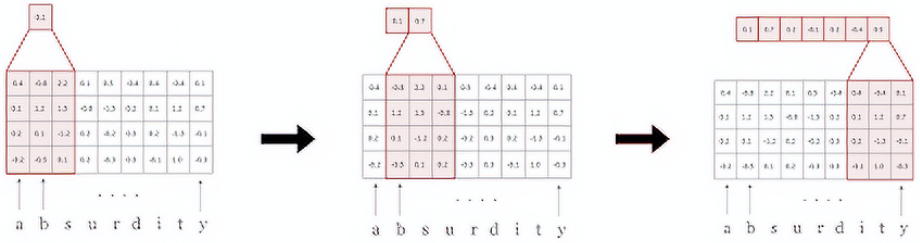

Figure 1: character embedding

We also use pre-trained word embeddings from GloVe, which are concatenated to the character embeddings before being fed to the Highway Network, a series of feed-forward or linear layers with a gating mechanism. Next, the output of the Highway Network is passed to the Contextual Embedding, the final embedding layer of the model. This is implemented as a Bidirectional LSTM, with input both context and query. So far we have computed features at different levels of granularity, similarly to multi-scale feature extraction in Computer Vision.

3.1.2 Attention Flow Layer

The inputs to the Attention Flow Layer layer are contextual vector representations of the context H and the query U. The outputs of the layer are the query-aware vector representations of the context words, G, along with the contextual embeddings from the previous layer. We use the shared similarity matrix, S with dimensions T × J, to compute attention for both directions, from context to query and vice versa. T and J are respectively the length of the context and the length of the query.

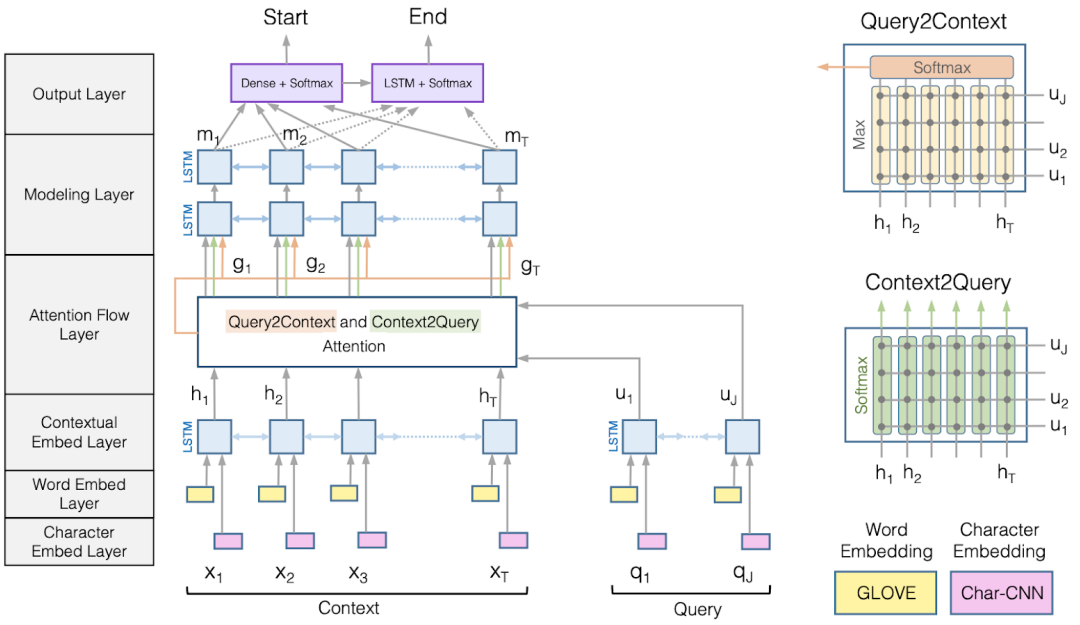

Figure 2: BiDAF architecture

3.2 Transformers

The Attention mechanism lets us overcome the limited range dependency of RNNs, enabling us to consider the whole input sequence. RNNs also suffer from scalability problems, being inherently sequential in computation. A new Neural Network architecture was introduced in 2017, levering attention mechanism while being able to process all input words in parallel. This architecture is the Transformer, an encoder-decoder architecture. Both the encoding and decoding parts use as the main building block of the multi-head self-attention module, jointly attending information from different representation subspaces at different positions. The use of positional encoding makes up for the lack of sequential information processing.

