Exploring Question Answering (QA) on Tweets using BERT and its variants

Judhajit Roy, Krishnan C.S., Shivali Singh

Department of Computer Science, University of Illinois at Chicago

Abstract

Social Media platforms have become an integral part of people's life over the last few years. They act as a provider of varied information, a subpart being the source of news and various real-time events that are happening around the globe. In this paper, we aim to explore Question Answering in a more informal space like twitter. Almost all of the previous research focused on the Question Answering task is done on datasets containing formally or academically written literature. The issue of unavailability of social media data for training models for question answering is addressed by using the TweetQA dataset. We study techniques like Data Augmentation to improve standard BERT models and observe how hyperparameter tuning effects the performance of the different models. The performance of models finetuned on SQuAD was compared with the vanilla models. We use the Exact Match and F1 score metrics and our results point to the need of improved QA systems targeting social media text.

1 Introduction

Question answering is a critical NLP problem, which is concerned with building systems that automatically answer questions posed by humans in a natural language. It has applications in many fields where a language model is trained on a large collection of articles/paragraphs and the model is able to answer questions pertaining to the paragraph accurately. Social media is now becoming an important real-time information source, especially during natural disasters and emergencies. It is now very common for traditional news media to frequently probe users and resort to social media platforms to obtain real-time developments of events. Previous datasets have concentrated on question answering (QA) for formal text like news and Wikipedia. In this project we will use TweetQA (Xiong et al.,2019), the first large-scale dataset for QA over social media data to compare different models of BERT. BERT's key technical innovation is applying the bidirectional training of Transformer, a popular attention model, to language modeling. This contrasts with previous efforts which looked at a text sequence either from left to right or combined left-to-right and right-to-left training. A bidirectionally trained language model can have a deeper sense of language context and flow than single-direction language models.

2 Related Work

2.1 TweetQA

Since the inception of language processing, several tools have been developed for handling noisy social media text, but little work has been done on question answering or reading comprehension over social media due to lack of available datasets. To overcome this hurdle TweetQA data set is created, which contains only tweets used by journalists in news articles, ensuring that the dataset contains accurate and relevant information.

For the task of Question Answering three strong neural models namely Generative QA, BiDAF and Fine-Tuned BERT on TweetQA have been experimented with before. Results show that models that work well on datasets collected from formal sources don't work as well on tweets, as tweets are informal in nature, suggesting the need for a better model that is more specifically for social media data.

Related/related works in the field include a Twitter part-of-speech tagger made by Gimpel et al. (2011) using a dataset of 1,827 manually annotated tweets. Ritter et al. (2011) created another Twitter dataset by annotating 800 tweets and performed part-of-speech tagging and chunking on it. They also worked on the task of Twitter Named Entity Recognition on a dataset comprised of 2400 annotated tweets. Kong et al. (2014) built the first dependency parser for tweets using 929 annotated tweets as dataset and the Chinese counterpart was made using 1000 annotated Weibo posts by Wang et al. (2014).

2.2 Reading Comprehension

There have been primarily two styles of Reading Comprehension datasets. The cloze-style (Hermann et al., 2015; Hill et al., 2015) is aimed at producing single-token answers from automatically constructed pseudo-questions and the quiz-style problems (Richardson et al.,2013; Lai et al.,2013) focus on selecting an answer from multiple candidates. However, these styles cannot serve as standard QA benchmarks due to their unnatural settings which lead to the compilation of the popular crowdsourced datasets containing questions answered by human annotators. Examples of such datasets include SQuAD (Rajpurkar et al., 2016), MS MARCO (Nguyen et al., 2016), NewsQA (Trischler et al., 2016), all of which are in formal language as they comprise of passages derived from Wikipedia, news articles or fiction.

2.3 Transformer Based QA

Previously, for question answering tasks on social media data BERT (J. Devlin et al., 2019), ALBERT (Lan et al.,2019) and SpanBERT(Butt et al.,2021) have been trained on TweetQA dataset and the models are compared using ROUGE, METEOR and BLEU metrics. We have explored additional models like DistilBERT (Sanh et al.,2019) and RoBERTa (Liu et al., 2019) and compared all of them using F1 and exact-match score.

