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

Introduction:

We are using “Question and Answering Model” in nlp using pre-trained models to get the expected outputs. Initially we are passing a context and we would be giving a question based on the context or not. If the question’s context matches with the given para context it returns the output string based on the context if the context is not matched nothing will be returned as an output since the given question on the context are not related to each other. The scenario of checking a question without a given context would not be a possible scenario.

Goals and Objectives:

Motivation:

The interest to study models including "BERT", "Albert", and answer questions also with least amount of human contact (designed to automate "Question-Answering System") motivated us to chose this one for our research. " Machine learning(ML)" and "Natural Language Processing(NLP)" methods are only now being employed as "chatbot" and also as medium for 'retrieving information'.

We are seeing great deal of potential in this, and it can affect things such as 'customer service' (understanding consumer needs to be involved potential buyers and discover potential prospects), 'adminitration and marketng,' and 'awarenes generation' (QA NLP mechanisms demonstrated to be rather helpful even during 'covid-19 Epidemic'.

It could provide answers to inquiries such as "How can I avoid corona?" "But what were the signs/ symptoms?" "How do I do home isolation?" and etc. As a result, our study would refine current models that answer queries from a specific context. Its actual purpose of the application is to assist an users in getting a correct answer.

Significance:

We are quite well informed that now the "Question-Answering nlp System" indeed has wide range of uses in throughout many sectors such as "Healthcare", "Business", "Search-engines", and so on. We are attempting to construct model (which we will later deploy as 'Web-Application') in which we will receive the answer to a question once we transfer paragraph together with the question.

It could be quite useful in instances where a human isn't really present to respond to the enquiry because there will be no wait in receiving a reply, making things easier. Understanding client demands in order to involve prospective consumers and recognize new opportunities, QA-NLP system proven to be rather beneficial during in 'covid-19' outbreak, for example.

Objectives:

We'll be collaborating on this project to complete one shared task: 'Answering-Questions'. To address questions based on 'stories' from "CoQA dataset", we are using a previously "Fine Tuned BERT Model" from "Hugging-face-transformer's" library.

By glancing just at code, I'm confident you'll see just how straight forward it's to employ "Fine Tuned Model" for our requirements. We'll undertake a similar study using "Squad2 Dataset" upon "roberta Model" in next edition of just this project.

Features:

we first used the "CoQA Dataset", which would be "conversational question answering" Dataset published by the "Stanford NLP". It's a large data set that can be used to create "conversationl question-answering system".

The purpose of this particular Dataset would be to see how effectively the "Automatd Systems" can understand textual passage plus react to a sequence of interconnected queries that arise during a discussion.

Each discussion throughout this Datast is gathered by pairing 2 crowd workers that converse about something like an information form of "Questions and Answers", resulting in 'Conversaton Inquiries" and exchanges. This is ideal for us. As features, we'll take the tale, the inputs text from either the questionnaire, and the answers again from "JSON Dataset" to generate the Data-Frame.

We wouldn't need the columns edition, thus it's gone. The literature sets the scene and deduces the solution to a specific question.

Background:

A "Question-Answering Implementation", which is basically an online software, may create its replies by asking "Structured Database" containing facts or information, which is typically a source of information.

"Knowledge Bases" for narrower categories of knowledge were created in late 1970s. The "Question-Answering Systems" created to connect with some of these intelligent systems resulted in more consistent and reliable responses to inquiries within a given field of knowledge.

Except for underlying internal structure, such 'expert systems' were very similar to modern "Question Answering Systems". Many modern "Question Answer System" depends involves the use of statistical analysis of a huge, unorganized "Natural Language" text corpus, whereas expert systems rely significantly on specialist and structured domain knowledge.

The establishment of detailed ideas in language modeling in "1970s and 1980s" contributed to the creation of ambitious goals in reading instruction and "Question Answering".