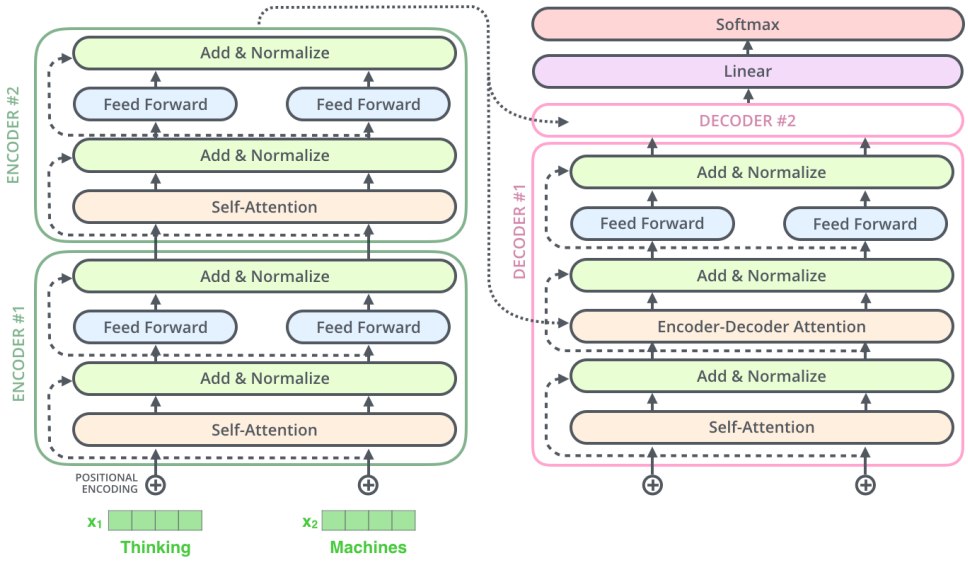

Figure 3: Transformers architecture

3.2.1 BERT

The key innovation of the BERT model is applying bidirectional training of Transformers to language modelling. The entire input sequence though is read a once, so it would be more correct to define it non-directional. BERT is basically an Encoder stack of transformer architecture. BERTBASE has 12 layers in the Encoder stack while BERTLARGE has 24 layers in the Encoder stack. These are more than the Transformer architecture described in the original paper (6 encoder layers). BERT architectures (BASE and LARGE) also have larger feedforward-networks (768 and 1024 hidden units respectively), and more attention heads (12 and 16 respectively) than the Transformer architecture suggested in the original paper. Bert model is pre-trained on 2 tasks: Masked LM, where a percentage of the input tokens are masked and then predicted, and Next Sentence Prediction, in order to learn the relationship between two sentences, which is not directly modelled by language modelling.

3.2.2 DistilBERT

DistilBERT is a small, fast, cheap and light Transformer model trained by distilling BERT base. It has 40% less parameters than bert-base-uncased, runs 60% faster while preserving over 95% of BERT's performances as measured on the GLUE

language understanding benchmark. Knowledge Distillation (KB) could be seen as a transfer learning technique, even though it has a different aim. Knowledge Distillation is a form of compression from a huge high precision model to a smaller one, without losing too much in generalization. It is a reinforcement learning technique that wants to reduce the dimension of huge models (like BERT) and the training time by transferring knowledge between two models. In particular, the Teacher-Student paradigm is composed of a small network, the Student, and a bigger one, the Teacher, which should be trained on the complete dataset. The training phase usually needs a considerable amount of time for models. The final aim is to teach the Student how to simulate the Teacher's behaviour, but with a model smaller in size and computation.

