Universiteit Leiden

Master Computer Science

Deep learning for stance detection in the news domain

Name:	Xiao Zhang
Student ID:	s2696959
Date:	[24/07/2022]
Specialisation:	Artificial Intelligence
1st supervisor:	Suzan Verberne
2nd supervisor:	Johan Bos

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

Stance detection (SD) determines whether a piece of text is in favor, against, neutral, or unrelated to a specific target. Recently, SD has been applied in a variety of fields (social media, news, online debates, etc.) with favorable practical results. In the news domain, the application of SD assists the news recommendation system in increasing the diversity of news displayed to the public. Despite the significant progress in SD models, previous work has mostly concentrated on single semantic model. For instance, BertEmb, the most advanced model for the STANDER dataset, mainly employs sentence embeddings and simple neural layers. Therefore, the performance achieved by this model has not been close to the upper bound. In this thesis, we propose several methods for fusing different models and a syntactic model to seek better model performance.

Our first model is CLS-transfer BERT, which reuses the CLS token of a BERT model fine-tuned by another task. In practice, we create two new tasks to yield transferable CLS tokens. Our second model is CLS-concat BERT, which combines the BertEmb and BERT by concatenating the CLS embedding of BERT and the embeddings of news sentences. Our third model, Dependency-Based Graph Convolutional Network (DB-GCN), takes the syntactic structure of sentences into account. Subsequently, we tweak the CLS-concat BERT slightly and try to concatenate the node embeddings of DB-GCN and CLS embedding for possible improvements.

We use STANDER as the experiment dataset for evaluating our models. It provides 3,291 pieces of news with expert-annotated stance labels (Favor, Against, Neutral, and Unrelated). To evaluate the proposed models, we use the BertEmb and the popular pre-trained model BERT as two baselines.

According to the average F1 scores of multiple experiments, CLS-transfer BERT does not improve the performance of BERT. We discover that random CLS embedding still yields similar results, implying that the CLS token lacks sufficient stance information for further fine-tuning. Furthermore, we visualize twenty samples that are trained with different CLS embeddings to identify that this type of transfer is ineffective. On the other hand, our CLS-concat BERT overperforms BertEmb by 12.7% points and BERT by 1.4% points, demonstrating that the combination of information at different levels is useful for stance detection. Furthermore, since only syntactic information is considered, the F1 score of the DB-GCN model is lower than the baseline (about 4% points lower than BertEmb) and the concatenation between the node embeddings and the CLS embeddings is shown to provide no improvement.

Keywords. Stance Detection; Transfer Learning; Transformer models; Dependency Parsing; Graph models

Contents

Chapter 1

Introduction

1.1 Problem Statement

News, as a special type of information with a distinct social function, informs citizens about what events are important, contested, or issues that should be widely debated by the public (Bernstein et al., 2020). The diversity of news on social medias is always citical to meetting the communication needs of individuals and society. It presents a diverse range of social perspectives to facilitate public debate and deliberation on current events. However, the explosive growth of news today poses challenges to news diversity, and news platforms must recommend different news content to users across a wide range of topics. A recommendation system guides users to find interesting news in a personalized way based on their historical behaviors and preferences (Shao et al., 2021), but this type of personalization may reduce the news diversity and thus produce polarization (Reuver et al., 2021), Filter Bubbles (Pariser, 2011), and Echo Chambers (Jamieson et al., 2008), all of which have negative effects on the right of citizens to social information (Eskens et al., 2017). In this case, a more in-depth examination of news articles is cirtical for the diversity of results produced by news recommendation systems.

According to the deliverative model of democracy, which states that citizens in a democratic nation must be exposed to a variety of views and arguments (Manin, 1987; Helberger, 2019), many Natural Language Processing (NLP) techniques are proposed to help identify claims, stances, and argumentation in news articles (Reuver et al., 2021), such as Viewpoint Detection, Argument Mining (Lawrence et al., 2020) and Stance Detection (Küçük et al., 2020). These automatic classifications of semantic information in text can assist to quickly identifying and recommending news with various points of view or stances to users. These techniques enable recommended systems to recommend various news articles, thereby protecting democracy to some extent.

In this thesis, we focus on deep-learning based stance detection task in the news domain, which is a representative argument analytical paradigm for various social and political democracy. The definition of SD varies depending on the application scenario, while the most common and basic definition is (Küçük et al., 2020):

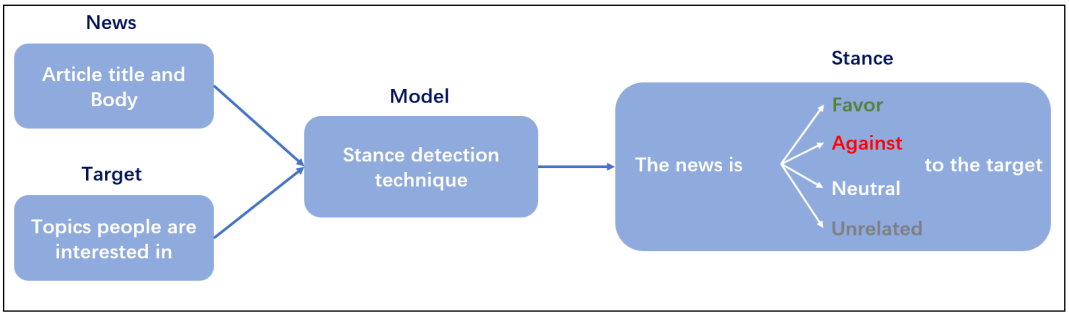


FIGURE 1.1: Definition of Stance Detection

FIGURE 1.2: An example from STANDER dataset

"Automatic classification of the stance of the producer of a text towards a target, into one of these three classes: Favor, Against, Neither."

In the news domain, *producer* refers to the writer, editor or reviewer of the news, while *text* refers to the title and body of the news. *Target* are the topics that the general public is interested in. *Stance* has many definitions, but the most basic common ground that stance is the someone's attitude and judgement towards a proposition (Biber et al., 1988). In some applications (Conforti et al., 2020a), there are four types of news stance towards a target. Instead of considering the *Neither* stance, *Neutral* and *Unrelated* are taken into account. The general news stance detection procedure and an example are shown in Figure 1.1 and Figure 1.2.

1.2 Research Questions

Conforti et al. (2020a) presented a challenging English dataset STANDER together with some baselines, with the goal of stance detection and evidence retrieval. Their research, however, is primarily focused on the dataset rather than on the models. Therefore, they did not tune the hyperparameters of the models, nor tried other cutting-edge models, such as pre-trained models and syntactic-based models. As a result, the performance of these models still has potential for improvement. In this case, we use the STANDER dataset for our research and address research questions related to five aspects: adjustment of basic models, application of pre-trained models, transfer ability of models, syntactic-based models and combinations of different techniques/models. Here we propose the five research questions listed below:

- BertEmb is the method proposed by Conforti et al. (2020a), that achieves the best

performance so far on STANDER according to their results. However, as previous stated, they did not concentrate on the models and the parameters tuning in their work. Then, from the perspective of adjusting the model, we arrive at the first question: **(1) Is it possible to improve the performance of the BertEmb method by identifying better hyperparameters (for example, learning rate and dropout rate)?**

- With the transformer-based pre-trained models achieving cutting-edge performance on several NLP tasks, we wonder **(2) How well the pre-trained model BERT (Devlin et al., 2019) performs after fine-tuning on the STANDER dataset?**
- Because of the difficulties in making expert annotations for datasets, the transferability of methods, which is also the focus of our research, is becoming increasingly important. We propose the model CLS-transfer BERT and two new tasks, inspired by Stance-Bert (Tian et al., 2020), which reuses the CLS tokens of BERT models fine-tuned by other tasks. Then, we address the question of **(3) To what extent the CLS token of BERT can transfer the stance information across different tasks? Will the CLS-transfer BERT improve the performance of BERT?**
- The lack of grounded supervision calls into the question that how well the BERT-based representations capture the meaning of sentences (Bender et al., 2020). As a result, syntactic models are being proposed gradually. In a similar task, sentiment analysis, integrating syntactic structure into the deep learning process has already yielded good results (Žunić et al., 2021). The author proposed a well-designed Dependency-based Graph Convolutional Network (GCN), so we can ask the fourth question that **(4) Can we achieve a considerable result on the stance detection task using a dependency-based model?**
- Another novel idea is to incorporate information from different level (sentence-level and document-level). On this basis, we wonder **(5) Can the model that combines the sentence embeddings in BertEmb and the BERT CLS embedding achieve the state-of-the-art performance and how about the model that combines the syntactic representation and the CLS embedding?**

1.3 Thesis Overview

Our research thesis is divided into five chapters. In Chapter 2, we provide the background information and elaborate the relevant datasets and algorithms, particularly deep learning techniques. In Chapter 3, we conduct an in-depth exploratory analysis of the methods, including BertEmb, BERT, CLS-transfer BERT, DB-GCN, and CLS-concat BERT. In Chapter 4, we go over the experiments and their corresponding results. The conclusion on analysis of results will be given in Chapter 5.

Chapter 2

Background

This chapter's scope are the theoretical foundations of NLP-related knowledge and the related work of stance detection (SD). To begin, in Section 4.3, due to the enormous amount of knowledge involved in NLP, we focus on the subjects that are frequently used in our work: attention, transformer, BERT (Bidirectional Encoder Representation from Transformers), sentence BERT and stance BERT. Following that, we introduced the related knowledge of dependency parsing and the graph model GCN. Secondly, we introduce the history of SD in Section 2.4 from four perspectives: early work, competitions, datasets, and methods. Finally, the evaluation metrics are mentioned in Section 2.5.

2.1 Neural Language Models

2.1.1 Attention

Since the first application on machine translation (Bahdanau et al., 2016), the attention mechanism (AM) has become a key component in deep learning models. It is widely used in a wide range of artificial intelligence applications (Chaudhari et al., 2021), including Natural Language Processing (NLP) (Galassi et al., 2021), Computer Vision (CV) (Wang et al., 2016) and Speech Recognition (Chorowski et al., 2015). AM is a network architecture component in charge of managing and quantifying interdependence. It enables models to dynamically focus only on specific parts of the input, similar to how humans perceive things. To be more specific, for each word in the input sentences, the AM component in a network maps the relevant words from the input sentence to it, assigning higher weights to more relevant words.

2.1.2 Transformer

Transformer is a well-known deep learning architecture that achieves remarkable results in a variety of domains, particularly in NLP (Lin et al., 2021). Transformer was initially proposed as a seq2seq model for machine translation (Vaswani et al., 2017). Recently, transformer-based pre-trained models (PTMs) have propelled NLP into a new era (Qiu et al., 2020) by achieving state-of-the-art performance on a variety of tasks. Traditional

loop architectures, such as RNN and LSTM, cannot be parallelized due to their reliance on sequential processing of inputs. To address this inefficiency, Vaswani et al. (2017) proposed the Transformer architecture, which has been demonstrated to achieve parallel processing and higher accuracy for the machine translation task.

The design of Transformer follows the encoder-decoder architecture, each of which is a stack of blocks. The encoder and decoder blocks are primarily composed of a multi-head self-attention module, a position-wise feed-forward network (FFN) and residual connections, followed by layer normalization (**ba2016layer**).

2.1.3 BERT

The most well-known Transformer-based model is BERT (Devlin et al., 2019). It achieves cutting-edge resutls on on numerous benchmarks. Fundamentally, BERT is made up of bidirectional transformer encoder layers that convert the word sequence to a vector and decoder layers that convert the vector back to a sequence of words. These layers are composed of multiple self-attention heads.

BERT takes a sequence as input. To properly understand input, BERT employs the special tokens: [CLS] and [SEP]. [CLS] is a special classification token, and the last hidden layer of BERT corresponding to this token is always used for classification. The [SEP] token should be appended at the end of a single input (sentence). When BERT is applied to a task that requires more than one input, [SEP] enables the model to recognize the end of one input and the beginning of another input. All Tokens are entered into BERT using Wordpiece embeddings.

The BERT training procedure consists of two steps: pre-training on unlabeled data and fine-tuning on labeled data for downstream tasks.

- In the pre-training step, BERT uses two training strategies on large corpora: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). Devlin et al. (2019) trained two versions of BERT: BERT-base [1] with 12 transformer layers and 768-long embeddings (110M parameters) and BERT-large [2] with 24 layers and 1024-long embeddings (340M parameters). This stage is primarily concerned with learning general language patterns.
- In the fine-tuning step, the model is simply added with one additional layer after the final layer of BERT and trained for few epochs. All of the parameters of BERT are fine-tuned using labeled data in the downstream language tasks, with the goal of learning language paradigms in specific data.

Following that, we introduce two extensions of the BERT that are frequently used in our work: Sentence BERT and Stance BERT.

