REPORT R3

Devarsh Shah (dshah3)

Karan Dave (kdave)

Introduction

Question Answering is the field in Natural Language Processing which deals with building systems that automatically answer questions posted by humans in natural language. With increase in data, Question Answering systems have emerged as powerful platforms for efficiently retrieving relevant pieces of information. It enables asking questions and retrieving answers using natural language queries. In order to answer these questions, the machines need to understand the natural language and the context. It needs to translate the sentences into an understandable representation so that it generates valid answers. Question answering spans over research topics such as semantic parsing, knowledge representation and reasoning, sentiment analysis and image captioning. Facebook has launched its DRQA system for question answering tasks.

Question answering systems are of 2 major types: knowledge based and information retrieval systems. The system focuses on Question Processing, Information Retrieval and Answer Extraction. Question answering systems are of 2 types: open and closed domain. We focus currently on closed domain systems which try to predict short answers. We are planning to gain insights into Google NQ dataset which is open domain by predicting long and short answers.

Project Proposal

The project aims on building a question answering model to extract relevant answers based on the question provided by the user. Sometimes, it becomes a tedious process for the reader to go through an entire article. We plan to generate answers from the text passages. A reading - comprehension based question answering model would help users to get details from articles by asking questions. This project aims to give an easy understanding of the entire article by getting answers to relevant questions.

Motivation

Question Answering systems are very useful as they allow users to get insights of an article by asking few questions. Also, most of the natural language processing problems can be modeled as question answering problems. Consider the example of Text Summarization where a user can ask 'What is the summary of the article?' and the QA system can answer by providing a summary for it. Question Answering systems can be used as a part of chatbots. QA systems can be very useful to blind people when they are integrated with speech recognition systems. Getting relevant information based on the user's question would be very helpful to them. One of the best examples for QA systems is IBM Watson which helps to answer questions in natural language..

Dataset

SQUAD (Stanford Question Answering Dataset) is a reading comprehension dataset which consists of 1,00,000+ questions posed by crowdworkers on a set of Wikipedia Articles, where the answer to each question is a segment of text from the corresponding reading passage. SQUAD contains nearly 107,785 question-answer pairs on 536 articles. The questions in the dataset are generally focused on 'What' questions.

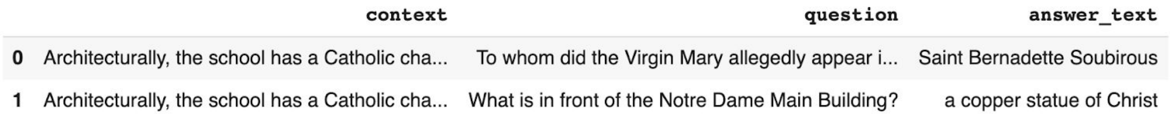
The SQUAD dataset contains triplets in the form of (context, question, answer). Each context (sometimes called a passage, paragraph or document in other papers) is an excerpt from Wikipedia. The question (sometimes called a query in other papers) is the question to be answered based on the context. The answer is a span (i.e. excerpt of text) from the context. The SQuAD dataset consists of a training and development set. The training set consists of 87,599 training points whereas the development set consists of nearly 10,570 data points.

Hypothesis:

The new methods which we are planning to use are BERT and transformer models. The transformer models are a pair of encoders and decoders. We are planning to innovate compared to RNN as RNN gives equal importance to all the tokens in the sentence making it suffer from vanishing gradient problems. Further we plan to fine tune the BERT models in order to achieve better results.

Data Extraction and Preprocessing:

We extracted the data from json files and preprocessed it to create 3 lists which consists of context, question and answer. We tried to remove white spaces and remove unwanted symbols from the context and the questions. We also tried to identify the different categories of questions i.e 'Why', 'What', 'Where', 'How' and 'When' questions. Further we also analysed the range of length of answers for all the questions.



