Fact Extraction And Verification Using Deep Learning

BuddhaTeja
201IT115
Information Technology
NIT Karnataka
Surathkal, India

Aditya Rama Hegde
201IT105
Information Technology
NIT Karnataka
Surathkal, India

Bharath C Kulkarni
201IT213
Information Technology
NIT Karnataka
Surathkal, India

Dhruv Nagesh Pai
201IT118
Information Technology
NIT Karnataka
Surathkal, India

***Abstract*—In this paper, we have designed a model to predict whether a given Claimed claim is true or false using a transformers-based DeBERTa model, which takes the text encodings of the evidences that are extracted and their labels for further fact verification. And in the case of test data which only contains the claim, the evidences are extracted based on the keywords and phrases in the claim and they are ranked based on their context similarity with the claim(as only limited evidences are to be considered).Also This paper highlights the challenges of the fact extraction and verification problem and presents the proposed approach's results.**

***Index Terms*—Fact Extraction and Verification (FEVER), Wikipedia, Claim, Evidences, Machine Learning, Deep Learning,Debarta,context extraction,ranking.**

I. Introduction

In the recent years and in our daily day-to-day life, the amount of wrong and misleading content on the Internet and the web has significantly increased. We are hearing the words hate speeches and fake news these days very frequently. It will change public opinion and modifies society. It also causes defamations and false perceptions about individuals. The idea of fake news has been in existence even before the emergence of the web and the Internet as publishers used false and misleading information to aim towards their interests.But at that time it was not harming the society much as compared to now.

Following the recent craze and popularity of the internet, more and more users also began forsaking the traditional media channels used to spread information for online platforms. These kind of false and misleading information could produce harmful effects on society. The impact of incorrect information can place many wrong beliefs and views in individuals and can also lead to an unsuccessful public, undermining the democratic process. So considering these all as a result, information verification in terms of fact-checking has become very much important as it allows to verify of controversial claims stated on the web and corrects and punishes the people who are doing this kind of fake news spread. However, because of the large number of fake news and misleading articles published on the internet daily, the old and traditional manual fact-checking is no longer feasible and useful. Therefore, for all of these problems, this paper proposes a step by step approach for the fact extraction and verification problem using the FEVER 2018 dataset.

Fig. 1: Fake information Spread

In this paper, we have proposed a step-by-step approach for the fact extraction and verification problem. We have used the Fact Extraction and VERification (FEVER) 2018 dataset. The train dataset contains 185,445 claims, each of which comes with several evidence sets from Wikipedia. An evidence set consists of facts, i.e. sentences from Wikipedia articles that jointly support or contradict the claim. On the basis of its evidence sets, each claim is labeled as Supported, Refuted, or NotEnoughInfo. If the sentences in the evidence sets to support the given claim, then it is labeled as supported. If the sentences in the evidence sets oppose the claim then it is labeled as refuted. Claim is labeled as NotEnoughInfo if no decision about the veracity of the claim can be made or the evidences are not enough.

Test dataset consists of claims without any evidence or labels. The task is difficult in two aspects. First, accurate selection of potential evidence from a huge knowledge base i.e wikipedia, with respect to an arbitrary claim requires a thoughtful system design and results in a trade-off between computational resources and computational resources. Moreover, even with ground truth evidence, the verification sub-task of predicting the relation between evidence and the

claim is still a long-existing open problem.

There are two major steps that should be done by our program to handle it. Firstly, the extraction of evidence from the given claim (Fact Extraction), then verify the claim with the obtained evidence and tell whether the evidence supports or refutes the claim. In the train data, we have been given Wikipedia page id for each evidence. We have extracted the summary of those Wikipedia pages. Then for fact extraction step, we are given claims. For a given claim we extract keywords from the claim and get the summary of Wikipedia pages as evidences using Wikipedia API. Using context similarity, we will consider only those 3 evidences for whom the context similarity with the claim was maximum. We trained the training dataset using deberta which is a Transformer-based neural language model that aims to improve the BERT and RoBERTa models with two techniques: a disentangled attention mechanism and an enhanced mask decoder.

II. Problem Statement

To design and implement a system to take input of a claim from the user and give the respective supports or refutes for the given claim using that evidences for the given claim.

A. Objectives

- Taking the input of a claim from the user.
- Extracting only respective useful evidences for the given claim using Wikipedia API based on the key words and key phrases in the claim which have high contextual similarity with the claim.
- Predicting the Label(truthness) of the claim

III. Dataset

The total number of samples that are present in the dataset(training dataset) are nearly 146K and the main attributes it contains are Verifiability(if the given claim is verifiable by the given evidences), Claim(The statement whose truthness is to be checked), Evidence IDs(The page ids of the Wikipedia which are related to the claim), Label(The truthness of the claim) in which nearly 35K are non-verifiable samples.