2.4 Easy Data Augmentation

One of the most popular data augmentation techniques for NLP tasks is Easy Data Augmentation, proposed in the paper "EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks" . Easy Data Augmentation is designed to work by augmenting text for classification tasks, taking in one sentence from the training set and performing one of the following operations chosen at random: synonym replacement (SR), random swap (RS), random insertion (RI), and random deletion (RD) to create a new augmented example (Wei, J et al.,2019). EDA is known to improve the performance of machine learning classifiers and in our project we alter this technique to work for Question Answering and augment questions in the dataset to increase data.

3 Problem Statement

To compare the performances of extractive transformer models like BERT and its variants on TweetQA dataset.

4 Technical Approach

We are using BERT and its variants borrowed from the literature and training them on the TweetQA dataset to evaluate their performance. Question answering datasets that have answers extracted from the content/paragraph have the span of the answer present. TweetQA is an abstractive question answering dataset, it consists of paraphrased answers. We had to modify the dataset into the squad format for training. Through modification we obtained the span of the answers present in the context(tweet) and removed the rest of the answers. We developed an original code that iterates through the answers in the TweetQA and searches for its position in the tweet. The method returns the starting index of the answer and returns -1(index of last character) in case it doesn't find any such instance in the tweet. After starting indexes are available, we run another bit of code that iterates through all the tweets, questions and answer pairs and modifies the format to match the squad format.

We used pretrained HuggingFace models and Simple transformers library to fine-tune and test the performance on the modified TweetQA dataset. To establish the baseline, we used the smallest and quickest versions of BERT – tinyBERT(Jiao et al.,2019) and miniBERT. The input for models is the tweet, question and answer span. Further, experimentation was done on standard Bert, DistilBERT, ALBERT and RoBERTa as well as models fine-tuned on Squad 2.0. The models are then tested on the dev set and metrics used for evaluation are exact match and F1-score.

4.1 Data Augmentation

The TweetQA dataset is a smaller dataset than the SQUAD dataset with 10k tweets and BERT models might perform better with more data. Therefore, we experimented with data augmentation on the dataset. Easy Data Augmentation is a state-of-the-art algorithm for data augmentation for binary text classification. EDA makes use of four simple operations to generate augmented sentences given a basic sentence. The algorithm takes in one sentence from the training set and performs one of the following operations chosen at random:

synonym replacement (SR), random swap (RS), random insertion (RI), and random deletion (RD) (Wei, J et al.,2019). However, the data format in Question Answering is different and any modification in the tweet might change the meaning and impact the answer span. Therefore, we decided to augment only the questions corresponding to the tweet using synonym replacement. We would choose 1 word in a question that is not a stop word and replace it with a synonym chosen at random (Mofid, N et al.).

After the TweetQA dataset was molded into the SQUAD format we passed the questions in the dataset into a synonym replacement method we adapted from the EDA paper and its according Github using nltk WordNet to produce synonyms. From here, we augment 5% to 20% of the total data and append the generated questions to the dataset as a new entry. From there, we trained the models that had the best performance previously dataset on the augmented and evaluated it on the dev set using the same metrics.

5 Experimental Setup

5.1 Data

Column Name	Description	Type
Tweet	Contains the content of the tweet	String
Question	Question based on the tweet	String
Answer(s)	Possible answer(s)	String
qid	Unique identifier of the question-answer pair	String

Table 1: TweetQA input data format

The dataset used was taken from the paper TweetQA, which is a dataset focused on Social Media Question Answering. The data consists of tweets used by journalists to write news articles, and questions and answers written by human annotators. The dataset has 3 files namely train.json, dev.json and test.json. The training data consists of 10k+ entries, the dev data consists of 1k+ entries and the test data consist of 2k+ entries. We have used train.json and dev.json for training and evaluation respectively.