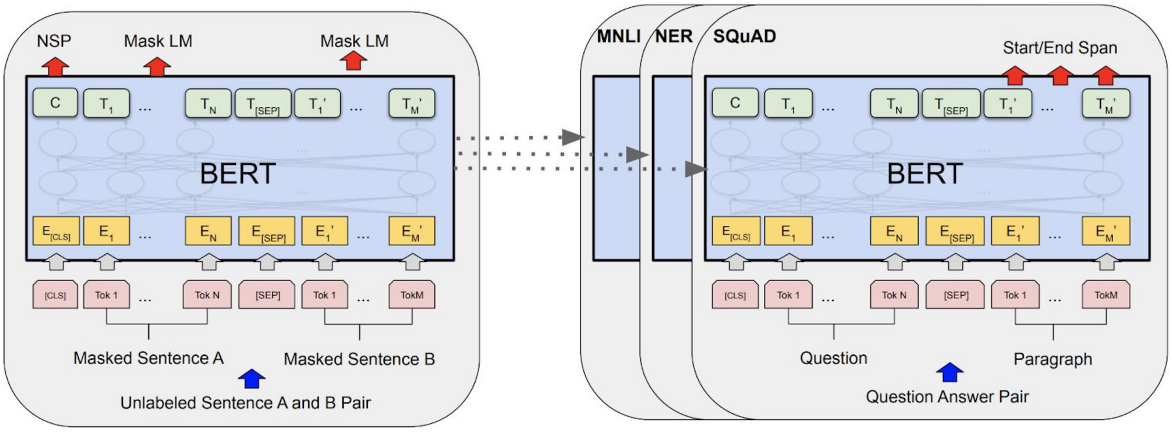
	context	question	answer_text
0	Architecturally, the school has a Catholic cha...	To whom did the Virgin Mary allegedly appear i...	Saint Bernadette Soubirous
1	Architecturally, the school has a Catholic cha...	What is in front of the Notre Dame Main Building?	a copper statue of Christ

Baseline Model(Doc2Vec)

We evaluated SQUAD dataset against baseline Doc2vec with Cosine Similarity. The doc2vec model was trained on the context and the questions provided in the dataset. Doc2vec model was trained on the dataset and it returns word embeddings of dimension 125. Answer index is predicted by finding the index of the sentence with maximum cosine similarity with the question. Our model infers vectors for new context and questions and displays the answer after finding the sentence with the highest cosine similarity.

Proposed Methodology

Our baseline model did not perform well and provided very low results. In order to achieve better results, we proposed to tune the state of the art model ‘BERT’.
BERT stands for Bidirectional Encoder Representation from Transformers. The Bert model consists of transformer blocks and trains using the left and right context of every word. The model uses an attention mechanism which is used in learning contextual relationships between the words in text. Bert also uses a masked language model which masks some random tokens and later tries to predict the mask tokens from its surroundings. There are 2 steps in the BERT framework: pre-training and fine-tuning. We fine-tuned the pretrained model made available by Google Research. The figure shows how BERT works for question answering.



Pre-training Fine-Tuning

Our model pretrains BERT using two unsupervised tasks:

1) Masked LM: It trains a deep bidirectional representation by masking a percentage of the input tokens at random and predicting those masked tokens. The final hidden vectors corresponding to this mask token are fed into output

softmax over the vocabulary. We masked 15% of wordpiece tokens in each sequence at random.