4 Experimental setup and results

In order to improve the DistilBERT baseline model we have experimented with different question heads which differ in the number of layers, activation function and overall structure. We implemented 4 different question heads, two models with 2 fully connected layers, and two models with 3 fully connected layers.

4.1 Data Loading and Split

We wanted to decode the JSON training set into a fitting table, which proved useful also for subsequent pre-processing tasks. We allocated one row for each context/question/answer triple, thereby replicating the same context and question multiple times, since each context has multiple question/answer pairs and each question may have multiple answers.

In terms of the dataset splitting method, we chose 20% of the total dataset for validation, and, as test set we used the official SQuAD v1.1 dev set. This dataset differs from the training dataset in that the same question may have multiple correct answers.

4.2 Vanilla DistilBERT QuestionAnswering Head

We fine-tuned the vanilla version of DistilBERT for question answering so as it is implemented in the HuggingFace transformers library. It uses a single linear layer, with no activation function, to reduce the 768 output dimension that comes from the DistilBERT backbone to 2.

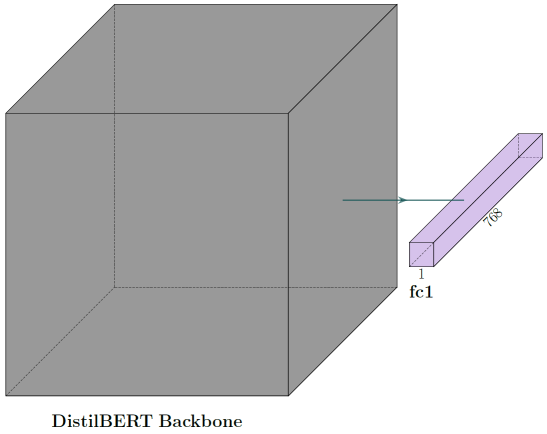

Figure 4: QADistilBERTVanilla model schema

4.3 DistilBERT with custom QuestionAnswering Head

To explore the benefit of having more layers, we added two layers on the top of DistilBERT and tried different activation functions to understand which one works better. We did this in order to achieve better results on question answering. We also tried different configuration shapes and we will present only the best ones. The idea was to have a conic function reduction of the dimension, in this way, we could have a more gradual reduction from the DistilBERT output dimension to 2.

4.3.1 2L + tanh

In the version with Tanh function we opted for a reduction, firstly, from 768 to 384, and then from 384 to 2.

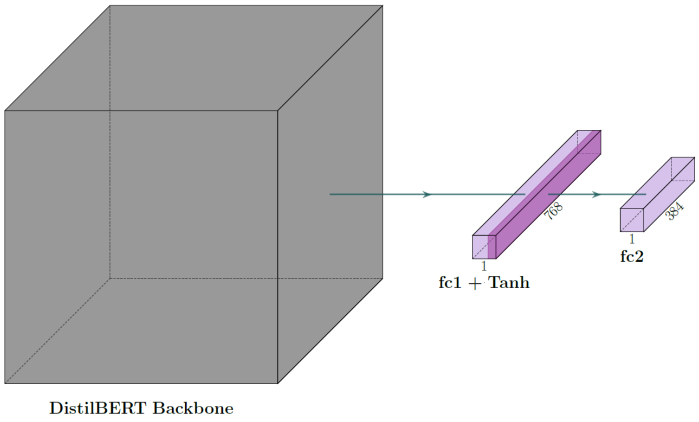

Figure 5: QADistilBERT2LTanh model schema

4.3.2 2L + GELU

The second version of this model uses the GELU(Gaussian Error Linear Units)[5], a high-performing neural network activation function which is $x\Phi(x)$, where $\Phi(x)$ is the standard Gaussian cumulative distribution function, used in DistilBERT. With the GELU activation function the same input dimension was kept in both layers but the skip connection idea was introduced. The output of DistilBERT is fed to the first layer and then added to its output before feeding everything to the second layer.

