Stance Detection

CSE 573 - Group 15 - Project 23

Collin Wood*, Eric Waters*, Mahidher Duraisamy Krishnan*, Mukesh Jha*,
Saurabh Balasaheb Mohite* and Viraj Thakkar*

School of Computing and Augmented Intelligence, Arizona State University, Tempe, USA
{cjwood8, eswaters, mdurais1, mjha8, saurabhmohite, viraj.dt}@asu.edu

***Abstract*—Information extraction from texts has been one of the most important research topics of Natural Language Processing over the last several years. Among these topics, Stance Detection has been one of the main research problems and plays a major role in analytical studies measuring public opinion on social media, particularly on political and social issues. Stance detection is a method used to decide the stance of a text with respect to a particular topic. This is a popular technique to build personality profiles for people on social media better than its sub-optimal counterpart - sentiment analysis. In this project, we have tried to tackle this problem by using Decision Trees and BERT, to build generalized models to predict the stance of the query with respect to a specific topic. We have further tried to improve the models and data to enable few-shot learning and zero-shot learning which could help to predict the stance on a topic that was never seen by the models before. Later we also discuss and compare the results obtained from the different techniques applied in this study.**

***Index Terms*—Stance detection, Back Translation, Style Transfer, Transfer Learning, Decision Tree, BERT**

I. Introduction

In recent years, Social Media has quickly become the primary form of interaction for a large proportion of the global community. Platforms like Facebook, Twitter, and Instagram, have become everyday tools for disseminating information and spreading ideas. According to the Pew Research Center, 80% of Americans were using Social Media in 2020, and noted that this trend of rapid adoption was likely to continue [1]. Each platform has their own methods for delivering information to its users, for validating the authenticity of each account, and for acting when incursions on their social space disrupt the ability of users to trust or use their platform. Given the increasing dependency that we humans have on Social Media, it's now possible to conduct research on the large data provided by these platforms' users, for use in building models on human behavior and interaction, and for use in the detection of an individual's opinions, beliefs, or stance on a given topic.

Stance detection is typically described as the automatic assessment of a post owner's position (as for or against) toward a particular target, based on the post's content. Eliciting relevant information from the underlying text is vital when dealing with contentious topics or elections/referendums, as well as a number of related challenges like sentiment analysis, controversy identification, and argument mining.

This stance can help identify whether an opinion is in favor of an idea or opposes it and is known simply as Stance Detection. It is quite similar to sentiment analysis except that it usually investigates the two-sided relationship between a question and an opinion. Position statements frequently include figurative language that is challenging for robots to decipher. Think about the text "We don't inherit the earth from our parents we borrow it from our children and Last time I checked, Al Gore is a politician, not a scientist". Such statements convey an interpretable viewpoint pertinent to the subject of climate change to the human viewer. However, a reader frequently draws on personal experience to infer greater context in order to appreciate rhetorical devices like sarcasm, irony, analogy, and metaphor. Using informal grammar, spelling, censorship, and vocabulary further complicates problems for machines.

Stance detection is becoming increasingly important for analytical purposes and for studying the interactions of individuals on social media platforms [2], more specifically for socio-political purposes. In the last decade alone, Americans have already seen the impact of social media on elections, political discourse, and the influence of political parties and this is only going to spread further to the rest of the world.

This has been noticed and in 2016, the SemEval-2016 stance dataset was created and further used to construct a model for identifying the stances of users about current presidential candidates and controversial subjects at the time from about 4000 tweets of training data [3]. It shows the desire for organizations and researchers to perform sentiment analysis for use in making political choices or predictions.

This paper primarily focuses on two methods to implement Stance Detection - Decision Trees and BERT. However, our contributions go deeper into Zero-Shot Learning (ZSL) and Few-Shot Learning (FSL) with the usage of an additional dataset generated by us, back-translation, and paraphrasing, to aid in improvement of our results. The simple end goal is to generate a stance (favor/against/neither) for any given text and target topic whether the target was in our dataset or not. The ZSL and FSL aspects are important to us in this study because there is a need to verify if the model is able to classify data with unseen topics, or if it needs some examples to fine-tune and then classify.

II. Problem Statement

We will define stance detection as "automatic classification of the stance of the producer of a piece of text, towards a target, into one of these three classes: Favor, Against, Neither" [4], and in the case of Social Media stance detection, we plan to identify the stance over a large variety of targets or 'topics'. Stance Detection generally involves a text input from a Social Media platform like a tweet (sometimes including a hashtag '#' or other platform-unique characters) that makes a claim about some topic. The text is then evaluated over some topic (methods for topic detection allow the model to simultaneously decide on the subject discussed) and an output is provided as either positive, negative, or neither.

