Question Answering System-NLP

Team members: Alex Weaver, Nitin Dunday Mohan, Diwash Humagain, Brinda Potluri

GitHub Link: https://github.com/Diwash95/NLP-project-Question-Answer-System

Motivation

Answering questions with the least possible human interaction (automating the question answering system) and the desire to explore models like BERT, Albert drove us to choose this for our project. Machine learning and natural language processing techniques are only beginning to be used as chatbots, & as a medium of information retrieval.

We see that there is a lot of potential here, and that it can have an impact on things like 'Customer Service' (Understanding customer's needs to engage prospective customers and identify new prospects), 'Administration and Marketing', and 'Awareness Generation' (QA nlp systems proved to be quite useful during the Covid-19 epidemic.

It could answer questions like "How to prevent corona?", "What are the symptoms?", "How to isolate at home?" and so on). Therefore, our project would use existing models and fine-tune them to answer questions from a given context. The actual usage goal is to help find a user an accurate answer.

Significance

We are well aware that the question answering NLP system has wide range applications in many industries like healthcare, business, search engines etc., we are trying to build a model (later deploy it as a web application) where when we pass a paragraph along with a question we will get its answer as an output.

It can be proved very helpful in situations like when a human is not available to answer the query, there won't be any delay i.e., quick response will be obtained making life easy. Understanding customer's needs to engage prospective customers and identify new prospects, QA nlp systems proved to be quite useful during the Covid-19 epidemic etc.

Objectives

In this project, we'll be working together to solve one common task: answering questions. We'll use an already fine-tuned BERT model from the Hugging Face Transformers library to answer questions based on the stories from the CoQA dataset. I'm convinced that by looking at the code, you'll notice how simple it is to use a fine-tuned model for our needs. In the next increment of this project, we will use the squad2 dataset on the Roberta model to do a similar analysis.

Features

For the first increment, we choose a dataset released by Stanford NLP called CoQA dataset, which is a Conversational Question Answering dataset. It's a big data set for creating Conversational Question Answering Systems. The goal of this dataset is to assess how well automated systems can comprehend a text passage and respond to a series of interrelated questions that emerge during a conversation.

Each conversation in this dataset is collected by matching two crowd workers to talk about a text in the form of questions and replies, so the inquiries/exchange are conversational. This fits our needs. The story, input text from the question, and answer from the JSON dataset will be used to create our data frame as features. The column version is deleted as we don't need it. The text provides the context and derives the answer to a given query.

Related Work

The question answering study addresses various types or forms of questions like 'why', 'facts', 'how', etc. This field of NLP has also been developed in the medical arena for example, 'alzheimer's disease'. The earliest 'question-answering' models "Lunar" and "Baseball" (where experts have given the 'hand written' database) paved the way for huge developments in research in question answering systems. Gradually it became more domain specific.

Earlier mostly expert's knowledge or handwritten scripts were used whereas now unstructured statistical data using corpus. The development of this has been continued since then, like automating geospatial questions or temporal etc.

Dataset

We use the CoQA dataset, which is a Conversational Question Answering dataset released by Stanford NLP, for the first increment. It's a large data set used to develop Conversational Question Answering Systems. The purpose of this dataset is to see how well automated systems can understand a text passage and react to a series of relevant questions that arise during a conversation.

Each discussion in this dataset is gathered by pairing two crowd workers to discuss a text in the form of questions and responses, resulting in conversational inquiries and exchanges. This is ideal for us. As features, we'll take the tale, the input text from the question, and the answer from the JSON dataset to generate our data frame. We don't need the column version; thus, it's gone. The literature sets the scene and deduces the solution to a specific question.

Detail Design of Features

It's a 'Closed Dataset', which means that answer to the 'question' is always part of context and that the context spans a continuous period of time. Getting starting index and ending index of 'context' that correlates to responses simplifies the difficulty of finding the answer.

We have implemented 2 models BERT (2 versions/ models one is using pre-trained and the other one using CoQA dataset), and ALBERT respectively.

Used CoQA to train the model along with the pre-trained finetuned-squad model.We gave a passage for which we passed a question and the model gave answers. When the passage had

more than 512 words the model threw an error because of which we built one more model called 'BERT' to overcome this issue.

In BERT model we used 'split' function and created multiple sub components where each component wouldn't cross 512 words and would give an output.

Analysis

We evaluate the models by comparing both the f1 score (the harmonic mean of the precision and recall) as well as the exact match score of the models run on the squad dev set

Implementation

We evaluate how 4 different pre-trained models perform on the dataset. The 4 models are all versions of the transformer-based machine learning model BERT. The models all differ in the number of parameters but have a similar architecture of multiple encoder, attention, and decoder layers.