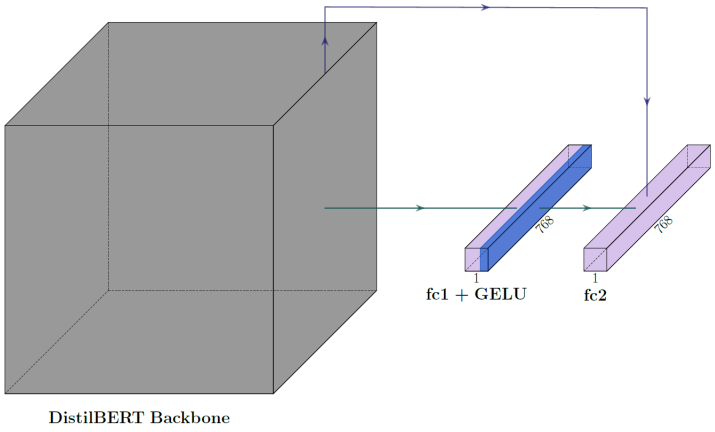

Figure 6: QADistilBERT2LGELU model schema

4.3.3 3L + GELU

This model uses the GELU activation function between the layers. In this case we have used 3 layers so we reduced the dimension from 768 to 512 and then from 512 to 32 and finally 32 to 2.

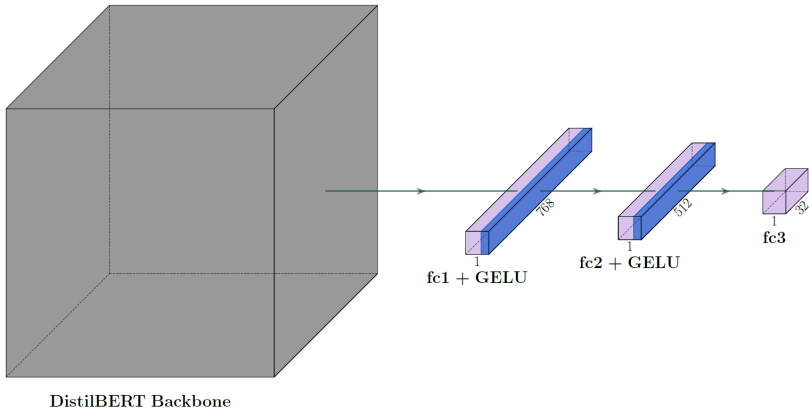

Figure 7: QADistilBERT3LGELU model schema

4.3.4 3L + GELU with layer Norm

We will introduce another idea which consists of using a Normalization Layer[1] as final layer. This layer, unlike batch normalization, directly estimates the normalization statistics from the summed inputs to the neurons within a hidden layer so the normalization does not introduce any new dependencies between training cases, it does not impose any constraint on the size of the mini-batch and it can be used in the pure online regime with batch size 1. We will use this question answering level as a baseline for our study because, looking at the results we obtained by fine-tuning DistilBERT with different configurations, this structure reached better results with respect to the others.

