Project Definition

TwitterQA: Question Answering in Social Media

CSI 5180

Kaishuo Wang, 300068284
Feier Zhang, 8589976

Mar 12, 2023

Description

Task of this project is to build a QA system and compare performances of different language models on TweetQA, it requires models to read a brief tweet and a corresponding question and generate a texture response as an answer, which may or may not be present in the original tweet.

Goals and limitations

Achievement

From this project, we will develop a QA system on TweetQA by going through the entire process of building a QA system, including data preprocessing, feature extraction, model selection, training and evaluation, this would provide us a valuable experience in developing a QA system and working with NLP techniques. Furthermore, we will fine-tune the pre-trained BERT models and compare the performance of different BERT models for this task, this could help us learn how to fine-tune BERT for a specific task and dataset, and gain a deeper understanding of the models and their capabilities. In addition, TweetQA is the data from social media, so we will have a better understanding of how social media data can be processed and analyzed for insights.

Prior knowledge

Firstly, since the goal of a QA system on TweetQA is to retrieve relevant information from a large corpus of tweets, we need to have knowledge of information retrieval techniques, such as query expansion and relevance feedback. Secondly, we also need to know how to handle informal language, slang, and noisy data and understand social media data and its characteristics. Thirdly, we need a good understanding of language models, this will help us better to choose some suitable models for this task. In order to evaluate the performance of different models of the QA system on TweetQA, knowledge of evaluation metrics such as accuracy, precision, recall, and F1 score would also be required. Finally, many NLP and machine learning libraries are available in Python, so a prior knowledge of Python programming would be useful.

Algorithms/Approaches

In this project, we will test the performance of BERT and its variants such as DistilBERT and RoBERTa on the TwitterQA dataset. We will use Transformers and PyTorch to fine-tune models on the TwitterQA dataset and use these fine-tuned models for inference. For the evaluation, we will be using Exact Match and F1-score. Exact Match equals 1 if the characters of the model's prediction exactly match with the characters of the true answer otherwise it will be 0. And the F1 score will be calculated based on the number of shared words between theoretical and predicted answers.

Final Deliverable

The final deliverable of this project is a question-answering system trained on TweetQA dataset which takes a tweet and corresponding question as input, and generates a textual response as the answer. Additionally, we will include a report about introducing different

BERT models we will use in this task, comparing the performance of these models using appropriate evaluation metrics, and analyzing the reasons causing the performance difference.

Contribution Field

The development of such systems could lead to advances in NLP techniques, especially in dealing with social media data, which is notoriously less formal and more challenging to process than other types of textual data. Also, it will enhance social media analytics, because there is a vast amount of noise and irrelevant content on social media and the ability to extract information from tweets could help businesses and researchers gain insights into consumer behavior, sentiment analysis, and other aspects of social media communication. Moreover, it can also be used by search engines to find relevant information from social media posts.

Project Boundaries

Since fine-tuning the BERT model can take a considerable amount of time, we limited the number of models we are going to test in this project to 3. However, in order to achieve the goal of this project which is to compare the performance of BERT and its variations on the question-answering system, we may adjust the number of BERT variants tested. That is to say, we will spend more time adjusting the training parameters when fine-tuning the BERT model instead of testing more models to achieve the goal of testing their performance.

Description and Justification of dataset

We will use TweetQA as our dataset. TweetQA is developed by Wenhan Xiong, Jiawei Wu, Hong Wang, Vivek Kulkarni, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang at the University of California and Santa Barbara IBM Research. It is a large-scale dataset designed for training and evaluating question-answering systems on social media data. The dataset is designed to test a system's ability to answer questions in a noisy and informal environment, where text can be shortened. Unlike other QA datasets such as SQuAD, where the answers are extracted, TweetQA permits abstractive answers.

There are some reasons why we choose TweetQA as the dataset:

1. Tweets are a form of informal text that is commonly used in social media. Therefore, developing QA systems that can understand and answer questions based on tweets is highly relevant to real-world scenarios.
2. Unlike other QA datasets that focus on extractive answers, TweetQA allows for abstractive answers. This means that the QA system must not only identify relevant information from the tweet but also generate a meaningful answer that may not be an exact match to the original text.
3. Tweets often contain complex linguistic phenomena, such as sarcasm, irony, and colloquial language. As a result, TweetQA provides a challenging task for QA systems, as they must be able to identify and understand these phenomena to produce accurate answers.

4. TweetQA covers a wide range of topics, including entertainment, politics, sports, and current events. This provides a diverse set of data for QA systems to train and test on, ensuring that they are robust and can handle a variety of topics.

Activity Table

Activity	Why	Time Planned	Deliverable
Read articles	Gather knowledge about how to fine-tune a BERT model and build a question-answering system	5h	Summary of the articles to be shared by the group with important ideas highlighted
Explore dataset	To determine how we are going to do the preprocessing	3h	A summary of observations about the dataset which could impact our future development
Write code for preprocessing dataset	To clean data and make data into the proper format for question-answering system	3h	Cleaned dataset in the format of Stanford Question Answering Dataset (SQuAD)
Develop baseline system using BERT	To have a system to compare to	5h	Baseline system
Develop question-answering system using DistilBERT	To compare with the baseline system	5h	Question-answering system using DistilBERT
Develop question-answering system using RoBERTa	To compare with the baseline system	5h	Question-answering system using RoBERTa
Adjust training parameters of each system	To improve the performance of each system	17h	Get the optimal performance of each system
Write final report	To compare the performance of each system and analysis the results	7h	Final report