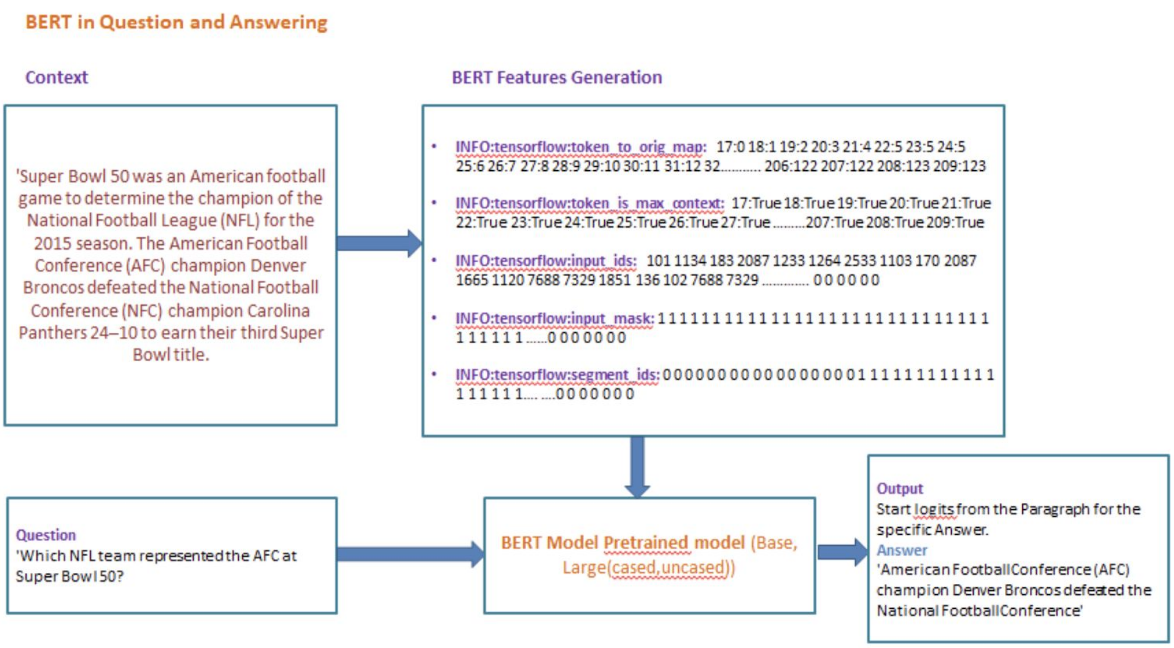
2) Next Sequence Prediction: We pre-train for a binarized next sentence prediction task that can be trivially generated from any monolingual corpus. Specifically, when choosing the sentences A and B for each pretraining example, 50 % of the time B is the actual next sentence that follows A (labeled as IsNext), and 50 % of the time it is a random sentence from the corpus (labeled as NotNext)

In the question answering system, BERT considers the question and context as a single sequence where the question is followed by the context. The embeddings are a mixture of both the token and segment embedding. A 'CLS' token which stands for classification is added at the start of the question and each sentence is expected to end with a 'SEP' token . The tokens are useful in indicating the start and end of both the question and the context. It also maintains a segment embedding which maintains a marker for the sentences. Thus, the tokens belonging to question are marked with A and tokens of context are marked with marker 'B'. BERT takes a sequence of words as input which keep flowing up the stack. Each layer applies self-attention, and passes its results through a feed-forward network, and then hands it off to the next encoder.

Implementation

In this section, we first describe the extraction of relevant data from the dataset and generation of feature records. Later we give the implementation details of the model. To implement the Question Answering model, we used the SQuAD Dataset.
We extract the question, document tokens, answer, start-end tokens of the answers and the context paragraph and store them in a list. We generate all the possible question and answer pairs for each article. These data points are stored in Squad Example instances. We then convert the training data points into tensorflow records. This step adds the 'CLS' and 'SEP' tokens between the question and the context. It further generates features such as token to origin map, token max content, Input ids, segment ids and Input mask. These features are then used as an input to the BERT model.

For implementation of the model, we used Tensorflow 2.1. We experimented with BERT uncased base model which consists of 12 transformer blocks and 110 million parameters with 768 hidden units on the SQUAD dataset. The entire model was trained on 20000 samples due to computational restrictions. We decompose the words that are not in vocabulary into subword and character words and generate embeddings for it rather than assigning 'OOV' or 'UNK' tags. We tokenize the context paragraph into a set of tokens. We trained our model by separating the tokens and assigning each word in the first sentence plus 'SEP' token 0 and all the tokens of the next sentence as 1. We converted the train examples to features as shown in figure 2 using max seq length of 256.


