Chaii – Hindi and Tamil Question answering

Harsh Panday, Vritangi Kansal, Nitish Readdy

Abstract

Machine reading context and question answering is an indispensable task in Natural Language Processing. Recently Google hosted a competition on Kaggle in which the task was to develop a model which can understand and comprehend the context in Hindi and Tamil language give answers to asked questions in the respected language.

There are many models for context-based question-answering in English, but research on context-based question answering hasn’t evolved that much, that’s the reason why we choose this competition as our AI project as we thought that this is a great initiative towards developing a natural language model for languages other than English. As we all are citizens of India and are well versed in Hindi and Tamil this is a great opportunity to work on a model for our native languages. There are almost 1 billion people in the world who speak Hindi and around 100 million who converse in Tamil, and yet these languages are underrepresented on the web and there are no reviews that cover Hindi or Tamil question answering systems or even tools, resources so far. With this model, we hope to advance the research in multilingual NLP models and also help Indian users to make the most out of the web.

1. Introduction:

Natural Language Processing is a branch of artificial intelligence that is concerned with giving machines the ability to understand human language and dialects. NLP combines linguistic rule-based modeling with deep learning in order to understand human language. NLP is mainly used in translation applications, speech-to-text applications, etc.

Question answering is a discipline within NLP concerned with developing models which can give relevant answers to questions posed by humans in their own language. The main task of a question answering model is to convert the natural language into machine language, find the most relevant answer to the asked questions and convert the answer back to human-readable language.

There are many types of question answering models, for this project we will be focusing on context-based question answers.

The main drawback in the current NLP models for question answering is that most of them are trained on a dataset which are in the English language and the sentence construction for Hindi and Tamil is very different from English. Let’s take an example in English - “*I went outside to play*” is similar to Hindi- “*मैं बहार खेलने गया*” both of these sentences mean the same but the literal translation of the Hindi sentence word to word in English will be “*I outside play went*”. This is one of the main reasons why we need a separate NLP model for Hindi and Tamil that is trained on a dataset that is in Hindi and Tamil and this is also one of the reasons why we choose BERT for our project instead of models like Open AI GPT as BERT is a bidirectional model and it simultaneously trains from left to right and right to left in order to understand the full context of the sentence but models like Open AI GPT are unidirectional models which train from left to right, this might be good for a language like English but for languages like Hindi and Tamil a bidirectional model is better suited.

2. About the dataset:

Our dataset has 6 different columns for question id, context, question, answer, answer start (position in the context from where the answer starts), and language. The training dataset contains data of both Tamil and Hindi languages. About 67 percent of the data is in Hindi and 33 percent is in Tamil. Apart from this Google has also provided a testing dataset; it only has 4 columns that are id, context, question, and language.

3. Literature Survey

In [1] Sorokin D et al. performed an entity linking prior to forming a SPARQL query. Specific characters were used instead of words as input. The proposed system worked well only for unidirectional languages but had major performance issues. In [2] the author created a two-stage network for question answering. The first stage dealt with the extraction of relevant span (evidence) to the question from the document. The second stage of the network is responsible for synthesizing the answer from the extracted sentences. The first stage of the network is a multi-task model focused on (1) evidence extraction and (2) passage ranking. The authors choose a passage ranking task for better evidence prediction. The synthesized model is a seq2seq learning framework to generate the answer by using the extracted evidence as an additional feature to the model. In [3] Seo M et al. developed a Bi-Directional Attention Flow model for reading comprehension and understanding the context better. BiDAF consists of a hierarchical architecture to encode the context representation at different levels of granularity. It encodes the words in question and context by three different levels of embeddings: character, word, and contextual. The selling point of the architecture is the use of bi-directional attention flow from a query (question) to paragraph and vice-versa, which provides complementary information to each other. With the help of bi-directional attention, they compute the query-aware context (paragraph) representation. The attention operation is performed at each time step and to obtain an attended vector. The obtained attended vector and representations from the previous layers are passed to the next layer in the architecture. In [4] Liu Y et al. worked on a Match-LSTM model which proposes a neural-based solution for machine comprehension tasks. The model provides two different ways to obtain the answer: sequence and boundary. In the sequence model, the proposed architecture predicts the sequence of answer tokens. In the boundary model, only predicts the start and end indices of the answer in the original passage. The words present between the start and end indices are considered to be the answer sequence. The boundary model performed better compared to the sequence model. In [5] Yufeng Diao et al. aimed to create a multi-dimensional question answering network for sarcasm detection, they used bi-directional LSTM along with an attention mechanism in order to achieve their goal. They concluded that even though their model outperformed most of the competitive baseline models there is still room for improvement in the overall performance of their deep learning approach. Yuan-ping et al. in [6] developed an attention based encoder-decoder model using bi-directional LSTM in order to bridge the lexical gap between questions and answers. They also used a step attention mechanism which allowed their model to focus on a certain part of the answer based on the question. In [7] Devendra Singh Sachan et al. propose an end-to-end differentiable training method for retrieval augmented open-domain question answering system that can combine information from multiple resources and retrieved documents when looking for answers. Qin Chen et al. in [8] proposes a positional attention based RNN model which incorporates the positional context of words that occurred in question into the answers' attentive representation. Even though their model didn’t perform as well as some of the benchmark models, their novel approach gave us some ideas regarding how to approach our problem statement. In [9] the authors try to improve the current question answering models by developing an unsupervised model that will decompose complex questions into simpler smaller questions that current QA models can answer easily. Their model worked well with yes/no questions but it didn't perform as consistently with the "wh-"starting question.

