chaii: Hindi and Tamil Question-Answering using XLM-RoBERTa

Project under the course IT412 - Natural Language Processing

Tarang Ranpara
ID - 202011057, DA-IICT, Gandhinagar, 202011057@daiict.ac.in
Akash Dhedhi
ID - 202011053, DA-IICT, Gandhinagar, 202011053@daiict.ac.in
Darshil Patel
ID - 202011034, DA-IICT, Gandhinagar, 202011034@daiict.ac.in
Dhyanil Mehta
ID - 202011032, DA-IICT, Gandhinagar, 202011032@daiict.ac.in
Kishan Vaishnani
ID - 202011004, DA-IICT, Gandhinagar, 202011004@daiict.ac.in

This report will discuss some of the approaches we experimented in the kaggle competition called “chaii” by Google India. The competition was about performing the question answering task on a mix of Hindi and Tamil data. Evaluation was done using a jaccard score. As transformers have almost replaced recurrence based architectures like LSTMs and GRUs, we chose to go with transformer based models. In the comparison of multilingual models like XLM-Roberta and IndicBERT, XLM-Roberta was found to be superior and considered for further experimentation.

GitHub Repo: https://github.com/kdv4/chaii-Q-A-Group-4

Additional Keywords and Phrases: transformers, chaii, question answering, XLM-Roberta

1 Introduction

With nearly 1.4 billion people, India is the second-most populated country in the world. Yet Indian languages, like Hindi and Tamil, are underrepresented on the web. Popular Natural Language Understanding (NLU) models perform worse with Indian languages compared to English, the effects of which lead to subpar experiences in downstream web applications for Indian users. To tackle this issue, a competition called “chaii” was organized by Google India and on Kaggle requiring participants to perform question answering task on newly open-sourced dataset called “chaii-1”.

Question answering is inherently a sequence to sequence problem where context and question is fed as an input in the encoder, and the decoder is supposed to produce the answer as an output sequence. we discuss all approaches in section 2 and discuss their results in section 3.

2 Experiments

2.1 Dataset

In this competition, our goal is to predict answers to real questions about Wikipedia articles. We will use chaii-1, a new question answering dataset with question-answer pairs. The dataset covers Hindi and Tamil, collected without the use of translation. It provides a realistic information-seeking task with questions written by native-speaking expert data annotators.

The chaii training dataset includes 1114 samples of Hindi and Tamil questions, context, and their answers. The public version of the test dataset contains only 5 samples. The full test dataset was kept private by the competition host. While submitting the predictions, it expects us to submit the question ID and the predicted answer text only. The training dataset contains six fields namely *id, context, answer_text, answer_start, and language*. The public test dataset is shown in Fig. 1.

	id	context	question	language
0	22bff3dec	ज्वाला गुट्टा (जन्म: 7 सितंबर 1983; वर्धा, महा...	ज्वाला गुट्टा की माँ का नाम क्या है	hindi
1	282758170	गूगल मानचित्र (Google Maps) (पूर्व में गूगल लो...	गूगल मैप्स कब लॉन्च किया गया था?	hindi
2	d60987e0e	गुस्ताव रॉबर्ट किरचॉफ़ (१२ मार्च १८२४ - १७ अक्...	गुस्ताव किरचॉफ का जन्म कब हुआ था?	hindi
3	f99c770dc	அலுமினியம் (ஆங்கிலம்: அலுமினியம்; வட அமெரிக்க ...	அலுமினியத்தின் அணு எண் என்ன?	tamil
4	40dec1964	கூட்டுறவு இயக்க வரலாறு, இங்கிலாந்து நாட்டில் ...	இந்தியாவில் பசுமை புரட்சியின் தந்தை என்று கருத...	tamil

Figure-1: overall architecture of qa pipeline

2.2 Approaches

IndicBERT was experimented for inference but was found to be producing repetitively wrong answers, and hence it was not considered for further experimentation. For all experiments we use PyTorch and transformers library.

2.2.1 Approach-1: Inference Pipeline

We used an XML-Roberta-Large that was fine tuned on the SQuaD-2 dataset. we provide context as an input, and receive answers as an output.