Since the 2016 SemEval model, many developments in Stance Detection have been made, including the ability for larger datasets to be fed into more complex models, some of which employ Regression Models, or other classifiers, most recently Deep Neural Networks. We propose and implement ideas for increasing the accuracy of such models, by expanding the scope of the dataset by combining other datasets and building a Neural Network for classification.

The main contributions of this work are as follows:

- Introducing data augmentation and pre-processing techniques on the dataset. Some of these techniques include Back Translation, Paraphrasing, Miscellaneous Changes, adding Neutral stances, and also generating new data using GPT-3
- Testing the performance of traditional machine learning models such as Decision Tree on the Semeval dataset
- Training the Bert model using the Semeval Dataset. The model generates Bert embeddings of the query with respect to the target, and these embeddings are pooled and used as a classifier
- Analyzing the ZSL and FSL feature of the Bert model on Procon and AI-generated tweets using Open AI GPT3.

III. Related Work

Distinctive characteristics of the initial studies on stance detection lie in (1) the text genre and annotation characteristics of the datasets that they use and (2) the types of stance detection classifiers and features used by these classifiers.

Common stance targets in online debates include diverse topics including evolution, gun rights, gay rights, abortion, healthcare, the death penalty, and the existence of God. Earlier work also demonstrates a slight diversity in the class names used for stance annotation, i.e., in place of the stance classes of Favor, Against, different studies use Support, Oppose, Pro, Con, and Pro, Anti, among others.

In earlier work on stance detection (as well as in recent related work), it is a common practice to employ various different classifiers and compare their performance rates or an ensemble of those classifiers [5]. The classifiers tested in earlier work include rule-based algorithms (such as JRip) supervised algorithms like SVM [6], naive Bayes, boosting, decision tree and random forest Hidden Markov Models (HMM) and Conditional Random Fields (CRF), graph algorithms such as MaxCut, and other approaches such as Integer Linear Programming (ILP) and Probabilistic Soft Logic (PSL). Several state-of-the-art machine learning algorithms are used to implement Stance Detection as follows:

A. *Supervised Machine Learning*

SVM is the most commonly employed approach for stance detection, being used in more than 40 studies on stance detection, either as the main best-scoring approach or as the baseline approach. Logistic Regression [7] is the second most frequent classifier used for stance detection, appearing in more than 15 on-topic studies [4]. Considering the related literature that we cover in this article, the probabilistic classifier, Naïve Bayes, is the third widely employed algorithm of the traditional feature-based learning genre, appearing in more than 10 related studies.

B. *Deep Learning*

Deep learning models such as recurrent neural networks (RNN) and their variants and convolution neural networks (CNN) have been used effectively in many NLP tasks that share similarities to fake news and consist of calculating semantic similarity between sentences and community-based question answering. Siamese MaLSTM [8] is used to compute the semantic similarity of question pairs. A deep neural network converts the text sequence into fixed-length vector representation which is then used to measure the relevance of two textual sequences, which is the relevance of each headline-body pair in our case.

C. *Transfer Learning*

In recent times, methods such as ULMFiT, OpenAI GPT, ELMo and Google AI's BERT [9] have revolutionized the field of transfer learning in NLP by using language modeling during pre-training, which has significantly improved the state-of-the-art for a variety of tasks in natural language understanding. It can be argued that the use of language modeling (which is not without its limitations) is one of the main reasons computers have shown great improvements in their semantic understanding of language.

D. *Unsupervised Learning*

Using dimensionality reduction [10] and clustering, unsupervised learning approaches map users onto a low-dimensional space, allowing us to identify core users who are typical of the many views. In comparison to previous techniques that rely on supervised or semi-supervised classification, this approach has three key advantages. First, instead of requiring users to be labeled beforehand, clusters are generated instead, which are significantly quicker to label manually later on, for example, in a matter of seconds or minutes as opposed to hours. Second, neither defining the pertinent positions (labels) nor carrying out the actual labeling require domain or topic-level understanding. Third, the current system is resistant to data skewness, such as when some users or stances have a more significant representation in the data. [11]