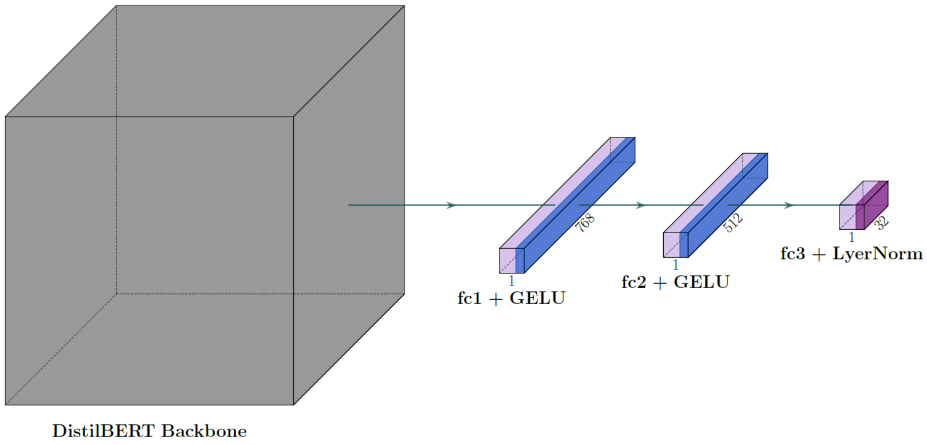

Figure 8: QADistilBERT2LGELU with Layer Norm model schema

4.4 Results

Our experiments started with BiDAF. Our implementation didn't perform as expected, with worse results with respect to the ones found in the literature, so we didn't include it in the comparison of our results, because we believe that it is not a fair baseline. Regarding DistilBERT, we aimed to understand if some benefit could be obtained by adding fully-connected layers to the vanilla question answering level. DistilBERT has been treated as a backbone so the focus is on the specific task of question answering. Firstly, a fine-tuning over the vanilla DistilBERT for question answering was performed to have a baseline. Then we created our DistilbertForQuestionAnswering model and fine-tuned it with the same hype-parameters. We measured the F1 score and the EM (Exact Match) score to have the same set of results as in the SQuAD paper. According to the paper, the F1 score measures the average overlap between the prediction and the ground truth answer, while EM measures the percentage of predictions that match exactly the ground truth answers.

The final objective was to compare the results of both versions on this specific task (QA). The next tables collect all the three fine-tuning we did 1, the highlighted line is the one that allowed us to get better results, we also compared our best model to other fine-tuned transformers, on the validation set 2 and on the SQuAD 1.1 dev set 3. Regarding SQuAD 1.1 dev set, F1 and Exact Match scores are higher because for each question there are multiple possible answers, for instance: ['X-Rays', 'x-rays', '"Roentgen rays" or "X-Rays"'].

Configuration	Learning Rate	Batch Size	Num Epochs
config1	2.00E-05	16	3
config2	5.00E-05	12	4
config3	**5.00E-05**	**32**	**3**

Table 1: Hyperparameters of the different configurations

Models	VanillaQADistilBERT		QADistilBERT	
Configurations	**F1(%)**	**EM(%)**	**F1(%)**	**EM(%)**
config1	75.09	59.37	75.68	59.87
config2	74.93	58.72	75.47	59.81
config3	75.62	59.98	76.04	60.17

Table 2: Comparison between VanillaQADistilBERT and our best model

VanillaDistilBERT		DistilBERT		ALBERT		ROBERTA	
F1(%)	**EM(%)**	**F1(%)**	**EM(%)**	**F1(%)**	**EM(%)**	**F1(%)**	**EM(%)**
75.09	59.37	75.68	59.87	81.76	65.94	82.49	67.13

Table 3: Comparison among different transformers

VanillaDistilBERT		DistilBERT		ALBERT		ROBERTA	
F1(%)	**EM(%)**	**F1(%)**	**EM(%)**	**F1(%)**	**EM(%)**	**F1(%)**	**EM(%)**
84.38	75.53	84.48	75.82	89.95	82.46	90.79	84.06

Table 4: Comparison among different transformers on the test set (SQuAD 1.1 dev set)

5 Analysis of results

We did not expect significant improvements because of our limited resources. Despite this, we got some interesting results. In our case the best performances came from the configuration with a batch size of 12. Other parameters which demonstrated to have an impact during the training are the learning rate and the number of epochs (this is a very difficult choice because it could lead to overfitting if the model is trained for too long, and to underfitting otherwise).

By studying the results, we have found an interesting behaviour, the frequency of the questions it's not correlated to the F1 score and exact match, as it can be seen in the plots below. It can also be noticed that more open questions like 'why', 'if' and 'what' on average performs worse with respect to closer questions like 'when', 'who' and 'which'.

