Stance Detection

Avish Khosla
School of Computing and Augmented Intelligence
Arizona State University
Tempe, USA
akhosla4@asu.edu

Baibhav Phukan
School of Computing and Augmented Intelligence
Arizona State University
Tempe, USA
bphukan@asu.edu

Gautham Maraswami
School of Computing and Augmented Intelligence
Arizona State University
Tempe, USA
gmaraswa@asu.edu

Sai Rathnam Pallayam Ramanarasaiah
School of Computing and Augmented Intelligence
Arizona State University
Tempe, USA
spallaya@asu.edu

Sai Vikhyath Kudhroli
School of Computing and Augmented Intelligence
Arizona State University
Tempe, USA
sv18@asu.edu

Tanuja Renu Sudha
School of Computing and Augmented Intelligence
Arizona State University
Tempe, USA
trenusud@asu.edu

***Abstract*— Stance detection is a challenging task in natural language processing that involves identifying the attitude or perspective of a speaker or writer towards a particular topic, claim, or entity. In this project, we explore two different techniques for stance detection: BERT with SVM and bi-directional LSTM. We use the SemEval-2016 dataset as the primary dataset for our experiments, and also further augment it with additional data from Twitter related to SemEval's topics as well as trending topics like Immigration and Gun laws. We use clustering algorithms to group the scraped tweets and manually assign a stance label to each cluster, which then serves as the ground truth for the classification task. We experiment with different feature sets and training strategies, including using pre-trained word embeddings and fine-tuning BERT. Our results show that both BERT with SVM and bi-directional LSTM achieve good accuracy on the SemEval-2016 dataset, with BERT performing slightly better. Overall, this project contributes to reproducing the state of the art in stance detection and advancing it a little further with our own modifications. This project demonstrates the potential of natural language processing for analyzing attitudes and opinions in text data.**

Keywords—component, formatting, style, styling, insert *(key words)*

I. Introduction

Stance detection is a relatively new research problem that falls under the umbrella of sentiment analysis. It aims to identify the author's stance towards a target mentioned or implied in the text, such as an entity, concept, event, idea, opinion, claim, or topic. It is also referred to as stance classification, identification, prediction, debate-side classification, or debate stance classification. Despite having a common purpose, the literature reports three main definitions of stance detection, including generic stance detection, rumour stance classification, and fake news stance detection. There are also two subclasses of the generic stance detection problem, which are multi-target stance detection and cross-target stance detection, based on the number of targets and the presence of the stance target in the training and testing datasets of the experimental settings.

Before introducing the definitions of all these different variations of the same problem, it would be beneficial to provide a linguistics-based definition of stance. Therefore, Du Bois defines stance as follows: "Stance is a public act by a social actor, achieved dialogically through overt communicative means, of simultaneously evaluating objects, positioning subjects (self and others), and aligning with other subjects, with respect to any salient dimension of the sociocultural field" [Du Bois 2007]. According to this definition, a stancetaker expresses their evaluation on an object and aligns themselves with others during a stance act. Figure 1 schematically demonstrates a generic stance detection task and identifies all the components of the task. There is a person or a stancetaker who has a piece of text associated to them. This text can be a tweet, a reddit post or comment, or a facebook post. There is also a target which is the topic of interest. And finally, there is a stance assigned to the person's text with respect to the target.

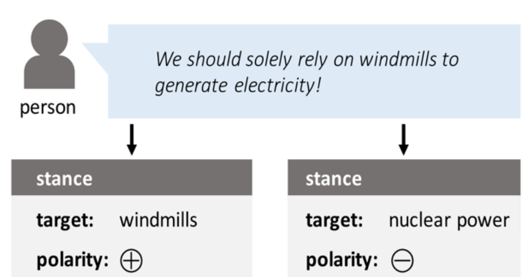


Fig. 1. A simple Stance Detection task

Returning back to the process of automatic stance detection, following are some definitions of stance detection:

- *Stance detection* is a classification problem that seeks to determine the stance of a text author towards a specific target or concept. The stance is categorized into labels such as "Favor," "Against," "Neither," and sometimes "Neutral." The target may or may not be explicitly mentioned in the text.

- *Multi-target stance detection* is a classification problem that seeks to determine the stance of a text author towards a set of related targets. For each target, the stance of the text author is categorized into "Favor," "Against," or "Neither." The classifications for each target can affect the stance classifications for the remaining targets.

- *Cross-target stance detection* is a classification problem that seeks to determine the stance of a text author towards a specific target when stance annotations are available for different (but related) targets. This is done in a setting where there is not enough stance-annotated training data for the target under consideration.

- *Rumour stance classification* is a problem that seeks to determine the position of the text author towards the veracity of a rumour. The stance is categorized into labels such as "Supporting," "Denying," "Querying," or "Commenting."

- *Fake news stance detection* is a classification problem that seeks to determine the stance of the body of a news article towards the claim of the headline. The stance is categorized into labels such as "Agrees," "Disagrees," "Discusses (the same topic)," or "Unrelated." This problem is defined to aid the detection of fake news.

According to our literature, the most common definition of automatic stance detection is the first one given above, which predicts one's stance on what they write. This is also the same definition that has been represented in Figure 1.