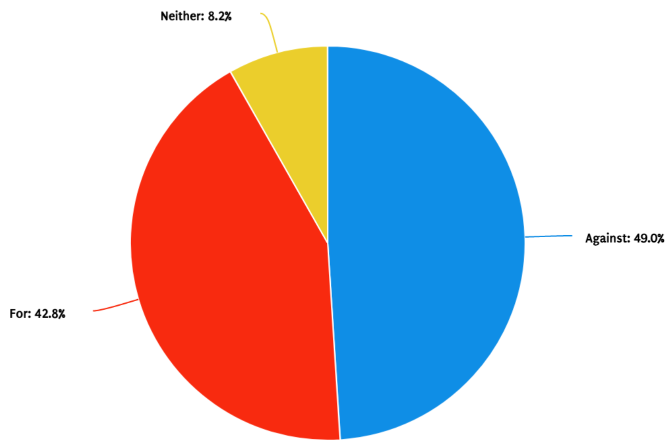


Fig. 1: Pie chart to vizualize dataset

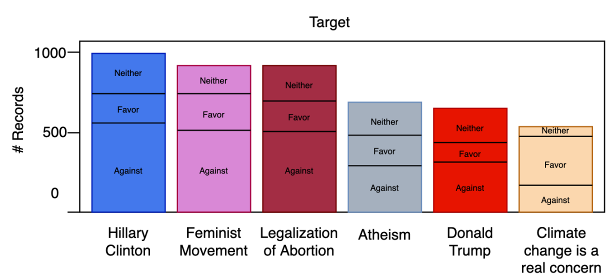


Fig. 2: Distribution of Targets and Stances in SemEval-2016

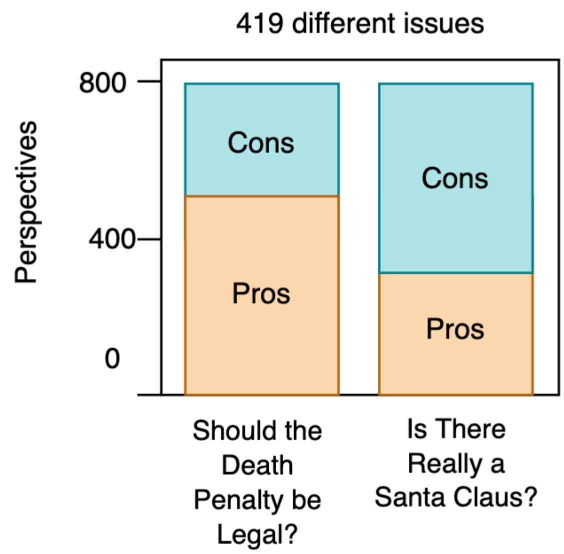


Fig. 3: Procon Dataset Structure

IV. Datasets

In pursuit of a model that can serve as a general stance detector for any topic, the dataset(s) to be used in its training was a key consideration. Ultimately, two datasets were selected. Further data augmentation was performed on this data to generate it for more objects as well.

A. The SemEval-2016 Stance Dataset

This dataset often referred to as the *Stance Dataset*, is a powerful tool containing 4870 pairs of tweets and their associated targets/topics. The tweets are labeled by an accuracy-checked human as positive, negative, or neither. The topics it contains are:

1) 'Atheism'
2) 'Climate Change is a Real Concern'
3) 'Feminist Movement'
4) 'Hillary Clinton'
5) 'Legalization of Abortion'
6) 'Donald Trump

SemEval-2016 [6] is a powerful tool to leverage because it provides nearly a thousand instances per target of hand-labeled tweets, which would theoretically show the model minute differences in tweets concerning the same topic and how to discern different stances among them. However, this dataset will not suffice if the model is to perform well when faced with a topic it has not yet seen. In order to achieve this, the model ought to be exposed to far more than six different targets.

B. The Procon20 Dataset

Procon.org [12] is a website that presents a wide variety of controversial topics in a question format from 'Should the Death Penalty be Legal?' to 'Is There Really a Santa Claus?'. They then present a list of pros and cons for each topic. Since these pros and cons are inherently labeled - the website provides 419 different issues, this data is a powerful tool in training a generalized stance detection model.

In total, the dataset contains 6,094 different "perspectives" (pro or con) across 419 issues. Generally, these perspectives are 100-200 words written by experts from procon.org in favor of or against a certain topic. This dataset will be highly valuable in training the model because it covers 70x more topics/targets than the SemEval-2016 Dataset. Although there are far less perspectives per target, exposing the model to a wider array of targets is vital to improving the accuracy of a general stance detector for any target.

