Question Answering on Query like inputs

Abstract

Question Answering is an important task in Natural Language Processing where in a system given a question in natural language can extract give an answer. The aim of our study has been to improve question answering model for Natural Questions(NQ) [4]. The basic design of the study is to see how Bidirectional Encoder Representations from Transformers (BERT) [3] would perform on NQ. First, I experimented with hyper-parameter tuning of BERT to understand how it would perform on NQ. Second, I explored how baseline DecAtt + DocReader performed on NQ. We found that hyperparameter tuning of BERT on NQ to be saturated for this task and found ensemble modelling of BERT and DecAtt + DocReader to be a useful next step in improving the performance.

1. Introduction

Modern day search engines (like Google, Bing, DuckDuckGo) are getting increasingly better at interpreting queries and extracting relevant text from results. For example, a google search for "Make Tea" results in an instruction snippet on how we should go about making tea. But a query like "nearest RMV" results in not an extracted text, rather it gives us list of web pages with the relevant information. However, searching for "Where is the nearest RMV?" gives us the full address of the nearest RMV office. In the first case, the search engine was able to extract an answer to the query from a document, while in the latter case it wasn't able to do so. Our aim is to improve the Question Answering models to perform better on natural queries which come from actual need of people rather than synthesized questions after looking at passages.

We saw that the existing question answering models perform really well (F1 score of 92.215) on SQuAD. But if we see it's performance on Natural Question dataset [4], it gives an F1 score of 52.66. It is an interesting task because QA systems that can answer Natural Language Questions are more realistic and have a bigger impact on the way we perceive information.

2. Background/Related Work

The release of BERT has substantially advanced the state-of-the-art in question answering. For example, currently, the top 25 systems on the SQuAD 2.0 leaderboard and the top 5 systems on the CoQA [7] leaderboard are all based on BERT. The results obtained by BERT-based question answering models have surpassed if not come close to human performance on these tasks; with 2.5 F1 points of room left on SQuAD 2.0 and 6 F1 points on CoQA. The Natural Questions (NQ) might represent a substantially harder research challenge as compared to the SQuAD 2.0 and CoQA.

2.1. What is BERT?

BERT is a deep learning model that has given state-of-the-art results on a wide variety of natural language processing tasks. It stands for Bidirectional Encoder Representations for Transformers. It has been pre-trained on Wikipedia and BooksCorpus and requires (only) task-specific fine-tuning.

BERT is based on the Transformer architecture, and it is bidirectional which means that it learns information from both right and left side of the token thus giving the context. BERT is pretrained on 42GB of unlabelled text which includes the Wikipidea and Book corpus (3300 million words). To understand how large the dataset is, an avid reader that reads a book a day cannot read that much text over their lifetime. Pretraining our model on such a huge dataset is important because it starts learning the deeper and intimate understanding of how language works.

BERT is a multilayer Bidirectional Transformer encoder where each encoder has two layers: self-attention layer and a feed forward neural network. The self-attention layer gets the preprocessed text such that it has token, segment and position embeddings. While the feed forward network is independent of the embeddings and as a result can be applied in parallel which speeds up the performance.

2.2. DecAtt + DocReader

DecAtt+DocReader is a pipeline model that draws inspiration from natural language inference (NLI). The long