Most of these existing studies are in resource-rich languages like English, which is difficult to port into the other relatively low-resource language (Hindi/Tamil). Most of these works made use of machine translation, where questions and/or documents in less-resourced languages were translated to resource-rich language(s) like English. The motivation has been to utilize the resources and tools available in resource-rich languages.

4. BERT:

NLP or Natural Language Processing is a substantially growing field. Originally, simple Recurrent Neural Networks (RNN) was used for training text data and contexts. But in recent years, there have been many research publications that provide exemplary results. One of such is BERT, but before diving into BERT's architecture we need to understand Transformers.

4.1 Transformers:

Google launched the transformer architecture in the well-known paper "Attention is all you need". Transformers basically use a self-attention mechanism, which is suitable for language understanding and such tasks.

We can look at an example to understand the need of attention. There's a statement - "I went to Horsley hills past summer, and it was pretty well flourished in the view of last time I was there". The last word - "there" refers to the "Horsley hills". But to really comprehend this, recollecting the first few parts are essential. To attain this, the attention system decides at each step of an input sequence which other parts of the sequence are important.

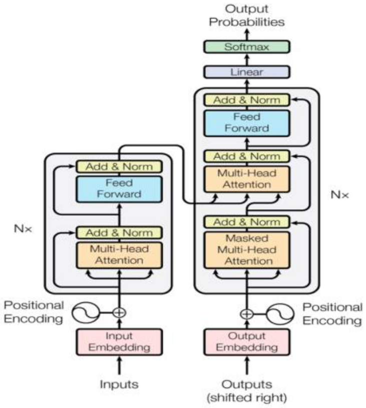


Figure 1 The Transformer model architecture

As we can see in the above model architecture the transformer has an encoder-decoder engineering. They are made up of modules that contain feed-forward and attention layers.

4.2 BERT's architecture:

BERT is a multi-layered encoder. In the initial publication, two models were introduced, BERT base and BERT large. The BERT large even doubled the layers compared to the foundation model. Layers were used to indicate transformer blocks. BERT-base is trained on 4 cloud-based. BERT-large is trained on 16 TPUs.

- BERT base has a total of 12 layers and 12 attention heads
- BERT Large has a total of 24 layers and 16 attention heads.

BERT employs the power of Transformer, an attention apparatus that grasps contextual relations between words (or sub-words) in a text. In its rustic form, Transformer incorporates two separate mechanisms — an encoder that reads the text input and a decoder that generates a prediction for the task. Since BERT's aim is to create a language model, only the encoder mechanism is requisite.

Opposite to directional models, which read the text input consecutively (left-to-right or right-to-left), the Transformer encoder reads consecutive words at once. Hence, it is appraised bidirectional, though it would be more precise to say that it’s non-directional. This attribute allows the model to learn the setting of a word based on its entire environment (left and right of the word).

Apart from generated layers, similar architectures are used in both pre-training and fine-tuning. The exact similar pre-trained model parameters are used to initialize models for, unlike downstream tasks. In the course of fine-tuning, all parameters are fine-tuned. [CLS] is a particular symbol added in front of each input example, and [SEP] is a noteworthy separator token (e.g. separating questions/answers).

