NLP Final Project: Claim Detection

Siddhant Agarwal and **Saksham Bhupal** and **Shantanu Dixit** and **Aditya Nangia**
CSE556 : GROUP 1
IIIT-Delhi, India
{siddhant20247, saksham20573, shantanu20118, aditya20168}@iiitd.ac.in

Abstract

A claim is an assertive statement that may or may not be proven. In today's world, most people express their opinions and thoughts on online social media sites. This results in the generation of large amounts of data on both claims and opinions. The task at hand is to detect whether a given tweet is a claim or not. This is a binary classification task. Claim Detection is the first step to deal with the current global infodemic, which can help with further downstream resource-intensive tasks such as check-worthiness of a claim and, finally, true or false claim classification. We tried various baseline models to compare our proposed architecture against, such as - Support Vector Classifier, Logistic Regression, AdaBoost Classifier and MLP CLassifier. Our Proposed Model finetunes the BERT transformer on the Twitter dataset for Claim Detection. It achieved the best results among all our experiments. The model may be helpful in claim detection tasks and in helping fact-checkers fight against misinformation on online social media platforms.

1 Introduction

1.1 Problem Definition

A claim is defined by Toulmin (2003) as "an assertion that deserves our attention". Through this task of claim detection, we aim to provide claims on online social media platforms the 'attention' they 'deserve'. Online social media platforms are filled with users generating vast amounts of data in the expression of their claims as well as opinions. This task aims to separate out "claims" from this online content that may be a precursor to essential tasks such as fake news detection and fact-checking. Whether the fact stated by the individual/group to express themselves is a claim regarding a particular topic or not, furthermore detection of a claim can be more problematic when it consists of a concept which may have legitimate merit in itself, but with no measurable outcomes to support the claim being made regarding this topic.

Formally, the task at hand may be described as - Given an input tweet, the task is to detect whether the tweet is a claim or not. A claim is an assertive statement that may or may not be proven. This is a binary classification task.

1.2 Motivation

Nowadays, most people express themselves on online social media sites, with Twitter leading the way. Users express themselves in the form of tweets on various topics ranging from personal to political. These tweets can be assertive. Individuals tend to "claim" something that conforms to their interests and biases. Claims could be knowingly or unknowingly targeting an individual or a group of people, thus possibly creating social unrest. Thus the current situation demands an automated method to detect such tweets and take appropriate actions if required to maintain peace and order in the digital sphere. The following task is of social importance; it makes it challenging, pursuable, and worth devoting time to.

2 Related works

Gupta et al. (2021) proposed LESA (Linguistic Encapsulation and Semantic Amalgamation Based Generalised Claim Detection from Online Content), a one of its kind generalised claim detection system capable of predicting "claims" and "not claims" both on structured as well as unstructured data. The model categorises the text as tweets, comments and essays based on noise; each category is modelled separately and joined together using attention layers. In brief, the model captures the linguistic level properties using POS and dependency tree, combines it with BERT (Bidirectional Encoder Representations from Transformers), and combines it with the attention layers with a softmax unit at the end to generate the prediction.

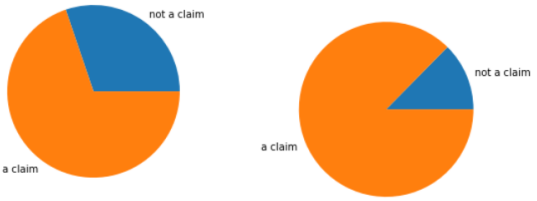

Figure 1: Data distributions before and after data augmentation

Sundriyal et al. (2022b) further the task of claim detection in the broader context of its impact on society by checking the veracity of the claims. They proposed an end-to-end claim verification model for verifying claims made on Twitter against high-quality editorial articles. They also released an API for easy claim verification at the user level.