Stance detection is a challenging task due to the ambiguity and subjectivity of language. The same text can be interpreted differently by different people depending on their background, beliefs, and context. Moreover, the expressions of stance can be implicit and indirect, making them hard to capture using traditional rule-based or keyword-based methods. As a result, researchers have explored various techniques and models for stance detection, ranging from traditional machine learning methods to deep learning approaches.

In recent years, deep learning models, such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and transformer-based models like BERT, have shown promising results on stance detection tasks. These models can capture complex patterns and relationships in language and learn representations that are useful for classification. For instance, BERT is a pre-trained transformer-based model that has achieved state-of-the-art performance on various natural language processing tasks, including sentiment analysis, question answering, and stance detection.

In this project, we explore different techniques and models for stance detection, with a focus on comparing the performance of BERT with SVM and bi-directional LSTM on the SemEval-2016 dataset. The SemEval-2016 dataset is a widely used benchmark dataset for stance detection that contains tweets related to five topics: Atheism, Climate Change, Feminism, Hillary Clinton, and Abortion. We also expand the dataset by using Twitter scraping and clustering to obtain additional data related to SemEval's topics as well as other trending topics like Immigration and Gun Laws. We use the clustering results to manually assign a stance label to each cluster, which serves as the ground truth for the classification task. Our goal is to evaluate the effectiveness of different techniques and models on stance detection and investigate the impact of data diversity and domain specificity on the performance of the models.

To accomplish our goal, we first preprocess the dataset by cleaning the tweets, tokenizing the text, and performing normalization and stemming. We then experiment with different feature sets and training strategies, and go on to build two models - BERT with SVM and a bi-directional LSTM. We train the models on the SemEval-2016 dataset and evaluate their performance using various metrics, including accuracy, F1-score, and confusion matrix. We also conduct an error analysis to investigate the types of errors made by the models and the limitations of the dataset. Our findings suggest that the performance of the models is influenced by the quality and representativeness of the data, as well as the choice of hyperparameters and classification threshold.

II. Problem Statement

The widespread adoption of social media platforms has transformed the way people communicate and consume information. As more and more people rely on online sources for news, opinions, and product reviews, the volume of user-generated content has increased significantly. While this democratization of information has some clear advantages, it also presents a challenge. The sheer volume of content generated every day makes it impossible for individuals to fact-check and verify all the information they come across. Consequently, there is a growing concern about the reliability and trustworthiness of the information available online.

One of the major problems with online content is the presence of biased or fake information. With so many different viewpoints and sources available online, it can be challenging to discern the truth from falsehood. This is particularly true when it comes to sensitive topics like politics, religion, and social issues. In such cases, people may intentionally or unintentionally present a skewed or one-sided perspective, making it difficult for readers to form an objective opinion.

To address the problem of biased information, researchers have been pushing the horizon on automatic stance detection. This process involves using computational techniques to analyze text data and determine the writer's attitude or stance towards a specific topic, product, or service. By analyzing the stance of a writer towards a particular topic, it is possible to determine if the information they are providing is biased or fake. For instance, if a writer consistently takes a negative stance towards a particular political party or candidate, their opinions on that topic may be influenced by their biases, and their content may not be objective.

In addition, stance detection can also be used to identify fake news. Fake news is often designed to mislead people by presenting false or misleading information as if it were true. By analyzing the stance of a writer towards a particular topic, it is possible to determine if their content is reliable or not. If a writer consistently takes a stance that is contrary to the facts, their content may be considered unreliable, and may be flagged as potentially fake news.

Stance detection is also a powerful tool that has the potential to transform the way organizations approach marketing and advertising. By identifying an individual's stance towards a particular topic, product, or service, organizations can create personalized and targeted marketing strategies that are more likely to resonate with their target audience. For instance, if an individual has a positive stance towards healthy eating, a company that sells organic foods or health supplements could use this information to recommend their products to the individual. This can result in higher conversion rates, as individuals are more likely to purchase products that align with their values and beliefs.

Finally, stance detection can also be used to identify an individual's stance towards broader social or political issues. This information can be invaluable for organizations looking to tailor their marketing and advertising campaigns to better align with the values and beliefs of their target audience. For example, a clothing company that is committed to diversity and inclusivity could use stance detection to identify individuals who are passionate about social justice issues and use this information to promote their products to this audience.

Stance detection is particularly relevant in today's digital age, where individuals are increasingly using social media and online platforms to express their opinions and beliefs. By analyzing the language used in social media posts, stance detection algorithms can accurately identify an individual's stance towards a particular topic, product, or service. This can help organizations stay ahead of the curve and develop marketing strategies that are more attuned to the evolving preferences and beliefs of their target audience.

The purpose of our project is to evaluate the effectiveness of different techniques for stance detection and to explore the potential of expanding the dataset through data augmentation techniques like web scraping. By doing so, we aim to contribute to the growing body of research on automatic stance detection and to provide insights into its application for real-world problems. We believe that automatic stance detection has significant implications on all the practical applications we've discussed throughout this section, including targeted marketing and identifying biased or fake information.

