NATIONAL INSTITUTE OF TECHNOLOGY ANDHRA PRADESH

MINI PROJECT

Cross Domain Label Adaptive Stance Detection

Kaushal Tiwari(420245)
Burli Sai Tharun(420118)
Medisetty Sriram(420208)

Department of Computer Science and Engineering

Supervised by
Dr K Himabindu

Table of Contents

List of Figures

List of Tables

Nomenclature

i.e	that is
BERT	Bidirectional Encoder Representations from Transformers,
LEL	Label Embedding Layer
MGTAB	Multi-Relational Graph-Based Twitter Account Detection Benchmark

Abstract

Stance Detection is an NLP task which aims to find the stance attitude of a sentence with respect to a given target, it is one of the most important modern NLP tasks as it can be used to analyse social media data (As seen in the Turkish or Argentinian Elections). Our project report focuses on the various modern methodologies used in Stance detection, which includes Multi Domain Label Adaptive Stance Detection [1] , Wiki Enhanced Stance Detection [2] and User Account Graph Based Stance Detection [3]. The Multi Domain Task is novel due to its idea to assimilate various different domains into one model and thus being able to train the model on a larger and more robust dataset. In order to obtain this goal the model was trained on a total of 16 datasets.

The central aim of the Wiki enhanced stance detection task was to provide the model with some knowledge related to the task, this performed better when the knowledge was given as an embedding rather than a Knowledge Graph. The model trained on various twitter based datasets, but we worked with the vast dataset while training the model.

Finally, rather than focusing on the target and words in the sentence if we are somehow able to include Knowledge about the author of the provided sentence then we could have a user based stance detection idea, this is really helpful in identifying swinging opinions and other ideas. This idea was explored in the MGTAB [3] paper, which also provided a dataset (MGTAB) along with the model.

We were able to implement the wiki-enhanced Stance Detection task which worked on the VAST Dataset and the MGTAB task which was trained on the (MGTAB) Dataset. We are still working with the Multi Domain Datasets in order to achieve satisfactory results for the Multi Domain Task.

Chapter 1

Introduction

With the rise of both the internet and social media, social networks have become a part of people's daily life, Twitter is one of the most used social media site currently and has become a forum for arguements,debates and opinions of various kinds.

Besides the stance of a sentence if seen on a large scale also gives us a peek into certain events an example would be the Turkish,Argentinian and Indian elections.

A lot of people often say "Data is the currency of the next generation " and hence to be able to mine and understand the patterns within the data can be instrumental to take artificial intelligence to the next level.

Stance detection has a wide range of applications, from analyzing political discourse to improving customer engagement in online communities. As the amount of online communication continues to grow, the ability to automatically detect and understand the attitudes and opinions of individuals and groups will become increasingly important for businesses, organizations, and governments alike.Additionally, Graph Neural Networks are a cutting-edge technology that are widely used, and we attempted to cite relevant articles so that we could apply these concepts to NLP jobs as well.

We also tried to use account detection based ideas through graph neural network.

Account detection can be performed using a variety of techniques, including text analysis, social network analysis, and metadata analysis. These techniques can help to identify patterns in the language, behavior, and network connections of social media accounts, which can be used to make inferences about the author's stance and the context in which their statements were made.

This can then be used to generate a larger version of the dataset, where we could identify the stance of an author with respect to past tweets, this idea can draw similarity for Collaborative Filtering where we try to find unknown or unfamiliar values in the dataset.

Chapter 2

Literature review

Stance Detection is generally seen as a very important NLP task as it can be used to identify user sentiment along with other attributes and can grant us a peek in the future.

Stance can be useful in understanding the sentiment of the people with respect to a certain target. Our main aim during this review was to investigate these ideas and try to present a hollistic view of the stance detection task.

2.1 Stance Detection

The main aim of stance detection is to predict the sentiment of a comment with respect to a given target or the sentence's subject.

Prior works on Stance detection have been by various authors on topics such as arguement mining [4] and others on debates such as [5] , have been widely applauded and have shown the usability of stance in the modern world of social media.

The focus mostly has been on features, since they could be used to find the embeddings of a sentence and thus provide a soft metric for the similarity between the target and the given sentence.

Here we try to focus on three approaches, one based on multi domain task learning approaches [1], the second based on knowledge enhancement [2] and finally one based on graph based approaches [6] [3].

Given below is the amalgamation of the ideas that are currently being used in the stance detection task, and could be enhanced further to get better results.

2.2 Multi Domain Learning

Multi-domain learning, also known as cross-domain learning or domain adaptation, is a subfield of machine learning that deals with the transfer of knowledge across different domains or datasets. It was discussed by [7] for Natural Language Processing Tasks and various other papers such as [1],[8] as well. In many real-world scenarios, models trained on one domain often struggle to generalize well to new, unseen domains due to differences in data distribution and characteristics. Multi-domain learning aims to address this challenge by enabling models to leverage knowledge from multiple domains to improve their performance on target domains.

Recently domain adaptation was applied on pre trained transformers by [9] and was used by [10] who investigated an unsupervised multi-source approach with Mixture of Experts and domain adversarial training.

2.3 Feature Based Methods for Stance Detection

A popular method for determining the mood or opinion expressed in text towards a specific topic or entity is to use feature-based algorithms for stance detection. In this method, the text's linguistic, syntactic, and semantic traits that could point to a specific viewpoint or attitude are retrieved as features.