[1] https://huggingface.co/bert-base-uncased
[2] https://huggingface.co/bert-large-uncased

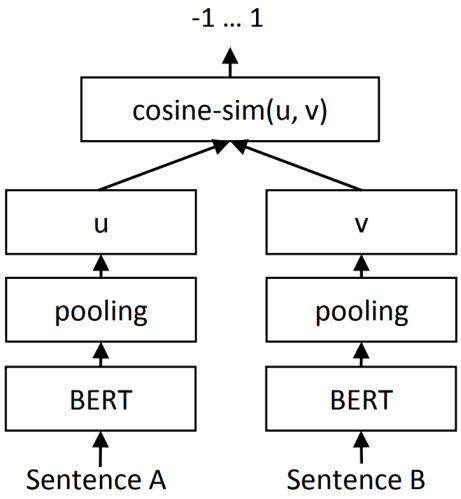


FIGURE 2.1: Sentence BERT architecture at inference (Reimers et al., 2019).

2.1.3.1 Sentence BERT

BERT models have outperformed other NLP models in many tasks, including text similarity computing. However, when computing the similarity, the BERT model must be fed with two sentences at the same time, resulting in significant computational overhead. For example, finding the most similar pair in 10,000 sentences, which requires calculations of $\frac{10{,}000 \cdot 9{,}999}{2} = 49{,}995{,}000$, will take approximately 65 hours. In order to overcome this limitation of BERT, Reimers et al. presented Sentence BERT, a variant of BERT that uses siamese and triplet structures to generate sentence representations which can be compared using cosine similarity. In this case, Sentence BERT takes only 5 seconds to complete the same task of finding the most similar pair.

The basic architecture of Sentence BERT consists of two BERT models (The two BERTs share the parameters), as shown in Figure 2.1.

2.1.3.2 Stance BERT

Stance BERT (Tian et al., 2020) is designed for early rumor detection on Twitter, which is the idea of our CLS-transfer BERT comes from. It is composed of two BERT models, as shown in Figure 2.2. The goal of the left BERT is to capture the stance distribution for a tweet and its comments. The inputs of Stance BERT are pairs of tweets from the SemEval-16 dataset. After the left BERT has been fine-tuned, its CLS embedding is then used to replace the original CLS embedding of the right BERT in order to further fine-tune it for rumor classification.

It is worth noting that the labels of the pairs for left BERT are six combinations of original stance: Favour-Favour, Favour-Neither, Favour-Against, Against-Against, Against-Neither and Neither-Neither. In the original paper, it is demonstrated that the CLS token obtained

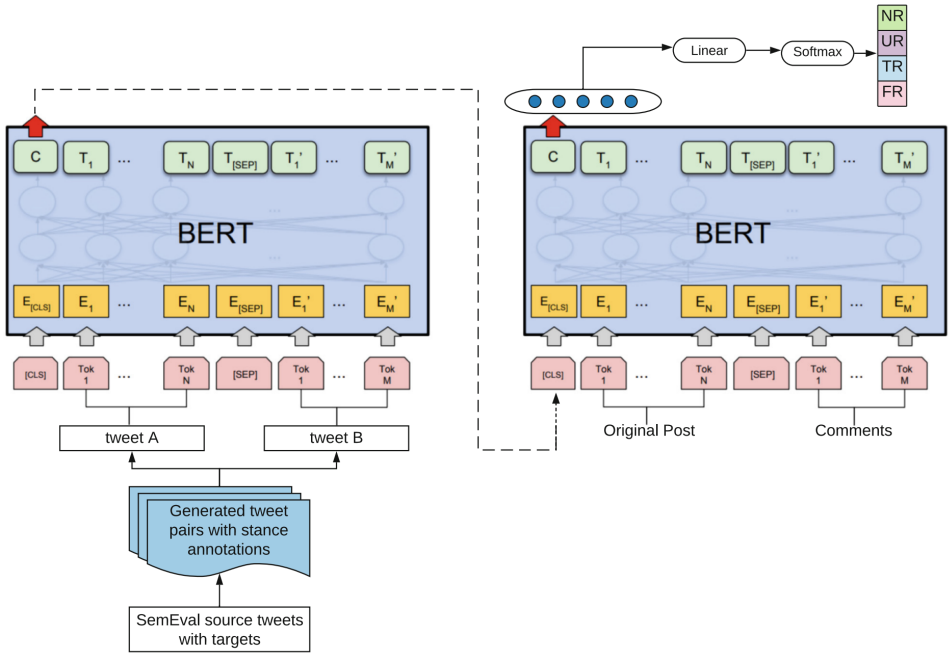


FIGURE 2.2: The architecture of our Stance-BERT model

through this fine-tuning task contains stance information, which can significantly improve the model's performance in the rumor classification task.

2.1.4 Dependency Parsing

Most language theories place a premium on syntactic structure. The models of syntactic analysis are relatively complex because they cannot be directly observed and require a great deal of linguistic knowledge. In linguistics, one of the methods of syntactic structure analysis is Dependency Parsing (DP), that describes the syntactic structure of a sentence in terms of binary grammatical relations between words. Chen et al. (2014) provided fast transition-based parser, which produces dependency parses for Universal Dependencies. As an example, consider the following sentence from the news article in STANDER, as shown in Figure 2.3.

In the diagram above, we choose one part to explain. There is a connection between *reviews* and *antitrust* because *antitrust* alters the meaning of *reviews*. In this case, *reviews* serves as the head, with *antitrust* as a dependent of the head. The relationship type between these two words is called *amod*, which stands for "Adjectival Modifier". It is a noun-modifying adjective.

The linguistic knowledge of dependency is fully explained in "Speech and Language Processing" by Jurafsky et al. (2021). Our introduction to dependency trees' relationship with neural networks will be the focus of this section.

A dependency tree can be represented as a graph $G = (V, E)$, where V is a set of vertices and E is a set of edges, which are corresponding to the words and arcs respectively in Figure

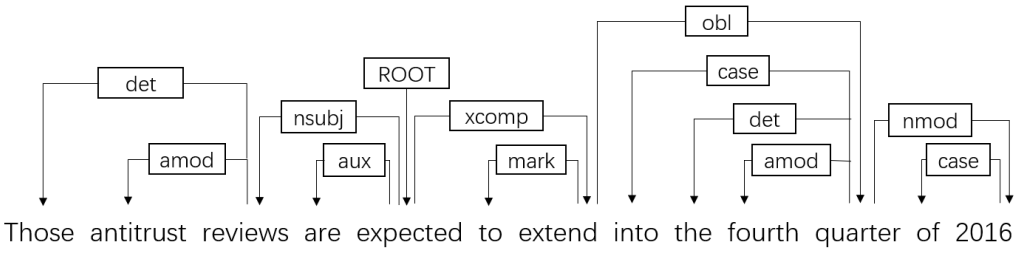


FIGURE 2.3: Dependency Tree of a sentence in the STANDER dataset. *det*: Determiner, *amod*: Adjectival modifier, *nsubj*: Nominal subject, *aux*: auxiliaries, *xcomp*: Open clausal complement, *case*: Prepositions, postpositions and other case markers, *nmod*: Nominal modifier. More relations are introduced by Marneffe et al. (2014)

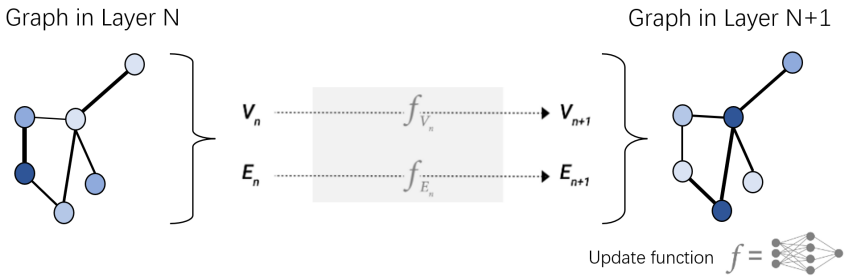


FIGURE 2.4: A single layer of a simple GNN

2.3. Importantly, the arcs in edge set E can capture the grammatical relationships between the words in vertex set V (Jurafsky et al., 2021). The dependency tree is commonly used as a graph as an input into the models (Graph Neural Networks/Graph Convolutional Networks) that introduced in the following section.

2.1.5 Graph Convolutional Network

Graph structure has demonstrated good capability in modeling structural information, because it can fully exploit the structural features of text (Yang et al., 2021). There are many studies proving that Graph Neural Networks (GNN) perform well in graph-structured text modeling (Yang et al., 2021). The basic structure of GNN is depicted in Figure 2.4.

Graph Convolutional Networks (GCNs; Kipf et al., 2016), as a member of the GNN family, also aim at the node classification on graphs. GCNs are similar to convolutions in images (CNNs) in that the parameters of filters are generally shared over all locations in the graph. Message passing (Gilmer et al., 2017) is used by GCNs during training: the vertices exchange information with their neighbors and send specific messages to each other. In most cases, the message is a word embedding. Specifically, in the work procedure of GCNs, each node first generates a feature vector (embedding) that represents its key message. Then the messages are sent to the neighbors, implying that the node will receive one message from each adjacent node.

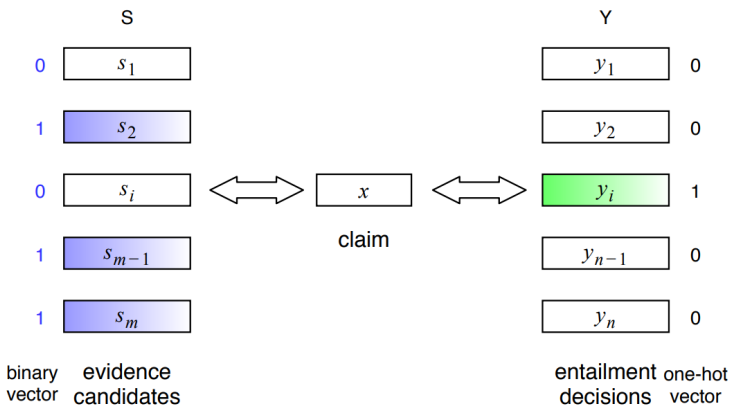


FIGURE 2.5: The architecture of the TWOWINGOS model (Yin et al., 2018).

Dependency trees containing structural and semantic attributes are used as input graphs for GNNs, which, after modeling, learn to obtain more efficient representations of nodes or entire graphs.

2.2 Two-wing Optimization Strategy

Two-wing Optimization Strategy (TwoWingOS) is a component used in BertEmb method. It is a system that can identify evidence for a claim and determine whether the evidence supports the claim (Yin et al., 2018). As shown in Figure 2.5, given a set of evidence sequence $S = \{s_1, ..., s_{m1}, s_m\}$ and a decision set $Y = \{y_1, ..., y_n\}$ (right), the TwoWingOS model predicts a binary vector p that represents a subset as evidence, and a one-hot vector o represents a single decision. Then the learning task becomes optimizing these two vectors to ground-truth vectors.

Importantly, one of the most significant advantages of TwoWingOS is the inclusion coarse-grained and fine-grained representations. The coarse-grained representation concatenates the representations of the sentence s_i and the claim x. While the fine-grained representation uses each word in s_i to create the sentence representation r_i. The BertEmb method makes use of a coarse-grained representation, which is described in detail in Chapter 3.

2.3 Principal Component Analysis

Principal Component Analysis (PCA) is a well-known unsupervised machine learning algorithms which we use for visualizing the embeddings. Hotelling (1933) proposed the modern instantiation of PCA: It is a mathematically technique for reducing the dimensionality while retaining as much variance as possible. It looks for linear combinations which holds the

highest variances, and divides them into principal components to obtain the most important information. PCA is used in a wide range of applications, including multidimensional data visualization, information compression, data denoising and dimensionality reduction.

2.4 Related Work

2.4.1 Early work on stance detection

For a long time, stance detection has been a key component in the analysis of online debates. Many datasets and methods are proposed to recognize the stance taken by documents or articles in online debates (Lin et al., 2006; Somasundaran et al., 2009; Murakami et al., 2010; Walker et al., 2012b). These online debate data were gathered and analyzed from a variety of websites: such as *bitterlemons*[3] which presents the viewpoints of Israeli and Palestinian on issues, *collaborationjam*[4] which provides post-event analytic report and *4Forums*[5] which shows comments from the Internet. Digging deeper into the data, we discovered that there are many common goals in their data, including topics such as *evolution*, *abortion*, *healthcare*, *gun rights* and so on. Additionally, it is also worth noting that stance annotation in early work only considered two stances: Favor and Against. As for the earlier research methods, they are mainly based on rule-based algorithms (Anand et al., 2011; Murakami et al., 2010; Walker et al., 2012a), machine learning algorithms such as Naive Bayes (Anand et al., 2011; Walker et al., 2012a; Hasan et al., 2013; Rajadesingan et al., 2014), Support vector machine (Thomas et al., 2006; Somasundaran et al., 2010; Walker et al., 2012b; Hasan et al., 2013), Random Forests (Misra et al., 2013), Conditional Random Fields (Hasan et al., 2013) and graph algorithms (Murakami et al., 2010; Walker et al., 2012a).