III. RELATED WORK

Current stance detection approaches employ many language characteristics such as word/character n-grams, dependency parse trees, and lexicons [Sun et al., 2018; Sridhar et al., 2015; Hasan and Ng, 2013; Walker et al.]. End-to-end neural network models that learn subjects and opinions individually and merge them using memory networks [Mohtarami et al., 2018], or neural attention [Du et al., 2017] are used in certain techniques. To jointly describe topics and opinions, several neural network models include lexical characteristics [Riedel et al., 2017; Hanselowski et al., 2018] and a consistency requirement. However, none of these approaches, have employed bidirectional transformers and large language models with sentiment and emotion in a shallow neural network. The emphasis has been universally on detecting the position in nonpartisan talks that are generally 100-200 words long.

According to Hasan and Ng's classification in 2013 [Hasan and Ng, 2013], debate attitude identification research has mostly concentrated on legislative floor debates [Thomas et al. 2006], company-internal talks [Murakami and Raymond 2010], and online social, political, and ideological disputes [Anand et al. 2011; Somasundaran and Wiebe 2010; Walker et al. 2012a]. Earlier research also included product-related online arguments [Somasundaran and Wiebe 2009]. Additional research on stance detection in spontaneous speech [Levow et al. 2014], student essays [Faulkner 2014], and tweets [Rajadesingan and Liu 2014] have been conducted since 2013. Because of related contests, the number of studies on posture detection for tweets has expanded dramatically. However, online ideological and social debate research [Sridhar et al. 2014, 2015] continues to be important in stance detection research. This survey article discusses the many types of stance detection investigations that have been undertaken, as well as their frequency.

Previous studies and new research in stance detection often compare the performance of several classifiers. Rule-based algorithms such as JRip [Anand et al. 2011; Murakami and Raymond 2010; Walker et al. 2012a, 2012b], supervised algorithms such as SVM [Hasan and Ng 2013; Somasundaran and Wiebe 2010; Thomas et al. 2006; Walker et al. 2012b] and naive Bayes [Anand et al. 2011; Hasan and Ng 2013; Rajadesingan and Liu 2014; Walker et al. 2012b], boosting [Levow et al. 2014], decision tree and random forest [Misra and Walker 2013], Hidden Markov Models [Murakami and Raymond 2010; Walker et al. 2012a] and CRF, graph algorithms such as MaxCut, and other techniques such as Integer Linear Programming [Somasundaran and Wiebe 2009] and Probabilistic Soft Logic [Sridhar et al. 2014, 2015] are examples of classifiers. Previous studies on posture identification for a range of contexts, including legislative floor debates, firm internal talks, and online social, political, and ideological conflicts, have used these classifiers.

The utilization of inter-post information, such as agreement or disagreement links, reply links, rebuttal material, and retweeting activity, was a key aspect of previous stance identification investigations. These experiments show that incorporating such collective information enhances stance identification performance when compared to evaluating each post separately. Word n-grams, cue or topic words, dependencies, argument-related features, emotion or subjectivity characteristics, and frame-semantic features are all commonly employed in these investigations.

IV. DATASET

A. *SemEval 2016*

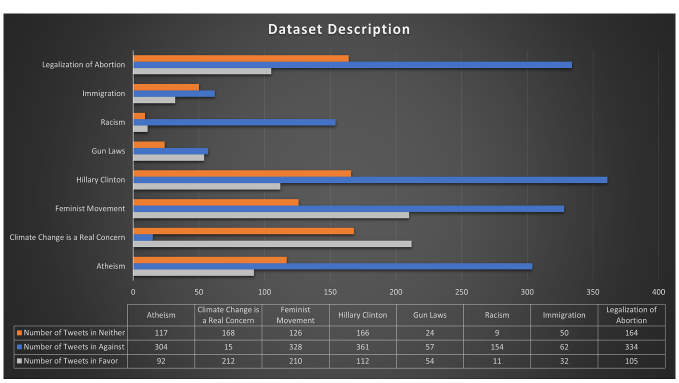
The SemEval 2016 dataset for Stance Detection is a benchmark dataset that contains tweets expressing either support, denial, or query for a specific target, such as a person, organization, or policy. The dataset was developed for the SemEval-2016 Task 6: Detecting Stance in Tweets challenge, which aimed to encourage research in automatic stance detection in social media.

The dataset consists of 2,914 tweets, with each tweet annotated by multiple human annotators for stance towards the given target. The dataset includes a total of 5 targets, ranging from political figures to social issues, such as climate change, feminism, and abortion. For each target, the dataset provides annotations of the tweet stance, where a tweet can be classified as either supportive, denying, or questioning towards the target.

The SemEval 2016 dataset has been widely used in research on automatic stance detection in social media. It has been used to evaluate various machine learning models and deep learning techniques for stance detection, including SVMs, LSTMs, and BERT-based models. Moreover, the dataset has been extended and adapted for various domains, such as the detection of stance towards specific events, products, or services. Therefore, this serves as an excellent dataset for our project as well, and we will be training our models on it.

However, we have also realized that the SemEval-2016 dataset is limited in size and may not fully represent the complexities and nuances of stance detection in real-world scenarios. To overcome this, we have expanded our dataset by scraping more tweets related to SemEval's topics and other trending using the Twitter API for scraping called Tweepy. To ensure the quality and reliability of our expanded dataset, we have run some basic clustering algorithms to find reasonable clusters among the scraped tweets. We then manually reviewed each cluster to assign a stance to it, which serves as the label for the classification task. This step was crucial in ensuring the accuracy of our results, as clustering and labeling tweets is a complex and subjective task that requires human expertise.