4.3 BERT Pre-training:

BERT pre-trains based on two strategies that are

4.3.1 Masked LM (MLM)

Before inputting word sequences into BERT, 15% of the words in each sequence are changed with a [MASK] token. The model then aims to predict the original value of the masked words, conditioning on the context provided by the other, non-masked, words in the sentence. In ML terms, the prediction of the output words requires:

- Appending a classification layer on top of the encoder output.
- Multiplying the output vectors by the embedding matrix, modifying them into the vocabulary dimension.
- Computing the probability of a piece word in the vocabulary with softmax.

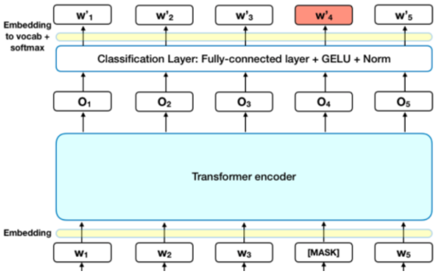

w'₁
w'₂
w'₃
w'₄
w'₅
Embedding to vocab + softmax
Classification Layer: Fully-connected layer + GELU + Norm
O₁
O₂
O₃
O₄
O₅
Transformer encoder
Embedding
w₁
w₂
w₃
[MASK]
w₅

Figure 2 BERT transformer encoder

4.3.2 Next Sentence Prediction (NSP)

In the BERT training procedure, the model accepts pairs of sentences as input and masters to predict if the second sentence in the pair is the succeeding sentence in the original document. At the time of training, 50% of the inputs are a pair in which the second sentence is the succeeding sentence in the original document, whereas in the other 50% a random sentence from the collection is chosen as the second sentence. The conjecture is that the random sentence will be detached from the first sentence.

To assist the model to differentiate between the two sentences in training, the input is handled in the following way before entering the model:

- A [CLS] token is placed at the start of the first sentence and a [SEP] token is pushed at the end of each sentence.
- A sentence embedding referring to Sentence A or Sentence B is connected to each token. Sentence embeddings are alike in concept to token embeddings with a vocabulary of 2.
- A positional embedding is pushed to each token to indicate its location in the sequence. The concept and design of positional embedding can be read in the Transformer paper.

To predict if the second sentence is indeed attached to the first, the below steps are performed:

- The whole input sequence will enter the Transformer model.
- The output of the [CLS] token is altered into a 2×1 shaped vector, using a straightforward classification layer (learned matrices of weights and biases).
- Computing the probability of IsNextSequence with softmax.

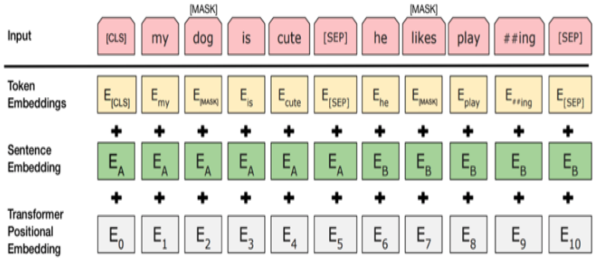

Input
[MASK]
[MASK]
[CLS]
my
dog
is
cute
[SEP]
he
likes
play
##ing
[SEP]
Token Embeddings
$E_{[CLS]}$ E_{my} $E_{[MASK]}$ E_{is} E_{cute} $E_{[SEP]}$ E_{he} $E_{[MASK]}$ E_{play} $E_{\#\#ing}$ $E_{[SEP]}$
Sentence Embedding
E_A E_A E_A E_A E_A E_A E_B E_B E_B E_B E_B
Transformer Positional Embedding
E_0 E_1 E_2 E_3 E_4 E_5 E_6 E_7 E_8 E_9 E_{10}

Figure 4 Embeddings for BERT

5. Why BERT is best suited for our project

The main reason why we choose BERT for our project instead of models like Open AI GPT as BERT uses a bidirectional transformer that means it simultaneously trains from left to right and right to left in order to understand the full context of the sentence but models like Open AI GPT use a unidirectional transformer which trains from left to right, this might be good for a language like English but for languages like Hindi and Tamil a bidirectional model is better suited.