2.4.2 Competitions and Datasets

With the popularity of social media, there has been a growing research trend to analyze the public stance towards various social and political issues (AlDayel et al., 2020) by detecting the stance on social platforms, especially after the International Workshop on Semantic Evaluation in 2016 (SemEval-2016[6]) and Fake News Challenge in 2017 (FNC-2017[7]).

- *SemEval-2016 Task 6: Stance Detection of Tweets*. The organizers provided a dataset with six targets: Atheism, Climate Change is a Real Concern, Feminist Movement, Hillary Clinton, Legalization of Abortion, and Donald Trump. There are three possible outcomes: In-Favor, Against, and None.

[3] http://www.bitterlemons.org
[4] https://www.collaborationjam.com
[5] http://www.4forums.com
[6] https://alt.qcri.org/semeval2016/task6/
[7] http://www.fakenewschallenge.org/

This task is split into two subtasks: **A.** supervised stance detection and **B.** weakly supervised stance detection. For subtask A, the participants were provided with a training set containing 2,814 tweets and a test set of 1,249 tweets for five targets. While for subtask B, an unlabeled set of 78,000 tweets and a test set of 707 tweets were given (Mohammad et al., 2016).

- *FNC-2017 Phase 1 Shared Task: Stance Detection*. In this competition, stance detection is used as the first step in identifying fake news. It assists people in understanding the topics that news publishers are discussing by estimating the stance of news articles based on their headlines.

  The distinction is that this task involves four stances: agree, disagree, neutral, and irrelevant. The proposed dataset includes 300 topics represented by claims, with 5-20 news articles per topic.

Competitions in other languages (Chinese, Spanish, Catalan and etc.) have also received a lot of attention, such as *NLPCC-ICCPOL-2016 task: Stance Detection in Chinese Microblogs* and *IberEval-2017 task: Stance Detection in Spanish and Catalan Tweets*. To summarize, these competitions introduced expert-annotated datasets in multiple languages and inspired many cutting-edge approaches, greatly facilitating the research on stance detection.

The Semeval-2016 and FNC datasets have become the standard datasets since they were published. However, aside from the social media datasets mentioned above, significant effort has also been devoted to the annotation of stance data in other domains. *MeTooMA* (Gautam et al., 2019) is a dataset related to the (Me Too) movement that contains approximately 9,000 tweets annotated with stance. Conforti et al. (2020b) presented two datasets, one containing approximately 51,000 tweets in the financial domain and the other, called STANDER, including about 3,000 news articles covering the health industry. Furthermore, STANDER is adopted as an experimental dataset in our work and the details of it will be introduced in Chapter 4.

2.4.3 Stance Detection Approaches

According to the survey of Küçük et al. (2020), machine learning (ML) methods are the mainstream of current research and have achieved optimal results on the majority of datasets. In this case, we are primarily concerned with machine learning algorithms used for SD. The ML algorithms for stance detection can be divided into two main types: supervised learning and unsupervised learning. Supervised learning algorithms are the most common algorithm in SD task, which is also the focus of our research. We refer to unsupervised learning as a potential improvement and future work.

2.4.3.1 Supervised learning

Supervised learning is the most frequently used approach for SD (ALDayel et al., 2021). As mentioned in Early Work section, many SD studies employ traditional feature-based supervised ML algorithms. Therefore, we list the following commonly used algorithms:

- **Naive Bayes**, as the simplest ML algorithm, has been employed in SD for a long time. In addition to the early studies mentioned in Early Work section, it is still used as a baseline in some recent studies (Addawood et al., 2017; Simaki et al., 2017).
- **Logistic Regression** is another commonly used classifier that has been shown to perform well in related studies (Ferreira et al., 2016; Kucher et al., 2018; Zhang et al., 2018).
- **Support Vector Machine (SVM)** is the most common baseline in related studies and is the best performing ML model for many tasks. In SemEval-2016, SVM is used as a baseline, outperforming many other proposed approaches (Mohammad et al., 2016). Other works also tend to conclude SVM as a baseline (Gadek et al., 2017; Sen et al., 2018; Siddiqua et al., 2018; Conforti et al., 2020a).
- Other popular traditional ML algorithms that appear in recent studies are **Decision Tree** (Addawood et al., 2017; Simaki et al., 2017), **maximum entropy** (Xu et al., 2016), **kmeans clustering** (Simaki et al., 2017).

It is worth noting that the aforementioned algorithms employ a wide range of features, including lexical features, sentiment/argumentation features, word vectors, topic modeling features and features based on part of speech tags/named entities.

Nowadays, neural models, as the most popular ML models, have been widely applied in classification problems, such as Recurrent Neutral Network (RNN; Cho et al., 2014), Long short-term memory (LSTM; Hochreiter et al., 1997), Graph Neural Network (GNN; Zhou et al., 2020) and the most popular transfer learning model: Bidirectional Encoder Representation from Transformers (BERT; Devlin et al., 2019).

- **RNN & LSTM** have been commonly used in the NLP domain. Basic RNN has already demonstrated good performance in SD (Sobhani et al., 2017; Benton et al., 2018; Rajendran et al., 2018).As a variant RNN model, LSTM also achieves excellent results on a variety of datasets: In SemEval-2016, a model was proposed that used bidirectional LSTM with a fasttext embedding layer for the SD task (Siddiqua et al., 2019). Another multitask learning model on this task used a bidirectional gate that to detect the stance based on sentiment (Li et al., 2019). In FNC-2017, Hanselowski et al. proposed a feature-rich stacked LSTM model which performs on par with the best system (Hanselowski et al., 2018).
- **Transfer Learning (TL) methods** are conducted to solve the issue of insufficient labeled data. In transfer, the relevant knowledge gained by an algorithm from one

task is transferred to another similar task. For example, in the task of stance detection, the idea of transfer is applied to detect different objects. By maximizing the value of the available data, TL becomes a high-performance technology for SD.

For the transfer learning on SD, several studies have introduced techniques to enrich texts and targets representation. Mitre et al. (the winner of SemEval-2016 SD task) proposed the RNN initialized with features learned by distant supervision on unlabeled data (Zarrella et al., 2016). Ghosh et al. used the BERT model to achieve the most effective performance on the SemEval-2016 SD dataset and online news articles (Sen et al., 2018; Ghosh et al., 2019). Giorgioni et al. trained a specific UmBERTo based sentence classifier from three tasks: sentiment, irony and hate-speech (Giorgioni et al., 2020). Kawintiranon et al. presented a novel BERT-based fine-tuning method that enhances knowledge by training the transformer with masked language modeling (Kawintiranon et al., 2021) and they also obtained promising results.

- **GNN** is currently not used for SD tasks. However, in a similar study, aspect-based sentiment analysis, Žunić et al. achieved remarkable results (Žunić et al., 2021). The authors classified the sentiment of a given aspect through sentence-dependent parse trees and graph convolutional networks.

2.4.3.2 Unsupervised learning

Recently, researchers have also begun to investigate unsupervised stance detection models, where clustering techniques are widely used in stance detection. Trabelsi et al. proposed a purely unsupervised model for viewpoint identification using a clustering model at the discourse level. The model takes the text as input and assigns a viewpoint to the text. Then the model gives a viewpoint label to each occurrence of the unigram words (Trabelsi et al., 2018). Another work presents a highly effective unsupervised framework for detecting stance on the controversial Twitter topics (Darwish et al., 2019). The authors used dimension reduction to project users into a low-dimensional space to find core users representing different stances. They discovered that including retweets as a feature would benefit the clustering algorithm, and it even outperformed other supervised methods (FastText/SVM).

2.5 Evaluation Methods

The evaluation phase is one of the fundamental parts of any project and is the follow-up to the modeling step. This section presents various mainstream approaches to evaluate the performance of deep learning methods on classification tasks.

To get an overview of the performance of a specific algorithm, accuracy, recall, precision, and F score are good choices and can be determined for a classification problem.

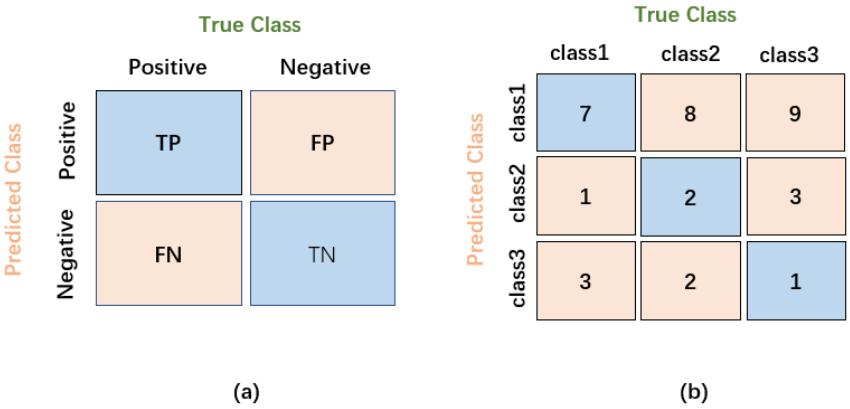


FIGURE 2.6: (a) Confusion Matrix for Binary Classification (b) Confusion Matrix for Multi-Class Classification

Confusion Matrix, as shown in Figure 2.6, is a tabular representation of the performance of a classification model. Each entry in a confusion matrix represents the number of predictions made by the model in which the classes are correctly or incorrectly classified.

Accuracy is determined by dividing the sum of the diagonals by the total number of entries in the confusion matrix.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.1}$$

Recall is a measure that computes the percentage of relevant instances chosen by the algorithm. Thus, it evaluates how many True Positives (True Postive) are present in a given class.

$$Recall = \frac{TP}{TP + FN} \tag{2.2}$$

Precision is another measure for evaluation. It calculates the percentage of True Positives (TPs) in the set where the model has classified all instances as positive.

$$Precision = \frac{TP}{TP + FP} \tag{2.3}$$

F-measure represents a harmonic mean between recall and precision. The weighting can be adjusted by the parameter β. This makes F_β adoptable to different data mining tasks, for example, in a search engine, where recall could be more important than precision. In our work, we keep β as 1.

$$F_\beta = \frac{(1 + \beta^2) * precision * recall}{\beta^2 * (precision + recall)} \tag{2.4}$$

Chapter 3

Methodology

The following chapter presents the models we use for news stance detection. First, we employ the general architecture of BertEmb proposed by Conforti et al. (2020a) as the first baseline model and elaborate on the details based on of our understanding. We also use the pre-trained BERT as the second baseline model. Then, in order to verify the transferability of the CLS token, we introduce the CLS-transfer BERT and devise two distinct tasks to obtain transferable CLS tokens. We then introduce the CLS-concat BERT, which concatenates the sentence embeddings with the CLS embeddings. Finally, we go over the model details of DB-GCN and how it solves the problem of multi-sentence input.

3.1 Baseline1: BertEmb

Conforti et al. proposed the BertEmb model, a language model based on Sentence BERT with TwoWingOS (Yin et al., 2018). BertEmb is reported to have the best results compared to other baselines on the STANDER dataset (Conforti et al., 2020a). Because the model architectures are not the focus of their research, some model details are omitted. We therefore elaborate on BertEmb in detail as follows.

BertEmb uses Sentence BERT as the encoder to generate embeddings of sentences, while using a simplified TwoWingOS and linear layers as the decoder. The architecture of the entire model is depicted in Figure 3.1.

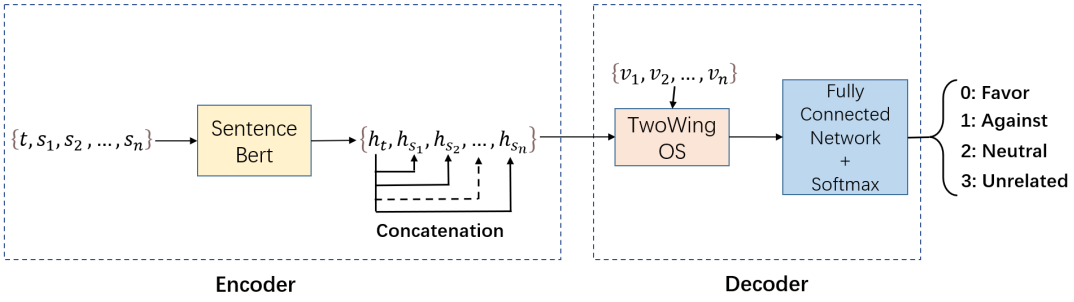


FIGURE 3.1: The architecture of BertEmb

Input & Output The model receives a raw sentence sequence $S = \{t, s_1, s_2, ..., s_n\}$, where t represents the target and s_i represents a sentence from the article or its title. The output is a class label from $\{0, 1, 2, 3\}$, representing four different stances (*Favor*, *Against*, *Neutral*, *Unrelated*) of these sentences towards the target t.