We converted the TweetQA dataset to SQuAD dataset format which has the following fields, ‘context’ (in this case tweet) and ‘qas’. qas contains the questions, question ids and their corresponding answers with the start index of the answer within the tweet. We are not performing any other preprocessing like removing punctuation, special characters and numbers so as to not affect any information present in hashtags, emoticons, ids etc. as they might help provide additional meaning to the model for this task.

Apart from that we applied data augmentation to generate 5% to 20% additional questions by replacing a word in question with it’s synonym.

5.2 Research questions

1. Which variant of BERT performs better on this dataset?

   BERT has many variants depending on the number of layers and modifications made in the design of the network. We thought that it is essential to determine the performance of the variants on this dataset to narrow down the best performing.

2. Does fine tuning the model on SQuAD help increase its performance on TweetQA?

   The SQUAD dataset is a trademark dataset for Extractive Question Answering and we have modified TweetQA dataset into SQuAD format. We hypothesize that any model that is already finetuned on SQUAD should perform well on our modified dataset.

3. Does the performance of BERT models change when a single-line tweet is the input? (As opposed to a paragraph of context)

   The context is a large paragraph with multiple questions and their corresponding answers in the original SQUAD dataset whereas in our modified dataset the context is a 280-character tweet with a single question and answer. We wanted to observe if word length impacts the performance in any way.

4. How much does Data Augmentation enhance the performance of the models?

   Data augmentation is an useful method to generate additional data replicating the existing data. As, TweetQA consists of around 10k+ tweets we wanted to experiment if data augmentation has an effect on the overall performance of the models.

5. Does hyperparameter tuning help improve BERT performance?

There are several hyperparameters such as number of epochs, learning rate, batch size etc. We wanted to see how the model reacts to changes in hyperparameter values.

5.3 Evaluation Metrics

We have used F1 and Exact Match scores as evaluation metrics to assess and compare the performance of different models. The F1-score is a way of combining the precision and recall of the model, and it is defined as the harmonic mean of the model's precision and recall. It is a looser metric that measures the average overlap between the prediction and ground truth answer. The Exact Match metric measures the percentage of predictions that exactly matches the ground truth answer.

5.4 Experiments

We initially trained multiple variants of uncased BERTs on the TweetQA dataset and compared their performance on the dev set. Then, we compared the performances of BERT variants finetuned on SQUAD. We further tried hyperparameter tuning on the models that had best performance on the dataset. For BERT, we finally used a learning rate of 3e-5 for 3 epochs over 2 batches and for DistilBERT we used a learning rate of 5e-5 for 3 epochs. After tuning the models, we trained the models with best hyperparameters on the augmented dataset with a range 5% to 20% augmentations and compared the model performances again.

6 Results

Model	F1	Exact match
tinyBERT	0.336	0.177
miniBERT	0.534	0.354
ALBERT	0.642	0.317
DistilBERT	0.652	0.481
RoBERTa	0.654	0.327
BERT	**0.694**	**0.538**

Table 2: Model Performance on dev set

We were able to train multiple BERT models and obtained comparable results on this task. According to our experiment the result, as shown below, is quite satisfactory. Overall, we can see that mini-Bert and tiny-Bert were the poorest performing models. This could be due to these models being a compressed variant of BERT with very few parameters used to design them. BERT was the best model among all it's variants, closely followed by DistilBERT, ALBERT and ROBERTa. We observed a similar order when models fine-tuned on SQuAD were evaluated, with the performances of the models generally improving by 0.01 points apart from DistilBERT which had a increment by 0.05 points. Finally, we evaluated the best models on the augmented data, BERT finetuned on SQuAD was again the best performing model followed closely by DistilBERT finetuned on SQuAD.