Prabhakar et al. (2020) present work on the FEVER (Thorne et al., 2018) dataset, which is one of the most extensive datasets for claim detection. They propose a claim detection model by employing transfer learning and fine-tuning BERT and DistilBERT models for claim detection. They achieve an F1-score of 97.99% on the prediction of 50% of the data as a claim and 50% as a non-claim. However, this result is not directly comparable to our task as the dataset in our task is more skewed towards the positive class of claim.

Nikolov et al. (2020) focus on detecting check-worthy tweets related to COVID-19 on the dataset CLEF-2020 CheckThat! This check-worthiness task may depend on a claim detection model to detect claims which may be considered for check-worthiness. The model uses a variety of classifiers like SVM and logistic regression on the generated word embeddings from Sent2Vec, FastText and GloVe. The work incorporates preprocessing techniques like keyword extraction, hashtag unifications, and Corona and SW+C (combines Corona and stopword removal) pre-preprocessing.

Sundriyal et al. (2022a) take the task of claim detection a step further by proposing a framework for 'claim span detection'. They propose to 'identify argument units' of claims in this task. They proposed CURT, a dataset annotated for claim-span detection on Twitter data. This work furthers the overall motivation of the task by enabling fact-checkers to perform fact-checking faster and easier by providing them with spans of text to focus on.

Figure 2: Wordcloud corresponding to claim tweet

Figure 3: Wordcloud corresponding to not a claim tweet

3 Methodology

3.1 Data Pre-processing

1. Data Augmentation: Since the data was complete with no missing values, no label decision for a tweet was required. The data contained usernames as well as hyperlinks which were removed through regular expressions. The data was highly skewed, with the tweets corresponding to "claim" being almost eight times the number of tweets corresponding to the tweets of "not a claim" hence it was necessary to perform data augmentation corresponding to "not a claim" tweets. The library "nlpaug" was used to generate the new tweets; since the tweet generated would contain a similar sentiment (in this case, claim/not claim), the label from the original tweet was used.

2. URL removal - We used a Regular Expression in python for the removal of URLs from the tweets.

3. HTML Tag removal - We used a Regular Expression in python to remove HTML tags from the tweets.

4. Username Removal - We used a Regular Expression in python to remove Usernames from the tweets.

5. New line and Whitespaces removal - We used a Regular Expression in python to remove new line tags and extra whitespaces from the tweets.

3.2 Baseline Models

The following machine learning models were used after generating sentence embeddings using distil-roberta model:

1. SVC: Support Vector Machines are a set of Supervised Machine Learning algorithms. This algorithm aims to create the optimal decision boundary - hyperplane - to separate the n-dimensional space into classes. Since our task is a binary classification task, a support vector classifier is an appropriate model. We ran the sklearn.svm.svc implementation of support vector classifier with C = 1.0, gamma = 'auto' and kernel = 'rbf'.

2. Logistic Regression: Since our task is a binary classification task, hence our dependent variable is categorical and we can apply Logistic regression for our task. Using the sklearn implementation with C = 1.0, penalty = 'l2', solver = 'lbfgs', random_state = 0.

3. MLP classifier: MLPClassifier stands for Multi-layer Perceptron classifier. Unlike other classification algorithms such as Support Vectors Classifier, MLPClassifier relies on an underlying Neural Network to perform the classification task. We use the sklearn implementation of MLPClassifier with one hidden layer of 100 neurons, activation = 'relu', max_iter = 1000 with early_stopping set to True.

4. ADA boost classifier: An AdaBoost classifier is a meta-estimator that begins by fitting a classifier on the original dataset. It then fits additional copies of the classifier to the same dataset but with the weights of instances that were incorrectly classified adjusted, so subsequent classifiers would concentrate more on challenging cases. We train our model using the sklearn's implementation of AdaBoost classifier with base estimator set as a Decision tree classifier initialised with max_depth = 2, n_estimators = 100, learning_Rate = 1.0, algorithm = "SAMME.R", and random_state = 0.