Encoder & Decoder Sentences in S are transformed to fixed length vectors $V=\{h_t, h_{s_1}, h_{s_2}, ..., h_{s_n}\}$ by a pre-trained Sentence BERT model. To combine sentences with the target, each sentence embedding h_{s_i} is concatenated with the target embedding h_t, which generates an article-target representation sequence $H=\{h_1, h_2, ..., h_n\}$. The TwoWingOS receives H and calculates a score $\alpha_i \in (0, 1)$ for each sentence, as formulated in Equation 3.1:

$$\alpha_i = sigmoid(v \cdot h_i) \tag{3.1}$$

In matrix form, we have:

$$A = sigmoid(v \cdot H^T) \tag{3.2}$$

where v is parameter vector with the same dimensionality as h_i and is learned from training; $A = \{\alpha_1, \alpha_2, ..., \alpha_3\}$ is the set of scores. Combining the score α_i with the embedding h_i generates a combination representation as e:

$$e = \sum_{i=1}^{n} \alpha_i \cdot h_i = A \cdot H \tag{3.3}$$

e is transformed into a class number by fully connected layers and a softmax layer, as described in Equation 3.4.

$$\hat{y} = Softmax(W\dot{e} + b) \tag{3.4}$$

where W and b are the weights and bias, respectively, and $\hat{y}$ is the class number.

3.2 Baseline2: BERT

As described in section 2, we apply the BERT model to this SD task. The architecture of BERT for news stance detection is shown in Figure 3.2.

Input & Output We fine tune the BERT model on the STANDER dataset. BERT receives the raw text t of the article, title, and target of each sample, where the article is concatenated with its title. Then the raw text t is tokenized using the built-in wordpiece tokenizer. The model produces the same results as BertEmb: four class numbers from $\{0, 1, 2, 3\}$ to represent the stance.

Pooler Output & Feed Forward Network The pooler output, which is the CLS embedding of the final layer of the BERT model, is always used to solve classification problems. Therefore, we feed the CLS embedding to the linear and softmax layers.

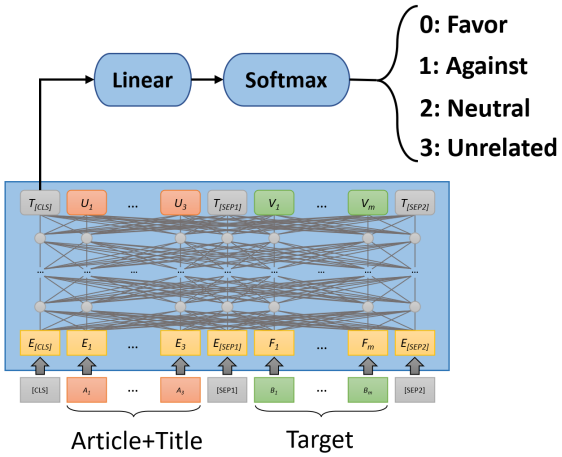


FIGURE 3.2: Architecture of BERT in news stance detection. We use the BERT-base model, where the length of the word/token embedding is 768d

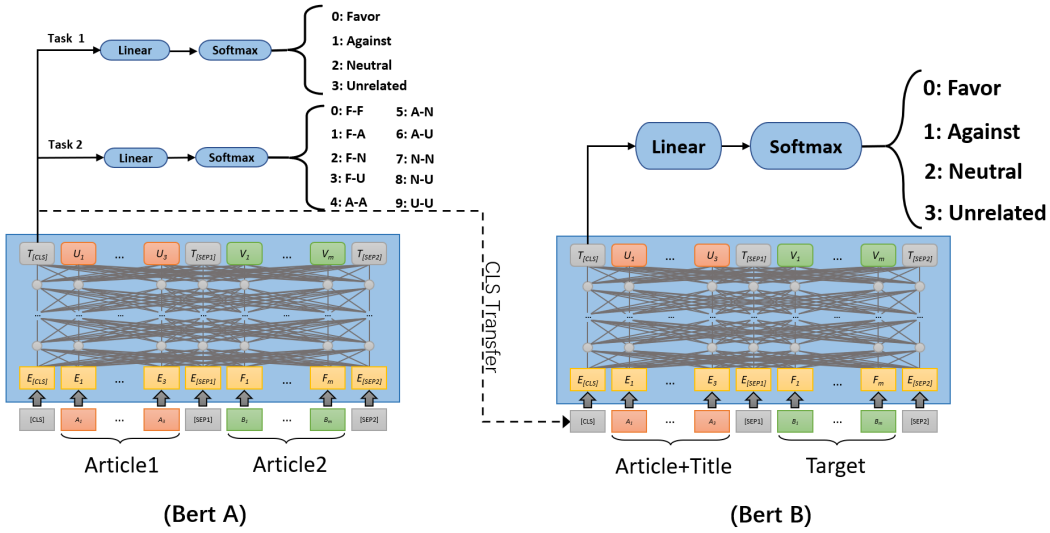


FIGURE 3.3: Architecture of CLS-transfer BERT

3.3 CLS-transfer BERT

We design the CLS-transfer BERT, as shown in Figure 3.3, in response to the Stance BERT model introduced in Section 2, which demonstrates that CLS embedding can be a carrier of stance knowledge. The CLS-transfer BERT is made up of two independent BERT models: BERT A, which is fine-tuned on two auxiliary tasks and BERT B, which is fine-tuned on the stance detection task. The embedding of the CLS token in BERT A is expected to contain stance information after fine-tuning. To transfer prior stance knowledge, we use it to replace the original CLS embedding in BERT B.

Two auxiliary tasks. We created two new tasks based on the STANDER dataset. In this case, they can also be viewed as two methods for expanding the dataset.

- **Task 1.** We construct pairs of articles from the STANDER dataset and label them

Article 1	Article 2	A1 to A2		Article 1	Article 2	A1 to A2
Favor	Favor	Favor		Neutral	Favor	Neutral
Favor	Against	Against		Neutral	Against	Against
Favor	Neutral	Against		Neutral	Neutral	Favor
Favor	Unrelated	Unrelated		Neutral	Unrelated	Unrelated
Against	Favor	Against		Unrelated	Favor	Unrelated
Against	Against	Favor		Unrelated	Against	Unrelated
Against	Neutral	Against		Unrelated	Neutral	Unrelated
Against	Unrelated	Unrelated		Unrelated	Unrelated	Unrelated

TABLE 3.1: Parallel table for the types of stance between two articles

on the relationship between stances towards a target. For example, if article 1 has a favor stance towards a target, and article 2 also holds the same stance, we define that article 1 is in favor of Article 2. If article 1 holds the favor stance towards a target, but article 2 holds the against stance, we define that article 1 is Against Article 2. Similarly, to define the stance of one article toward another on the same target, we create the parallel table shown in Table 3.1.

- **Task 2.** The input is also made up of pairs of articles constructed from the STANDER dataset. What distinguishes task 2 from task 1 is the label. Task 2 is less concerned with the relationship between two articles and more concerned with the combinations. For example, if article 1 holds Favor stance towards a target while article 2 also holds Favor stance for the same target, then they are labeled as Favor-Favor(F-F). Similarly, if article 3 has Favor stance to a target but article 4 has Against stance to this target, then we generate an instance (article 3, article 4) with the label Favour-Against(F-A). In the STANDER dataset, there are four stances, which means there are ten label combinations of article paris: Favour-Favour(F-F), Favour-Against(F-A), Favour-Neutral(F-N), Favour-Unrelated(F-U), Against-Against(A-A), Against-Neutral(A-N), Against-Unrelated(A-U), Neutral-Neutral(N-N), Neutral-Unrelated (N-U) and Unrelated-Un- related(U-U).

Feed Forward Network. The linear and softmax layers of BERT A and B are similar to those described in Section 3.2. The only difference is that the label count of BERT A varies depending on the tasks. For the large number of parameters in BERT that require fine-tuning, we only connect the pooler output with one linear layer and a softmax layer to reduce the number of variable parameters.

CLS transfer. The usability of the CLS-transfer BERT can be evaluated in two ways. The first and most obvious method is to compare the performance with BERT. Furthermore, we generate a random vector with the same size as the CLS embedding and transfer it to BERT B. In this case, this model is called as CLS-random BERT. This enables us to investigate the significance of the auxiliary training (part A) of the CLS embeddings in the architecture. In addition, to make the results more visual, we present an analysis based on PCA. We select 20 samples at random to observe their pooler outputs with different CLS embeddings. Because the outputs are in 768 dimensions (BERT base), we apply PCA

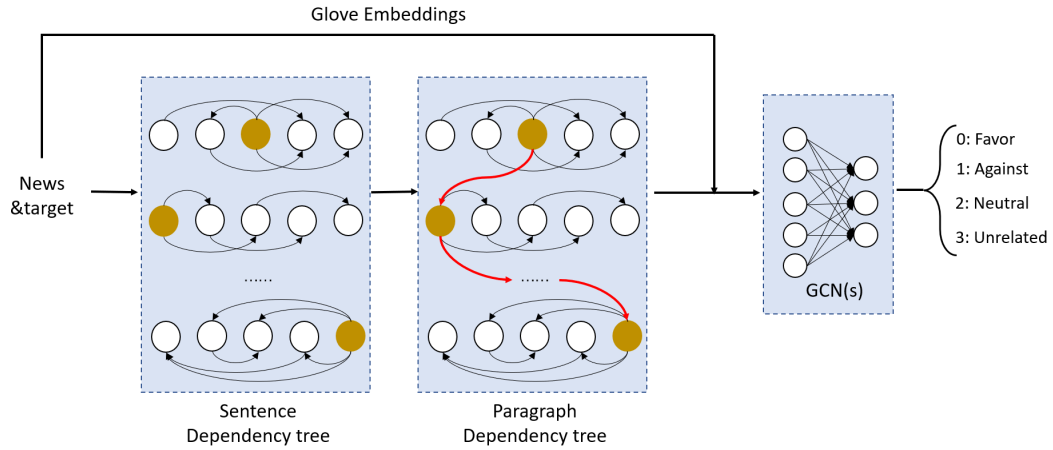


FIGURE 3.4: Dependency-based GCN system for stance detection in news

on these outputs to reduce the dimension to 2D and visualize them. By comparing the distribution of these samples before and after fine-tuning, we can indirectly observe the difference.

3.4 DB-GCN

Dependency-based GCN is a syntactic model which is inspired by the aspect-based sentiment analysis with GCN taking precedence over syntactic dependency (Žunić et al., 2021). The model is made up of two components: dependency parser and GCN model. Figure 3.4 depicts the overall system architecture.

First, the input data is processed by a dependency parser (Marneffe et al., 2014), which converts a single sentence into a dependency graph. Words in sentences, i.e. vertices in the dependency graph, are mapped to embeddings. We use GLOVE (Pennington et al., 2014) to obtain word embeddings. As a result, each input sentence is represented as a sequence $S = (w1, w2, ..., wn)$, where w is of dimensionality d of the word embedding, that is, this representation yields a $n\dot{d}$ matrix.

In our task, each news article contains multiple sentences. Therefore, we need to establish the relationship of the dependency graph of each sentence. As illustrated in the paragraph dependency tree in Figure 3.4, we connect the roots of each sentence, so that the sentences will affect each other when the passing information in GCN.

3.5 CLS-concat BERT

The fusion of various levels of knowledge has gained widespread attention (Wang et al., 2020). Following this line of thought, we see that BertEmb is a sentence-level model. It focuses more on the representations of sentences in articles and targets while ignoring the

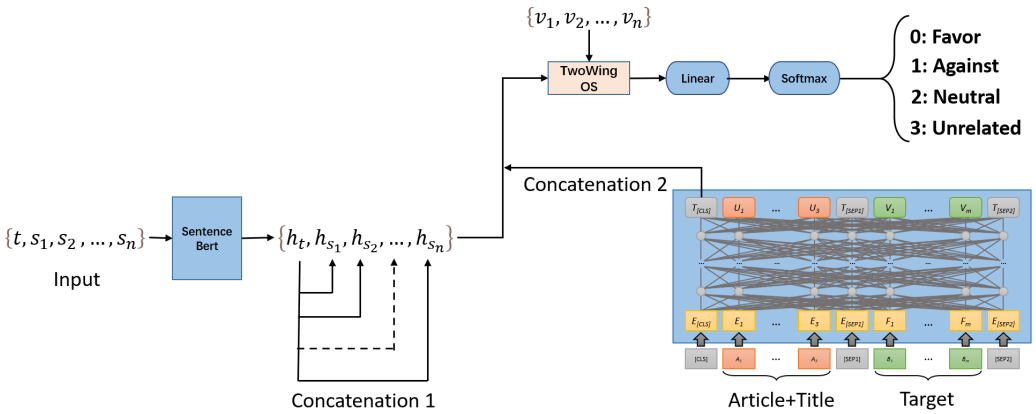


FIGURE 3.5: The architecture of the CLS-concat BERT: the combination of BertEmb and BERT

relationships between sentences. However, BERT can be used to capture the stance of the entire article towards a target in the document level. In this case, combining the sentence-level information in BertEmb with document-level information in BERT is a potentially useful solution. We call this combined model as CLS-concat BERT, the architecture of which is depicted in Figure 3.5.