With 348 extra tweets scraped across topics like Gun Laws, Immigration and Racism, we finally prepared our dataset of total 3262 tweets across 8 topics. We have visually represented our dataset and all the individual numbers per each stance (In favor, Against, Neutral) across all topics below.



	Atheism	Climate Change is a Real Concern	Feminist Movement	Hillary Clinton	Gun Laws	Racism	Immigration	Legalization of Abortion
Number of Tweets in Neither	117	168	126	166	24	9	50	164
Number of Tweets in Against	304	15	328	361	57	154	62	334
Number of Tweets in Favor	92	212	210	112	54	11	32	105

Fig. 2. Distribution of the dataset

As you can see from the figure, we have relatively much more tweets in against of each topic as compared to those in favor and neutral. This imbalance further explains our model performance and its inherent bias towards classifying the Against tweets more correctly predominantly over the other two classes.

B. *Data Augmentation*

In order to preprocess the SemEval 2016 dataset for stance detection, we carried out several steps to clean and transform the text into a suitable format for analysis. Firstly, we divided the data into a training set containing 80% of the data, and a test set containing the remaining 20% of the data.

Next, we conducted data cleaning by removing URLs and mentions and replacing emoticons with their corresponding sentiment to eliminate any noise in the dataset. Additionally, we corrected any spelling errors in the text to ensure consistency and accuracy.

Further, we carried out tokenization to split the text into individual words. This enabled us to represent the text as numerical vectors, which are essential for training machine learning models. In order to focus more on the important and informative parts of the text, we removed common words such as "and", "the", "a", "an", etc. This is known as stop word removal and helps in reducing the dimensionality of the dataset and improving the model's performance.

Finally, we carried out stemming and lemmatization, which involves clustering similar words together to pinpoint pertinent keywords and extract the most significant features. Stemming reduces inflectional forms and derivatives of words to their base or root form, whereas lemmatization transforms words to their dictionary form. By performing stemming and lemmatization, we were able to reduce the variability of words and group similar words together to extract more relevant features for the model training process.

C. *Data Preprocessing*

In addition to preprocessing the SemEval dataset, we have also carried out data augmentation techniques to enhance the dataset and improve the performance of the stance detection model. Firstly, we expanded the dataset by scraping Twitter for more tweets related to the existing topics as well as for the recently trending topics. This step was necessary to make the dataset more representative of current opinions and trends.

Next, we carried out synonym expansion by including synonyms to capture diverse opinions and understand more nuanced contexts. This step enabled the model to capture a wider range of opinions and improve the accuracy of stance detection. We also carried out phrase expansion to expand the use of related phrases, which helped to improve the comprehensiveness of the analysis.

Lastly, we performed query reformulation by reformulating the query, enabling the model to become more adaptable to predict the stance regardless of the specific way in which an opinion is expressed. These data augmentation techniques were crucial for enhancing the quality and diversity of the dataset, which ultimately contributed to improving the performance of the stance detection model.

V. Architecture and Algorithms

A. System Architecture

The architecture for our stance detection system can be described as a pipeline of sequential steps shown in Figure 3 below. The first step involves dataset preparation, which consists of collecting tweets from SemEval and additional tweets obtained through Twitter API scraping. A basic clustering algorithm is run on the additional tweets to identify clusters among them. Then we have manually reviewed each cluster to assign the most reasonable stances to each of them. This is important because the manually assigned stances serve as the ground truth for classification.

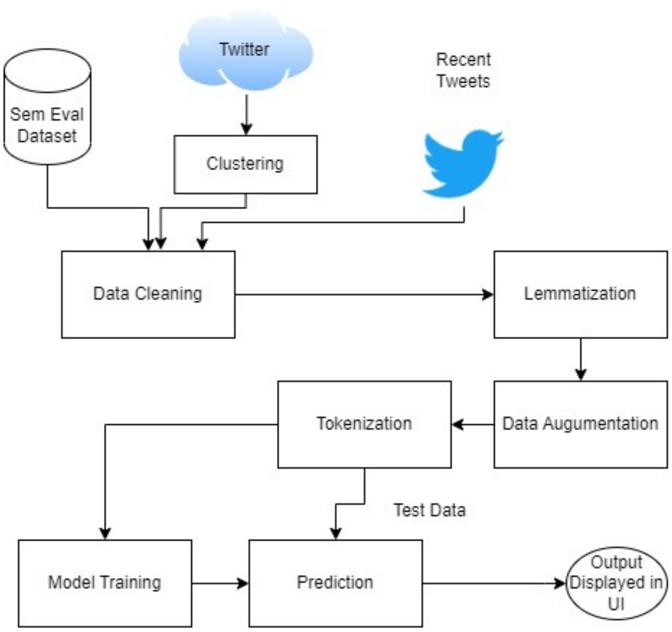


Fig. 3. System Architecture

The tweets are then standardized by lemmatization, which is a process of reducing words to their base form to ensure that similar words are grouped together.

Next, Data augmentation techniques are applied to expand the dataset and capture diverse opinions and nuanced contexts. Synonym expansion, phrase expansion, and query reformulation are used to increase the variety and richness of the dataset.

After data augmentation, tokenization is performed to convert the textual data into numerical vectors that can be used for model training.

We have then trained the dataset on 2 models - BERT with SVM and Bi-LSTM, both of which have shown promising results in previous research on stance detection.