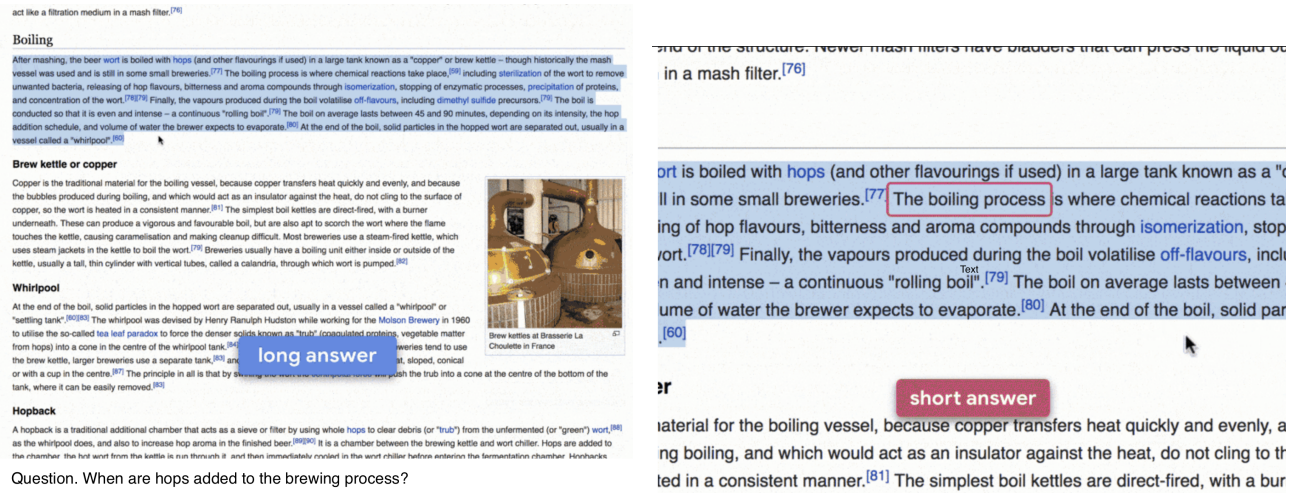


Figure 1. Natural Question with long and short answer respectively

answer and short answer prediction can be looked as different tasks. The long answer prediction can be looked at as an inference task and it needs to have sufficient information to answer the question. While, short answer does not need need to contain all the information, it just needs to be surrounded by the long answer. Using this inspiration, pipelined approach can be used where a model drawn from the NLI literature selects the long answers. And then the short answers are selected from the long answers using a model drawn from the short answer extraction literature. So, long and short answers are predicted sequentially and independent of each other using a DecAtt [5] NLI model and a DocReader [2] model respectively.

3. Approach

3.1. The baseline: Bert joint

We use the BERT$_{joint}$ baseline model. In that we utilize BERT model pretrained on SQuAD2.0 and then finetune it on NQ dataset. To obtain the baseline model:

- The training set for the Natural Question is quite large so we precompute all the features for it as tensorflow examples.
- Used a pretrained model on SQuAD2.0 and evaluate it on the "tiny" dev set to verify that everything is working correctly
- Trained the above mentioned pretrained model on Natural Questions which took 5-7 hours to run on a TPU
- In the paper [4], it is recommended to finetune on different hyperparameters and save the best model. We have performed this hyperparameter tuning and described it in section 4.3.

model	learning rate	#epochs	batch size
BERT$_{joint}$	0.0001	1	24
BERT$_{joint}$	0.005	1	24
BERT$_{joint}$	0.0005	3	32
BERT$_{joint}$	0.005	1	32

Table 1: Summary of hyperparameter tuning

3.2. Hyperparameter tuning: Bert joint

Our hyperparameter tuning was very straighforward. We tried 2 different learning rates and 2 different number of epochs. This was because for most BERT-based models, we found this as a common practice. We tried the batch_size of 24 & 32, epoch of 1 & 3 and learning rate between 1e-4 and 5e-3, where each run took a few hours. We found that training for 1 epoch with an initial learning rate of 0.005 was the best setting and hence we kept that constant. The F1 scores obtained on dev set was 64.7 and 52.7 for long and short answers. The summary of this has been shown in the Table 3

3.3. Kaggle challenge

Oct 10 is when we submitted the project proposal and Oct 28 Tensoflow released a challenge to identify the answers to real user questions about Wikipedia page content. The dataset in this challenge is provided by Google's Natural Questions, but contains its own unique private test set. Since the task is the same as what our project is about, we wanted to see how our fine-tuned model performs on the Kaggle scoreboard. We then performed the following steps:

- Set up our model baseline on Kaggle Kernel.
- Read in the test set.