We also experiment with the combination of DB-GCN and BERT. We concatenate the average of target node embedding obtained from GCN convolution with CLS embedding of BERT and feed them into linear and softmax layers, as shown in Figure 3.6.

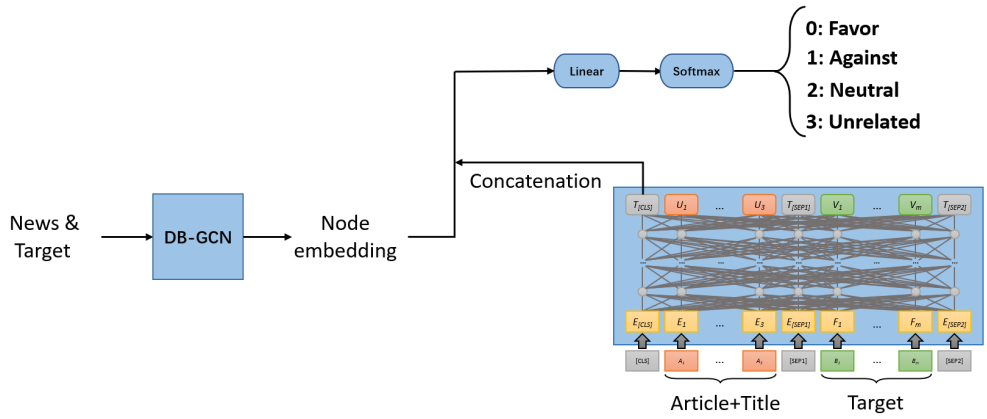


FIGURE 3.6: The architecture of the CLS-concat BERT: the combination of DB-GCN and BERT

Chapter 4

Experimental Results

This chapter is devoted to the dataset, the experiments for the models mentioned in chapter 3 and the results. First, we introduce the STANDER dataset and analyze it statistically. Then we sketch the experiments which we conduct for answering the research questions. The performance of these models is also organized and provided for further comparison and analysis. Our code is available now[1].

4.1 Dataset

STANDER (STANCE Detection & Evidence Retrieval) (Conforti et al., 2020a) is an expert-annotated dataset for detecting news stances and retrieving evidence. Because of the high quality of the news content and expert annotation, the dataset has the potential to become a standard for stance detection in the news domain. The data content and statistics are provided in the subsections that follow.

4.1.1 Data Content

The news in STANDER is about four mergers of six companies (Buyer and Target) in the healthcare industry in the United States, as shown in Table 4.1. These four merges were crawled, processed and annotated from newspapers, journals and magazines by the authors, and they were hot topics of widespread concern.

In the dataset, each news article includes the body and title of the article, as well as the target. The target is the merger event described in the newspaper article. It can take one of four possible stances towards the target: Support, Refute, Comment and Unrelated.

[1] https://github.com/LastDance500/StanceDetection

Merger	Buyer	Target Company	Target
AET_HUM	Aetna	Humana	Aetna (AET) will merge with Humana (HUM).
ANTM_CI	Anthem	Cigna	Anthem (ANTM) will merge with Cigna (CI).
CI_ESRX	Cigna	Express Scripts	Cigna (CI) will merge with Express Script (ESRX).
CVS_AET	CVS	Aetna	CVS (CVS) will merge with Aetna (AET).

TABLE 4.1: Four mergers in *STANDER* (Conforti et al., 2020a).

Stance	**refute**
Target	**Aetna (AET) will merge with Humana (HUM)**
Title	Financial Post Investing U.S. officials target health mega-mergers; Deals would see top five insurers reduced to three
Body	U.S. antitrust officials on Thursday moved to block an unprecedented consolidation of the national health insurance market,filing suit against Anthem Inc.'s proposed purchase of Cigna Corp. and Aetna Inc.'s planned acquisition of Humana Inc.[...]
Stance	**support**
Target	**Anthem (ANTM) will merge with Cigna (CI)**
Title	Anthem Seals Deal With Cigna Amid Industry Shake-Up
Body	Anthem Inc. agreed to buy Cigna Corp. for $48 billion, capping months of merger frenzy among top U.S. health insurers that is set to reshape the industry.[...]
Stance	**comment**
Target	**Cigna (CI) will merge with Express Script** (ESRX)
Title	Extra Cigna eyes 'sustainable healthcare system' with Express Scripts deal
Body	The need to create a "more sustainable healthcare system" was one of the key drivers that led Cigna Corp. to agree to combine with pharmacy benefit manager Express Scripts Holding Co. in a massive $67 billion transaction, according to Cigna President and CEO David Cordani.[...]
Stance	**unrelated**
Target	**CVS (CVS) will merge with Aetna (AET)**
Title	Financial Post Walmart reportedly eyes deal with insurer Humana
Body	Walmart Inc. is in talks with health insurer Humana Inc. for a closer partnership to provide health care to consumers at home and prevent illness, according to a person familiar with the matter.[...]

TABLE 4.2: Four samples in STANDER: A refuting, a supporting, a commenting, and a unrelated sample.

These stances are only named differently than the previous introduced ones: Support = favor, refute = against, comment = neutral, and unrelated is the same. In Table 4.2, we provide four examples, each with its own body, title, target, and stance. To be more comprehensive, we choose samples from four different kinds of stance and four different mergers. For the sake of the illustration, we only include the first sentence of the body of each article in Table 4.2. As input for our models we use the complete articles.

4.1.2 Data Statistics

In Table 4.3, we provide the distribution of the stances and mergers. STANDER contains 3291 labeled news. Among the samples, for the merger, the number of ANTM_CI is the greatest, while CI_ESRX has the fewest examples. For the stance, support has the most diverse samples, far more than unrelated stance.

Merger	support	refute	comment	unrelated	Total
AET_HUM	463	313	197	5	978
ANTM_CI	367	537	248	14	1166
CI_ESRX	207	64	70	5	346
CVS_AET	372	104	294	31	801
Total	1409	1018	809	55	3291

TABLE 4.3: Stance distribution for stances and mergers (Conforti et al., 2020a). The blue numbers are corrections of the original paper.

According to the distribution of stance, we notice that there is an imbalance in the data. The Unrelated stance has far fewer instances than the other three stances, which may lead to insufficient prediction ability of the models for the Unrelated label.

4.2 Experiments

4.2.1 Environment

All experiments are conducted on the duranium server provided by the Leiden Institute of Advanced Computer Science. In order to make the experiment more efficient, we run multiple experiments on 6 NVIDIA GTX 980 Ti and 2 NVIDIA Titanium GPUs simultaneously in most of time.

The programming language is Python 3.8 and the deep learning framework is PyTorch 1.11. The evaluation metrics are implemented by scikit-learn[2]. We only employ two pre-trained models, BERT base[3] and Sentence BERT[4], in our experiments. All pre-trained models are provided by HuggingFace.

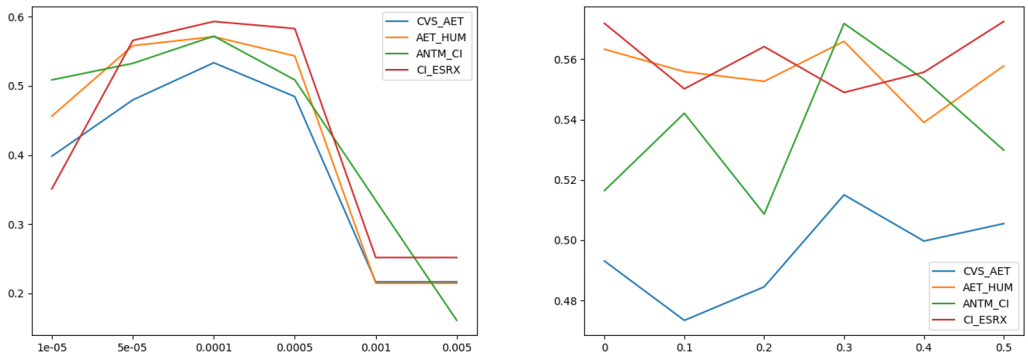
4.2.2 Experiment setup & Results

For comparison, we keep the majority of the experimental setup used by Conforti et al. (2020a). For each training, we train the model on three of these mergers while testing on the fourth. For instance, we train the models on AET_HUM, ANTM_CI and CI_ESRX mergers and test on the CVS_AET merger. Differently with original paper, we set batch size to 8 and epochs to 10 rather than 32 and 70. This is due to the limitations of GPU device. We also employ early stopping with patience of 10. For each experiment, we repeat it five times and calculate the average precision, recall, and f1 score.

4.2.2.1 Exp 1: Hyperparameters of BertEmb

There are two main hyperparameters that affect the performance of BertEmb: learning rate and dropout rate. In our experiments, we use 10% of test set as the validation set to

[2] https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics
[3] https://huggingface.co/bert-base-uncased
[4] https://huggingface.co/sentence-transformers/bert-base-nli-mean-tokens



(a) Learning rate **(b) Dropout rate**

FIGURE 4.1: The fluctuation of f1 score with two hyperparameters: learning rate and dropout rate

find the best configurations of BertEmb.

Learning rate. Learning rate (lr) is a parameter in an optimization algorithm. It determines the step size while moving to a minimum of the loss function. There is a trade-off between the convergence and overshooting when setting the rate. A rate that is too low may cause the model become stuck in an local minimum while too high rate may cause the model to skip over the global minimum. In our experiments, we test the learning rate for BertEmb in $(5^{-2}, 10^{-3}, 5^{-3}, 10^{-4}, 5^{-4}, 10^{-5})$, as plotted in Figure 4.1a.

Dropout rate. Large neural networks trained on relatively small datasets may suffer the overfitting problem. Therefore, dropout is proposed as a regularization method that randomly selects neurons and ignores them during training (Srivastava et al., 2014). In our experiments, we tested the dropout rate for BertEmb in $(0, 0.1, 0.2, 0.3, 0.4)$ increments, as plotted in Figure 4.1b.

In Table 4.4, we list the performance of original BertEmb (first row) and the performance after tuning the hyperparameters (second row).

4.2.2.2 Exp 2: BERT

As a second baseline, we use the pre-trained BERT-base model and fine-tune it on the train set. For the hyperparameters of BERT, we set the learning rate to 10^{-5} and dropout rate to 0.3. We use a lower learning rate here because the BERT model has far more parameters than BertEmb model. And for the large amount of parameters, we only add one single linear layer and softmax layer to process the CLS embedding. The result of the BERT model on the test set is shown in the third row of Table 4.4.

4.2.2.3 Exp 3: CLS-transfer BERT

In this experiment, we train three models: CLS-random BERT and two CLS-transfer BERTs which are the models for tasks 1 and 2, as described in Section 3.3. We save the CLS

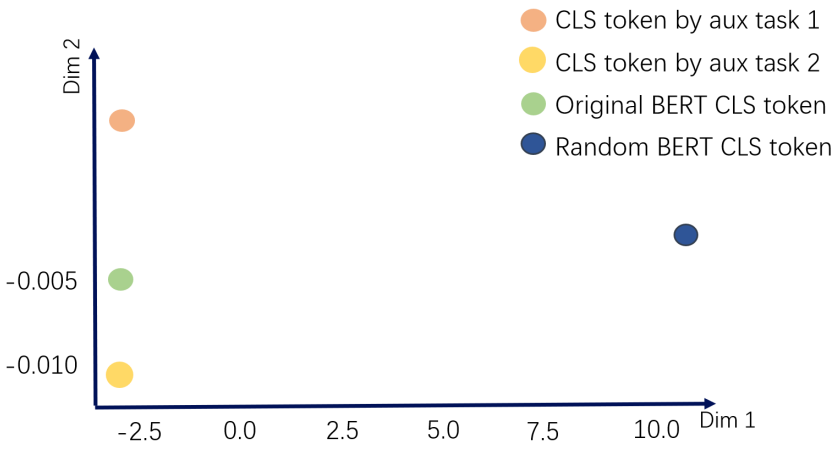


FIGURE 4.2: PCA results of four embeddings: original CLS, CLS based on task 1, CLS based on task 2 and random CLS

embeddings of these models and use PCA to visualize them, as shown in Figure 4.2. Then we transfer these four embeddings to BERT B and use PCA to reduce the 768-d pooler outputs to 2-d with PCA, as shown in Figure 4.3. For all models, we fine-tune them on train set and test on test set, just as we do with the BERT model. The performance of these three models is shown in the third to fifth rows of the Table 4.4.

