Natural language processing

Q&A CHATBOT

Nathan Margni

INDEX

INTRODUCTION

Chatbot answering question based on set of documents.

Two main approaches:

- **Extractive:** Finds the most relevant answer span within the given context or documents.

- **Generative**: Generates a new answer text based on the input context and question.

EXTRACTIVE MODEL

- Baseline algorithm, bm25 ranking to find best sentence without semantics understanding.

- BERT model extract part of text representing the answer.

- f1 = 93.15, exact_match = 86.91 on squad dataset, versus 73 and 69 with bert base, *(hugging face, SQuAD Model Comparison paper)*

MODEL

- BERT pretrained model, unsupervised training on large corpus.
- Supervised fine-tuning on Squad dataset for question answering.

TOKENIZATION

- Tokenization of documents with nltk tokenizer.
- Tokenize with bert apposite tokenizer. (using special tokens [CLS], [SEP]

DOCUMENT RANKING

- Rank documents with BM25 algorithm and use the most relevants.
- BM25 is an extension of TF-IDF with saturation and normalized for document lenght.

ANSWER EXTRACTION

- For each sentence in the relevant documents formulate the answer by computing start and end tokens scores using the bert model.
- Keep the answer with the highest start and end score!

CODE EXAMPLE

```
# Load the BERT model and tokenizer
model = BertForQuestionAnswering.from_pretrained("bert-large-uncased-whole-word-masking-finetuned-squad")
tokenizer = BertTokenizer.from_pretrained("bert-large-uncased-whole-word-masking-finetuned-squad")
```

[3] ✓ 8.1s Python

```
documents =  ["The Eiffel Tower is a wrought iron lattice tower on the Champ de Mars in Paris, France. It is named after the engineer Gustave
    "The Great Wall of China is a series of fortifications made of stone, brick, tamped earth, wood, and other materials, generally built alon
    "The Leaning Tower of Pisa is the campanile, or freestanding bell tower, of the cathedral of the Italian city of Pisa, known worldwide for
    "Mount Everest is Earth's highest mountain above sea level, with a peak that rises to an elevation of 8,848.86 meters. It is located in th
    "The Amazon Rainforest, also known as the Amazon Jungle, is a moist broadleaf forest that covers most of the Amazon basin of South America
```

[4] ✓ 0.0s Python

```
# tokenize documents
tokenized_documents = [nltk.word_tokenize(doc) for doc in documents]

# initialize bm25 with the documents
bm25 = BM25Okapi(tokenized_documents)
```

[5] ✓ 0.0s Python Python

CODE EXAMPLE

Relevant document: "The Leaning Tower of Pisa is the campanile, or freestanding bell tower, of the cathedral of the Italian city of Pisa, known worldwide for its nearly four-degree lean, the result of an unstable foundation. Construction of the tower began in 1173 and took nearly 200 years to complete.",

```
question = "why the pisa tower is leaning?"

k = 2
top_k_documents = [documents[i] for i in bm25.get_top_n(nltk.word_tokenize(question), range(len(documents)), k)]
```

✓ 0.1s

```
# split the documents
sentences = [sentence for document in top_k_documents for sentence in nltk.sent_tokenize(document)]

# iterate through sentences and calculate start and end scores with BERT
best_score = -1
best_sentence = None

for sentence in sentences:
    inputs = tokenizer.encode_plus(question, sentence, return_tensors="pt", max_length=512, truncation=True)
    outputs = model(**inputs)
    start_scores, end_scores = outputs.start_logits, outputs.end_logits

    # get tokens with highest start and end scores
    start_index = torch.argmax(start_scores)
    end_index = torch.argmax(end_scores)

    # check if current sentence has better score
    combined_score = start_scores[0][start_index].item() + end_scores[0][end_index].item()
    if combined_score > best_score:
        best_score = combined_score
        answer_tokens = tokenizer.convert_ids_to_tokens(inputs["input_ids"][0][start_index:end_index + 1])
        best_sentence = tokenizer.convert_tokens_to_string(answer_tokens)

print(f"Answer: {best_sentence}")
```

✓ 3.5s

```
Answer: the result of an unstable foundation
```

Question: why is the tower of pisa leaning?

Answer: the result of unstable foundation

PROS

- Understanding of context.
- Direct answer.
- Accurate and coherent answer with the document.

CONS

- Sensitive to document quality.
- Can cut relevant information
- Limited to answers present in the input context

GENERATIVE MODEL

- Chatgpt model generate an answer given question and context
- Chatwithpdf current solution

MODEL

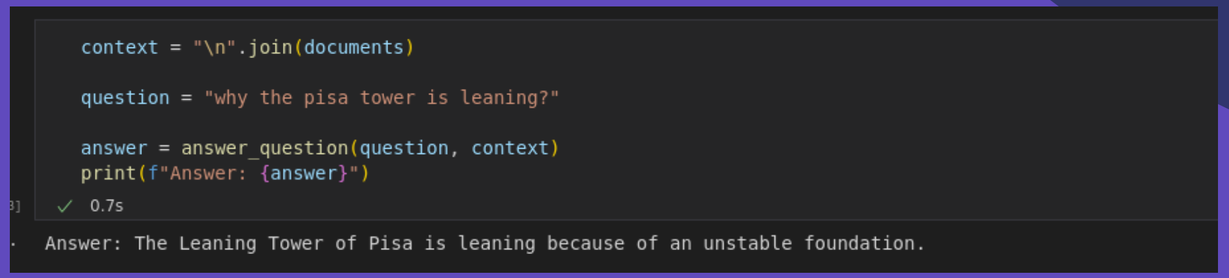
- Access to chatgpt-3 model with api.
- Pre-trained on large corpus and finetuned for question answering.

ANSWER GENERATION

- Most relevant documents can firstly be selected using some methods (b25, word embeddings etc.)
- Give to the model context (documents) and it generates the answer computing the most likely sequence.

EXAMPLE

Relevant document: "The Leaning Tower of Pisa is the campanile, or freestanding bell tower, of the cathedral of the Italian city of Pisa, known worldwide for its nearly four-degree lean, the result of an unstable foundation. Construction of the tower began in 1173 and took nearly 200 years to complete.",



Question: why is the tower of pisa leaning?
Answer: The leaning tower of pisa is leaning because of an unstable foundation.

PROS

- Flexibility and reasoning.
- Syntetization.
- Understanding of context and semantics

CONS

- Slow and complex.
- Need to send requests
- Inaccurate or off-topic reponses (allucinations)

QUESTIONS?

References:
https://huggingface.co/
https://arxiv.org/abs/2212.00857
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (paper)
chatgpt-3