Previous research works [11] used machine learning algorithms and deep learning methods such as Support Vector Machines (SVM), Recurrent Neural Networks (RNNs) [12], and Convolutional Neural Networks (CNNs) to automatically learn latent features from a large amount of raw data. Several recent works focused on the use of bidirectional encoder representations from transformers (BERT) [13] on stance detection. [14] explored stance detection based on transfer learning, and [7] explored BERT-based data augmentation models

2.4 Label Embeddings

Label embeddings can capture, in an unsupervised fashion, the complex relations between target labels for multiple datasets or tasks.

Label embeddings refer to the representation of labels or classes in a machine learning or deep learning model as continuous vectors in a low-dimensional embedding space. These embeddings capture the semantic relationships and similarities between different labels, allowing for more effective and expressive modeling of categorical information.

[15] used label emebeddings for text classification tasks along with Convolutional Neural Networks whereas it was used alongside Transformers for similar tasks by [16].

2.5 Wikipedia Encodings.

[17] uses Wikipedia pages of a news medium as an additional source of information to predict the factuality and bias of the medium. However, they use static pretrained BERT [13], embeddings of the Wikipedia pages without finetuning, failing to align the pretrained embeddings to the domain of the target task.However the authors only consider the promote/suppress relations between the texts and Wikipedia, which require a large amount of manual annotations to extract; in addition, a substantial amount of knowledge that is not captured by such relations is ignored; in contrast, WS-BERT utilizes the original Wikipedia textual knowledge and does not proactively exclude any information.

[18] utilises commonsense knowledge from a knowledge graph by extracting the two-hop paths between entities in the targets and in the documents; however, the existence of such paths do not always hold true and we found that a well-finetuned BERT without external knowledge can achieve performance comparable with it, as shown in [2]

2.6 Graph Based Stance Detection

Graph Neural Networks (GNNs) are powerful deep learning models designed to operate on graph-structured data. By leveraging message passing and graph-level aggregation mechanisms, GNNs enable effective learning and inference on graphs, leading to state-of-the-art performance on various graph-related tasks. They are also being introduced to NLP tasks as well in papers like [19],[3] etc. Most studies on stance detection have focused on text-based features [12]. However, recent work has demonstrated the effectiveness of using user network graphs as features [20]. Graph Neural Networks (GNNs) [21] have become the preferred model for account detection due to their ability to process graph information. [19] first achieved stance and rumor detection using a GNN-based architecture that efficiently captured user interaction characteristics. However, the lack of graph structure in existing stance detection datasets hinders the development of graph-based detection methods

The paper by [3] provides a deeper insight on the usage of Graphs in order to find user related information for both Bot and Stance Detection.

Chapter 3

Methodology

3.1 Problem Statement

The main aim of the project was to try and learn about various stance detection techniques and then try to assimilate them so that we can find a better model than the current state of the art models. We found three very important papers [3] [1] [2] that helped us understand different ways to approach the same problem.

The mathematical definition of the task can be defined as Given a set of target labels Y such that $Y = [y_1, y_2, ...y_n]$ a sentence S and a target t we need to find a discrete value function f such that

$$f : (S, t) - > Y$$

The function f takes a Sentence and a Target as an input and returns the most probable label that could be assigned to it, also it is not necessary that the target needs to be present in the sentence. We also wanted to be able to implement different kinds of stance detection models, test them on various benchmark datasets and if possible make any improvements to the current state of the art models. The main aim of our project was to look into Multi Domain Learning [1] , Knowledge Enhanced Stance Detection [2] and finally a Graph Based Approach to Stance Detection [3].

The main method of our project was in dealing with data(quantitative), however when we used datasets such as MGTAB we also added qualitative research approaches to our study.

3.2 Dataset Processing

The section provided below deals with the various kinds of datasets that are currently

3.2.1 Collecting Datasets

The main way to collect data was through online forums and github, most of the data we used is open source and available online. The dataset for vast which was used in [2] was provided by the author themselves along with a general way for getting the tweets in the dataset.

The tweets could either be requested from the author or we could fetch them using an api like tweepy.

We also collected the dataset for the MGTAB task [3] from the github repo of the author.

The major problem was in collecting the datasets for the cross domain task was we had to deal wuth 16 datasets, for this we tried to work with smaller, more refined versions of the dataset.

In order to obtain datasets we also sent a request to the authors of the paper [8] and they sent us the pretrained BERT embeddings of the model. We used these embeddings as a form of oracle against which we would test other models.

3.2.2 Reducing/Preprocessing Datasets

The datasets that we obtained through the internet were not suitable for execution on a local machine. This made us go with reducing the size of the obtained datasets, also while dealing with the vast stance detection dataset we obtained a wiki model of similar type this helped us in reducing the time required to train the model for the wiki embeddings.

3.2.3 Information Regarding the datasets.

VAST

The Varied Stance Topics dataset consists of topic–comment pairs from the The New York Times Room for Debate section. The dataset covers a large variety of topics in order to facilitate zero-shot learning on new unseen topics.