The "Unix Consultant (UC)", created by "Robert Wilensky" at "U.C. Berkeley" throughout late 1980s, is a representation of system. The system responded to questions about "unix Operating-System". It intended to wording the response to satisfy diverse types of consumers, because that had a vast customized depth of knowledge among its domain.

In recent decades, "Question Answering Systems" have indeed been expanded to include more knowledge categories. Systems are created to autonomously respond to "temporal-geospatial" inquiries, "definition-terminology" questions, "biographical queries", "multilingual queries",

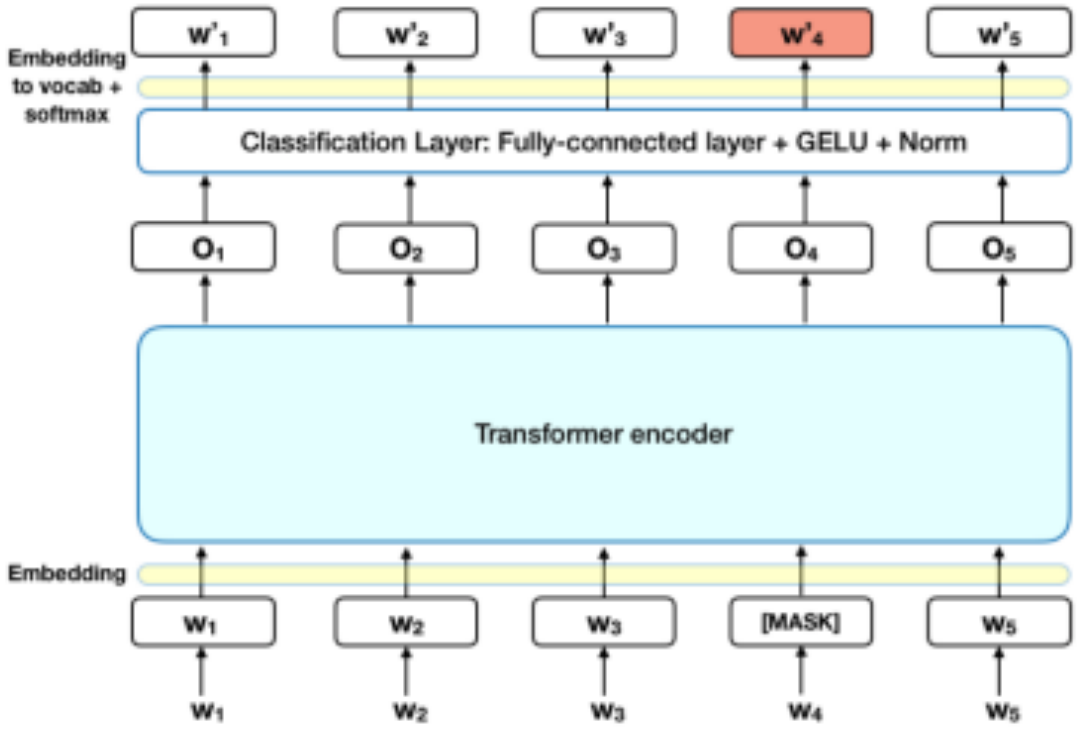
and uncertainty regarding the topic of sound, photos, and videos, for instance.

Your Model:

Because this task is relatively mature in that there are multiple high performing models, we selected a couple pre-trained models to use for our question answering engine. Using pre-trained models cuts down on the energy consumption and consequently has a lower environmental impact.

We tested both the Roberta and GPT2 pretrained models for question answering. Roberta is a variation of the google BERT model that optimizes the selection of hyperparameters during training and uses significantly more training data to achieve better results on many applications.

Like BERT, Roberta uses language masking training strategy, which forces the system to predict hidden sections of text in a semi-unsupervised fashion.



Transformer Model

Both models use bi-directional transformers and the encoder-decoder architecture, which makes it very suitable for the reading comprehension task.