C. Augmentation & Preprocessing

1) Query Augmentation: Going forward, with the supervised learning algorithm, the main issue here is the shortage of annotated data. So we would use some data augmentation techniques to increase the size of the dataset. We will be using the following methods -

- Back Translation [13]
- Text Paraphrasing [14]
- Introduce typing mistakes

As seen in the Fig. 4, only sentences with good bleu [15] scores (for this study we considered anything above 0.75 as a good score) will be added to the training data, and with effective permutations of these methods, we can obtain 3 to

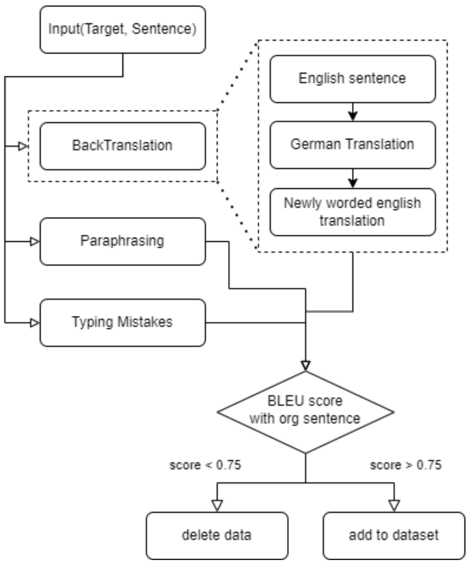


Fig. 4: Augmentation Flowchart

15 times more data. Later, we discovered using 3 times data is more effective because as we increase similar kinds of data in the training dataset, the model was overfitting on it and underperforming on the test data.

2) 'Neutral' Data: Unlike the Sem-eval dataset, the Procon20 dataset does not have a 'Neutral' stance. So we plan on adding enough data from the other topics to a particular topic and label them as 'Neutral' so long as the dataset is not imbalanced.

3) Data Generation: Data generation for new/unknown topics is done using OpenAI GPT-3 to set baselines for the decision tree. We also use OpenAI GPT3 to generate tweets regarding those targets which have fewer data.

V. System Architecture & Algorithms

A. Decision Tree

The first approach chosen by us was to use a Decision Trees. As the name suggests, these decision trees make a series of 'decisions' for each attribute we provide to it. These decisions can be said to be a bunch of if and else statements put together, with the decision tree setting the actual condition. This condition is chosen with the help of Information Gain. Due to the powerful visualisation they can provide for such models, it would be quite easy to understand fulfilling the criteria we want and giving us a deeper look into what is required and what is lacking.

The plan is to convert words to tokens using Word2Vec which would also take into account how most general words will behave with each other. And the same Word2Vec model will also allow us to have vectorized distances between any 2 chosen words which allows for target identification.

For the data in the dataset, the plan was to generate a unique decision tree each for the targets and store them. As for the targets not present in the dataset; the plan to implement them was unique. Since we cannot have a decision tree for

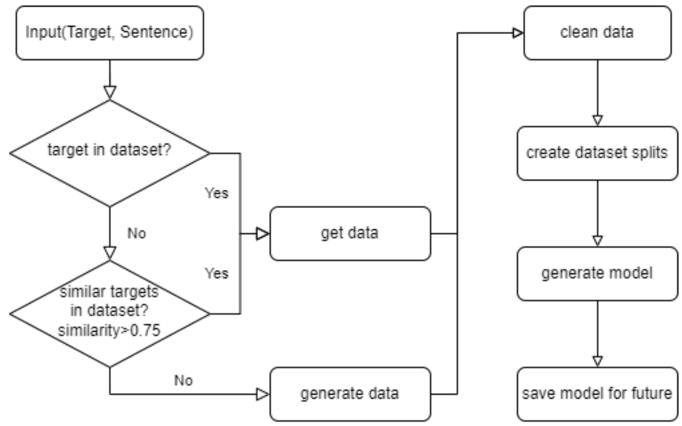


Fig. 5: Decision Tree based model Architecture

every target in the English dictionary, we would instead cluster multiple targets together. So, using Word2Vec [16], we check which of our already existing targets are closest to the decision tree model and if that vectorized distance meets our criteria, we use the same tree. If it does not, we use OpenAI's GPT3 to generate new data and make a new tree using that, and this decision tree will be used to predict outcomes. This particular architecture is what we see in Fig 5.

However, after more consideration and testing, this model falls very short when it comes to proper nouns and this approach does not yield results that can be touted to be great in general cases as well. This led us to trying to implement more complex models like BERT.