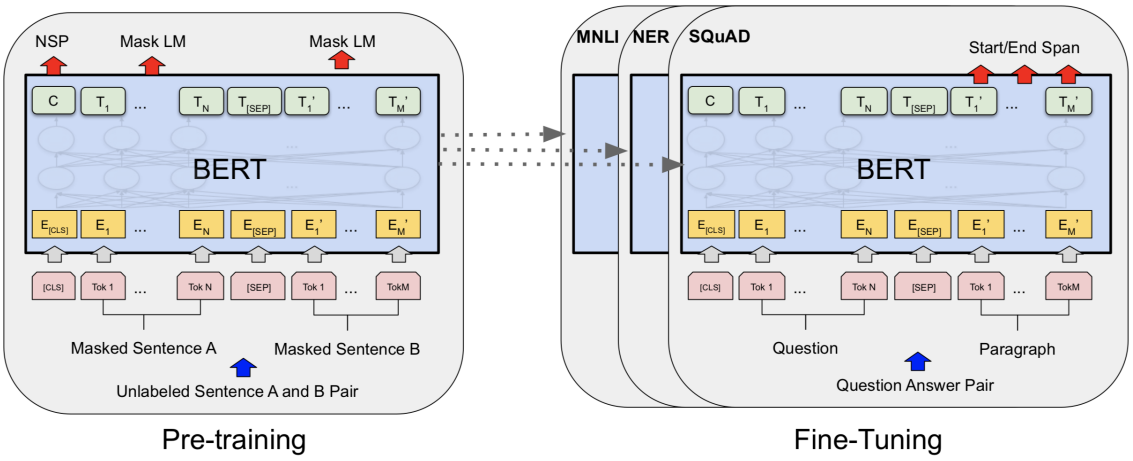


Figure 2. General method of fine-tuning BERT

Baseline Models	Long Answer Dev	Long Answer Test	Short Answer Dev	Short Answer Test
DecAtt+DocReader	54.8	55.0	31.4	31.5
BERT$_{joint}$	64.7	66.2	52.7	52.1

Table 2: Harmonic Mean (F1) of all baseline

- Run it past the pre-built Bert model to create embeddings
- Use those embeddings to make predictions
- Write those predictions to predictions.json
- Further output processing to create submissions.csv file for Kaggle.

The evaluation metric of Kaggle Submissions uses micro F1 between the predicted and expected answers. This has been described in Section 5. The metric in this competition diverges from the original metric in two key respects:

- short and long answer formats do not receive separate scores, but are instead combined into a micro F1 score across both formats, and
- this competition's metric does not use confidence scores to find an optimal threshold for predictions.

When we submitted the baseline on kaggle, we got a micro_f1 score 0.22. We believe the drop in accuracy is due to the difference in the metric. We further want to improve our model such that it can perform better on this score. This is because having one score for both long and short answer seems more intuitive to us to judge our model.

3.4. The baseline: DecAtt + DocReader

As mentioned above we take the Google DecAtt+DocReader (custom pipeline) model as another baseline. DecAtt is used for long answer selection and then DocReader is used on those long answers to extract the short answers. To obtain this baseline model:

- We preprocessed the data for long answer and short answer pipeline model
- Used glove pre-trained word embeddings [6]
- Trained the long answer and short answer pipeline model and save them

3.5. Ensembling BERT and DecAtt+DocReader

After obtaining baseline models of BERT and DecAtt+DocReader, we use each model to get predictions on test set. Nobody(that we can find) has created a BERT and DecAtt+DocReader ensemble. Our inspiration behind this ensemble is that in some cases DecAtt+DocReader could outperform BERT [1] and we want to capitalize on it to improve the accuracy. We didn't get the chance to try different stochastic ensembles of BERT & DecAtt+DocReader and evaluate them when there is a disagreement between the predictions of both the models. However, it is a future work that we want to find the best ratio to ensemble the two models together, for example the ensemble model will accept BERT 95% of the times(due to obvious outperformance of BERT) and DecAtt+DocReader the 5% of the times.