MGTAB

MGTAB [3] is the first standardized graph-based benchmark for stance and bot detection. MGTAB contains 10,199 expert-annotated users and 7 types of relationships, ensuring high-quality annotation and diversified relations.

semeval2016t6

The SemEval-2016 Task 6 dataset provides tweet–target pairs for 5 targets including Atheism, Feminist Movement, and Climate Change.

semeval2019t7

The SemEval-2019 Task 9 dataset aims to model authors' stance towards a particular rumour. It provides annotated tweets supporting, denying, querying, or commenting on the rumour

3.3 Approach and Modelling for Different Embeddings

3.3.1 Approach

The central approach while analysing the data, was to train the model in such a way that it would be able to understand the intricate relations between the data features while also respecting the time and space constraints of the machine.

3.3.2 Word Embeddings

Though their are various approaches to obtain word embeddings, we decided to focus on using BERT for all the models as it would provide uniformity across all the experiments, without being biased towards any of them.

The label embeddings layer (LEL) for the multi domain learning task [1] were also obtained through their BERT embeddings rather than training them. This reduced the efficiency of the model but also decreased an additional layer of parameters which were their in LEL.

3.3.3 Wiki Embeddings

The Wiki Embeddings were obtained by the wiki models, these were provided in a pickle file by the authors [2] and can be obtained through the Github repository of the paper.

The embeddings are useful to add knowledge to the stance detection models, we also hope to use them along with Graph Neural Networks in order to provide more information about the topic to the network.

3.3.4 Graph Embeddings

In the [3], the author proposes two approaches to create the Multi-Relational Graph, implicit and explicit respectively. Most of our work was done through the explicit approach in order to create the graph, we are still trying to replicate the implicit technique and would report the results whenever they are obtained.

3.4 Cross Domain Label Adaptive Stance Detection [1]

The main aim of this task was to create a multi domain model trained on various datasets for Stance Detection. The model also used Domain Adversarial Learning (3.4.4) in order to train a Domain-Invariate Model, The model used a Label Embedding Layer to map the in-domain labels to the out-domain labels. Our implementation of this was rather simple, as we did not have the raw computation power required to deal with all the datasets available , we worked with smaller datasets in order to achieve the required objectives.

This meant that the obtained results were inferior to the actual values however we did get a basic idea of the functioning of the task.

3.4.1 Main aim

The paper presented by [1] trained the model on 16 datasets with different domain labels, the main aim of this paper was to learn stance based tasks independent of the label

3.4.2 Mixture of Experts

The paper proposes a Mixture of Expert in order to find the probability vectors for the given sentence with respect to the target.

The MOE is an ensemble model, and is useful for multi domain adaptation. It was introduced to the large transformers by [22]. The main aim is to fine-tune k different transformers along with one global model and then predict the final output as the average over all these models. The equation for the mixture of experts model can be shown as

$$p_A(x, \bar{K}) = \frac{1}{|\bar{K}|+1} \sum_{k \in \bar{K}} p_k(x) + p_g(x) \tag{3.1}$$

3.4.3 MOE with Label Embeddings

The author proposed changes to the MOE architecture in order to allow the possibilty for learning cross domain labels.

Here the primary focus is to have the Label Embedding i.e treat the labels as embeddings themselves this allows us to do mathematical operations on the labels themselves. The sentences are sent through their [CLS] tokens which is found through a common encoder block. This helps in reducing the no of parameters in the model. The input is embedded through the ROBERta Model, which is then passed to the domain specific transformers, this is finally sent to the label embedding layer which is able to map them onto the required outputs.

The obtained vectors are then passed through a softmax or linear layer in order to get the probability output.

3.4.4 Domain Adversarial Classification

As the dataset is from multiple domains, we need to train the model to be domain invariant for both the source and the target. This leads to the idea of Domain Adversarial Training as it forces the model to learn domain-invariant representations, both for the source and for the target domains. The latter is done with an adversarial loss function, given in equation 3.2 .

The paper by [22] also tries to use this technique. The main objective is to minimize the task objective f_g and maximize the domain confusion f_d

$$\mathcal{L}_D = \max_{\theta_D} \min_{\theta_G} -d \log f_d \left(f_g(x) \right) \tag{3.2}$$

3.4.5 Label Embedding

In multi-task learning, each task is given a label which can be predicted in a joint label distribution on the union of all labels across all domains this however may not be the best practice as not all labels are orthogonal. Hence in order to keep the correlation between the labels we try to learn a label embedding L such that

$$p = softmax(Lh) \tag{3.3}$$

where L is the shared embedding for all the datasets and p is the final probabilty vector.

3.4.6 Label Adaptive Prediction

The main aim of this task was to predict labels independent of the training dataset, we trained the dataset with domain adversarial training, similarly for predictions we go with 2 major ways.

Hard Mappings :- Here we supervise the labels and put them in the corresponding blocks where they belong. The table 3.1 shows a similar distribution.

Group	Assigned Labels
Positive	argmin-argument for, emergent-for, fnc1-agree, iac-pro, mtsd-favor
Negative	arc-disagree, argmin-argument against, emergent-against, fnc1-disagree, iac1-anti, ibmcs-con, mtsd-against, perspectrum-undermine, poldeb-against,
Discuss	arc-discuss, emergent-observing, fnc1-discuss, rumor-question, semeval2019t7-query, wtwt-comment

Table 3.1: Hard Mapping of Labels [1]

Soft Mappings :- Here the set of labels are predicted by using the similarity between the label embeddings seen.