4.2.2.4 Exp 4: DB-GCN

The parser we use for dependency parsing is Stanford CoreNLP (Chen et al., 2014), which returns the dependency tree of each sentences. The edges between roots are added to the edge set E to connect sentences. As a result, the trees of multiple sentences are combined into a tree of paragraph. This dependency tree is fed into GCN, with each vertex corresponds to a 300-d glove vector. After two layers of convolution, we take the average of the vertices corresponding to the target and feed it into the linear layer and softmax layer. In addition, GCN is consistent with BERT for all other hyperparameters.

4.2.2.5 Exp 5: CLS-concat BERT

In this experiment, we compare two combined models that combine CLS embedding with other kinds of embeddings. The first is that we combine the BertEmb model and the BERT model and concatenate the CLS embedding with the sentence embeddings, as described in the Section 3.5. We keep the hyperparameters of the BertEmb and BERT model and apply the same training paradigm as BERT: fine-tune the model on the training set and test on the test set. The second model is basically the same as the first, in that we combine the GCN and BERT model by concatenating the CLS embedding with the average embedding of the target nodes. The results of the CLS-concat BERT models can be seen in the last two rows of the Table 4.4.

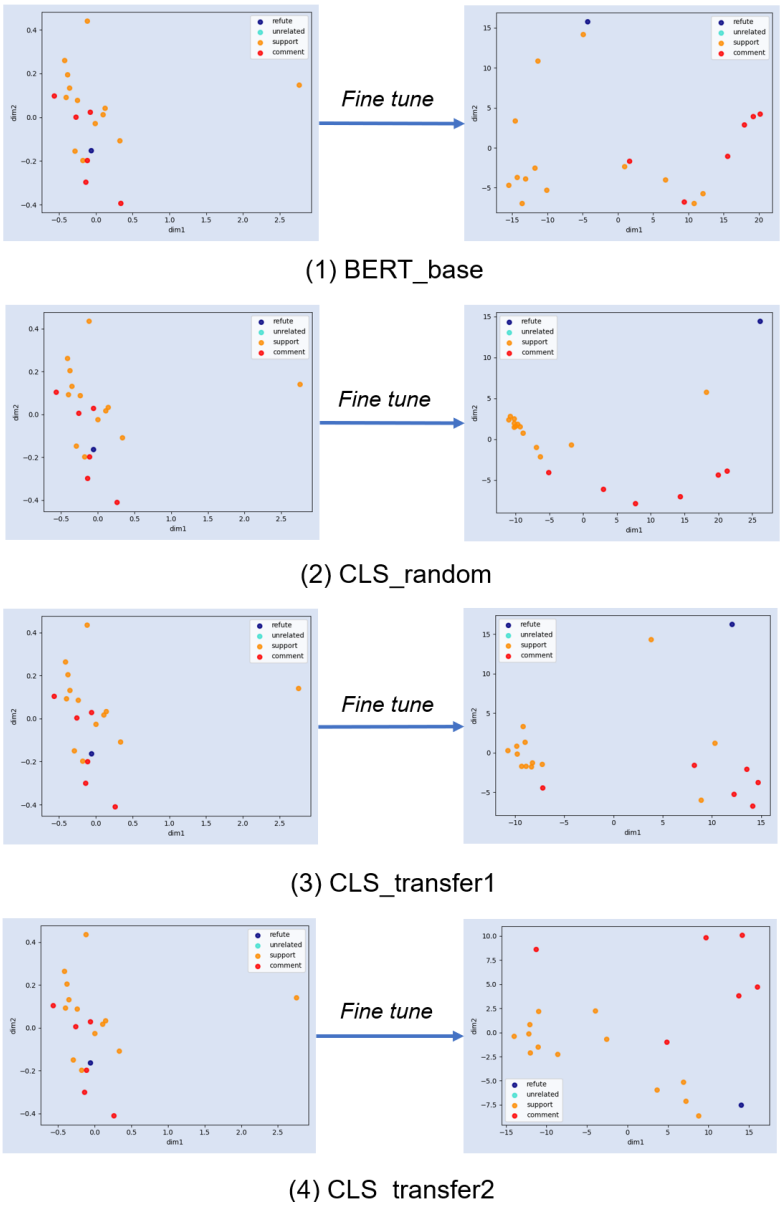


FIGURE 4.3: PCA results of 20 samples by four different CLS embeddings

4.3 Overall Performance and Analysis

The overall performance is shown in Table 4.4 and Figure 4.4.

For the hyperparameter tuning of the BertEmb model, we discover that a learning rate of 10^{-4} and a dropout rate of 0.3 are optimal. This result matches our expectations, as mentioned in Section 4.2, moderate learning rate and dropout rate help in model convergence. Applying this set of hyperparameters to the BertEmb model, we obtain better results (avgF1: 43.2) than the original paper (avgF1: 43.2).

The BERT model shows further improvement on this task, as shown in the third row of table 4.4. Compared to BertEmb (avgF1: 43.2), the BERT model (avgF1: 68.2) achieves

Model	F1 for the mergers				Avergae		
	CVS_AET	CI_ESRX	ANTM_CI	AET_HUM	avgP	avgR	avgF1
BertEmb(Ori.)	42.5	33.2	46.4	43.9	50.5	45.6	43.2
BertEmb(Opt.)	53.7	59.4	57.2	57.0	58.0	57.6	56.8
BERT	69.9	73.8	64.3	64.7	69.0	69.4	68.2
CLS-random BERT	70.3	73.4	63.9	64.3	68.5	69.0	67.9
CLS-transfer BERT1	71.7	72.4	**64.4**	64.4	69.3	69.2	68.2
CLS-transfer BERT2	69.4	74.7	64.2	63.9	68.4	69.6	67.9
DB-GCN	36.3	37.3	31.0	42.5	39.3	37.7	36.7
CLS-concat(BertEmb)	**72.4**	**76.4**	**64.4**	**64.9**	**69.2**	**71.2**	**69.5**
CLS-concat(GCN)	71.4	72.2	63.9	60.7	68.3	69.7	67.0

TABLE 4.4: Results of experiments on stance detection on STANDER. The bold numbers indicate that the corresponding models achieved the best results

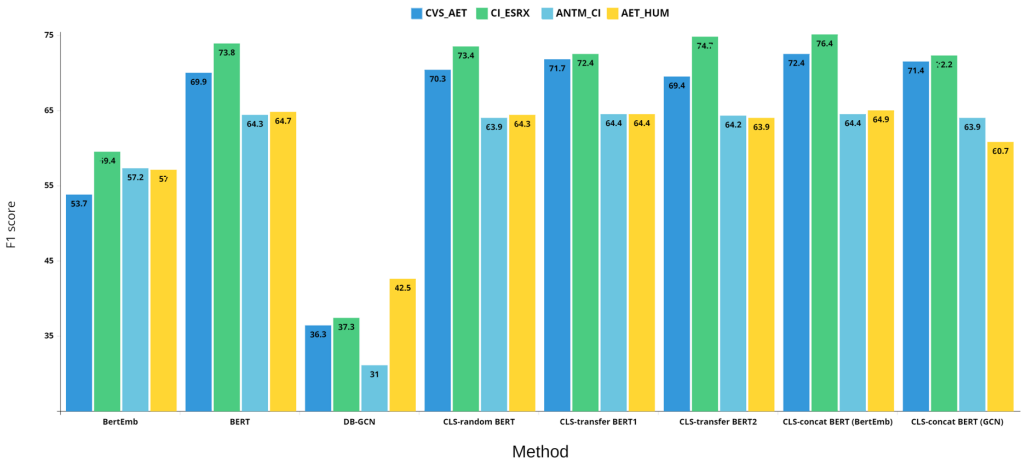


FIGURE 4.4: Bar graph representation of experimental results

more than 10% point improvement. Investigating the reasons for this enhancement, we contend that, as described in Section , the transformer used by BERT is a more advanced architecture, and pre-training enables BERT to contain rich semantic information and to capture the context of words. Furthermore, the pre-training corpus of BERT model includes BookCorpus[5] and English Wikipedia[6], which contains a large number of formal and regular sentences, similar to sentences in the news.

As for CLS-transfer BERTs, we discover that they do not improve the BERT model. Our experiments in Section 4.2 also support this observation. As shown in the figure, the CLS embeddings based on tasks 1 and 2 are very similar to the original embeddings, while the random CLS embedding is completely different and further away from the other three embeddings. Due to the randomness of CLS-random BERT, its performance should be relatively low. However, as we can see in Figure 4.3, the distributions of outputs of CLS-transfer BERTs, CLS-random BERT and BERT are nearly identical. That's why they perform similarly (avgF1: 68.2, 67.9, 68.2, 67.9) in the Table 4.4. These four similar f1 scores indicate that the CLS token does not bring useful stance information to the second BERT model. In this case, we think transferring the CLS tokens is not valid, which contradicts what the author of stance-BERT (Tian et al., 2020) stated. We believe this is because, in a large number of parameters and multi-layer transformers, only changing the CLS embedding (768 parameters) before fine-tuning will not have a significant impact on the fine-tuning result (that is, the CLS embedding of the last layer).

DB-GCN (avgF1: 36.7) fails to meet expectations of performance, performing significantly worse than BERT and even worse than the first baseline BertEmb(avgF1: 56.8). As a result, we believe that relying mainly on sentence syntactic structure information is insufficient for classification tasks, at least the SD task. In DB-GCN, we only use the dependencies between words, ignoring information such as part of speech. At the same time, our method of connecting different sentences is straightforward, and there is no syntactic theoretical basis to demonstrate that the relationship between sentence roots can represent the relationship between different sentences. All of the aforementioned reasons could account for the lower result.

The CLS-concat BERT model, which combines sentence embeddings and CLS embeddings, produces the best results of all the models we discussed. The model can better predict news stances by combining sentence-level and document-level embeddings. However, concatenating the embeddings obtained by GCN does not achieve an improvement and even degrade the performance of the BERT model. We attribute the difference between the results of these two CLS-concat BERTs (avgF1: 69.5, 67.0) to the comparability of the concatenated embeddings. To begin with, each GLOVE embedding is a vector of a single word, and their average is quite different from the paragraph meaning contained in the CLS embedding form BERT. Furthermore, the lengths, training methods, and training models of CLS embedding and Glove embedding are not the same. In contrast, the sentence embeddings in the first CLS-concat model are from Sentence BERT, which is

[5] https://yknzhu.wixsite.com/mbweb
[6] https://en.wikipedia.org/wiki/English_Wikipedia

structurally similar to BERT, indicating that fusion between two similar models may yield good results, but models that are too different do not.

Chapter 5

Conclusions and Future work

In this chapter, we give our answers to the research questions and summarize our work. Subsequently, we present some potential improvements as future work at the end of this chapter, which also serves as the end of the thesis.

5.1 Answers to Research Questions

The main findings of our work can be summarized by the answers to the five research questions introduced in Section 1.2. The answers are elaborated as follows:

Q1. Is it possible to improve the performance of the BertEmb method by identifying better hyperparameters (for example, learning rate and dropout rate)?

A1: **Yes. We show that moderate learning and dropout rates can effectively improve the performance of the BertEmb model by more than 10% points (compared to the original paper (Conforti et al., 2020a)) through experiments on learning rate and dropout rate.**

Q2. How well does the pre-trained model BERT perform on STANDER dataset?

A2: **The result of BERT is as expected. We fine-tune the pre-trained model BERT to this stance detection task. The results of experiments show that BERT model has achieved well performance as expected, which is about 12% points higher than BertEmb after hyperparameter tuning.**

Q3. How far can the CLS token of BERT transfer the stance information across different tasks? Will the CLS-transfer BERT improve the performance of BERT?

A3: **Our experiments with the CLS-transfer BERT prove that the CLS token cannot transfer the information of the stance. The results of random-generated CLS token further prove that changing the CLS token alone has little effect on the fine-tuning.**

Q4. Can we achieve a considerable result on the stance detection task using a dependency-based model?

A4: **The result demonstrates that dependency-based GCN can be used for SD task. However, the result of DB-GCN is below that of the simpler models. We have discussed the following possible reasons in Section 4.2: lack of semantic information and linguistic basis of connecting sentences.**

Q5. Can the model that combines the sentence embeddings in BertEmb and the BERT CLS embedding achieve the state-of-the-art performance and how about the model that combines the syntactic representation and the CLS embedding?