4. Experiment

4.1. Experimental Details

Two experiments for the project:

- Hyperparameter tuning
  We tried batch_size of 24 & 32, epoch of 1 & 3 and learning rate between 1e-3 and 1e-5. We found out that learning rate: 5e-3, epoch:1 and batch_size: 32 gives the best accuracy
- Ensemble modeling
  As detailed above, we have the BERT and DecAtt+DocReader trained on NQ dataset. The ensembling would be applied after DecAtt+DocReader baseline and BERT models made their predictions.

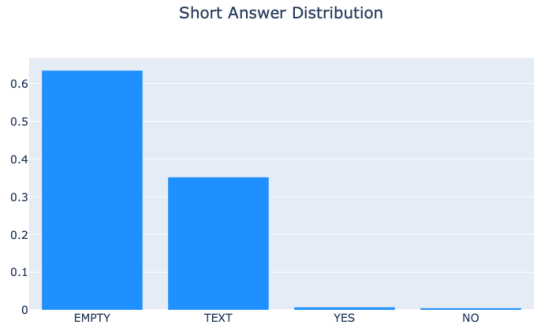
4.2. Dataset

Natural Questions dataset created by Google for Open Domain Question Answering. It contains questions in the form of real queries from users and contains 307K training examples, 8K examples for development, and a further 8K examples for testing. An example is shown in Figure 1. Each example is comprised of a google.com query and a corresponding Wikipedia page.
What makes NQ the best dataset for our task:

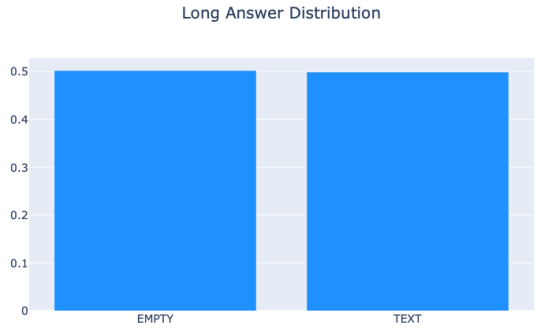
- The questions on NQ are questions asked by people out of need, like we google all the time.
- The questions were not formed by people after looking at the document. Which is how we go about it in real life.
- The answers to these questions come from documents which are much longer than what SQuAD deals with.

There is a kaggle challenge on the same Natural Questions dataset where we are tasked with selecting the best short and long answers from Wikipedia articles to the given questions. In the challenge, there are two samples of data: simplified-nq-train.jsonl(training data) and simplified-nq-kaggle-test.jsonl(testing data). Each sample contains a Wikipedia article, a related question, and the candidate long form answers. The training examples also provide the groundtruth i.e correct long and short form answer or answers for the sample, if any exist. Number of samples in training dataset is 307373 and number of samples in test dataset is 345.

Analysed the training dataset to see the type of answers and their numbers. There are two types of answers - short and long. The long and the short answer annotations can however be empty. If they are both empty, then there is no answer on the page at all. Finally 1% of the documents have a passage annotated with a short answer that is "yes" or "no", instead of a list of short spans.

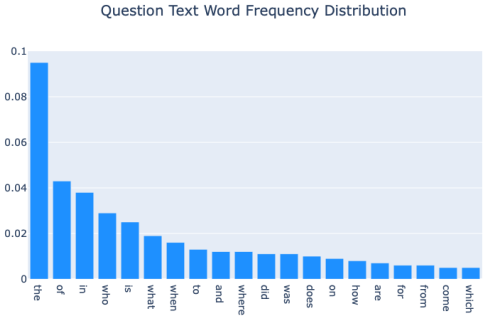


Also analysed the data to see how many long answers are empty/non-empty and how many short answers are empty/yes-no/non-empty. This is summarized in the images below.



As we see in the figures below, the highest frequency words are functional words. We had analysed when submitting the proposal of the project the following:

- Took questions from SQuAD dataset.
- Removed the functional words from the questions. Let's call them generated-queries.
- Then we gave the generated-queries to the model as input. This was in replacement of the original questions.