B. BERT

After trying simpler machine learning models like decision trees, it was clear to us that the simple nature of these models cannot represent the complex contextually varying nature of our queries. It might have performed better if it was a straightforward sentiment prediction task, by identifying certain keywords and focusing on them, but the stance aspect of the classification made it difficult for the linear models to identify the keywords for different topics and then predict the stance. Thus, we decided using deep neural network models was inevitable and we came across BERT.

BERT is a pre-trained transformer-based language model for English and is trained on a ton of data, so by applying transfer learning to such a model we can acquire high-quality predictions. [17] The two techniques for training a BERT model are masked language modeling and next-sentence prediction together. In the case of classification, the BERT model is finetuned using the next sentence prediction training technique. This enables the classifier to predict if the sentence is connected to the previous sentence.

In our case, it would predict the connection between the tweet or sentence and the target (i.e. topic). Suppose we have a sentence - "The ocean levels are rising, and the weather is changing constantly; the earth needs to be saved." and the target is "Climate change is a real concern" then the model

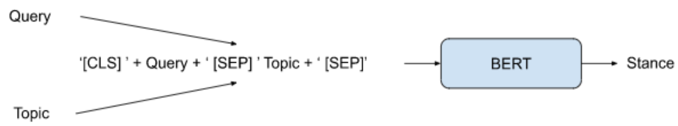


Fig. 6: BERT based model Architecture

will predict the stance based on the connection between the target and sentence.

Thus, to train a stance detection model (BERTStance), the BERT model can be finetuned for the classification task using both the sentence and the target as input. As shown in Fig 6 the input to the BERT model will be query i.e. the sentence and the target i.e. the topic, to get the stance as output

Input - '[CLS]' + Query + '[SEP]' + Topic + '[SEP]'

Output - Stance

To further illustrate the problem with classifying 'neutral' stance queries we even modeled a BERTStanceYN model that only predicts the stance as either 'favor' or 'against'.

VI. Evaluations

A. Decision Tree

We evaluate these models primarily based on F1 scores that we generate for varied test cases. For targets that is already present in the dataset, the implementation yielded us results as seen in Table I. And for targets that were not seen by the model before, we use 2 methodologies, in case pre-generated models are used and in case data-generation is used. Both of these can be seen in Table II respectively.

Target	F1-score	
	Original	Back-translated
Atheism	0.34	0.41
Feminist Movement	0.34	0.21
Hillary Clinton	0.52	0.46
Climate change is a real concern	0.27	0.59
Legalization of abortion	0.25	0.50

TABLE I: Decision Tree for SemEval Test A

Target	F1-score	
	pre-existing data	generated data
Donald Trump	0.29	0.33

TABLE II: Decision Tree for SemEval Test B

We can also see the Area under curve, False positive rate, and True positive rate evaluations with their plot in Fig. 7a, Fig. 7b and Fig. 7c which gives us an insight on the ratio of True and False Positives for different scenarios.

B. BERTStance

The BERTStance model is initially evaluated using the accuracy as well as the F1 score of the model on each target. In III you can see the test accuracies and the F1 score of the model on each topic from the semeval dataset. The topics in III are topics that are present in the training data, so the model has already been trained to predict stances for these topics.

Target	Accuracy	F1-score
Atheism	0.71	0.68
Feminist Movement	0.62	0.62
Hillary Clinton	0.73	0.73
Climate change is a real concern	0.84	0.82
Legalization of abortion	0.65	0.64

TABLE III: BERTStance for SemEval Test A

To further understand the robust nature of the model we plot the ROC curve for each stance "favor", "against" and "neither" in Fig. 8a, Fig. 8b and Fig. 8c respectively. If we compare these curves with those obtained in Fig. 7a, Fig. 7b and Fig. 7c we can see a significant improvement in the area under the ROC curve which helps to aggregate the performance of the classifier.

Further, we perform to evaluate the model's performance on unseen topics. This is essentially the ZSL performance of the BERTStance model. In Table IV the topic is "Donald Trump" which was not used during the training and the model performs on this topic with an accuracy of 0.41 and F1 score of 0.38 as shown in the table.