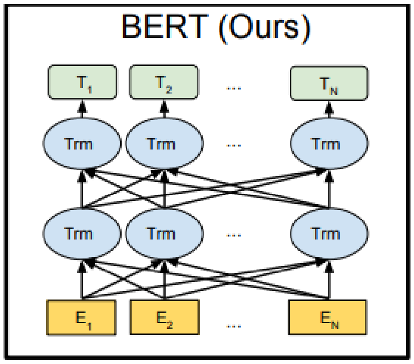


Figure 3 BERT [10]

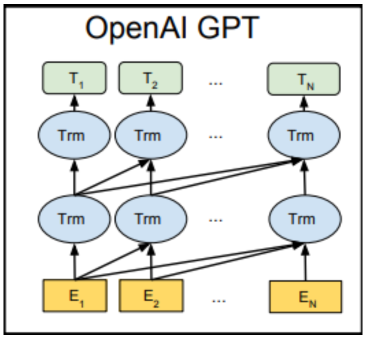


Figure 5 OpenAI GPT model [10]

6. Our methodology:

6.1 Data Preparation:

Before giving the data to BERT we need to make certain changes in the data. The first step is to use the pre-trained tokenizer and create embeddings for our input data that is for context and the questions. The next step is to mark the answer in the context so that BERT can learn while training; this is where we used the "answer start" column in our training dataset. As we were given the answer and the answer start position using them we calculated the position where the answer ended and then marked the answer as 1 in the context and the rest of the context was marked 0. We do this because we want our model to give answers based on the context and not just learn the answers to different questions, marking the answers in the context and giving the answer start and answer end token to our model will make our help our model to find answers in the context rather than just learning answers. Then we store the embeddings, the masks, and the answer start and end token in a dictionary so that we can feed it to BERT to train.

6.2 Building Model:

For our model, we decided to use Sparse Categorical Entropy as our loss function. The reason behind that is traditional categorical entropy requires data in one-hot encoded form but as we prepared the data for BERT it is not in one-hot encoded form, thus we used Sparse Categorical Entropy as our loss function.

$$CE = -\sum_{i}^{C} t_i log(f(s)_i)$$

Figure 6 Sparse Categorical

Now after taking input our model gives start logits and end logits that is our model telling us the probability of a token being part of the answer, we then apply softmax activation on the start and end logits to get the probability of each token and select the tokens with the highest start and end probability as our answer.

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

Figure 7 Softmax activation

Our model takes in the context and masters to predict the answer of the question provided from the context; it then compares the answer it predicted with the actual answer given in the training dataset.

7. Results:

We divided our training dataset into 80/20 and combined the 20% of the training dataset with the small testing dataset given by Google in order to create our final testing data, after running 15 epochs our chaii-Bert model gave the following results

	Loss	Accuracy
Train	0.6738	85.01%
Test	1.3141	81.56%

Table 1 Loss and Accuracy

As we can see from the below graphs our loss was consistently dropping while our model's accuracy was consistently increasing, as BERT is a big model with over 177 million parameters it only took 15 epochs to get decent accuracy.

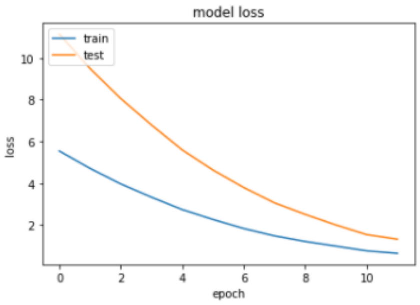


Figure 8 Model loss

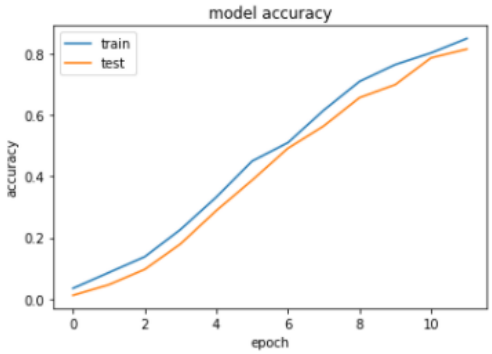
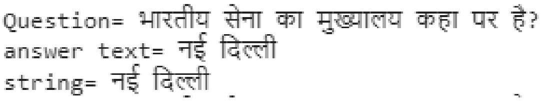


Figure 9 Model Accuracy

Let's analyze some great results given by our model. In the example below, the question is "*Where is India's Army base situated?*" and the answer is "*New Delhi*", as we can see in the context there is a line given which answers the question "New Delhi situated Army base", this is where the bidirectional transformer of BERT is very useful, as it can understand the context from left to right and right to left simultaneously allowing it to understand that *"Army base is situated in New Delhi"*.