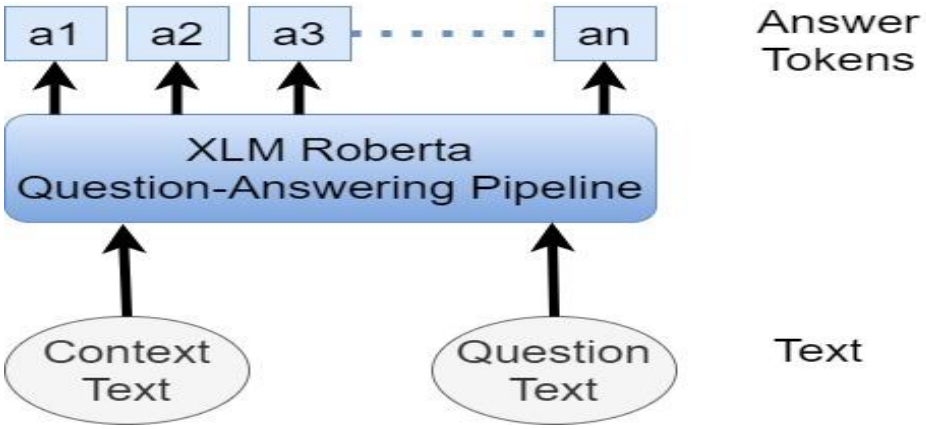

Figure-2: overall architecture of qa pipeline

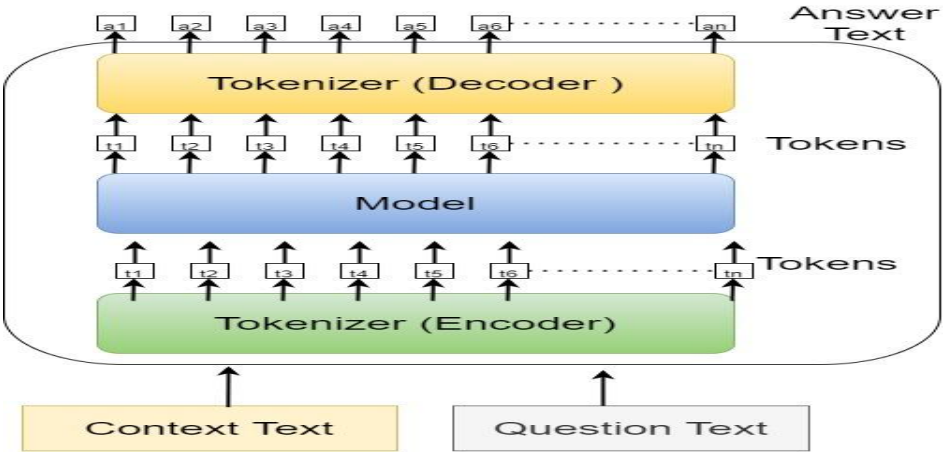

Figure-3: inside the qa pipeline

2.2.2 Approach-2

We added 2 extra linear layers on top of XLMRoberta to output start and end position from the set of tokens.We fine-tuned our model on combined chaii-1 + MLQA dataset. Maximum acceptable sequence length was 512 but in many cases, context length in our dataset was exceeding that limit. Thus, context was split into **n** number of sub-parts, and instead of inputting one **(context, question)** pair, we input **n** such pairs.