Target	Accuracy	F1-score
Donald Trump	0.41	0.38

TABLE IV: BERTStance for SemEval Test B

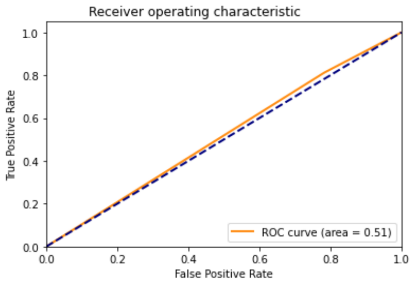
We further compare the model's Few Shot Learning (FSL) performance [18]. In Table V the FSL performance over ZSL performance signifies how quickly the model learns. To finetune on new topics we only used 5 queries per new target to train, validate and test the model. The new targets we introduced here were - "Raising minimum wage", "Human cloning", "Defund the police", "Underage drinking", and "Mandatory vaccinations". We used sentences from the Procon dataset and tweets generated by the OpenAI GPT3 model. This result also demonstrated that the model performs much better on relatively smaller queries (tweets/sentences).

	Accuracy	F1-score
Zero-shot learning	0.46	0.45
Few-shot learning	0.96	0.95

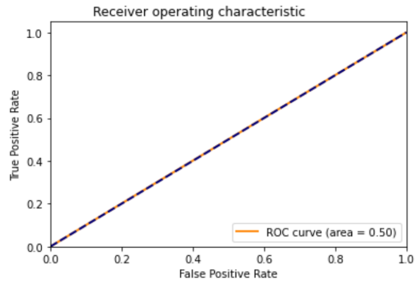
TABLE V: Comparing Zero-shot and Few-shot learning

C. BERTStanceYN

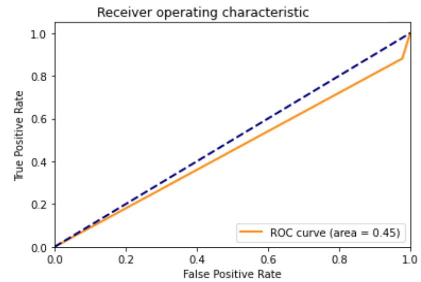
The special case model we created using just the "favor" and "against" stances, had higher accuracy and F-1 score. This model made it clearer how the language modeling is done for stance detection using BERT is actually working. There is a lack of neutral data in most topics which creates highly imbalanced training data. We also noticed some mislabeling of neutral data present in the dataset. In Table VI it can be



(a) ”Favor” stance

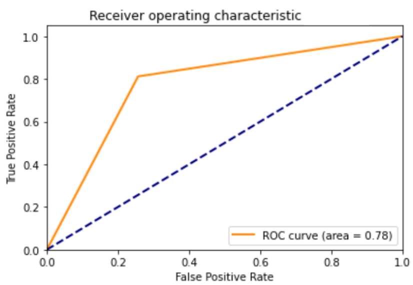


(b) ”Against” stance

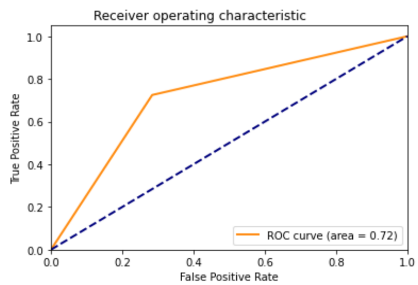


(c) ”Neither” stance

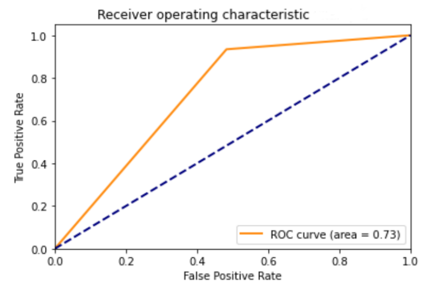
Fig. 7: Decision Tree ROC curves



(a) ”Favor” stance



(b) ”Against” stance



(c) ”Neither” stance

Fig. 8: BERTStance ROC curves

Target	Accuracy	F1-score
Atheism	0.74	0.72
Feminist Movement	0.74	0.73
Hillary Clinton	0.80	0.77
Climate change is a real concern	0.84	0.82
Legalization of abortion	0.81	0.79

TABLE VI: BERTStanceYN for SemEval Test A

seen that the accuracies and F1 scores for every target's stance prediction are improved.

The results in Table VI is only on targets that have been seen by the model, but the results in Table VII display the model's zero-shot learning performance with an accuracy of 0.68 on unseen target and 0.48 F1 score.