ID	**Context**	**Question**	**Answer**	**Answer start**	**Language**
612852f39	…नई दिल्ली में स्थित सेना मुख्यालय से सीधे जुड़ा हुआ है….	भारतीय सेना का मुख्यालय कहा पर है?	नई दिल्ली	6371	hindi



```
Question= भारतीय सेना का मुख्यालय कहा पर है?
answer text= नई दिल्ली
string= नई दिल्ली
```

Figure 9 BERT predicting answer during training

Google provided a small test dataset containing some Hindi and Tamil context and questions, our model predicted very accurate answers for the questions asked in Hindi and fairly accurate answers for questions asked in Tamil. The result of the test dataset is given in below.

ID	Question	Start index	End Index	Answer (predicted)	Answer (actual)
22bff3dec	ज्वाला गुट्टा की माँ का नाम क्या है,	68.0	78.0	की माँ का नाम येलन गुट	येलन गुट
282758170	गूगल मैप्स कब लॉन्च किया गया था?	8.0	11.0	8 फरवरी 2005	8 फरवरी 2005
d60987e0e	गुस्ताव किरचॉफ का जन्म कब हुआ था?,	18.0	20.0	१८२४	१८२४
f99c770dc	அலுமினியத்தின் அணு எண் என்ன?	1.0	331.0	மாய் பயன்படுத்தி வந்துள்ளனர்.13 அலுமினியத்தின் ம	13
40dec1964	இந்தியாவில் பசுமை புரட்சியின் தந்தை என்று கருதப்படுபவர் யார்?	46.0	222.0	ந்திக்கப்பட்டு, 1844ஆம் ஆண்டில் அவரது நண்பர்கள திரு.சுவாமிநாதன்	திரு.சுவாமிநாதன்

8. Conclusion and Future Work:

Context-based question and answering models are a very important part of Natural Language processing. Our model works very well with Hindi and Tamil languages, but there is still scope for improvement for example currently our model gives an offset within the context, we can fine-tune our model, even more, to give just the exact answer to the question instead of giving a line or a slice of the relevant section of the context. We can improve our model by training it on a bigger dataset with different dialects of Hindi and Tamil, as India is a big country every region has its own dialect and since Hindi and Tamil are under-represented on the web hence it is difficult to find a good dataset which contains different dialects of these languages but if such a dataset is created we can train several NLP models on that dataset and create better models for languages like Hindi and Tamil.

9. Contributions:

All of the group members researched BERT and other models together then divided the work in the following way:

Harsh Panday - Harsh wrote the Data preparation section of the code and fine-tuned BERT according to our project's need and wrote parts of the report.

Vritangi Kansal – Vritangi wrote the code for building the training and testing model for our project and worked on the presentation.

Nitish Readdy – Nitish worked on the report and prepared the presentation as well.

10. References:

[1] Sorokin D, Gurevych I. End-to-end representation learning for question answering with weak supervision. InSemantic Web

Evaluation Challenge 2017 May 28 (pp. 70-83). Springer, Cham.

[2] Zhou Q, Yang N, Wei F, Tan C, Bao H, Zhou M. Neural question generation from text:A preliminary study. In National CCF Conference on Natural Language Processing and Chinese Computing 2017 Nov 8 (pp. 662-671). Springer, Cham.

[3] Seo M, Kembhavi A, Farhadi A, Hajishirzi H. Bidirectional attention flow for machine comprehension. arXiv preprint arXiv:1611.01603. 2016 Nov 5

[4] Liu Y, Sun C, Lin L, Wang X. Learning natural language inference using bidirectional LSTM model and inner-attention. arXiv preprint arXiv:1605.09090. 2016 May 30.

[5] Diao, Y., Lin, H., Yang, L., Fan, X., Chu, Y., Xu, K. and Wu, D., 2020. A Multi-Dimension question answering network for Sarcasm Detection. IEEE Access, 8, pp.135152-135161.

[6] Nie, Y.P., Han, Y., Huang, J.M., Jiao, B. and Li, A.P., 2017. Attention-based encoder-decoder model for answer selection in question answering. *Frontiers of Information Technology & Electronic Engineering*, *18*(4), pp.535-544.

[7] Singh, D., Reddy, S., Hamilton, W., Dyer, C. and Yogatama, D., 2021. End-to-End Training of Multi-Document Reader and Retriever for Open-Domain Question Answering. Advances in Neural Information Processing Systems, 34.

[8] Chen, Q., Hu, Q., Huang, J.X., He, L. and An, W., 2017, August. Enhancing recurrent neural networks with positional attention for question answering. In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 993-996).

[9] Perez, E., Lewis, P., Yih, W.T., Cho, K. and Kiela, D., 2020. Unsupervised question decomposition for question answering. arXiv preprint arXiv:2002.09758.

[10] Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.