More precisely, we measure the similarity between the names of the labels across datasets. This is an intuitive approach for finding a matching label without further context, e.g., for is probably close to agree, and refute is close to against. In particular, given a set of out-of-domain target labels $Y^\tau \in \{y_1^\tau, \ldots, y_k^\tau\}$, and a set of predictions from in-domain labels $P_\delta \in \left\{p_1^\delta, \ldots, p_m^\delta\right\}, p_i^\delta \in \left\{y_1^\delta, \ldots, y_j^\delta\right\}$, we select the label from Y_τ with the highest cosine similarity to the predicted label p_i^δ :

$$p_i^\tau = \underset{y^\tau \in Y^\tau}{\arg\max} \cos\left(y^\tau, p_i^\delta\right)$$

where k is the number of out-of-domain labels, m the number of out-of-domain examples, and j the number of in-domain labels.

3.4.7 Training

The model trains with the loss function

$$\mathcal{L}_s = \frac{1}{N} \sum_i y_i \log p_X\left(x, S'\right) \tag{3.4}$$

$$\mathcal{L}_t = \frac{1}{N} \sum_i y_i \log p_t(x) \tag{3.5}$$

$$\mathcal{L} = \lambda \mathcal{L}_s + (1-\lambda)\mathcal{L}_t + \gamma \mathcal{L}_D \tag{3.6}$$

We sum the source-domain loss $\mathcal{L}_s$ with the meta-target loss from the domain expert subnetwork $\mathcal{L}_t$, where the contribution of each is balanced by a single hyper-parameter λ , set to 0.5. Next, we add the domain adversarial loss $\mathcal{L}_d$, and we multiply it by a weighting factor γ , which is set to a small positive number to prevent this regulariser from dominating the overall loss

3.4.8 Architecture

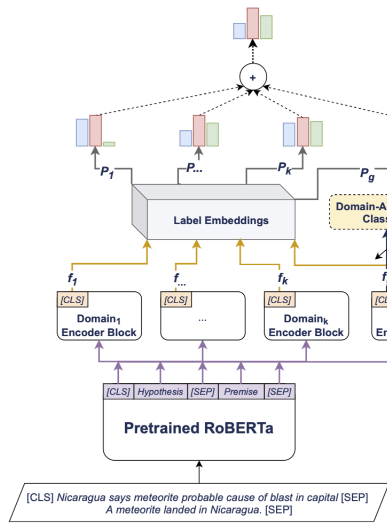

Figure 3.1: Architecture of the Cross Domain Label Adaptive Stance Detection [1]

The authors propose a novel end-to-end framework for cross-domain label-adaptive stance detection. The architecture is based on input representations from a pre-trained language model, adapted to source domains using MOE and domain adversarial training . We further use self-adaptive output representations obtained via label embeddings, and domain adversarial training 3.4.4. The output representations are self-adaptive and hence can be used for unsupervised alignment between seen and unseen target labels for out-of-domain datasets.

3.4.9 Algorithm

The given algorithm, works for a given input, we can add a for loop that would loop over all the inputs in the datasets.

Input : A sentence of the form [CLS] Sentence [SEP] Target
Output : A probability vector on the different labels defined.
$Model \leftarrow$ The Model Trained on the given Datasets.
$inputEmbed \leftarrow$ Embeddings of the Input from ROBERTa Model
$outputEmbed \leftarrow$ Output over all the domain Encoders.
$O \leftarrow [\ ]$
$lr \leftarrow$ Learning Rate
$\gamma \leftarrow$ hyperparameter

O is a vector of dimension k where k is the number of Labels
$\lambda \leftarrow$ hyperparameter
$Loss \leftarrow \infty$
$threshold \leftarrow$ Some very small threshold Value
while $Loss \geq threshold$ **do**
    for k in domains **do**
        $o \leftarrow$ Encoding for Input from Domain Encoder k
        $O \leftarrow O + o$
    end for
    $O \leftarrow softmax\ (O)$
    $L_t \leftarrow$ CrossEntropyLoss(O,Original Value)
    $L_s \leftarrow$ Source Adaptation Loss
    $d \leftarrow$ Predicted Domain
    $L_d \leftarrow$ Domain Adversarial Loss($Model, d$) (3.2)
    $Loss \leftarrow \lambda \cdot L_s + (1 - \lambda) \cdot L_t + \gamma \cdot L_d$
    $Model \leftarrow GradientDescent(Model, Loss)$
end while

3.5 Wiki Enhanced Stance Detection [2]

3.5.1 Problem Definition

Let $D = \{(x_i = (d_i, t_i, w_i), y_i)\}_{i=1}^{N}$ denote N examples, with input x_i consisting of a document d_i, target t_i, and Wikipedia text w_i about the target, and a stance label $y_i \in \{$ favor, against, neutral $\}$ as output. The goal is to infer y_i given x_i.

3.5.2 Approach

The main aim here was to train a model that did not look at the target as a word but rather the meaning of the target, this meaning was achieved through the Wikipedia articles on the target.This would help the model in working with the semantic meaning behind the target and the sentence rather than just the syntactic meaning behind it. The central ideas behind the model have been discussed in 2.5, we tried to implement the model in a similar fashion to the authors implementation while reducing the dataset size and the number of epochs. The author also provided a pickle file which was trained on the wikipedia articles of the various targets. The implementation was quite simple as it used the ideas of transfer learning, where we finetuned the model on the vast or other datasets. The pretrained models used were BERT(for Formal Training Texts) and BERT-Twitter(for Informal Training Texts). The figure given below provides a basic insight to the Wiki Enhanced Stance Detection Task.