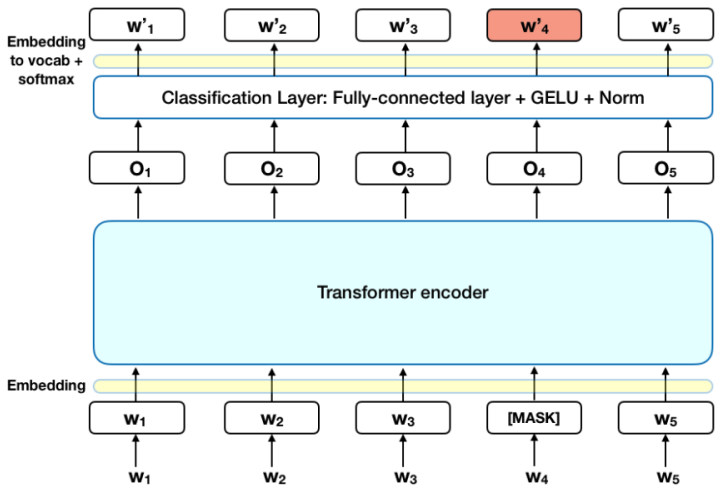

Fig.:1 Transformer Model

Preliminary results for analysis:

```
text = """Elon Reeve Musk FRS (/ˈiːlɒn/; born June 28, 1971) is an entrepreneur, investor, and busi
question = "Where did elon musk attend?"
print(question)
questionanswer(question, text)
```

```
Where did elon musk attend?

Answer:
University of pennsylvania
```

Fig.2: Output

Work completed:

NAME	TASK
Diwash Humagain	Built a BERT model using CoQA dataset, Objectives, features, preliminary results for analysis, references.
Nitin Dunday Mohan	Built another BERT model which overcomes the issue of the previous BERT model, Motivation, significance, related work, references.
Brinda Potluri	Built an ALBERT model, Dataset, detail design of features, work completed, work to be completed,
Alex Weaver	Built an DiBERT model, Analysis, Implementation, Preliminary results for analysis

Work to be completed

We will create a web application in which the user will be given the option of choosing between models before being asked to give a paragraph and a question for that.

1. We'll make a restful API which takes 2 parameters as an input of context and question and gives an answer in response to the question given.
2. A front-end web app using streamlit where user can input question and context to get an answer in response.
3. We'll try a new GPT-2 model to get much more satisfying responses (more accurate) and compare with the previous model which we have implemented.
4. Try to analyze different context on same question and figure out which model works better on different domain of data such as BERT model is working better on economic/ financial data.

Issues:

1. For the same question with different context we are getting different answer responses as the model is getting confused that what is the right entity to be extracted.
2. The paraphrased question with the same context is working in some situations but in other it doesn't give the same answer.
3. In a restful API which we are trying will give the response with a latency of minimum 120millisecs as the model is taking time to give response.

References

[1] Z. Zhang, H. Zhao, and R. Wang, “Machine Reading Comprehension: The Role of Contextualized Language Models and Beyond,” CoRR, vol. abs/2005.06249, 2020, [Online]. Available: https://arxiv.org/abs/2005.06249
[2] Y. Liu et al., “RoBERTa: A Robustly Optimized BERT Pretraining Approach,” CoRR, vol. abs/1907.11692, 2019, [Online]. Available: http://arxiv.org/abs/1907.11692
[3] J. Ni, T. Young, V. Pandelea, F. Xue, V. Adiga, and E. Cambria, “Recent Advances in Deep Learning Based Dialogue Systems: A Systematic Survey,” CoRR, vol. abs/2105.04387, 2021, [Online]. Available: https://arxiv.org/abs/2105.04387
[4] P. Dwivedi, “NLP - building a question answering model,” Medium, 11-Jul-2018. [Online]. Available: https://towardsdatascience.com/nlp-building-a-question-answering-model-ed0529a68c54. [Accessed: 15-Apr-2022].
[5] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, “SQuAD: 100,000+ Questions for Machine Comprehension of Text,” arXiv e-prints, p. earXiv:1606.05250, 2016.
[6]Squad V2 dataset card: https://huggingface.co/datasets/squad_v2
[7]Huggingface documentation: https://huggingface.co/docs/datasets/tutorial
[8]Transformers library documentation: https://huggingface.co/docs/transformers/index
[9]Deepset roberta base model: https://huggingface.co/deepset/roberta-base-squad2
[10]https://www.analyticsvidhya.com/blog/2021/11/end-to-end-question-answering-system-using-nlp-and-squad-dataset/
[11]https://blog.paperspace.com/how-to-train-question-answering-machine-learning-models/
[12]https://paperswithcode.com/sota/question-answering-on-squad20
[13]https://blog.seeburger.com/natural-language-question-answering-systems-get-quick-answers-to-concrete-questions/