Tri-Lingual Question-Answering System

Nithin Chandra Gupta Samudrala
200110076

Motivation

- A trilingual question answering system provides students with the ability to ask questions and gain a deeper understanding of content across multiple languages.

Problem Statement

Building Question-Answering System in Telugu, English and Hindi.

For each observation in the training set we have a context, question, and answer.

The goal is to find the answer for any new question and context provided.

I used “Extractive Question Answering” Method. This involves posing questions about a document and identifying the answers as spans of text in the document itself. This makes it a closed dataset meaning that the answer to a question is always a part of the context and also a continuous span of context.

Problem Statement - Contd.

Building Question-Answering System in Telugu, English and Hindi.

The context can be in any of the three languages: Telugu, English and Hindi.

The question can also be in any of the three languages mentioned above.

I used google-translate API to translate any text to English before passing it to our model.

Literature Survey

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, Percy Liang:

SQuAD: 100,000+ Questions for Machine Comprehension of Text

Link to paper

Dataset

Stanford Question Answering Dataset (**SQuAD**) is a new reading comprehension dataset, consisting of questions posed by crowdworkers on a set of Wikipedia articles, where the answer to every question is a segment of text, or span, from the corresponding reading passage. With 100,000+ question-answer pairs on 500+ articles, SQuAD is significantly larger than previous reading comprehension datasets.

Dataset Contd.

Loading the Dataset:

```
from datasets import load_dataset

raw_datasets = load_dataset("squad")
```

print(raw_datasets):

```
DatasetDict({
    train: Dataset({
        features: ['id', 'title', 'context', 'question', 'answers'],
        num_rows: 87599
    })
    validation: Dataset({
        features: ['id', 'title', 'context', 'question', 'answers'],
        num_rows: 10570
    })
})
```

Dataset Contd.

The training dataset has only one possible answer while the validation dataset has several possible answers, which may be same or different

```
raw_datasets["train"].filter(lambda x: len(x["answers"]["text"]) != 1)
```

```
Dataset({
    features: ['id', 'title', 'context', 'question', 'answers'],
    num_rows: 0
})
```

```
print(raw_datasets["validation"][0]["answers"])
print(raw_datasets["validation"][2]["answers"])
```

```
{'text': ['Denver Broncos', 'Denver Broncos', 'Denver Broncos'], 'answer_start': [177, 177, 177]}
{'text': ['Santa Clara, California', "Levi's Stadium", "Levi's Stadium in the San Francisco Bay Area at Santa Clara, California."], 'answer_start': [403, 355, 355]}
```

Model

- I have used a BERT Model (Bidirectional Encoder Representations from Transformers) and fine-tuned it for our task.
- The Architecture of this model is shown in the picture on the right-side.

```
BertModel(
  (embeddings): BertEmbeddings(
    (word_embeddings): Embedding(28996, 768, padding_idx=0)
    (position_embeddings): Embedding(512, 768)
    (token_type_embeddings): Embedding(2, 768)
    (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
    (dropout): Dropout(p=0.1, inplace=False)
  )
  (encoder): BertEncoder(
    (layer): ModuleList(
      (0-11): 12 x BertLayer(
        (attention): BertAttention(
          (self): BertSelfAttention(
            (query): Linear(in_features=768, out_features=768, bias=True)
            (key): Linear(in_features=768, out_features=768, bias=True)
            (value): Linear(in_features=768, out_features=768, bias=True)
            (dropout): Dropout(p=0.1, inplace=False)
          )
          (output): BertSelfOutput(
            (dense): Linear(in_features=768, out_features=768, bias=True)
            (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
            (dropout): Dropout(p=0.1, inplace=False)
          )
        )
        (intermediate): BertIntermediate(
          (dense): Linear(in_features=768, out_features=3072, bias=True)
          (intermediate_act_fn): GELUActivation()
        )
        (output): BertOutput(
          (dense): Linear(in_features=3072, out_features=768, bias=True)
          (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
          (dropout): Dropout(p=0.1, inplace=False)
        )
      )
    )
  )
  (pooler): BertPooler(
    (dense): Linear(in_features=768, out_features=768, bias=True)
    (activation): Tanh()
  )
)
```

Metrics

- **Exact match** : This metric calculates exact similarity between predict answer and the ground truth. For question answering, it serves as a good metric as we need answer that has to be accurate to question asked
- **f1** : it is the harmonic mean of the precision and recall of the predicted answer and ground truth answer.