Finally, the prediction phase involves inputting a text into the trained model to predict the stance of the text. And lastly, the output is displayed on the user interface, which provides a clear and easy-to-understand representation of the stance prediction results.

B. BERT with SVM

BERT, short for Bidirectional Encoder Representations from Transformers, is a pre-trained language model developed by Google. It is trained on a large corpus of text to learn the language patterns and generate contextualized embeddings for each word in the text. These embeddings can then be fine-tuned on a specific task, such as sentiment analysis or stance detection, by adding a task-specific layer on top of the pre-trained model.

BERT has shown to be effective for a range of natural language processing (NLP) tasks, including sentiment analysis and stance detection. In the case of stance detection, BERT is able to capture the contextual meaning of words and phrases, which is crucial for accurately detecting the stance of a piece of text.

When using BERT for stance detection, the model is first fine-tuned on a stance detection dataset. The input text is tokenized and fed into the pre-trained BERT model, which generates contextualized embeddings for each token in the text. The embeddings are then passed through a task-specific layer, which is trained to predict the stance label (e.g., favor, against, neutral) for the input text. Figure 4 shows how the input is broken into tokens and passed through layers of BERT embeddings for classifying given text.

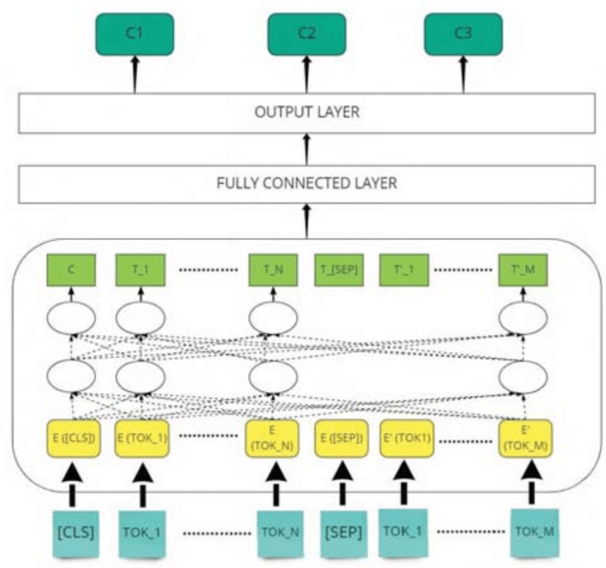


Fig. 4. Working of BERT

Coupling BERT with SVM for stance detection has been shown to improve the performance of the model. SVM, short for Support Vector Machines, is a classic machine learning algorithm used for classification tasks. It works by finding the hyperplane that best separates the data points of different classes. In the case of stance detection, SVM can help to classify the contextualized embeddings generated by BERT into the different stance categories.

By using BERT to generate contextualized embeddings and SVM to classify the embeddings into stance categories, the model is able to capture both the semantic meaning of the text and the underlying patterns in the data. This approach has shown to achieve state-of-the-art performance on various stance detection datasets.

C. Bi-directional LSTMs

Bi-LSTM stands for Bidirectional Long Short-Term Memory. It is a type of recurrent neural network (RNN) architecture that is often used in natural language processing tasks like sentiment analysis, text classification, and language translation.

The Bi-LSTM architecture consists of two layers: a forward LSTM layer that reads the input text from left to right, and a backward LSTM layer that reads the input text from right to left. This allows the model to capture both past and future dependencies in the text, making it well-suited for sequence labeling tasks like stance detection. Figure 5 shows the architecture of a Bidirectional LSTM.

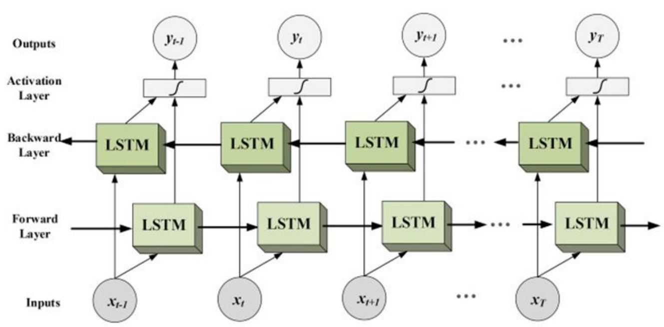


Fig. 5. Architecture of Bi-LSTMs

The LSTM layer within the Bi-LSTM architecture is designed to handle the vanishing gradient problem that is often encountered in RNNs. This problem occurs when the gradients of the network become too small during training, resulting in a slower convergence or even an inability to learn long-term dependencies in the data.

One of the main reasons why Bi-LSTM is a popular choice for stance detection is its ability to capture context-dependent information in the input text. Since stance detection involves analyzing the context and identifying the relevant aspects of the text that express a particular viewpoint or sentiment, Bi-LSTM is a suitable choice due to its ability to consider both the past and future contexts.

The model can be trained using the labeled dataset and can then be used to predict the stance of a new piece of text. However, it is worth noting that Bi-LSTM models can be computationally expensive and time-consuming to train, especially on large datasets. Therefore, it is important to consider the trade-off between performance and computational resources when choosing a model architecture for stance detection.

VI. EVALUATIONS

Regarding the evaluation of our stance detection models, we adopted two different approaches. The first one consisted of using common statistical metrics. The second approach involved comparing the performance of our model with existing models.