It is observed that there are an average of 2.6 evidences per claim and each evidence has on average of nearly 250 words(when extracted). We also modified the dataset for data cleaning for the evidences and claim. We combined the claims and evidences before sending to the model.

IV. Literature Survey

UCL Machine Reading Group: Four-Factor Framework For Fact Finding (HexaF) :-
This paper basically gave a four-stage model consisting of document retrieval, sentence retrieval, natural language inference, and aggregation. In this, they did Document retrieval attempts to find the name of a Wikipedia article in the claim and then rank each article based on capitalization, sentence position, and token match features. A set of sentences are then retrieved from the top-ranked articles, based on token matches with the claim and position in the article. The NLI model is subsequently applied to each of the retrieved sentences paired with the claim, giving a prediction for each potential evidence sentence. They also used the respective predictions to make it aggregated using a Multi-Layer Perceptron (MLP), and the sentences are finally reranked so that the evidence which is consistent with the final prediction are placed at the top.

V. Methodology

For the train data :

For every sample, there is a verifiable status which is either verifiable or non-verifiable, and all the verifiable samples contain evidence ids which are ids for the data that is required for the claim verification(whether the claim that is being claimed is true or false(2 labels)) through the wikipedia data source. It is observed that all the Non-verifiable samples doesn't contain enough information about the topic, i.e; there are no evidence ids for the same. Evidence ids are the IDs of the Wikipedia pages that can be accessed using a wikipedia API.

The original data that is considered is from the workshop conducted by fever.ai which provided the raw data(train.jsonl) and that is refined(like all the outliers are eliminated) to obtain the dataset.csv (the main dataset for the model)

Fact Extraction:

Evidence id for a claim is the wikipedia page id of wikipedia page which contains the information related to the claim. It is observed that there are an average of 250 words per summary of the Evidence id, for each evidence id of the claim(before the data cleaning-like removing the stop words, punctuations and applying stemming and lemmatization techniques)

After obtaining the evidence from each evidence id of the claim, they are concatenated to form a single large evidence and then this evidence is combined with the claim(before the text cleaning) and then this cleaned Claim Evidences are encoded accordingly(like the sentence is split into words(relevant words) in a way that they support the aspects like self-attention, easy computation) with the help of transformers(which also gives the corresponding attention masks for the samples accordingly) and along with them, their labels are also given as an input to the DeBERTa(Decoding-enhanced BERT(Bi-directional Encoder Representation for Transformers) with Disentangled Attention) Model

Fact Verification :

DeBERTa model is an extension of the BERT model which has gained popularity in the NLP domain over the past few years, and DeBERTa model significantly outperformed most of the recent BERT models,

The major advantages of DeBERTa over BERT include that it is training efficient(as DeBERTa model eliminates the unnecessary or redundant data over time which helps in overfitting and improves the efficiency), Using various Embedding techniques(like contextual, positional,...), Dis entangled attention mechanism(which can address the redundant cases of the attention heads), Enhanced-decoding techniques...

For training of the data, the model is trained on the train data, and the data is split in the ratio 80:20 (In here for now, like for the training data the model is trained and tested on train data itself, for the model to get accustomed because for the test data(that is provided) the labels are not mentioned for experimenting purposes).

The model is trained on instances of 2K samples(due to the extremely large data) and saved the model accordingly after every instance for the further instances of the data and is run for the data of 20K samples, for 10 epochs and batch size of 32 (after tuning of the hyperparameters).

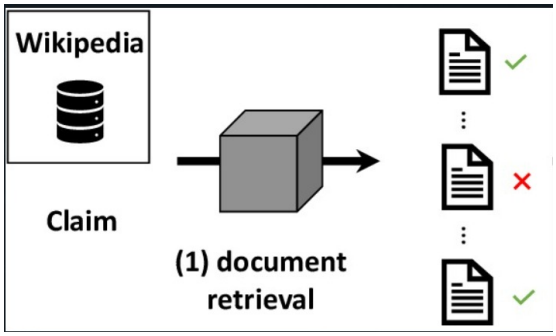

Fig. 2: Retrieving the Evidences

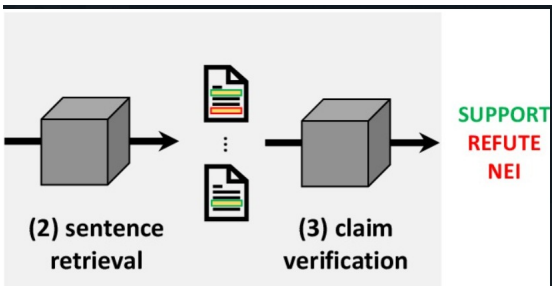

Fig. 3: Verifying the Claim against Evidences

For the test data :

Fact Extraction :

For the test data, there are only claims that are provided, from which evidences for the claimed claim are to be extracted and then those are used for the prediction of the truthness of the claim.