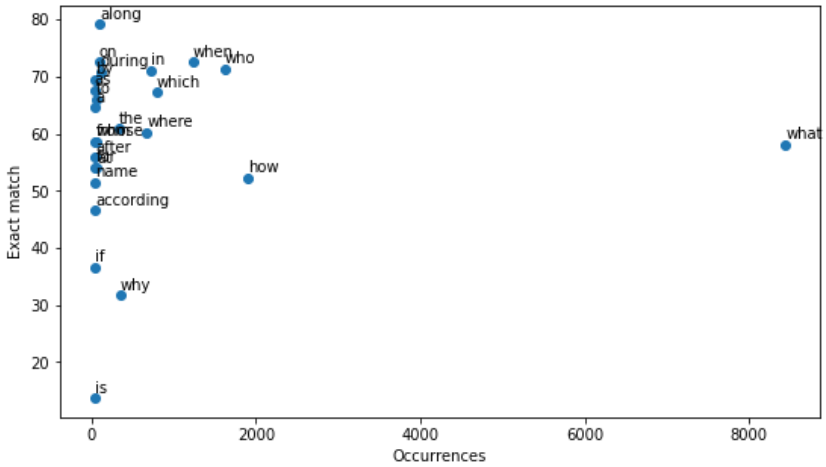

Figure 9: Plot of Exact Match compared to the number of occurrences per type of question

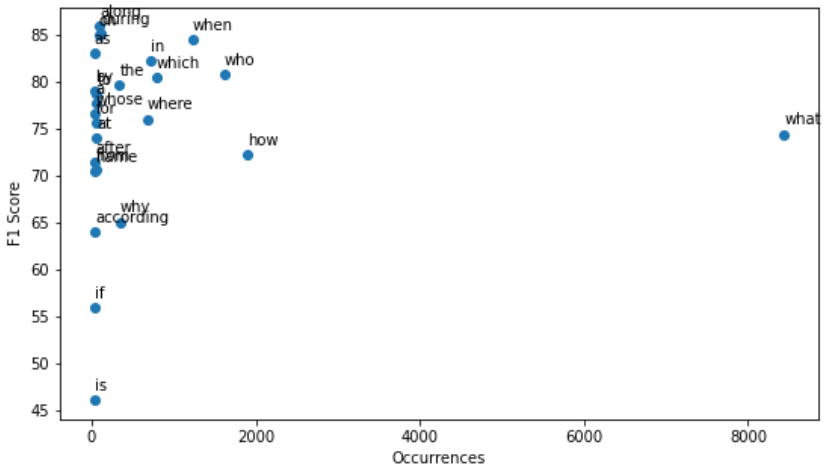

Figure 10: Plot of F1 Score compared to the number of occurrences per type of question

By examining the most common type of errors on the validation set, we have found that they could be divided in 3 main different categories:

Error type: Too generic answer

Context: The most widely used symbol is the flag of Greece, which features nine equal horizontal stripes of blue alternating with white representing the nine syllables of the Greek national motto Eleftheria i thanatos (freedom or death), which was the motto of the Greek War of Independence. The blue square in the upper hoist-side corner bears a white cross, which represents Greek Orthodoxy. The Greek flag is widely used by the Greek Cypriots, although Cyprus has officially adopted a neutral flag to ease ethnic tensions with the Turkish Cypriot minority – see flag of Cyprus).
Question: Have the people of Greece done anything to make the matter more palatable for the people of Turkey ?
Correct answer: Cyprus has officially adopted a neutral flag

Prediction: The Greek flag is widely used by the Greek Cypriots, although Cyprus has officially adopted a neutral flag to ease ethnic tensions with the Turkish Cypriot minority – see flag of Cyprus).