To evaluate the performance of our stance detection system, we have some of the commonly used statistical metrics - accuracy, precision, recall, and F1 score. Accuracy is a measure of how often the model makes correct predictions and is calculated by dividing the number of correct predictions by the total number of predictions. Precision is the proportion of true positive predictions out of all positive predictions, while recall measures the proportion of true positive predictions out of all actual positive instances. The F1 score is the harmonic mean of precision and recall, and it gives a more balanced view of the model's performance when compared to accuracy, especially when dealing with imbalanced datasets. Figure 6 summarizes the evaluation metrics for both of the models.

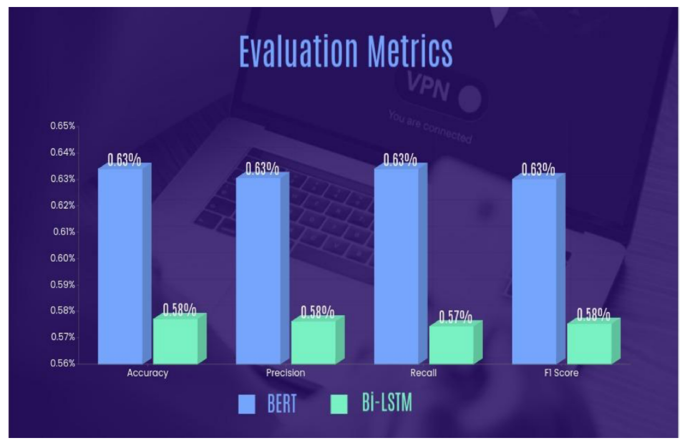


Fig. 6. Comparision of the evaluation metrics for both the models

The accuracy of our model using BERT was found to be 63%, whereas the Bi-LSTM approach yielded an accuracy of 58%. Accuracy measures the percentage of correct predictions out of the total number of predictions. It is a widely used metric for evaluating classification models. Accuracy is useful as it provides an overall picture of the model's performance.

Since the SemEval dataset that we used had the class imbalance problem, we decided to evaluate our model using precision and recall as well to make sure that the model is not biased towards a particular class. The precision of our BERT model was around 62%, whereas the Bi-LSTM model had a precision of 57.8%. Precision measures the fraction of true positives out of the total number of predicted positives. It is a useful metric when the goal is to minimize false positives. For example, in a stance detection task, a false positive would mean that the model predicted a stance that is not actually present in the text. In this case, precision would help in identifying the model's tendency to make false positive errors.

The recall of our model using BERT was found to be 63%, whereas the Bi-LSTM approach yielded a recall of 57%. Recall measures the fraction of true positives out of the total number of actual positives. It is a useful metric when the goal is to minimize false negatives. For example, in a stance detection task, a false negative would mean that the model failed to predict a stance that is actually present in the text. In this case, recall would help in identifying the model's tendency to miss important stances.

Finally, the F1 score is the harmonic mean of precision and recall. It is a useful metric when we want to balance the tradeoff between precision and recall. Our BERT model had an F1 score of 63% and our Bi-LSTM model resulted in an F1 score of 58%.

Figure 7 and Figure 8 below demonstrate the confusion matrices for the predictions from both the BERT and Bi-LSTM models. Both models are performing extremely well in predicting the tweets that are against the target, predominantly because of the class imbalance explained earlier. Both models correctly predicted the stance for more than 200 tweets. However, Bi-LSTM tends to perform better than BERT for the tweets that are in favor of the target, which can be observed by the numbers in the center cell of the matrix. Whereas BERT performs considerably better than BERT when it comes to tweets that do not relate to the target (neutral in stance). As you can see BERT predicted 85 tweets correctly that they are not related to the specified target whereas Bi-LSTM only predicted 46 correctly.

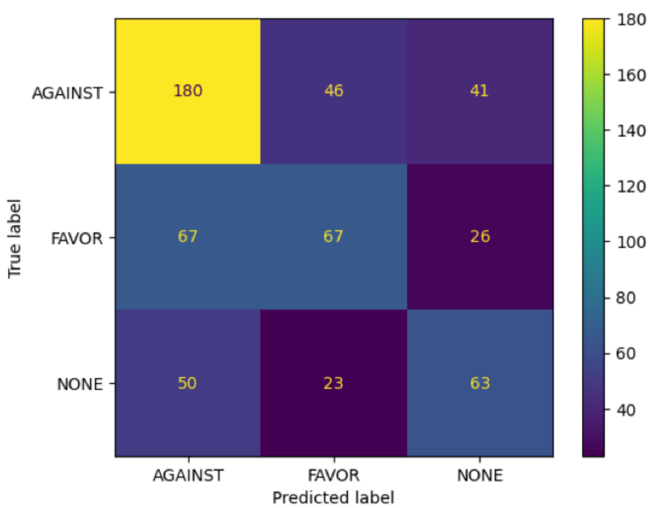


Fig. 7. Confusion Matrix for the predictions from BERT with SVM

Lastly, we have also compared our model against BERT-based models publicly available on the Hugging Face website. We found out that even though the training accuracy of those models was higher on most occasions, our model performed slightly better when it came to tweets outside of the SemEval dataset. While we are unsure as to why our models performed slightly better than those models, we speculate this is possibly because of our models not over-fitting on our training data.