A5: **Yes, concatenating the sentence embeddings and the CLS embedding improves the results by approximately 1.4% points. Concatenating the syntactic representation with CLS embedding, on the other hand, yields no benefits. For the difference in these two results, we give an explanation of the comparability between the representations in Section 4.2, arguing that concatenation requires similar representations, which means that these representations should preferably come from the same or similar model architectures.**

5.2 Contributions

In this thesis, we test multiple deep learning models, especially BERT-based models, on the STANDER dataset. Our main contributions can be summed up as:

- We used BertEmb and improved on the results of the original paper (Conforti et al., 2020a) by selecting a moderate learning rate and dropout rate.
- We fine-tuned the pre-trained model BERT and discovered that it outperforms BertEmb by about 12.7% points.
- We devised two distinct tasks to obtain the transferable CLS tokens for further fine-tuning. Through experiments with various CLS embeddings, we proved that CLS-transfer BERT cannot improve the performance of BERT, indicating that the CLS token is not capable of containing transferable stance-related information.
- We applied the dependency-based system proposed by Žunić et al. (2021). However, we employed an ingenious technique to connect the dependency graph of target and the sentences in news articles, thereby eliminating the issue that different sentences (particularly the target and the articles) do not affect each other in convolution. This system is functional, but it is no better than the baseline.
- We proposed CLS-concat BERT, which combines sentence embeddings and CLS embedding. When compared to other methods, CLS-concat BERT successfully achieves the best f1 score, demonstrating that the combination of information at different levels is useful for stance detection. Simultaneously, combining the node embeddings with CLS embedding does not improve the performance of BERT, indicating that embeddings from models that are too different are not comparable.

5.3 Limitation & Future work

In our work, there are two key aspects where we can improve: the dataset and the model.

We only test our model on STANDER for the dataset. As a result, the following issues may have an impact:

- **Single data type**. It means that we only conduct experiments in the news domain. However, other types of data, such as tweets, are not involved in our work. As a result, this may cause our model to only work on specific datasets.
- **Small amount of data**. STANDER only contains 3,291 pieces of news, and for each merger, the number is even lower. An insufficient amount of data may cause our models to be less robust, which is undesirable for future use.
- **Imbalance of data**. In Section 4.1, we mention that there is an imbalance in STANDER. In subsequent experiments, we also discover that the model rarely predicts unrelated labels, owing to the fact that there are far fewer samples of unrelated labels than of the other labels.

We concentrate on BERT and its associated improvements for the model. For other kinds of models, we only use dependency parsing and the GCN model. This may lead us to simply modify BERT rather than starting from theory in order to develop a widely applicable model.

Given the aforementioned issues, we believe that some improvements can be made in the following areas:

- **Add multiple datasets**. As we mentioned in the Section 2.4, many datasets related to stance detection have been proposed and applied, such as: semeval-16 and FNC. Experiments on these datasets can verify the robustness and practicality of our models.
- **Try different methods**. We can further analyze DS-GCN using different syntactic analysis methods to obtain graph representations, such as: Abstract Meaning Representations (Flanigan et al., 2016) and Discourse Representation Structures (Basile et al., 2011). It is worth noting that the processing object of DRS can be multiple sentences, implying that it has a high potential for solving the problem of plurality input sentences.

Appendix A

Details of Experiments

learning rate	1e-5	5e-5	1e-4	5e-4	1e-3	5e-3
CVS_AET	0.476	0.389	0.457	0.537	0.217	0.217
AET_HUM	0.456	0.558	0.571	0.543	0.215	0.215
ANTM_CI	0.509	0.532	0.571	0.509	0.334	0.161
CI_ESRX	0.351	0.566	0.593	0.582	0.251	0.251

TABLE A.1: The F1 scores of different learning rates on four mergers

dropout rate	0	0.1	0.2	0.3	0.4	0.5
CVS_AET	0.499	0.491	0.488	0.503	0.484	0.505
AET_HUM	0.565	0.558	0.562	0.565	0.548	0.563
ANTM_CI	0.522	0.539	0.528	0.546	0.540	0.529
CI_ESRX	0.575	0.558	0.564	0.552	0.551	0.568

TABLE A.2: The F1 scores of different dropout rates on four mergers

Experiments	Exp1	Exp2	Exp3	Exp4	Exp5
CVS_AET	0.687	0.664	0.721	0.730	0.699
AET_HUM	0.645	0.639	0.657	0.625	0.646
ANTM_CI	0.630	0.648	0.644	0.649	0.633
CI_ESRX	0.725	0.739	0.754	0.750	0.726

TABLE A.3: The F1 scores of five repeated experiments on four mergers by BERT

Experiments	Exp1	Exp2	Exp3	Exp4	Exp5
CVS_AET	0.687	0.664	0.720	0.730	0.698
AET_HUM	0.652	0.640	0.651	0.636	0.634
ANTM_CI	0.647	0.642	0.626	0.638	0.645
CI_ESRX	0.749	0.766	0.725	0.746	0.684

TABLE A.4: The F1 scores of five repeated experiments on four mergers by CLS-random BERT

Experiments	Exp1	Exp2	Exp3	Exp4	Exp5
CVS_AET	0.733	0.735	0.726	0.714	0.726
AET_HUM	0.632	0.649	0.652	0.638	0.649
ANTM_CI	0.637	0.647	0.636	0.657	0.647
CI_ESRX	0.733	0.725	0.718	0.739	0.706

TABLE A.5: The F1 scores of five repeated experiments on four mergers by CLS-transfer BERT1

Experiments	Exp1	Exp2	Exp3	Exp4	Exp5
CVS_AET	0.690	0.727	0.617	0.719	0.714
AET_HUM	0.630	0.647	0.634	0.638	0.647
ANTM_CI	0.628	0.647	0.624	0.646	0.665
CI_ESRX	0.764	0.749	0.731	0.753	0.737

TABLE A.6: The F1 scores of five repeated experiments on four mergers by CLS-transfer BERT2

Experiments	Exp1	Exp2	Exp3	Exp4	Exp5
CVS_AET	0.382	0.368	0.357	0.350	0.372
AET_HUM	0.405	0.411	0.423	0.445	0.439
ANTM_CI	0.300	0.322	0.299	0.319	0.312
CI_ESRX	0.373	0.382	0.396	0.357	0.369

TABLE A.7: The F1 scores of five repeated experiments on four mergers by DB-GCN

Experiments	Exp1	Exp2	Exp3	Exp4	Exp5
CVS_AET	0.725	0.746	0.702	0.728	0.721
AET_HUM	0.643	0.649	0.671	0.643	0.648
ANTM_CI	0.644	0.657	0.637	0.625	0.646
CI_ESRX	0.768	0.764	0.766	0.756	0.763

TABLE A.8: The F1 scores of five repeated experiments on four mergers by CLS-concat BERT(BertEmb)

Experiments	Exp1	Exp2	Exp3	Exp4	Exp5
CVS_AET	0.712	0.722	0.709	0.706	0.713
AET_HUM	0.616	0.609	0.599	0.611	0.604
ANTM_CI	0.640	0.639	0.632	0.637	0.640
CI_ESRX	0.729	0.718	0.727	0.712	0.727

TABLE A.9: The F1 scores of five repeated experiments on four mergers by CLS-concat BERT(GCN)

Bibliography

[1] Aseel Addawood, Jodi Schneider, and Masooda Bashir. "Stance classification of Twitter debates: The encryption debate as a use case". English (US). In: *8th International Conference on Social Media and Society*. ACM International Conference Proceeding Series. Publisher Copyright: © 2017 Copyright is held by the owner/author(s). Publication rights licensed to ACM.; 8th International International Conference on Social Media and Society, SMSociety 2017 ; Conference date: 28-07-2017 Through 30-07-2017. Association for Computing Machinery, July 2017. DOI: 10.1145/3097286.3097288.

[2] Abeer AlDayel and Walid Magdy. "Stance Detection on Social Media: State of the Art and Trends". In: *CoRR* abs/2006.03644 (2020). arXiv: 2006.03644.

[3] Abeer ALDayel and Walid Magdy. "Stance detection on social media: State of the art and trends". In: *Information Processing Management* 58.4 (2021), p. 102597. ISSN: 0306-4573. DOI: https://doi.org/10.1016/j.ipm.2021.102597.

[4] Pranav Anand, Marilyn Walker, Rob Abbott, Jean E. Fox Tree, Robeson Bowmani, and Michael Minor. "Cats Rule and Dogs Drool!: Classifying Stance in Online Debate". In: *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA 2.011)*. Portland, Oregon: Association for Computational Linguistics, June 2011, pp. 1–9.

[5] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. "Neural Machine Translation by Jointly Learning to Align and Translate". In: 2016. arXiv: 1409.0473 [cs.CL].

[6] Valerio Basile and Johan Bos. "Towards Generating Text from Discourse Representation Structures". Dutch. In: *Proceedings of the 13th European Workshop on Natural Language Generation (ENLG)*. 2011/johan.bos/pub004. 2011, pp. 145 – 150.

[7] Emily M. Bender and Alexander Koller. "Climbing towards NLU: On Meaning, Form, and Understanding in the Age of Data". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 5185–5198. DOI: 10.18653/v1/2020.acl-main.463.

[8] Adrian Benton and Mark Dredze. "Using Author Embeddings to Improve Tweet Stance Classification". In: *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 184–194. DOI: 10.18653/v1/W18-6124.

[9] Abraham Bernstein, Claes H. de Vreese, Natali Helberger, Wolfgang Schulz, Katharina Anna Zweig, Christian Baden, Michael A. Beam, Marc P. Hauer, Lucien Heitz, Pascal Jürgens, Christian Katzenbach, Benjamin Kille, Beate Klimkiewicz, Wiebke Loosen, Judith Möller, Goran Radanovic, Guy Shani, Nava Tintarev, Suzanne Tolmeijer, Wouter van Atteveldt, Sanne Vrijenhoek, and Theresa Zueger. "Diversity in News Recommendations". In: *CoRR* abs/2005.09495 (2020). arXiv: 2005.09495.

[10] Douglas Biber and Edward Finegan. "Adverbial stance types in English". In: *Discourse Processes* 11.1 (1988), pp. 1–34. DOI: 10.1080/01638538809544689. eprint: https://doi.org/10.1080/01638538809544689.

[11] Sneha Chaudhari, Varun Mithal, Gungor Polatkan, and Rohan Ramanath. "An Attentive Survey of Attention Models". In: 2021. arXiv: 1904.02874 [cs.LG].

[12] Danqi Chen and Christopher Manning. "A Fast and Accurate Dependency Parser using Neural Networks". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 740–750. DOI: 10.3115/v1/D14-1082.

[13] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation". In: *CoRR* abs/1406.1078 (2014). arXiv: 1406.1078.

[14] Jan Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, KyungHyun Cho, and Yoshua Bengio. "Attention-Based Models for Speech Recognition". In: *CoRR* abs/1506.07503 (2015). arXiv: 1506.07503.

[15] Costanza Conforti, Jakob Berndt, Mohammad Taher Pilehvar, Chryssi Giannitsarou, Flavio Toxvaerd, and Nigel Collier. "STANDER: An Expert-Annotated Dataset for News Stance Detection and Evidence Retrieval". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, Nov. 2020, pp. 4086–4101. DOI: 10.18653/v1/2020.findings-emnlp.365.

[16] Costanza Conforti, Jakob Berndt, Mohammad Taher Pilehvar, Chryssi Giannitsarou, Flavio Toxvaerd, and Nigel Collier. "Will-They-Won't-They: A Very Large Dataset for Stance Detection on Twitter". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 1715–1724. DOI: 10.18653/v1/2020.acl-main.157.

[17] Kareem Darwish, Peter Stefanov, Michaël J. Aupetit, and Preslav Nakov. "Unsupervised User Stance Detection on Twitter". In: *CoRR* abs/1904.02000 (2019). arXiv: 1904.02000.

[18] 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)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423.

[19] S. Eskens, N. Helberger, and J. Moeller. "Challenged by news personalisation: Five perspectives on the right to receive information". English. In: *Journal of Media Law* 9.2 (2017). None, pp. 259–284. ISSN: 1757-7632. DOI: 10.1080/17577632.2017.1387353.

[20] William Ferreira and Andreas Vlachos. "Emergent: a novel data-set for stance classification". In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, June 2016, pp. 1163–1168. DOI: 10.18653/v1/N16-1138.

[21] Jeffrey Flanigan, Chris Dyer, Noah Smith, and Jaime Carbonell. "Generation from Abstract Meaning Representation using Tree Transducers". In: Jan. 2016, pp. 731–739. DOI: 10.18653/v1/N16-1087.

[22] Guillaume Gadek, Josefin Betsholtz, Alexandre Pauchet, Stéphan Brunessaux, Nicolas Malandain, and Laurent Vercouter. "Extracting Contextonyms from Twitter for Stance Detection". In: Jan. 2017, pp. 132–141. DOI: 10.5220/0006190901320141.

[23] Andrea Galassi, Marco Lippi, and Paolo Torroni. "Attention in Natural Language Processing". In: *IEEE Transactions on Neural Networks and Learning Systems* 32.10 (2021), 4291–4308. ISSN: 2162-2388. DOI: 10.1109/tnnls.2020.3019893.