Model	F1	Exact match
tinyBERT	0.427	0.234
miniBERT	0.617	0.426
ALBERT	0.653	0.321
RoBERTa	0.664	0.327
DistilBERT	0.700	**0.537**
BERT	**0.704**	0.522

Table 3: Model Performance finetuned on SQuAD

Model	F1	Exact match
ALBERT	0.642	0.317
DistilBERT	0.670	0.518
BERT	0.696	0.543
DistilBERT finetuned SQuAD	**0.701**	**0.540**
BERT finetuned SQuAD	**0.711**	**0.549**

Table 4: Model Performance after Data Augmentation

Model	F1	Exact match
BERT + SR	0.696	0.543
DistilBERT SQuAD	0.700	0.537
BERT SQuAD	0.704	0.522
BERT SQuAD + SR	**0.711**	**0.549**

Table 5: Final Results

7 Conclusion

Our experiments show that BERT finetuned on SQuAD, trained on the data augmented with synonym replacement was the best performing model. We achieved a F1 score of 0.711 and Exact match of 0.549. We observed that the models finetuned on SQuAD performed comparatively better than the standard models. One explanation to this could be that SQuAD is a very balanced extractive question answering dataset and that prior knowledge obtained from it improves the efficiency of the model. Another observation was that tuning hyperparameters did not affect the model performance considerably. This goes to show that models like BERT and its variants are robust in nature and do not require hyperparameter tuning to improve their performance. Lastly, we found that training on more data via data augmentation helped increase the performance of the models by at least 0.01 points.

8 Future Work

Next steps to this experiment could be to experiment with assembling the models that performed well and assess the outcome against the standalone models. Social media platforms like Twitter usually contain special characters like hashtags, usernames, emojis etc. Currently, we are training the models on raw text that contain these special characters and the only preprocessing done is converting it to lower case. We could also try and remove these special characters to determine how that affects their performances. In this experiment we used the principle of synonym replacement from EDA for data augmentation. There several other techniques like Back-Translation and Contextualized Word Embeddings that can also be used to augment data. Further experiments using these techniques might help improve performance.

Overall Experience and Contributions

This project was a great learning experience for us. We learnt a lot and were able to apply techniques learnt in the course. The contribution of the group members are as follows:

Judhajit Roy: Preprocessing, data augmentation, Training and testing several BERT models, report work (technical approach, research questions)

Krishnan CS: Preprocessing, setting up the testing environment, Helper methods for evaluation metrics (F1 score and Exact Match), formatting code, Training and testing several BERT models, report work (results, experimental setup)

Shivali Singh: Hyperparameter tuning, formatting report, Training and testing several BERT models, report work (abstract, introduction, related work)

References

Xiong, W., Wu, J., Wang, H., Kulkarni, V., Yu, M., Chang, S., Guo, X., & Wang, W.Y. (2019). TWEETQA: A social media Focused Question Answering Dataset. *ACL*.

P. Rajpurkar, R. Jia, και P. Liang, ‘Know What You Don’t Know: Unanswerable Questions for SQuAD’,*CoRR*,τ.abs/1806.03822, 2018

Butt, Sabur & Ashraf, Noman & Fahim, Hammad & Sidorov, Grigori & Gelbukh, Alexander. (2021). Transformer-Based Extractive Social Media Question Answering on TweetQA. Computación y Sistemas. 25. 10.13053/cys-25-1-3897.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *ArXiv, abs/1910.01108*.

J. Devlin, M.-W. Chang, K. Lee, και K. Toutanova, ‘BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding’, στο *NAACL*, 2019.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2020). ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *ArXiv, abs/1909.11942*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. *ArXiv, abs/1907.11692*.

Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., & Liu, Q. (2020). TinyBERT: Distilling BERT for Natural Language Understanding. *ArXiv, abs/1909.10351*.

Wei, J., & Zou, K. (2019, August 25). Eda: Easy data augmentation techniques for boosting performance on text classification tasks. Retrieved May 5, 2022, from https://arxiv.org/abs/1901.11196

Mofid, N., Pinilla, E., & Freeman, E. (n.d.). Tackling SQuAD 2.0 with Ensemble Methods and Data Augmentation. Retrieved from https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1204/reports/default/report13.pdf