Thus for a given claim, all the required evidences that will be useful are extracted using the keywords, key phrases, and distinctive names that are in the claims

The Extraction of the evidence is done using these key phrases and words as evidence ids for the wikipedia API and thus are extracted individually, It is observed there are more key words for a claim with large lengths as most of the possible phrases are also considered, after the extraction of the information of the evidence id(if available in the wikipedia) only the top 3(at most) evidences are considered as the train data contained approximately 3 evidences per a claim

Thus all the evidences are extracted and are arranged according to their ranks by;

For the extracted evidences, the context of the evidence is extracted by summarizing the data of the evidences and then, the similarity of the context of the evidence is compared with that of the claim using cosine similarity and then based on the similarity they are ranked accordingly and the top three evidences are considered

Fact Verification:

For fact verification the three evidences are checked with the claim using the same model (DeBERTa trained with the train data) separately, i.e; the evidences are checked separately and it is given how likely a claim is true or false based on that particular source from which the evidence is collected

Thus provides three(at most) evidence and their relation with the claim

VI. Results and Conclusion

All the wikipedia pages are successfully extracted from their corresponding wikipedia page ids(evidence_ids) and then data cleaning is done on the obtained evidences which helps in reducing the unrelated and unrequired entities(which can also increase the efficiency), this data is passed onto the model after encoding and then the model is trained on for varying parameters to choose the best parameters

.

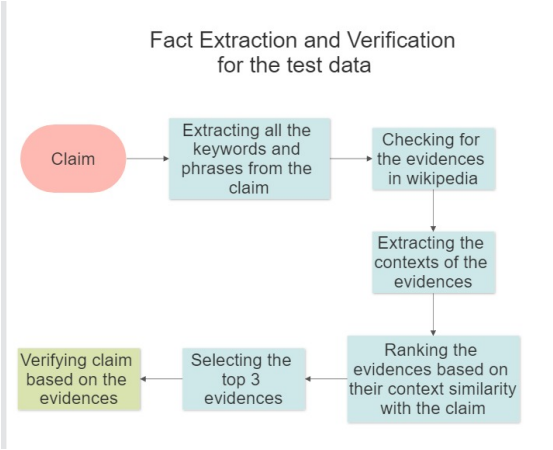

Fig. 4: For the Test Data)

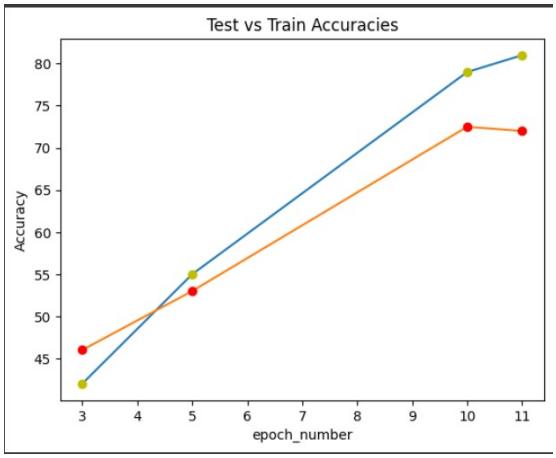

Fig. 5: Epoch Vs Accuracy

TABLE I: Epoch vs Accuracy

Epoch Number	Accuracy	
	Train	Test
3	42	46
5	55	53
10	79	72.5
11	81	72

The average accuracy of the model over the 20K samples of the data is 72.3%

For the test data when a Claim like 'The National Bank of Canada is the sixth largest theme park in Canada.' is given

Then the keywords that are extracted are 'sixth largest theme park', 'national bank', 'canada' and then the rank of the evidence ids are ;

['sixth largest theme park', 0.8153714699303967], ['national bank', 0.682946160738234], ['canada', 0.7959096802404682] where the lower rank indicates that the evidence is more related to the claim

Thus the facts related to the claims are successfully extracted via an API and then are efficiently verified with the claim to know the truthness of the claim that is claimed

VII. References

[1] Tokenization: https://neptune.ai/blog/tokenization-in-nlp

[2] FEVER(Fact Extraction and Verification) : fever.ai

[3] pytorch : https://pytorch.org/docs/stable/nn.html

[4] tensorflow : https://www.tensorflow.org/

[5] Keras : https://keras.io/

[6] DeBERTa : https://huggingface.co/docs/transformers/model_doc/debe

[7] BERT :https://huggingface.co/docs/transformers/model_doc/bert

[8] transforms :https://huggingface.co/docs/transformers/index
[9]Jupyter Notebook : https://docs.jupyter.org/

[10] Neural Networks : https://www.ibm.com/in-en/topics/neural-networks

[11] Dataset : https://fever.ai/dataset/fever.html