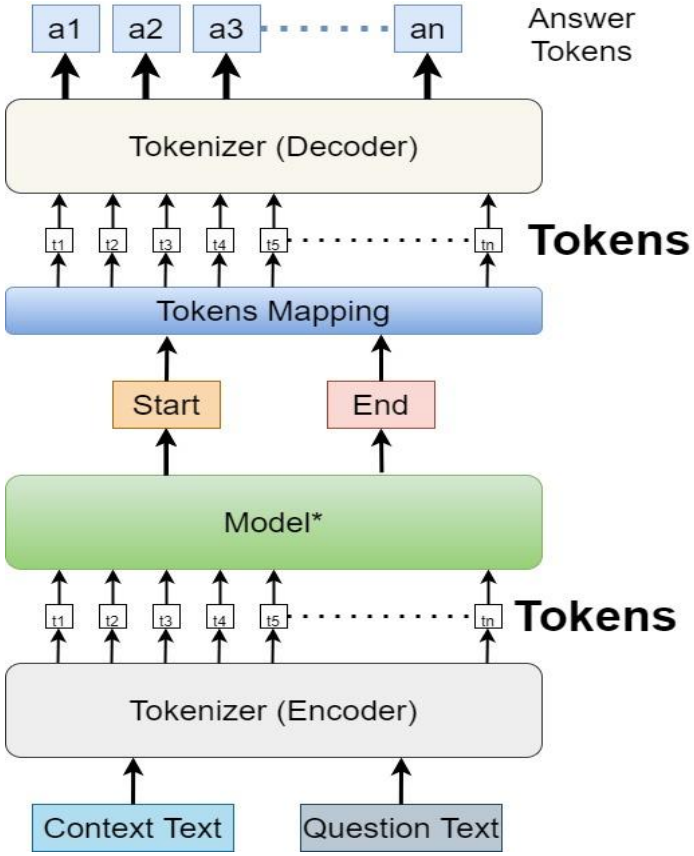

Figure-4: inside the modified QA pipeline

2.2.3 Approach-3

In this approach, each model was trained separately and their outputs were fed to ensemble model which then performed averaging over the given inputs.

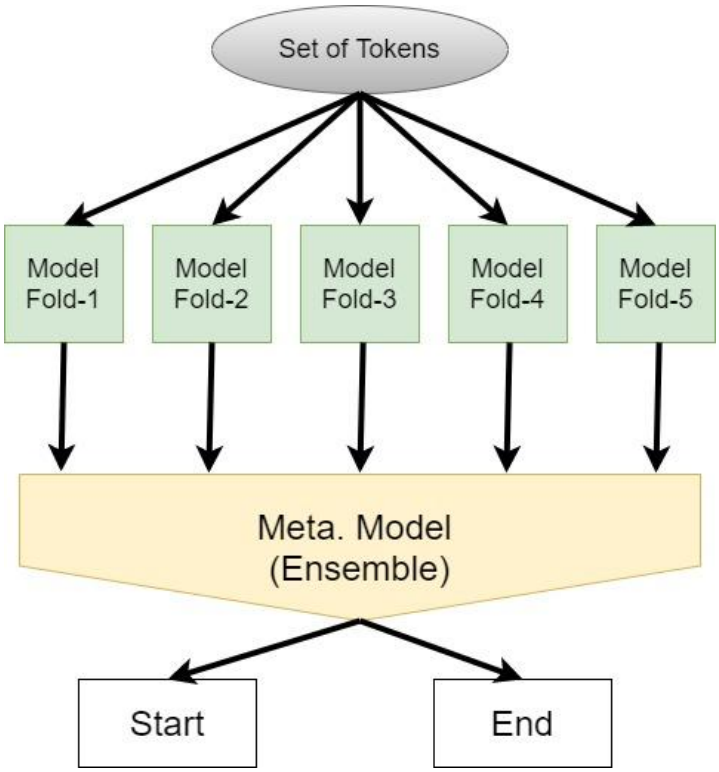

Figure-5: ensemble architecture we used

3 RESULTS

The approaches discussed in Section 2 were implemented and the corresponding jaccard scores were calculated based on the evaluation on the public and private test datasets. Table 1 shows the results for each submission made on the competition website.

Table 1. Evaluation results for the approaches

Approach	**Evaluation Score**
Approach-1	0.571
Approach-2	0.732
Approach-3	0.753

4 CONCLUSION

With nearly 1.4 billion people, India is the second-most populated country in the world. Yet Indian languages, like Hindi and Tamil, are underrepresented on the web. Popular Natural Language Understanding (NLU) models perform worse with Indian languages compared to English, the effects of which lead to subpar experiences in downstream web applications for Indian users. This competition was organized by **Google India** and hosted on **Kaggle.**