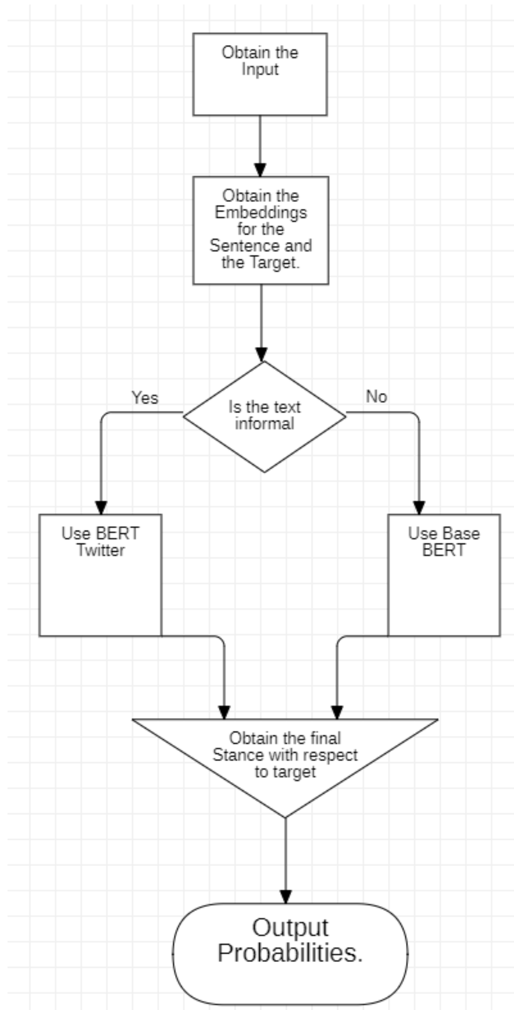

Figure 3.2: Wiki Enhanced Stance Detection Task. [2]

While dealing with stance detection, rather than having just the word of the target it would be nice if we could imbue it with some knowledge of the target.

Hence we try to add Wikipedia in order to give knowledge to the stance detection model.

In [2] the author tries to utilize background from wikipedia articles related to the topic in order to provide deeper understanding to the model.

3.5.3 Encoding Wikipedia Knowledge[2]

The idea of encoding knowledge was proposed by [18] in 2021, however this used the knowledge graph created from the relations between wikipedia articles.

The authors of [2] however propose that we could just take in the text in its raw form and still get similar or better results.

To do this we obtain the embedding of the article with a model like BERT or T5 , this embedding can then be sent with the original sentence and the target in order to predict the final output. Depending on the, textual style (formal vs. informal) of the documents, the author introduces two variants of WS-BERT, namely WS-BERT-Single, for dealing with formal documents, and WS-BERT-Dual, for dealing with informal documents. (WS stands for Wikipedia Stance) a) WS-BERT Single b) WS-BERT Dual

3.5.4 WS-BERT Single

The BERT-Single is best used on formal style documents which are similar to wikipedia style articles.

THe main aim is to encode the Document d, target t, Wiki Article w. Since BERT was originally created to deal with atmost two sequences, we put the document and target together creating the input of the format "[CLS] d [SEP] t [SEP]". The pooled output can then be sent to the final softmax/ feed forward neural network in order to classify the stance.

3.5.5 WS-BERT Dual

Social Media is used by many people to argue, give their opinion and is hence a valuable resource for any stance detection task, however the encodings need not be similar, thus it would be incorrect to use the same BERT model for encoding both the article and the sentence as their syntax are generally different.

To deal with this the author proposes a dual BERT model where the encodings for the sentence would be created by the BERT-Tweet model [23] The document pair and the target are then encoded through the BERT-Tweet Model whereas the wikipedia articles are encoded on the Basic BERT model.

The outputs are then concatenated in order to get the final predictions. The loss function used was the cross entropy function.

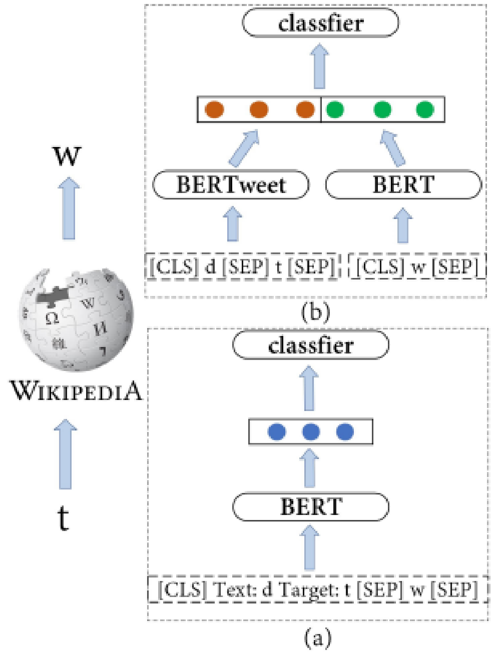

Figure 3.3: Architecture of a)WS-BERT Single b) WS-BERT Dual [2]

3.6 MGTAB [3]

The final problem that we dealt was that of Stance detection using user information, the central idea was to be able to create a relationship graph between different users based on these ideas. The Graph Embeddings were discussed in 3.3.4. The central idea was to shift from a feature based approach to a graph based approach as this helped in using unsupervised learning on the provided dataset, which meant a shift from data annotated by experts to data that could be obtained from a social media website like twitter, hence we considered this idea to be very important and focused on it. The paper also dealt with bot detection, which is a huge field as well but we looked only on the stance detection based tasks.