BERT in Question and Answering
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.
BERT Features Generation

```
• INFO:tensorflow:token_to_orig_map:  17:0 18:1 19:2 20:3 21:4 22:5 23:5 24:5
  25:6 26:7 27:8 28:9 29:10 30:11 31:12 32.......... 206:122 207:122 208:123 209:123

• INFO:tensorflow:token_is_max_context:  17:True 18:True 19:True 20:True 21:True
  22:True 23:True 24:True 25:True 26:True 27:True .........207:True 208:True 209:True

• INFO:tensorflow:input_ids:  101 1134 183 2087 1233 1264 2533 1103 170 2087
  1665 1120 7688 7329 1851 136 102 7688 7329 ............ 0 0 0 0 0 0

• INFO:tensorflow:input_mask: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
  1 1 1 1 1 1 ......0 0 0 0 0 0 0

• INFO:tensorflow:segment_ids: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1
  1 1 1 1 1 1.... ....0 0 0 0 0 0 0
```

Question
'Which NFL team represented the AFC at Super Bowl 50?
BERT Model Pretrained model (Base, Large(cased,uncased))
Output
Start logits from the Paragraph for the specific Answer.
Answer
'American Football Conference (AFC) champion Denver Broncos defeated the National Football Conference'

For model tuning, we experimented with a couple of hyperparameters such as training batch size, learning rate and epochs. For our final model, we choose a batch size of 128 with 2 epochs and a learning rate of 0.00003. The entire model was trained for 390 steps.

Technical Challenges: The major technical challenges that we faced in the project were the lack of computational resources to train our BERT model. The BERT model is a very resource expensive model and requires a large amount of memory. Also, training for even 20000 samples is a very expensive job as it took nearly 8-10 hours even on 102 GB RAM Instance. Also, with the large context size, the generation of features from the text is also a very time consuming process.

Other challenges we faced while working on the BERT model was a little lack of domain knowledge about neural networks which we can experiment with. Currently, we tried to tune the model with multiple parameters. However, with more detailed understanding of the neural network architecture, we can propose a more hybrid model which may use a combination of two architectures.

Metrics

To evaluate the performance of the model, we selected the F1 score and Exact match as the metric.

- F1 Score: The F1 score is the measure of average overlap between a model's prediction and the ground truth answer span. The Precision defines the ratio of correctly predicted words in the answer span to the total number of correctly predicted answer span words. Recall is defined as the ratio of correctly predicted answer span words to total number of words in answer span.
- Exact match: This metric measures the percentage of predictions that match any one of the ground truth answers exactly.

Results

For the baseline model, we preferred accuracy as an evaluation metric. The overall accuracy of the model is shown in the below table. The model does not seem to perform well. We can improve this accuracy by increasing the number of epochs in the Doc2vec model.

	Accuracy
Doc2vec Cosine	31.5

Results for Baseline Model

As mentioned in the evaluation metrics, we use F1 score and Exact match to measure the performance of the models. The model evaluation was done on the development dataset. Since the training was done on a subset of data i.e 20000 training points, for testing we evaluated our model by taking 2000 data points from the development dataset.

Our proposed fine tuned BERT model seems to perform well with the given training examples. Due to lack of computational resources and less number of training points, our model could not achieve scores better than other models. However, with the current number of samples the model still seems to give a good F1 score. The model can improve its performance by increasing the number of training examples.

Further, our results also showed that with increasing the number of training points and number of epochs, the model would perform better. The model is currently underfitting and it doesn't have enough data to learn the exact start and end points of the answer. However, we found that with a slight increase in the number of epochs and training points, it performed better. However, the process requires high memory GPU's to train the model faster.