3.3 Proposed Architecure

The model which we propose involves fine-tuning of the BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2018) for claim detection. Embeddings are obtained for each sentence from the "bert-base-uncased" model, and these embeddings are further used to classify the sentence as a claim or non-claim. The proposed architecture comprises three basic steps - a preprocessing step, a BERT encoder and a classification head on the BERT model. We use the Huggingface implementation of the transformer models for our implementation. The data is tokenised using a BERT Tokeniser with hyperparameters - max_length - 150, pad_to_max_length = True and add_special_tokens = True. The tokenized data is then fed to the BERT encoder consisting of 11 BERT layers - with each layer further having an attention, intermediate and output layer - and one pooling layer. The encoded output of the BERT model is used by the classification head - which contains a dropout layer with p set to 0.1 followed by a Linear Classifier layer input features = 768 and output features = 2 - for the classification task of claim/non-claim. The model is fine-tuned on the Twitter Dataset with an Adam Optimiser with a learning rate = 5e-5 and a batch size of 100 for three epochs.

The proposed model architecture is also given in Figure 4

4 Experimental Results

4.1 Basline Results

1. SVC: We obtained a training accuracy of 0.87 and a validation accuracy of 0.88 with a macro f1-score of 0.47 on our own dataset with a train-val split of 80-20%.

2. Logistic Regression: The model gave us a training accuracy of 0.88 and a validation accuracy of 0.89 with a macro f1-score of 0.57 on our dataset with a train-val split of 80-20%.

3. MLP classifier: The model gave us a training accuracy of 0.89 and a validation accuracy of 0.87 with a macro f1-score of 0.55 on our dataset with a train-val split of 80-20%.

4. ADA boost classifier: The model gave us a training accuracy of 0.98 and validation accuracy of 0.85 with a macro f1-score of 0.59 on our dataset with a train-val split of 80-20%.

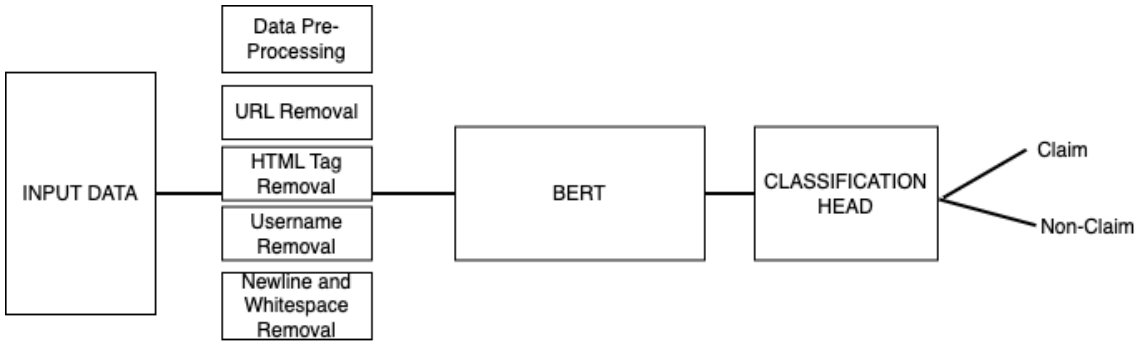

Figure 4: An illustration of the model's architecture, constituting Data Pre-Processing, BERT and a Classification Head.

4.2 Proposed Model Results

The model that was proposed used fine-tuned BERT for the given classification task. After running the data on "best-base-uncased" without any preprocessing for 2 epochs, a macro-F1 score of 0.7131 was achieved on 25 per cent of the test data. After applying the mentioned preprocessing steps, this score was increased to 0.73105.

After applying data augmentation, the F1-score did not improve and rather decreased to 0.68879. Other than data augmentation, we also attempted data resampling to overcome the class imbalance. However, the results weren't improved and stayed around 0.682.

These variations of the model were also compared by splitting the training data using an 80/20 train-val split. The model was trained on 80 per cent of the training data and validated on the remaining 20 per cent. The accuracy scores on the validation set for all the variations mentioned ranged from 88-90 per cent while the F1 scores ranged from 90-92 per cent.