Our main aim was to be able to understand the ideas proposed by the author , while also trying to explore newer ideas within the models.

3.6.1 Main Problem

It is very difficult to collect the target of a sentence as is requires a huge number of experts to complete the task.

However when dealing with such problems it is easier to find other ideas within the data and cluster them accordingly this helps in making the data more understable while adding newer dimensions to the model.

3.6.2 Account Detection

Account detection is used in various papers such as [24]. The central idea is to be able to train the model so that it is able to cluster the users according to their descriptions,priorities and other features.

Most papers that deal with stance detection refer to feature extraction which can then be used on various models like BERT or algorithms like SVM,Naive Bayes etc in order to get the required output.

Graph detection is a modern idea that is currently being used in many NLP and concurrent tasks. However the unavailabilty of correctly annotated datasets is an issue as it could lead to troubles training the dataset. The author along with proposing the model also provides us with a fully annotated dataset on which we can train our model.

Given below are a few ideas that the author presents.

3.6.3 Information Gain

In information theory and machine learning, information gain is a synonym for Kullback–Leibler divergence; the amount of information gained about a random variable or signal from observing another random variable.

We use this idea in order to find the correct features for detection which would optimise the accuracy / prediction of the model.

Given below are the equations that are generally used for information gain.

Use Y to denote the user's category, $H(Y)$ to represent the entropy of Y, and y is the value of

$Y, y \in \{y_1, y_2, \ldots, y_K\}$. In stance detection, K is 3 , and in bot detection, K is 2 .

$$H(Y) = -\sum_{k=1}^{K} p\left(y_k\right) \log_2 p\left(y_k\right)$$

$H(Y \mid X)$ denotes $H(Y)$ when the feature X is given and it can be computed by:

$$H(Y \mid X) = -\sum_{x \in \Phi} p_x \sum_{k=1}^{K} p\left(y_k \mid x\right) \log_2 p\left(y_i \mid x\right)$$

where x is the value of $X, x \in \Phi$. The $IG(X; Y)$ indicates that the category information increases (uncertainty decreases) after Y gets feature X :

$$IG(X; Y) = H(Y) - H(Y \mid X)$$

3.6.4 Dataset Feature Extraction

The main aim of this task is to find the correct feature representation in the Graph.

A user feature is the concatenation of the properties of the user and the embeddings of the tweet they make.

User Feature Extraction

User properties are obtained through the features from information gain, to obtain the representation of numerical feature r_{num} .

The selected boolean features are numericalized, where True and False are replaced with 1 and 0 , respectively, to obtain the representation of boolean feature r_{bool} . The representation of user property features is obtained by concatenating r_{num} and r_{bool} , $r_{\text{prop}} = [r_{\text{num}} \| r_{\text{bool}}]$.

Tweet features extraction

This is done through the embeddings obtained by BERT, or through other models. Generally a diverse model would be useful as it would be able to capture most data. This can be shown as r_{tweet} which can be then concatenated to the prior values obtained in order to gain the values.

3.6.5 Construction of the Graph

The author proposes to construct a User-User graph in order to work with the relationships among various users. There are two major types of constructions

Explicit

Explicit relations such as follower,friends,retweets are then constructed in order to derive their relationships. The edges are all directed except for the ones made by similar url or hashtags as shown in table 3.2

Table 3.2: Relations in the MGTAB Graph

Relation	Direction		Description
	Source	Target	
follower	user A	user B	user A is followed by user B
friend	user A	user B	user A follows user B
mention	user A	user B	user A mentions user B in tweets
reply	user A	user B	user A replies to tweet of user B
quote	user A	user B	user A quotes tweet of user B
URL	Undirected	Undirected	user A and user B have the same URL
hashtag	Undirected	Undirecteed	user A and user B have the same hashtag

Implicit

Two implicit relationships between the users was also found on the basis of url and hashtag co-occurrence, specifically between nodes v_i and v_j is given by the following equation.

$$\mathrm{W}\left(v_i, v_j\right)=\frac{1}{\left|\Psi_{\{i,j\}}\right|} \sum_{e_k \in \Psi_{\{i,j\}}} \log \frac{p\left(v_i, e_k\right) p\left(v_j, e_k\right)}{p\left(e_k\right)^2} \tag{3.7}$$

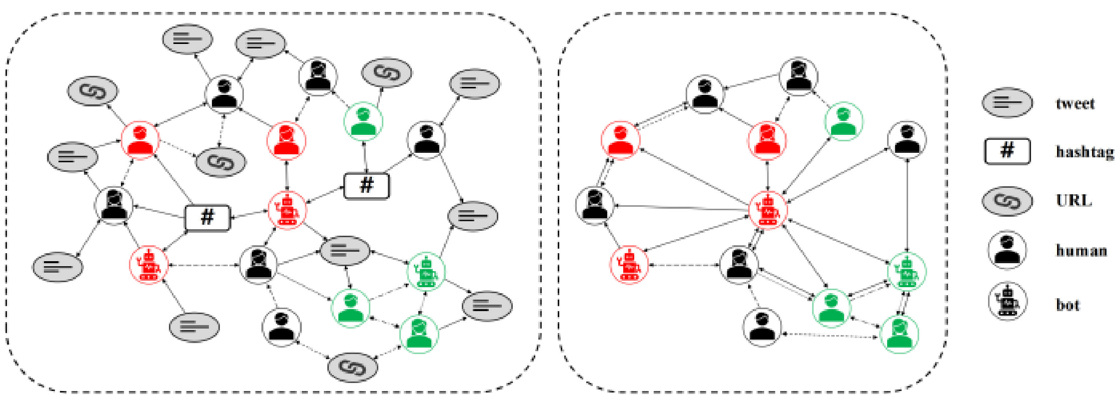

Figure 3.4: Representation of the user graph based on the relationships.