Model/Metric	F1 Score	Exact Match
Proposed Model	78.92	68.78
BiDAF-ELMO (Other models)	85.6	-
BERT LARGE (Other models)	90.9	84.1

Comparison of Proposed Model with other Models

Key Takeaways:

We have presented a fine tuned BERT model for span answer prediction using SQUAD dataset. We used a transformed based approach and investigated the effect of attention mechanisms in detail. Our model performs better than baselines and human evaluations which suffer from uni-directionality and vanishing gradient problems. The model is not producing astounding results because of the lack of deep networks and computational resources. We need better evaluation metrics which can efficiently predict the span generation task. Experimenting more with data preprocessing, hyper tuning parameters could yield better results.

For future work, we are considering expanding to open domain question answering systems like Google Natural Questions Dataset. This is more challenging because it predicts both long and short answers on the whole wikipedia article instead of a paragraph. The dataset is more balanced as compared to SQUAD dataset. We are planning to reduce candidate generation scope using TF IDF and then pass the set of candidates to BERT+BiGRU model as mentioned in the paper.

References:

1. Abdi, A., Idris, N., Ahmad, Z., 2016. Qapd: an ontology-based ques-

tion answering system in the physics domain. Soft. Comput., 1–18 https://doi.org/10.1007/s00500-016-2328-2.

1. Pranav Rajpurkar and Jian Zhang and Konstantin Lopyrev and Percy Liang, 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. https://arxiv.org/pdf/1606.05250.pdf
2. Natural Questions: A Benchmark for Question Answering Research. https://www.mitpressjournals.org/doi/pdf/10.1162/tacla00276
3. Natural Questions: A Benchmark for Question Answering Research. https://www.mitpressjournals.org/doi/pdf/10.1162/tacla00276
4. Ng, Hwee Tou, Leong Hwee Teo, and Jennifer Lai Pheng Kwan. ”A machine learning approach to answering questions for reading comprehension tests.” Association for Computational Linguistics, 2000.
5. Turian, Joseph, Lev Ratinov, and Yoshua Bengio. ”Word representations: a simple and general method for semi-supervised learning.” Proceedings of the 48th annual meeting of the association for computational linguistics. Association for Computational Linguistics, 2010.
6. Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova. ”BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”
7. M. M. Rahman, S. A. Khan and K. M. A. Hasan, ”Word Sense Disambiguation by Context Detection,” 2019 4th International Conference on Electrical Information and Communication Technology (EICT), Khulna, Bangladesh, 2019, pp. 1-6, doi: 10.1109/EICT48899.2019.9068810.
8. Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading. arXiv preprint arXiv:1601.06733, 2016.
9. Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, Hannaneh Hajishirzi. ”Bidirectional Attention Flow for Machine Comprehension”, ICLR Conference 2017
10. Caiming Xiong, Victor Zhong, Richard Socher. ”Dynamic Coattention Networks For Question Answering”, https://arxiv.org/abs/1611.01604, 2016
11. Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, Ming Zhou, ”Gated Self-Matching Networks for Reading Comprehension and Question Answering”,Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2017
12. Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, Ming Zhou, ”Gated Self-Matching Networks for Reading Comprehension and Question Answering”,Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2017
13. Wei Wang, Ming Yan, Chen Wu, ”Multi-Granularity Hierarchical Attention Fusion Networks for Reading Comprehension and Question Answering”, Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)
14. Wilson L Taylor. 1953. Cloze procedure: A new tool for measuring readability. Journalism Bulletin, 30(4):415–433.

15. XLNet: Generalized Autoregressive Pretraining for Language Understanding Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonel Ruslan Salakhutdinov, Quoc V, Le
16. https://arxiv.org/ftp/arxiv/papers/1911/1911.01528.pdf
17. https://web.stanford.edu/class/cs224n/reports/default/15792151.pdf