5 Analysis

We ran extensive tests to figure out the effectiveness of various components that we were adding to our models. Our first major insight was about the data imbalance. The data was highly skewed in favour of the positive class (Claim) while the negative class (Non-Claim) was only about 15% of the dataset. From the word clouds, we discovered that both Claim as well as Non-Claim tweets were centred around similar topics relating to the coronavirus pandemic. This also validated the data as a relatively new one as the coronavirus pandemic is a recent phenomenon.

Upon our analysis of the baseline results, we confirmed our hypothesis that simple ML models are ineffective in classifying data based on Natural Language and Large Language Models (LLMs) is important in understanding language at the pragmatic level.

Our experiments with fine-tuning BERT also highlighted how data preprocessing does not play a very important role in deep learning. Our results with and without preprocessing were approximately the same with minor differences in individual runs. Data augmentation and data resampling to artificially reduce the data imbalance problem also did not yield better results highlighting the difficulty of the chosen task. The task overall provided an excellent opportunity to learn and experiment with NLP models such as Transformers.

6 Contributions

The contributions by each member are listed as follows:

- Siddhant Agarwal (2020247) - Tried many variations of modelling such as trying ensembling of multiple models like BERT, RoBERTa and DistilBERT, tried models including dense feed-forward networks on top of BERT. Also participated in group discussions and report writing, formatted the report.
- Saksham Bhupal (2020573) - Ran extensive modelling tasks with models such as BERT, RoBERTa and DistilBERT. Prepared intial model for BERT fine tuning. Preprocessed the data. Participated in group discussions and report writing.
- Shantanu Dixit (2020118) - Performed data preprocessing and data augmentation, made visualisations, performed modelling and contributed to various other tasks. Also participated in group discussions and contributed to report writing. contributed to the report writing

- Aditya Nangia (2020168) - Tested baseline models - SVC, Logistic Regression, AdaBoost Classifier, MLP Classifier - with and without pre-processing. Prepared intial model for BERT encoder and performed data resampling tasks. Also participated in group discussions and contributed to report writing.

Acknowledgements

We would like to thank Professor Md. Shad Akhtar for his constant support during the semester. We implemented the topics discussed in the course with the help of the knowledge gained in the lectures delivered by sir. We would also like to thank all the TAs of the course for their systematic and effective course management.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Shreya Gupta, Parantak Singh, Megha Sundriyal, Md. Shad Akhtar, and Tanmoy Chakraborty. 2021. LESA: Linguistic encapsulation and semantic amalgamation based generalised claim detection from online content. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3178–3188, Online. Association for Computational Linguistics.

Alex Nikolov, Giovanni Da San Martino, Ivan Koychev, and Preslav Nakov. 2020. Team alex at clef checkthat! 2020: Identifying check-worthy tweets with transformer models. *ArXiv*, abs/2009.02931.

Acharya Ashish Prabhakar, Salar Mohtaj, and Sebastian Möller. 2020. Claim extraction from text using transfer learning. In *Proceedings of the 17th International Conference on Natural Language Processing (ICON)*, pages 297–302, Indian Institute of Technology Patna, Patna, India. NLP Association of India (NLPAI).

Megha Sundriyal, Atharva Kulkarni, Vaibhav Pulastya, Md. Shad Akhtar, and Tanmoy Chakraborty. 2022a. Empowering the fact-checkers! automatic identification of claim spans on twitter. *ArXiv*, abs/2210.04710.

Megha Sundriyal, Ganeshan Malhotra, Md Shad Akhtar, Shubhashis Sengupta, Andrew Fano, and Tanmoy Chakraborty. 2022b. Document retrieval and claim verification to mitigate COVID-19 misinformation. In *Proceedings of the Workshop on Combating Online Hostile Posts in Regional Languages during Emergency Situations*, pages 66–74, Dublin, Ireland. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Stephen E. Toulmin. 2003. *The Uses of Argument*, 2 edition. Cambridge University Press.