It is a "Transformer Architecture" that has been taught to master language patterns and is intended to be utilized as the primary architectural "nlp" tasks. It is distinguished from previous language models by the fact that its learnt representations include content through both sides of something like the phrases. Let's look just at training that's been suggested to understand these language structures, and then see how they're employed in those other 'nlp' tasks.

Dataset:

We used the Stanford Question answering Dataset (SQuAD V2) for our project. This dataset is designed to test ML reading comprehension tasks. It consists of crowdsourced questions that can be answered in a set of context articles from Wikipedia.

The questions are constrained to be answerable by a segment of text from the corresponding context passage. The 2.0 version that we use is augmented with over 50,000 crowdsourced questions that are not answerable by a passage.

The models, therefore, must also determine when no answer is found in addition to finding the answer segment in the context when possible. There are numerous advanced models that have been tested on the dataset, many achieving accuracy higher that human ability.

Analysis of Data:

As mentioned in the previous increment, the Stanford Question Answering Dataset (SQuAD2.0) is used for our project's model training. For instance, the fine-tuned GPT2 model would learn to analyze and answer given questions from the squad.

Then for the sampling, if our models work, we first provide them with a custom corpus and query, which is also tokenized and processed to put together the answer. The features extracted are questions and context columns which are indexed after tokenization.

Next, we also implement code to combine data from any website via link or path and subject them to similar processing explained above.

Implementation:

We are using the question and answer model using pertained models such as BERT, ALBERT, GPT2, ROBERT model where the data is taken from the user such as text inputs and then those are run with the pertained model.

The pre-trained model matches with the given data using natural language text processing where the unwanted texts are removed and then the text is preprocessed.

The processed text information is then uses the model to predict the which is the right phases of words used for the question asked from the given context data. The collected information is then processed and the output is displayed based on the given information.

The data is searched in the context and the output of the code is checked as the model is pre-trained. Mostly the output of the data is given as the textual information such as numeric data, string data.

The data once trained returns the output in a single sentence or a pair of words to make sure the given predicted text is match with the output based on the training data. BERT is a "Transformer Architecture" that has been taught to master language patterns and is intended to be utilized as the primary architectural "nlp" tasks.

It is distinguished from previous language models by the fact that its learnt representations include content through both sides of something like the phrases.

Let's look just at training that's been suggested to understand these language structures, and then see how they're employed in those other 'nlp' tasks.

BERT is taught with two goals in mind:

The "Modl Learns" to anticipate these words by masking certain tokens from either the input pattern "Masked-Model".

2 sentences were fed into the 'model' as input, as well as "Modl is Trained" for doing predictions whether one follow another.

BERT's input sequence is made up of two phrases separated by a token, as well as the initial Classificaton Token that will be utilized for the 'Prediction' later. Every token does have its own embedding, as well as a segments embed which identifies every phrase and a positional embedding that distinguishes each token's position. The sum of all of those embeddings is then calculated for every token.

Results:

We ran our pre-trained models with some variations, and the results from one of them, the GPT2 model, are displayed below. In the below diagram, the question generates an answer citing the context.

There are various elements added to our project milestones. For example, we're combining models to access any given website(data) as a context.

Also, we implemented a web platform to create a simplified system for a user to get their query answered from the appropriate model having the highest accuracy.

Project Management:

Implementation of status report

NAME	CONTRIBUTIONS/ WORK COMPLETED
Alex Weaver	DiBERT and Roberta Model was built, your model, dataset, and references.
Diwash Humagain	BERT and GPT-2 Model was built, analysis of data, results, and references.
Nitin Dunday Mohan	One more BERT Model was built to overcome earlier model's issue and Deployment was done using streamlit, introduction, work done, and references.
Brinda Potluri	ALBERT Model was built and Deployment was done using streamlit, implementation, background, and references.

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.

[5] https://towardsdatascience.com/question-answering-with-a-fine-tuned-bert-bc4dafd45626

[6]https://towardsdatascience.com/how-to-apply-transformers-to-any-length-of-text-a5601410af7f