REFERENCES

1) Ai4Bharat, IndicBert: https://bit.ly/311BO9y
2) Alvira Swalin , Build Q&A model , https://bit.ly/3CUf5JY
3) Zacchaeus,Clean Chai Dataset,https://bit.ly/3cRMVoo
4) **Google India**,Competition Link,https://bit.ly/30W4QYQ
5) PerceptiLabs,https://bit.ly/3cVkxl2
6) Frank Odom,Transformers from Scratch,https://bit.ly/3DX3dbx
7) Branden Chan,Multilingual XLM-RoBERTa large for QA on various languages,https://bit.ly/3r6D6eC
8) Branden chan,XLM-RoBERTa,https://bit.ly/3nU0uKn
9) Deepset-AI,https://bit.ly/313wPpo
10) How to Train a New Language Model ,https://bit.ly/3cQjhA0
11) Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. arXiv preprint arXiv:1911.02116 (2020).
12) Alexis Conneau and Guillaume Lample. 2019. Cross-lingual Language Model Pretraining. Advances in Neural Information Processing Systems (2019)

13) Simple-transformers (for fine-tuning any model),https://bit.ly/3CZq18Z
14) Jay Alammar,The illustrated Transformer,https://bit.ly/310HYXN
15) Kaggle save and load pretrained model,https://bit.ly/3HUrlZg

5 Individual Contribution

Individual contributions of each team member for the tasks are shown in Table 2.

Table 2. Individual Contributions

Tasks	TR	AD	DP	DM	KV
Finding task/problem statement from Kaggle a. Quora duplicate questions b. Nykaa app review sentiment **c. Chaii: Hindi and Tamil Question Answering by Google**	✓	✓	✓	✓	✓
Understanding Question-Answering task	✓	✓	✓	✓	✓
Understanding Basic Transformer				✓	
Understanding HuggingFace library ● Inference ● Fine-tuning	✓				
Finding pretrained transformers for the task	✓	✓	✓	✓	✓
Checking IndicBert by AI4Bharat					✓
Checking Multilingual XLM Roberta-base		✓	✓		
Checking Multilingual XLM Roberta-large			✓		
Exploring chaii dataset		✓	✓	✓	
Inferencing with IndicBert	✓				✓
Approach 1: Inferencing with XLM Roberta-large (Ideation)				✓	✓
Approach 1: Inferencing with XLM Roberta-large (Coding)	✓		✓		
Approach 1: Inferencing with XLM Roberta-large (Debugging)	✓	✓		✓	
Approach 1: Inferencing with XLM Roberta-large (Kaggle technical glitches + submission)				✓	✓

Approach 1: Inferencing with XLM Roberta-large (Updating PPT)	✓				✓
There was an issue in documentation of hugging face for XLM roberta. We raised an issue on github and it took 2 days to be resolved.	✓		✓	✓	
Creation of character to token mapping function	✓	✓	✓	✓	✓
Approach 2: Single fold Fine-tuning on XLM Roberta-large (Ideation)		✓			✓
Approach 2: Single fold Fine-tuning on XLM Roberta-large (Coding)	✓		✓		
Approach 2: Single fold Fine-tuning on XLM Roberta-large (Debugging)				✓	✓
Approach 2: Single fold Fine-tuning on XLM Roberta-large (Kaggle Submission)	✓	✓	✓		
Approach 2: Single fold Fine-tuning on XLM Roberta-large (Updating PPT)		✓		✓	
Approach 3: 5-fold Ensemble Fine-tuning on XLM Roberta-large (Ideation)		✓			
Approach 3: 5-fold Ensemble Fine-tuning on XLM Roberta-large (Coding)				✓	✓
Approach 3: 5-fold Ensemble Fine-tuning on XLM Roberta-large (Training)	✓	✓	✓	✓	✓
Approach 3: 5-fold Ensemble Fine-tuning on XLM Roberta-large (Debugging)	✓	✓			✓
Approach 3: 5-fold Ensemble Fine-tuning on XLM Roberta-large (Kaggle Submission)			✓		
Approach 3: 5-fold Ensemble Fine-tuning on XLM Roberta-large (Updating PPT)			✓		✓
Architecture Diagrams (Ideation - Rough Diagrams on paper)	✓			✓	
Architecture Diagrams (creation)		✓			
Preparing Interim PPT to be submitted	✓	✓	✓	✓	✓
Preparing interim report	✓	✓	✓	✓	✓

* TR - Tarang Ranpara, AD - Akash Dhedhi, DP - Darshil Patel, DM - Dhyanil Mehta, KV - Kishan Vaishnani