[24] Akash Kumar Gautam, Puneet Mathur, Rakesh Gosangi, Debanjan Mahata, Ramit Sawhney, and Rajiv Ratn Shah. "#MeTooMA: Multi-Aspect Annotations of Tweets Related to the MeToo Movement". In: *CoRR* abs/1912.06927 (2019). arXiv: 1912.06927.

[25] Shalmoli Ghosh, Prajwal Singhania, Siddharth Singh, Koustav Rudra, and Saptarshi Ghosh. "Stance Detection in Web and Social Media: A Comparative Study". In: *Experimental IR Meets Multilinguality, Multimodality, and Interaction*. Ed. by Fabio Crestani, Martin Braschler, Jacques Savoy, Andreas Rauber, Henning Müller, David E. Losada, Gundula Heinatz Bürki, Linda Cappellato, and Nicola Ferro. Cham: Springer International Publishing, 2019, pp. 75–87. ISBN: 978-3-030-28577-7.

[26] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. *Neural Message Passing for Quantum Chemistry*. 2017. DOI: 10.48550/ARXIV.1704.01212.

[27] Simone Giorgioni, Marcello Politi, Samir Salman, Roberto Basili, and Danilo Croce. "UNITOR @ Sardistance2020: Combining Transformer-based Architectures and Transfer Learning for Robust Stance Detection". In: *EVALITA*. 2020.

[28] Andreas Hanselowski, Avinesh PVS, Benjamin Schiller, Felix Caspelherr, Debanjan Chaudhuri, Christian M. Meyer, and Iryna Gurevych. "A Retrospective Analysis of the Fake News Challenge Stance-Detection Task". In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 1859–1874.

[29] Kazi Saidul Hasan and Vincent Ng. "Stance Classification of Ideological Debates: Data, Models, Features, and Constraints". In: *Proceedings of the Sixth International Joint Conference on Natural Language Processing*. Nagoya, Japan: Asian Federation of Natural Language Processing, Oct. 2013, pp. 1348–1356.

[30] Natali Helberger. "On the Democratic Role of News Recommenders". In: *Digital Journalism* 7 (2019), pp. 1012 –993.

[31] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (Nov. 1997), pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. eprint: https://direct.mit.edu/neco/article-pdf/9/8/1735/813796/neco.1997.9.8.1735.pdf.

[32] Harold Hotelling. "Analysis of a complex of statistical variables into principal components." In: *Journal of Educational Psychology* 24 (1933), pp. 498–520.

[33] K.H. Jamieson, J.N. Cappella, and D.A.S.C.J.N. Cappella. *Echo Chamber: Rush Limbaugh and the Conservative Media Establishment*. OUP E-Books. Oxford University Press, 2008. ISBN: 9780195366822.

[34] Dan Jurafsky and James H. Martin. *Speech and language processing : an introduction to natural language processing, computational linguistics, and speech recognition*. 2021.

[35] Kornraphop Kawintiranon and Lisa Singh. "Knowledge Enhanced Masked Language Model for Stance Detection". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, June 2021, pp. 4725–4735. DOI: 10.18653/v1/2021.naacl-main.376.

[36] Thomas N. Kipf and Max Welling. "Semi-Supervised Classification with Graph Convolutional Networks". In: *CoRR* abs/1609.02907 (2016). arXiv: 1609.02907.

[37] Dilek Küçük and Fazli Can. "Stance Detection: A Survey". In: *ACM Comput. Surv.* 53.1 (2020). ISSN: 0360-0300. DOI: 10.1145/3369026.

[38] Kostiantyn Kucher, Carita Paradis, and Andreas Kerren. "Visual Analysis of Sentiment and Stance in Social Media Texts". In: *EuroVis 2018 - Posters*. The Eurographics Association, 2018. ISBN: 978-3-03868-065-9. DOI: 10.2312/eurp.20181127.

[39] John Lawrence and Chris Reed. "Argument Mining: A Survey". In: *Computational Linguistics* 45.4 (Jan. 2020), pp. 765–818. ISSN: 0891-2017. DOI: 10.1162/coli_a_00364. eprint: https://direct.mit.edu/coli/article-pdf/45/4/765/1847520/coli_a_00364.pdf.

[40] Yingjie Li and Cornelia Caragea. "Multi-Task Stance Detection with Sentiment and Stance Lexicons". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 6299–6305. DOI: 10.18653/v1/D19-1657.

[41] Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu. "A Survey of Transformers". In: *CoRR* abs/2106.04554 (2021). arXiv: 2106.04554.

[42] Wei-Hao Lin, Theresa Wilson, Janyce Wiebe, and Alexander Hauptmann. "Which Side are You on? Identifying Perspectives at the Document and Sentence Levels". In: *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*. New York City: Association for Computational Linguistics, June 2006, pp. 109–116.

[43] Bernard Manin. "On Legitimacy and Political Deliberation". In: *Political Theory* 15.3 (1987), pp. 338–368. DOI: 10.1177/0090591787015003005. eprint: https://doi.org/10.1177/0090591787015003005.

[44] M.-C Marneffe, T. Dozat, N. Silveira, K. Haverinen, F. Ginter, Joakim Nivre, and C.D. Manning. "Universal Stanford Dependencies: A cross-linguistic typology". In: *Proceedings of the 9Th International Conference on Language Resources and Evaluation (LREC)* (Jan. 2014), pp. 4585–4592.

[45] Amita Misra and Marilyn Walker. "Topic Independent Identification of Agreement and Disagreement in Social Media Dialogue". In: *Proceedings of the SIGDIAL 2013 Conference*. Metz, France: Association for Computational Linguistics, Aug. 2013, pp. 41–50.

[46] Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. "A Dataset for Detecting Stance in Tweets". In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. Portorož, Slovenia: European Language Resources Association (ELRA), May 2016, pp. 3945–3952.

[47] Akiko Murakami and Rudy Raymond. "Support or Oppose? Classifying Positions in Online Debates from Reply Activities and Opinion Expressions". In: *Coling 2010: Posters*. Beijing, China: Coling 2010 Organizing Committee, Aug. 2010, pp. 869–875.

[48] E. Pariser. *The Filter Bubble: What The Internet Is Hiding From You*. Penguin Books Limited, 2011. ISBN: 9780141969923.

[49] Jeffrey Pennington, Richard Socher, and Christopher Manning. "GloVe: Global Vectors for Word Representation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. DOI: 10.3115/v1/D14-1162.

[50] Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. "Pre-trained Models for Natural Language Processing: A Survey". In: *CoRR* abs/2003.08271 (2020). arXiv: 2003.08271.

[51] Ashwin Rajadesingan and Huan Liu. "Identifying Users with Opposing Opinions in Twitter Debates". In: Feb. 2014. ISBN: 978-3-319-05578-7. DOI: 10.1007/978-3-319-05579-4_19.

[52] Gayathri Rajendran, Bhadrachalam Chitturi, and Prabaharan Poornachandran. "Stance-In-Depth Deep Neural Approach to Stance Classification". In: *Procedia Computer Science* 132 (2018). International Conference on Computational Intelligence and Data Science, pp. 1646–1653. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2018.05.132.

[53] Nils Reimers and Iryna Gurevych. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks". In: *CoRR* abs/1908.10084 (2019). arXiv: 1908.10084.

[54] Myrthe Reuver, Antske Fokkens, and Suzan Verberne. "No NLP Task Should be an Island: Multi-disciplinarity for Diversity in News Recommender Systems". In: *Proceedings of the EACL Hackashop on News Media Content Analysis and Automated Report Generation*. Online: Association for Computational Linguistics, Apr. 2021, pp. 45–55.

[55] Anirban Sen, Manjira Sinha, Sandya Mannarswamy, and Shourya Roy. "Stance Classification of Multi-Perspective Consumer Health Information". In: *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data*. CoDS-COMAD '18. Goa, India: Association for Computing Machinery, 2018, 273–281. ISBN: 9781450363419. DOI: 10.1145/3152494.3152518.

[56] Bilin Shao, Xiaojun Li, and Genqing Bian. "A survey of research hotspots and frontier trends of recommendation systems from the perspective of knowledge graph". In: *Expert Systems with Applications* 165 (2021), p. 113764. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2020.113764.

[57] Umme Siddiqua, Abu Nowshed Chy, and Masaki Aono. "Stance Detection on Microblog Focusing on Syntactic Tree Representation". In: June 2018, pp. 478–490. ISBN: 978-3-319-93802-8. DOI: 10.1007/978-3-319-93803-5_45.

[58] Umme Aymun Siddiqua, Abu Nowshed Chy, and Masaki Aono. "Tweet Stance Detection Using an Attention based Neural Ensemble Model". 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)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 1868–1873. DOI: 10.18653/v1/N19-1185.

[59] Vasiliki Simaki, Carita Paradis, and Andreas Kerren. "Stance classification in texts from blogs on the 2016 British referendum". In: *International Conference on Speech and Computer*. Springer. 2017, pp. 700–709.

[60] Parinaz Sobhani, Diana Inkpen, and Xiaodan Zhu. "A Dataset for Multi-Target Stance Detection". In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 551–557.

[61] 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*. Los Angeles, CA: Association for Computational Linguistics, June 2010, pp. 116–124.

[62] Swapna Somasundaran and Janyce Wiebe. "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*. Suntec, Singapore: Association for Computational Linguistics, Aug. 2009, pp. 226–234.

[63] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15.56 (2014), pp. 1929–1958.

[64] Matt Thomas, Bo Pang, and Lillian Lee. "Get out the vote: Determining support or opposition from Congressional floor-debate transcripts". In: *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. Sydney, Australia: Association for Computational Linguistics, July 2006, pp. 327–335.

[65] Lin Tian, Xiuzhen Zhang, Yan Wang, and Huan Liu. "Early Detection of Rumours on Twitter via Stance Transfer Learning". In: *Advances in Information Retrieval*. Ed. by Joemon M. Jose, Emine Yilmaz, João Magalhães, Pablo Castells, Nicola Ferro, Mário J. Silva, and Flávio Martins. Cham: Springer International Publishing, 2020, pp. 575–588. ISBN: 978-3-030-45439-5.

[66] Amine Trabelsi and Osmar Zaïane. "Unsupervised model for topic viewpoint discovery in online debates leveraging author interactions". In: June 2018.

[67] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention Is All You Need". In: *CoRR* abs/1706.03762 (2017). arXiv: 1706.03762.

[68] Marilyn Walker, Pranav Anand, Rob Abbott, and Ricky Grant. "Stance Classification using Dialogic Properties of Persuasion". In: *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Montréal, Canada: Association for Computational Linguistics, June 2012, pp. 592–596.

[69] Marilyn Walker, Jean Fox Tree, Pranav Anand, Rob Abbott, and Joseph King. "A Corpus for Research on Deliberation and Debate". In: *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*. Istanbul, Turkey: European Language Resources Association (ELRA), May 2012, pp. 812–817.

[70] Feng Wang and David M. J. Tax. "Survey on the attention based RNN model and its applications in computer vision". In: *CoRR* abs/1601.06823 (2016). arXiv: 1601.06823.

[71] Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. "K-Adapter: Infusing Knowledge into Pre-Trained Models with Adapters". In: *CoRR* abs/2002.01808 (2020). arXiv: 2002.01808.

[72] Jiaming Xu, Suncong Zheng, Jing Shi, Yiqun Yao, and Bo Xu. "Ensemble of Feature Sets and Classification Methods for Stance Detection". In: *NLPCC/ICCPOL*. 2016.

[73] Yiping Yang and Xiaohui Cui. "Bert-Enhanced Text Graph Neural Network for Classification". In: *Entropy* 23.11 (2021), p. 1536.

[74] Wenpeng Yin and Dan Roth. "TwoWingOS: A Two-Wing Optimization Strategy for Evidential Claim Verification". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 105–114. DOI: 10.18653/v1/D18-1010.

[75] Guido Zarrella and Amy Marsh. "MITRE at SemEval-2016 Task 6: Transfer Learning for Stance Detection". In: *CoRR* abs/1606.03784 (2016). arXiv: 1606.03784.

[76] Qiang Zhang, Emine Yilmaz, and Shangsong Liang. "Ranking-Based Method for News Stance Detection". In: *Companion Proceedings of the The Web Conference 2018*. WWW '18. Lyon, France: International World Wide Web Conferences Steering Committee, 2018, 41–42. ISBN: 9781450356404. DOI: 10.1145/3184558.3186919.

[77] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. "Graph neural networks: A review of methods and applications". In: *AI Open* 1 (2020), pp. 57–81. ISSN: 2666-6510. DOI: https://doi.org/10.1016/j.aiopen.2021.01.001.

[78] Anastazia Žunić, Padraig Corcoran, and Irena Spasić. "Aspect-based sentiment analysis with graph convolution over syntactic dependencies". In: *Artificial Intelligence in Medicine* 119 (2021), p. 102138. ISSN: 0933-3657. DOI: https://doi.org/10.1016/j.artmed.2021.102138.