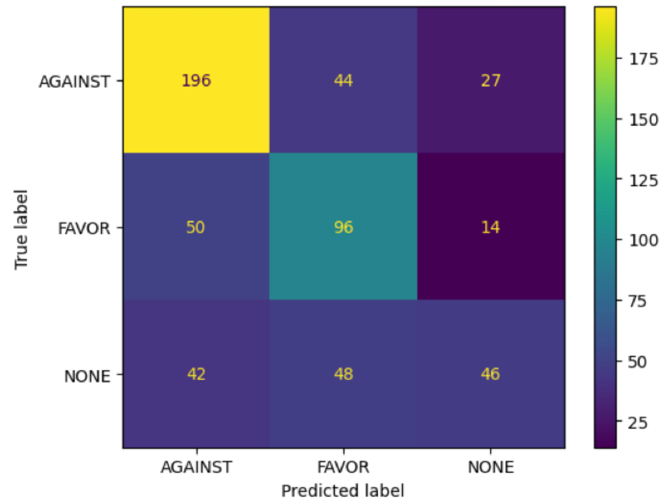


Fig. 8. Confusion Matrix for the predictions from Bi-LSTM

VII. User Interface

For our User Interface, we have built a simple web application using Dash by Plotly. Dash is a web application framework developed by Plotly for building interactive and customizable web applications with pure Python. It is built on top of Flask, React.js and Plotly.js, and provides a simple way to create interactive web applications with rich visualization capabilities.

The web application was simple and had just a handful of components. The first component was an input box where we can enter a piece of text whose stance we want to detect. Next, we had another dropdown component to select the target towards which we want to identify the stance of the initial text. Finally, once we click on the Call-to-action button, we concatenate the entered text and the selected target and pass it to a callback function within our Python application. Within the callback function, we would have serialized the deserialized Pickle file containing the trained model. Once the callback function receives the input, it uses the model to make a prediction on that input and then displays the prediction on the UI. We have used the BERT with SVM model for our prediction because it performed better than the Bi-LSTM model. Figure 9, 10 and 11 demonstrate a few instances being tested on the UI.

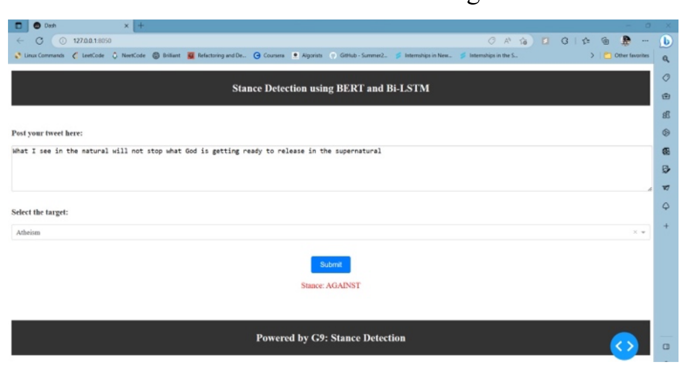


Fig. 9. Prediction for a tweet on the UI (1)

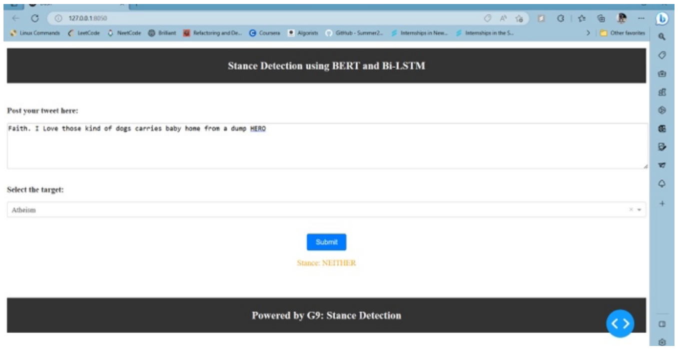


Fig. 10. Prediction for a tweet on the UI (2)

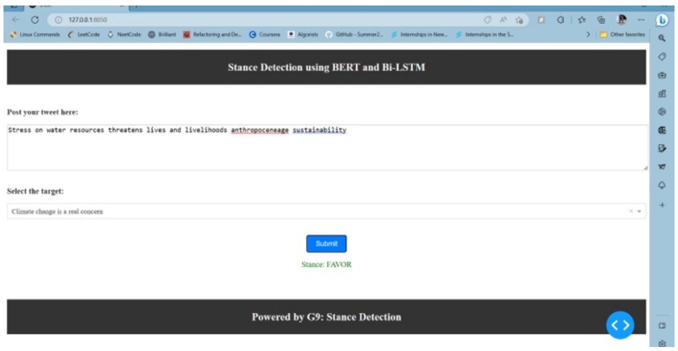


Fig. 11. Prediction for a tweet on the UI (3)

VIII. Division of Work

Table below shows the division of labor and the members that were responsible for each task's completion.