Target	Accuracy	F1-score
Donald Trump	0.68	0.48

TABLE VII: BERTStanceYN for SemEval Test B

VII. Model Reasoning Visualization

Although deep learning models are performing well on text data, obtaining the reasoning for the predictions is difficult because it is a black box model. To decode this black box model's reasoning we employed the metamorphic testing approach where we only consider the input and the output. We used a bag of words model disregarding the grammatical rules. The bag of words is a simple model, but useful in computer vision and document classification.

By using the bag of words model we were able to assign the contribution score to each token w.r.t. to the stance output. For instance, in Fig. 10 the query displayed is passed into the model to get the prediction and prediction confidence, then using this we try to assign word contributions to the words present in the input sentence. Here the topic is ”Atheism”, and the sentence here is against atheism. Under the stance - 'against' i.e. LABEL_0 the word ”believe” is highlighted green because it supports this stance w.r.t to the target, whereas the word ”freedom” is against the stance w.r.t. to the target. In short the word ”believe” suggests that the sentence is against atheism whereas the word ”freedom” supports that the sentence is in favor of atheism, but the overall stance is obtained as against atheism. The results in Fig. 9 and Fig. 11 are obtained similarly. For Fig. 10, Fig. 9 and Fig. 11 LABEL_0, LABEL_1, and LABEL_2 means ”Against”, ”Favor” and ”Neither” stances.

Fig. 9: Favour Output

The query below (Fig. 10) has a stance of 'against' wrt target 'atheism'. The model predicts correctly and highlights the reason which has a positive impact on the output with green, whereas red indicates a negative impact on the output.

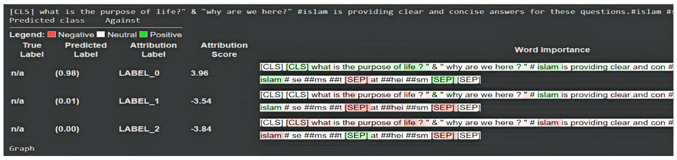


Fig. 10: Against Output

The query below (Fig. 11) has a stance of 'neither' w.r.t. target 'atheism'. The model predicts correctly and highlights the reason which has a positive impact on the output with green, whereas red indicates a negative impact on the output.

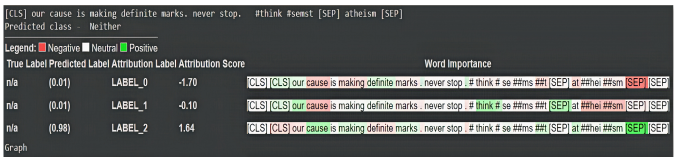


Fig. 11: Neither Output

VIII. Division of Work and Team Member Contributions

The implementation of stance detection of a tweet contains numerous steps like information gathering, data cleaning, model selection, and its research, model training, model evaluation, performance analysis, etc. To perform the steps efficiently, we divided the entire project into individual milestones with dedicated individuals for the same. The table VIII discusses the division of work for the project.

Task	**Assigned Group Member(s)**
Information Gathering & Research	Collin Wood, Mukesh Jha, Mahidher Duraisamy Krishnan
Data Cleaning & Pre-Processing	Collin Wood, Viraj Thakkar, Mukesh Jha
Model Selection & Research	Eric Waters, Mahidher Duraisamy Krishnan
Model Training & Evaluation	Saurabh Balasaheb Mohite, Viraj Thakkar
Evaluate Performance & Analysis	Eric Waters, Saurabh Balasaheb Mohite
Model reasoning visualization (Bonus)	Saurabh Balasaheb Mohite
Final Report	Everyone

TABLE VIII: Division of Work

The first step of information gathering and research involved going through various sources like [3], [6], and [12]. These sources were used to work on data generation, cleaning, and pre-processing in which we extracted the data as per the steps highlighted in the Augmentation & Preprocessing step. After this, we performed a survey of different methodologies for stance detection as listed under the Related works section in this paper. After finalizing several approaches we decided to move ahead with the decision tree and BERT models. The initial stage was training and testing the decision tree model followed by the BERT model to overcome the shortcomings of the decision tree model. After evaluating and testing the BERT model on seen target data and unseen target data (ZSL) we also evaluate the model's FSL aspect. After, this we tried to take a step towards explainable AI by employing the Bag of Words model on the model's performance to mimic the model in an explainable manner.

IX. Conclusion

In this study, we discuss the complex nature of the stance detection problem. The primary obstacle in stance detection is the contextual nature of the classification problem which is dependent on a target rather than a generalized way. Stance detection can be understood as a target-based sentiment analysis, which means the prediction made by the sentiment classifier is w.r.t. to the target.