We saw a drop in F1 score from 82.40 to 73.05. The graphs below also show the high frequency of functional

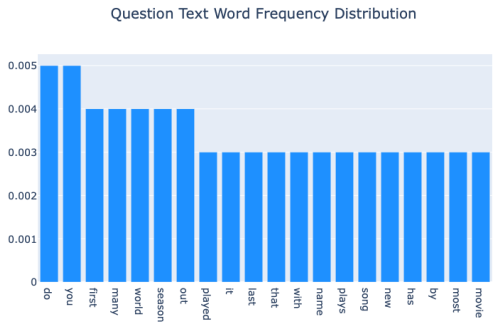
	document_text	long_answer_candidates	question_text	annotations	document_url	example_id
0	Email marketing - Wikipedia <H1> Email marketi...	[{'start_token': 14, 'top_level': True, 'end_t...	which is the most common use of opt-in e-mail ...	[{'yes_no_answer': 'NONE', 'long_answer': {'st...	https://en.wikipedia.org//w/index.php?title=Em...	5655493461695
1	The Mother (How I Met Your Mother) - wikiped...	[{'start_token': 28, 'top_level': True, 'end_t...	how i.met your mother who is the mother	[{'yes_no_answer': 'NONE', 'long_answer': {'st...	https://en.wikipedia.org//w/index.php?title=Th...	5328212470870
2	Human fertilization - wikipedia <H1> Human fer...	[{'start_token': 14, 'top_level': True, 'end_t...	what type of fertilisation takes place in humans	[{'yes_no_answer': 'NONE', 'long_answer': {'st...	https://en.wikipedia.org//w/index.php?title=Hu...	4435104480114
3	List of National Football League career quarte...	[{'start_token': 28, 'top_level': True, 'end_t...	who had the most wins in the nfl	[{'yes_no_answer': 'NONE', 'long_answer': {'st...	https://en.wikipedia.org//w/index.php?title=Li...	5289242154789
4	Roanoke Colony - wikipedia <H1> Roanoke Colony...	[{'start_token': 32, 'top_level': True, 'end_t...	what happened to the lost settlement of roanoke	[{'yes_no_answer': 'NONE', 'long_answer': {'st...	https://en.wikipedia.org//w/index.php?title=Ro...	5489863933082

Figure 3. Training data with groundtruth

	example_id	question_text	document_text	long_answer_candidates
0	-1220107454853145579	who is the south african high commissioner in ...	High Commission of South Africa , London - wik...	[{'end_token': 136, 'start_token': 18, 'top_le...
1	8777415633185303067	the office episode when they sing to michael	Michael 's Last Dundies - wikipedia <H1> Micha...	[{'end_token': 190, 'start_token': 23, 'top_le...
2	4640548859154538040	what is the main idea of the cross of gold speech	Cross of gold speech - wikipedia <H1> Cross of...	[{'end_token': 165, 'start_token': 12, 'top_le...
3	-5316095317154496261	when was i want to sing in opera written	Wilkie Bard - wikipedia <H1> Wilkie Bard </H1>...	[{'end_token': 105, 'start_token': 8, 'top_lev...
4	-8752372642178983917	who does the voices in ice age collision course	Ice Age : Collision Course - Wikipedia <H1> Ic...	[{'end_token': 287, 'start_token': 16, 'top_le...

Figure 4. Test data

words in the natural questions. This may have multiple implications. For example, if the NQ dataset has to come closer to natural queries people ask on Google, they will have to include queries shorter than 8 words. Right now, NQ has queries longer than 8 words only.



4.3. Evaluation metric

The metric in this competition diverges from the original metric in two key respects:

- short and long answer formats do not receive separate scores, but are instead combined into a micro F1 score across both formats
- this competition's metric does not use confidence scores to find an optimal threshold for predictions.