Results

```
predictions = model.predict(tf_eval_dataset)
compute_metrics(
    predictions["start_logits"],
    predictions["end_logits"],
    validation_dataset,
    raw_datasets["validation"],
)
```

```
677/677 [==============================] - 338s 495ms/step
100% 10570/10570 [00:20<00:00, 380.13it/s]
{'exact_match': 80.45411542100284, 'f1': 88.26667957260283}
```

Results and Analysis

- BERT worked pretty well in extracting answers for factoid questions like “Who is the current captain of CSK?” but fared poorly when given open-ended questions such as “Is Dhoni the best captain?”.
- This is because we trained the model on dataset which has fact-based questions with explicit answers. To answer open-ended questions, it needs better interpretation and understanding.
- Answering open-ended questions might also come under the category of generative tasks which is tough for BERT as it’s language and concepts of relations between words is limited to the training data and also due to its MLM objective.
- Although MLM objective is useful for learning representations of words and their relationships with other objects, it is not as effective for learning how to generate grammatically correct text.
- When model is asked out of context question, model is hallucinating which is undesired

Results

Context-Telugu

Question-English

Results

Context-Telugu
Question-Telugu

Streamlit Question Answering App

Enter Context:

महेंद्र सिंह धोनी एक भारतीय पूर्व अंतरराष्ट्रीय क्रिकेटर हैं, जो 2007 से 2017 तक सीमित ओवरों के प्रारूप में और 2008 से 2014 तक टेस्ट क्रिकेट में भारतीय राष्ट्रीय क्रिकेट टीम के कप्तान और सीएसके के वर्तमान कप्तान हैं। उन्होंने भारत को तीन आईसीसी ट्रॉफी में जीत दिलाई, जो किसी भी भारतीय कप्तान द्वारा सबसे अधिक है। भारतीय घरेलू क्रिकेट में उन्होंने बिहार और झारखंड क्रिकेट टीम के लिए खेला। वह इंडियन प्रीमियर लीग में चेन्नई सुपर किंग्स (CSK) के कप्तान हैं। उन्होंने आईपीएल लीग के 2010, 2011, 2018 और 2021 संस्करणों में चैंपियनशिप के लिए कप्तानी की। साथ ही उनकी कप्तानी में चेन्नई सुपर किंग्स (CSK) ने दो बार, 2010 और 2014 में चैंपियंस लीग टी20 जीता। लंका। [3] उन्होंने अपना पहला T20I भी एक साल बाद दक्षिण अफ्रीका के खिलाफ खेला। [4] 2007 में, उन्होंने राहुल द्रविड़ से ODI कप्तानी संभाली और उन्होंने इस वर्ष भारत के T20I कप्तान के रूप में भी चयन किया। [5] 2008 में, उन्हें टेस्ट कप्तान के रूप में चुना गया था। [6] टेस्ट प्रारूप में उनका कप्तानी रिकॉर्ड मिश्रित था, जिसने सफलतापूर्वक भारत को 2008 में न्यूजीलैंड के खिलाफ श्रृंखला जीत और ऑस्ट्रेलिया के खिलाफ बॉर्डर-गावस्कर ट्रॉफी (2010 और 2013 में घरेलू श्रृंखला) में श्रीलंका, ऑस्ट्रेलिया, इंग्लैंड और दक्षिण अफ्रीका से हार का सामना करना पड़ा। अवे कंडीशंस में बड़े अंतर से। [7] उन्होंने 30 दिसंबर 2014 को टेस्ट प्रारूप से अपनी सेवानिवृत्ति की घोषणा की, [8] और 2017 में T20I और ODI के कप्तान के रूप में पद छोड़ दिया।

Select Context Language:

Hindi

Enter Question:

ధోని ఎన్ని ఐసీసీ ట్రోఫీలు గెలుచుకున్నాడు

Select Question Language:

Telugu

Predict Answer

How many ICC trophies has Dhoni won?

Answer: three

Results

Context-Telugu
Question-English

Demo