Task	Members Responsible
Data cleaning and preprocessing	Tanuja Renu Sudha
Data Augmentation	Sai Rathnam Pallayam Ramanarasaiah
Web Scraping	Baibhav Phukan
Model Training	Sai Vikhyath Kudhroli, Avish Khosla
Performance Evaluation	Gautham Maraswami
Web Application and UI	Sai Vikhyath Kudhroli, Baibhav Phukan
Final Report	Sai Rathnam Pallayam Ramanarasaiah, Tanuja Renu Sudha, Gautham Maraswami, Avish Khosla

IX. Conclusion

In conclusion, our stance detection project achieved promising results using advanced deep learning techniques such as BERT with SVM and Bi-LSTM models. The accuracy of our model using BERT with SVM was 63%, while Bi-LSTM provided an accuracy of 58%. While these results are encouraging, there is still room for improvement, and further research can be done to improve the accuracy of the model.

One possible reason for not achieving a higher accuracy could be the complexity of the task. Stance detection is a difficult task that requires the model to accurately classify the opinions expressed in the text, which can be subtle and nuanced. In addition, the dataset used for training the model might not be comprehensive enough to capture the diversity of opinions and contexts in real-world scenarios.

Despite these challenges, our model shows promising results, and further improvements can be made by exploring new data augmentation techniques, experimenting with different neural network architectures, and fine-tuning the hyperparameters. With these improvements, our stance detection system can be applied to a wide range of real-world applications, including political analysis, social media monitoring, and market research, to name a few.

X. Future Work

While our stance detection project has shown promising results, there is always room for improvement and further development. One potential area for future work is we can also explore the use of techniques such as attention and explainable AI to make our models more transparent and interpretable. This could help users to better understand why a particular stance was predicted, and to gain insights into the underlying factors that contributed to the prediction.

Another area of future work is to explore additional data augmentation techniques to improve the diversity and comprehensiveness of the dataset. This could involve experimenting with other techniques such as back-translation or using pre-trained language models for data expansion.

In addition, we can also explore the use of techniques such as attention and explainable AI to make our models more transparent and interpretable. This could help users to better understand why a particular stance was predicted, and to gain insights into the underlying factors that contributed to the prediction.

Finally, we could also investigate the application of our stance detection model to other domains such as political debates or news articles, where understanding the different perspectives and stances is critical for accurate analysis and decision-making. Overall, there are many exciting avenues for future work in stance detection, and we believe that this project has laid a solid foundation for further research in this area.

References

[1] Kazi Saidul Hasan and Vincent Ng. 2013. Stance classification of ideological debates: Data, models, features, and constraints.In Proceedings of the International Joint Conference on Natural Language Processing. 1348-1356.J.

[2] Matt Thomas, Bo Pang, and Lillian Lee. 2006. Get out the vote: Determining support or opposition from congressional floor-debate transcripts. In Proceedings of the Conference on Empirical Methods in Natural Language Processing. 327-335.

[3] Akiko Murakami and Rudy Raymond. 2010. Support or oppose?: Classifying positions in online debates from reply activities

[4] and opinion expressions. In Proceedings of the International Conference on Computational Linguistics. 869-875.

[5] Pranav Anand, Marilyn Walker, Rob Abbott, Jean E Fox Tree, Robeson Bowmani, and Michael Minor. 2011. Cats rule and dogs drool!: Classifying stance in online debate. In Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis. 1-9.

[6] Swapna Somasundaran and Janyce Wiebe. 2010. Recognizing stances in ideological on-line debates. In Proceedings of the Workshop on Computational Approaches to Analysis and Generation of Emotion in Text. 116-124

[7] Marilyn A. Walker, Pranav Anand, Robert Abbott, and Ricky Grant. 2012a. Stance classification using dialogic properties of persuasion. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 592-596.

[8] Swapna Somasundaran and Janyce Wiebe. 2009. Recognizing stances in online debates. In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1. 226-234.

[9] Adam Faulkner. 2014. Automated classification of stance in student essays: An approach using stance target information and the Wikipedia link-based measure. In Proceedings of the International Florida Artificial Intelligence Research Society Conference. 174-179

[10] Ashwin Rajadesingan and Huan Liu. 2014. Identifying users with opposing opinions in Twitter debates. In Proceedings of the International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction. 153-160.

[11] Dhanya Sridhar, Lise Getoor, and Marilyn Walker. 2014. Collective stance classification of posts in online debate forums. In Proceedings of the Joint Workshop on Social Dynamics and Personal Attributes in Social Media. 109-117.

[12] Marilyn A. Walker, Pranav Anand, Robert Abbott, and Ricky Grant. 2012a. Stance classification using dialogic properties of persuasion. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 592-596.

[13] Marilyn A. Walker, Pranav Anand, Rob Abbott, Jean E. Fox Tree, Craig Martell, and Joseph King. 2012b. That is your evidence?: Classifying stance in online political debate. Decision Support Systems 53, 4 (2012), 719-729

[14] Amita Misra and Marilyn A. Walker. 2013. Topic independent identification of agreement and disagreement in social media dialogue. In Proceedings of the Conference of the Special Interest Group on Discourse and Dialogue. 41-50.

[15] Pranav Anand, Marilyn Walker, Rob Abbott, Jean E Fox Tree, Robeson Bowmani, and Michael Minor. 2011. Cats rule and dogs drool!: Classifying stance in online debate. In Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis. 1-9.

[16] Akiko Murakami and Rudy Raymond. 2010. Support or oppose?: Classifying positions in online debates from reply activities and opinion expressions. In Proceedings of the International Conference on Computational Linguistics. 869-875.

[17] Swapna Somasundaran and Janyce Wiebe. 2009. Recognizing stances in online debates. In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1. 226-234.