Evaluating using micro F1 between the predicted and expected answers:

$$F1 = \frac{2 * (precision * recall)}{precision + recall}$$

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

where TP: True Positive, FP: False Positive, FN:False Negative

'micro': This determines the type of averaging performed on the data: Calculate metrics globally by counting the total true positives, false negatives and false positives Table 3.

There is an answer	Long/Short + indices or Yes, No (and indices are correct)	True Positive
There is an answer	Long/Short + Blank (or indices are wrong)	False Negative
There is no answer	Long/Short + indices or Yes, No	False Positive
There is no answer	Long/Short + Blank	True Negative

Table 3: How the score is evaluated (Real - Prediction)

	question	respond	start_end_token	target
54	which is the most common use of opt-in e-mail ...	<P> A common example of permission marketing i...	1952:2019	1
15	how i.met your mother who is the mother	<P> Tracy McConnell , better known as `` The M...	212:310	1
24	what type of fertilisation takes place in humans	<P> The process of fertilization involves a sp...	319:438	1
59	who had the most wins in the nfl	<P> Active quarterback Tom Brady holds the rec...	509:576	1

Figure 5. Long Answer Result

4.4. Results

We prepared the data and evaluated BERT$_{joint}$ model on tiny dev set and were able to match the following F1 scores:

- long-best-threshold-f1: 0.6168
- short-best-threshold-f1: 0.5620

We also trained the DecAtt+DocReader on NQ dataset and saved the model. Evaluated the model on tiny dev set and were able to match the following F1 scores:

- long-best-threshold-f1: 54.8
- short-best-threshold-f1: 31.4

When we shifted to kaggle challenge our fine tuned BERT model which uses the micro-f1 metric it gave us combined score of 0.22. We believe an ensemble of Bert$_{joint}$ and DecAtt+DocReader will furhter improve the score.

Example Result
Question: which is the most common use of opt-in e-mail marketing
Short answer: a newsletter sent to an advertising firm 's customers
Long answer: $< P >$ A common example of permission marketing is a newsletter sent to an advertising firm 's customers . Such newsletters inform customers of upcoming events or promotions , or new products . In this type of advertising , a company that wants to send a newsletter to their customers may ask them at the point of purchase if they would like to receive the newsletter. $< /P >$

5. Conclusion

The motivation behind this project has been to try and create a system better than the existing systems to answer natural questions. We have at the least ruled out that hyperparameter tuning on BERT may have reached saturation on this task. We learned that it takes a lot of time and computational resources to fine-tune a model of the size of BERT. BERT also needs a huge amount of data to be trained, and hence is a limitation to models based on BERT.

We also learned that selecting the right evaluation metric is an immensely important task when we submitted the predictions from baseline to kaggle.

Due to saturation on hyperparameter tuning we moved towards an ensemble approach where we tried a novel ensemble of DecAtt + DocReader with BERT. Google has also recently released an Open Domain Question Answering challenge. And the leaderboard backs our idea of using ensemble models. The majority models on the leaderboard top 10 of this challenge are BERT ensemble with other models.

Ofcourse ensembling has been done before. And there are definitely much more precise ways to deal with it than just the stochastic approach of combining them together. If weight to each model was a learn-able parameter, that would make for a better, more robust system. And we propose this as future work to our task.

References

[1] C. Alberti, K. Lee, and M. Collins. A BERT baseline for the natural questions. *CoRR*, abs/1901.08634, 2019.

[2] D. Chen, A. Fisch, J. Weston, and A. Bordes. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[3] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[4] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, M. Kelcey, J. Devlin, K. Lee, K. N. Toutanova, L. Jones, M.-W. Chang, A. Dai, J. Uszkoreit, Q. Le, and S. Petrov. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*, 2019.

[5] A. Parikh, O. Täckström, D. Das, and J. Uszkoreit. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255, Austin, Texas, Nov. 2016. Association for Computational Linguistics.

[6] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[7] S. Reddy, D. Chen, and C. D. Manning. Coqa: A conversational question answering challenge. *CoRR*, abs/1808.07042, 2018.