Error type: Misunderstanding of the question's subject

Context: In association football, teams such as Manchester United, Bayern Munich, Liverpool, Arsenal, Toronto FC, and S.L. Benfica primarily wear red jerseys. Other teams that prominently feature red on their kits include A.C. Milan (nicknamed i rossoneri for their red and black shirts), AFC Ajax, Olympiacos, River Plate, Atlético Madrid, and Flamengo. A red penalty card is issued to a player who commits a serious infraction: the player is immediately disqualified from further play and his team must continue with one less player for the game's duration.
Question: How is an association football team impacted when a player is shown a red penalty card?
Correct answer: his team must continue with one less player for the game's duration
Prediction: immediately disqualified from further play

Error type: Misunderstanding of the question's type

Context: At over 5 million, Puerto Ricans are easily the 2nd largest Hispanic group. Of all major Hispanic groups, Puerto Ricans are the least likely to be proficient in Spanish, but millions of Puerto Rican Americans living in the U.S. mainland nonetheless are fluent in Spanish. Puerto Ricans are natural-born U.S. citizens, and many Puerto Ricans have migrated to New York City, Orlando, Philadelphia, and other areas of the Eastern United States, increasing the Spanish-speaking populations and in some areas being the majority of the Hispanophone population, especially in Central Florida. In Hawaii, where Puerto Rican farm laborers and Mexican ranchers have settled since the late 19th century, 7.0 per cent of the islands' people are either Hispanic or Hispanophone or both.
Question: Where are the biggest population of Puerto Ricans on the mainland?
Correct answer: many Puerto Ricans have migrated to New York City, Orlando, Philadelphia, and other areas of the Eastern United States
Prediction: over 5 million

Another thing to notice is that the exact match computed on the top 5 predictions it's about 20 points more, on the validation set, way higher than the one computed taking in consideration just the best one.

6 Discussion

We have presented a new question answering head, built on the top of the backbone of DistilBert, that seems to improve the performances of the Vanilla version presented in the Hugging Face library. To come up with the best model we have experimented with different question heads which differ in the number of layers, activation function and overall structure and we found the best one to be composed of 3 dense layers with GELU activation and Layer Norm at the end.

With respect to the Vanilla DistilBert of Hugging Face we obtain about 0.5 points more in F1 score and Exact Match, and, based on what we stated in the Analysis of results section, we think that our model could be strongly improved by defining a new answer selection algorithm because our top 5 predictions exact match is way higher with respect to the metric computed on the top 1 prediction, meaning that our model correctly predicts the answer among the most probable ones but it gives to low confidence.

For future works, this question answering head could be used on the top of other types of transformers to see if it improves also their performances.

Bibliography

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. *Layer Normalization*. 2016. arXiv: 1607.06450 [stat.ML].

[2] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. *Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context*. 2019. arXiv: 1901.02860 [cs.LG].

[3] Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. "Universal transformers. In 7th International Conference on Learning Representations". In: (2019). URL: https://openreview.net/forum?id=HyzdRiR9Y7.

[4] "DistilBERT for question answering". In: (). URL: https://huggingface.co/transformers/model_doc/distilbert.html#distilbertforquestionanswering.

[5] Dan Hendrycks and Kevin Gimpel. *Gaussian Error Linear Units (GELUs)*. 2020. arXiv: 1606.08415 [cs.LG].

[6] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. "SQuAD: 100,000+ Questions for Machine Comprehension of Text". In: (2016). URL: https://nlp.stanford.edu/pubs/rajpurkar2016squad.pdf.

[7] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter". In: *CoRR* abs/1910.01108 (2019). arXiv: 1910.01108. URL: http://arxiv.org/abs/1910.01108.

[8] Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. "Bidirectional Attention Flow for Machine Comprehension". In: *CoRR* abs/1611.01603 (2016). arXiv: 1611.01603. URL: http://arxiv.org/abs/1611.01603.

[9] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. "Bidirectional Attention Flow for Machine Comprehension". In: (2016). URL: https://arxiv.org/abs/1611.01603.

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. "Attention is all you need. In Advances in Neural Information Processing Systems". In: (2017). URL: http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

[11] Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. *LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention*. 2020. arXiv: 2010.01057 [cs.CL].

[12] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. *XLNet: Generalized Autoregressive Pretraining for Language Understanding*. 2020. arXiv: 1906.08237 [cs.CL].