Black Red and Green denote Neutral,Against and Support. [3]

The central approach of the paper was to use Graph Neural Networks in order to work with Natural Language Processing Tasks. We also explored this frontier as well and the results that we reproduced were satisfactory.

3.7 Experimentation

The central theme of experimenting was to test all the three models against some standard dataset however as they were trained on different datasets it is currently beyond the scope of our research, we hope to solve this problem in the future.

We tested the models provided for the given datasets, on a local machine however the constraints of both time and space bounded our exploration. In order to deal with this we reduced the size of the documents and also the number of epochs for training , this led to inferior results than the ones proposed in the models.

Midway through the allotted time,The central aim of our project shifted from trying to deal with just the first paper to trying and comparing the results with other papers.

The datasets on which the models were trained upon was the VAST dataset for the WIKI-Enhanced Stance Detection Task and the MGTAB for the corresponding task. We also hope to use twitter based datasets along with there Wikipedia articles in the future in order to deal with knowledge adaptation.

We also looked into the SEM-EVAL datasets as they were used as benchmark datasets by the author of [3].

Dataset	Samples	Annotation		Graph
		Instance	Expert-annotated	
SemEval-2016 T6	4,870	tweet	X	X
SemEval-2019 T7	7,730	tweet	X	X
COVIDLies	8,937	tweet	✓	X
COVID-19-Stance	7,122	tweet	X	X
COVMis-Stance	2,631	tweet	X	X
WT-WT	51,284	tweet		X
P-STANCE	21,574	tweet	X	X
Stance Dataset	4,870	tweeter	X	X
MGTAB (ours)	410,199	tweeter	✓	✓

Table 3.3: Comparison of the MGTAB against other stance based datasets [3]

The table above represents the comparison of the MGTAB dataset along with other datasets, Most of the current dataset are annotated through crowdsourcing which often makes them have low annotation quality and bad MGTAB is the first stance detection dataset with user network graphs. The large-scale and high-quality annotation of MGTAB will facilitate the development of user stance detection. Additionally, MGTAB provides opportunities for studying graph-based approaches in stance detection.

Chapter 4

Results

As mentioned in 3.7 we were able to replicate the results of the papers [3] [2] given below are the results and a few insights into the task.

The main aim of our project was to replicate the models proposed by the various authors [3] [1] [2], we also tried to find some additional theoretical models that could lead to better results.

The collection of the datasets has been discussed in the section 3.2, the results pertaining to the replication task are mentioned below.

4.1 Results for the wiki enhanced stance detection task

We tried to replicate the stance detection task using wikipedia embeddings given below are the results for the task. The dataset that we used for training the model this was the vast dataset. We are also trying to train the model using other twitter based datasets.

Note

We tried to reduce the size of the dataset, we also reduced the no of epochs, this lead to inferior results.

Metric	Our Result	Standard Result [2]
F1	0.556	0.745
F1 favor	0.494	0.763
F1 against	0.623	0.778

Table 4.1: Results for the Wiki Enhanced Stance Detection task [2]

As mentioned in the table 4.1 we can see that the model performs better when dealing with negative targets, we attribute this to the fact that negative sentiments are more easily sensed by the polarity of the sentence whereas it generally is hard to pick up the subtle hints of appreciation in sentences that are favorable.

4.2 Results for the MGTAB Task

We tried to replicate the MGTAB task, the source for the datasets are mentioned in the section 3.2

The results are mentioned in the table 4.2

Metric	Our Result	Standard Result [3]
Accuracy	81.90	87.8
Precision	81.38	
Recall	81.54	
F1	81.34	86.9

Table 4.2: Results for the MGTAB task

4.3 General Information regarding the results

The results obtained are not final and can still be improved using various techniques like increasing the data size, better representation of the word embeddings, adding more layers to the model.

4.3.1 Key Findings

Multi Domain Learning

The Multi Domain Learning [8] [1] , is a good way to collect more datasets and train better models, this new trained model could be used as a base on which we could build further models.

Further more, the label independence of the model makes it a key element in trying to apply unsupervised approaches to the Stance Detection Task.

Wiki Enhanced Stance Detection

Adding Knowledge to a Stance Detection Model was an idea that was provided to us by our Mentor, and we found various contemporary papers related to the idea as well.

The idea of adding knowledge to a model can be done either through knowledge graphs or through raw text. In their study the authors of the paper [2] found that raw text gave a better accuracy than knowledge graphs.

MGTAB

We found out that Graph Based Models generally tend to perform better than Feature Based Models, which can be seen in our results as well. Moreover using Graph Convolutional Networks, we are able to apply the power of Convolution to Graphical Models which helps increase the efficiency and accuracy of the said model. The key part that we would like to realize is that would it be possible to create a transfer learning paradigm on the top of these graph based neural networks.