Here, we have used classification techniques like decision trees and BERT for sequence classification to predict the stance of the query with respect to a specific topic. The decision tree model is a naive initial approach to understanding the complex nature of the problem. We also use the Bert language model to learn, with modified inputs - (Query, Topic) to predict the stance of the query w.r.t. the topic. The next sentence prediction training used to train and finetune the Bert model encodes the connection between the two sentences viz. the connection between the query and topic.

Although the BertStance model has a low ZSL accuracy and F1 score, it's still better than the state-of-the-art techniques used in the SemEval-2016 stance detection paper. [6] Also the FSL accuracy and F1 scores of the model makes it clear that the model is not a specific target classification model but a more generalized model that takes the target into consideration to make stance predictions.

References

[1] P. R. Center, "Social Media Fact Sheet." [Online]. Available: https://www.pewresearch.org/internet/fact-sheet/social-media/

[2] D. Pomerleau and D. Rao, "Fake news challenge stage 1 (fnc-i): Stance detection," *URL www. fakenewschallenge. org*, 2017.

[3] S. M. Mohammad, P. Sobhani, and S. Kiritchenko, "Stance and Sentiment in Tweets," *ACM Transactions on Internet Technology*, vol. 17, no. 3, pp. 1–23, Jul. 2017. [Online]. Available: https://dl.acm.org/doi/10.1145/3003433

[4] D. Küçük and F. Can, "Stance Detection: A Survey," *ACM Computing Surveys*, vol. 53, no. 1, pp. 1–37, Jan. 2021. [Online]. Available: https://dl.acm.org/doi/10.1145/3369026

[5] S. V. Vychegzhanin and E. V. Kotelnikov, "Stance detection based on ensembles of classifiers," *Programming and Computer Software*, vol. 45, no. 5, pp. 228–240, 2019.

[6] S. Mohammad, S. Kiritchenko, P. Sobhani, X. Zhu, and C. Cherry, "SemEval-2016 Task 6: Detecting Stance in Tweets," in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. San Diego, California: Association for Computational Linguistics, 2016, pp. 31–41. [Online]. Available: http://aclweb.org/anthology/S16-1003

[7] X. Jose, S. M. Kumar, and P. Chandran, "Characterization, classification and detection of fake news in online social media networks," in *2021 IEEE Mysore Sub Section International Conference (MysuruCon)*, 2021, pp. 759–765.

[8] E. Cohen, "How to predict Quora Question Pairs using Siamese Manhattan LSTM," Sep. 2018. [Online]. Available: https://blog.mlreview.com/implementing-malstm-on-kaggles-quora-question-pairs-competition-8b31b0b16a07

[9] K. Nagda, A. Mukherjee, M. Shah, P. Mulchandani, and L. Kurup, "Ascent of pre-trained state-of-the-art language models," in *Advanced Computing Technologies and Applications*. Springer, 2020, pp. 269–280.

[10] Y. Bian, C. North, E. Krokos, and S. Joseph, "Semantic explanation of interactive dimensionality reduction," in *2021 IEEE Visualization Conference (VIS)*, 2021, pp. 26–30.

[11] K. Darwish, P. Stefanov, M. Aupetit, and P. Nakov, "Unsupervised user stance detection on twitter," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 14, 2020, pp. 141–152.

[12] "ProCon.org - Pros and Cons of 100+ Topics." [Online]. Available: https://www.procon.org/

[13] N. Ng, K. Yee, A. Baevski, M. Ott, M. Auli, and S. Edunov, "Facebook FAIR's WMT19 News Translation Task Submission," in *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*. Florence, Italy: Association for Computational Linguistics, 2019, pp. 314–319. [Online]. Available: https://www.aclweb.org/anthology/W19-5333

[14] P. Damodaran, "Parrot: Paraphrase generation for nlu." 2021.

[15] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu, "Bleu: a method for automatic evaluation of machine translation," 10 2002.

[16] S. Sivakumar, L. S. Videla, T. Rajesh Kumar, J. Nagaraj, S. Itnal, and D. Haritha, "Review on word2vec word embedding neural net," in *2020 International Conference on Smart Electronics and Communication (ICOSEC)*, 2020, pp. 282–290.

[17] T. Zhang, F. Wu, A. Katiyar, K. Q. Weinberger, and Y. Artzi, "Revisiting few-sample bert fine-tuning," *arXiv preprint arXiv:2006.05987*, 2020.

[18] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata, "Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 9, pp. 2251–2265, 2019.