Chapter 5

Conclusion

In this project we were able to implement and test various stance detection techniques for multi domain datasets, we also used BERT embeddings along with preprocessing the input data.

The model evaluation was done using F1,Precision,Recall and Accuracy but we also tried to evaluate the potential of the models i.e how further could they be improved and in this regard we found the Graph Based Models to be the best approach going forward.

During the alloted time period we dealt with transfer learning and also many different methods for dealing with seq2label tasks.

We hope to improve the current state of the art models under the guidance of our mentor.

We also hope to use graphical neural network in our further ideas.

References

[1] Momchil Hardalov, Arnav Arora, Preslav Nakov, and Isabelle Augenstein. Cross-domain label-adaptive stance detection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9011–9028, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[2] Zihao He, Negar Mokhberian, and Kristina Lerman. Infusing knowledge from Wikipedia to enhance stance detection. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 71–77, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[3] Shi Shuhao, Qiao Kai, Chen Jian, Yang Shuai, Yang Jie, Song Baojie, Wang Linyuan, and Bin Yan. Mgtab: A multi-relational graph-based twitter account detection benchmark.

[4] Filip Boltužić and Jan Šnajder. Back up your stance: Recognizing arguments in online discussions. In *Proceedings of the First Workshop on Argumentation Mining*, pages 49–58, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[5] Swapna Somasundaran and Janyce Wiebe. Recognizing stances in ideological on-line debates. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 116–124, Los Angeles, CA, June 2010. Association for Computational Linguistics.

[6] Zhang Chong, Zhao Zhenkun, Peng Xingyu, and Xu Ke. Doubleh: Twitter user stance detection via bipartite graph neural networks.

[7] Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496, Florence, Italy, July 2019. Association for Computational Linguistics.

[8] Schiller Benjamin, Daxenberger Johannes, and Gurevych Iryna. Mstance detection benchmark: How robust is your stance detection?

[9] Chen Lin, Steven Bethard, Dmitriy Dligach, Farig Sadeque, Guergana Savova, and Timothy A Miller. Does BERT need domain adaptation for clinical negation detection? volume 27, pages 584–591, 02 2020.

[10] Dustin Wright and Isabelle Augenstein. Transformer based multi-source domain adaptation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7963–7974, Online, November 2020. Association for Computational Linguistics.

[11] Wan Wei, Xiao Zhang, Xuqin Liu, Wei Chen, and Tengjiao Wang. pkudblab at SemEval-2016 task 6 : A specific convolutional neural network system for effective stance detection. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 384–388, San Diego, California, June 2016. Association for Computational Linguistics.

[12] Guido Zarrella and Amy Marsh. MITRE at SemEval-2016 task 6: Transfer learning for stance detection. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 458–463, San Diego, California, June 2016. Association for Computational Linguistics.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[14] Anurag Roy, Shalmoli Ghosh, Kripabandhu Ghosh, and Saptarshi Ghosh. An unsupervised normalization algorithm for noisy text: A case study for information retrieval and stance detection. *ACM J. Data Inf. Qual.*, 13(3):17:1–17:25, 2021.

[15] Honglun Zhang, Liqiang Xiao, Wenqing Chen, Yongkun Wang, and Yaohui Jin. Multi-task label embedding for text classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4545–4553, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[16] Wei-Cheng Chang, Hsiang-Fu Yu, Kai Zhong, Yiming Yang, and Inderjit S. Dhillon. Taming pretrained transformers for extreme multi-label text classification. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining*, KDD '20, page 3163–3171, New York, NY, USA, 2020. Association for Computing Machinery.

[17] Ramy Baly, Georgi Karadzhov, Dimitar Alexandrov, James Glass, and Preslav Nakov. Predicting factuality of reporting and bias of news media sources. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3528–3539, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[18] Rui Liu, Zheng Lin, Yutong Tan, and Weiping Wang. Enhancing zero-shot and few-shot stance detection with commonsense knowledge graph. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3152–3157, Online, August 2021. Association for Computational Linguistics.

[19] Chen Li, Hao Peng, Jianxin Li, Lichao Sun, Lingjuan Lyu, Lihong Wang, Philip S. Yu, and Lifang He. Joint stance and rumor detection in hierarchical heterogeneous graph. *IEEE Transactions on Neural Networks and Learning Systems*, 33(6):2530–2542, 2022.

[20] Younes Samih and Kareem Darwish. A few topical tweets are enough for effective user stance detection. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2637–2646, Online, April 2021. Association for Computational Linguistics.

[21] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio', and Yoshua Bengio. Graph attention networks. *ArXiv*, abs/1710.10903, 2017.

[22] Dustin Wright and Isabelle Augenstein. Claim check-worthiness detection as positive unlabelled learning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 476–488, Online, November 2020. Association for Computational Linguistics.

[23] Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. BERTweet: A pre-trained language model for English tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14, Online, October 2020. Association for Computational Linguistics.

[24] Chen Li, Hao Peng, Jianxin Li, Lichao Sun, Lingjuan Lyu, Lihong Wang, Philip S. Yu, and Lifang He. Joint stance and rumor detection in hierarchical heterogeneous graph. volume 33, pages 2530–2542, 2022.