IMPROVING STANCE AND BIAS DETECTION IN TEXT BY MODELING SOCIAL CONTEXT

by

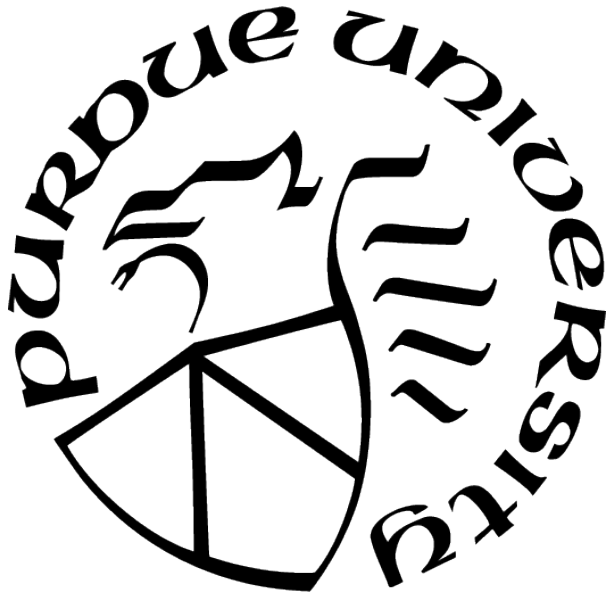
Chang Li

A Dissertation

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy



Department of Computer Science

West Lafayette, Indiana

May 2021

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Dan Goldwasser, Chair

Department of Computer Science

Dr. Jean Honorio

Department of Computer Science

Dr. Jennifer Neville

Department of Computer Science

Dr. Ming Yin

Department of Computer Science

Approved by:

Dr. Kihong Park

To my parents Xuejun Li and Jing Liu,
my wife Qian Chen and my daughter Alyssa Chen Li.

ACKNOWLEDGMENTS

First and foremost, I want to thank my advisor, Dr. Dan Goldwasser, for leading me into the field of machine learning and natural language processing and guiding me to explore and identify the problems discussed in this dissertation. From years of experience working with him, I really learned a lot about not only the model design and evaluation, but more importantly, the altitude and methodology toward research. He is always open to new ideas and good at dealing with the difficulties we are facing from different angles. He encourages me to focus on providing insights about the task, instead of just the tiny performance changes. All of these will be of great use in my future career as well.

I am very pleased to have Dr. Jennifer Neville, Dr. Jean Honorio, and Dr. Ming Yin serve on my final exam committee. They lent me their expertise to provide feedback and suggestions as I complete this dissertation. My appreciation also goes to many other faculty members in the Department of Computer Science, for the courses they offered and the interesting talks they gave.

I am also fortunate to be able to work with other members of the NLP group: Kristen Johnson, Xiao Zhang, Maria Leonor Pacheco, I-Ta Lee, Aldo Porco, Rajkumar Pujari, Shamik Roy, Maryam Davoodi, Nikhil Mehta, Prerit Gupta, Tunazzina Islam, Younghun Lee, and Abhishek Sharma. I want to thank them for their discussion, inspiration, and collaboration on research ideas. Many other people helped me during my graduate study at Purdue and made my time here an unforgettable experience. My gratitude goes to all of them as well.

All of my love goes to my amazing family who provides endless support and encouragement for me to work through all the difficult times. I want to thank my parents, Xuejun Li and Jing Liu, who taught me about the importance of hard work and curiosity. I still remember visiting my parents' offices when I was a little kid, where I get access to computers for the first time. To my wife, Qian Chen, thank you for your love and support. We learn from each other and grow to better persons together. I am grateful to have you on my side now and forever.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Understanding the stance and bias reflected in the text is an essential part of achieving machine intelligence. Successful detection of them will not only provide us with a huge amount of insights about public opinion and sentiment but also lay the foundation for serving the most reliable and accurate information to meet people's needs. Traditionally, this problem is often modeled merely as a text classification task. However, it is highly challenging due to the huge variation involved in opinion expressions as well as the need for background knowledge and commonsense reasoning. Meanwhile, just as we want to understand a word based on its context, we also have social contexts for a piece of text, including its author, its sharing pattern online, and its narrative about notable entities and events. These important factors have been largely ignored in previous work. In this dissertation, we tackle this problem by proposing three novel neural network models. Each of them capturing one important social context that can provide rich signals for the detection of stance and bias. The first model aims at predicting the stance of posts from online debate forums. We proposed a structured representation learning model that can make use of the authorship relation and conversational structure in debates. It takes advantage of both collective relational classification methods and distributed representation learning. The performance boost after the inference that is defined over the embedding space. The second model focuses on bias detection in news articles. We identify the social context available for many news articles, which is the engagement pattern over social media. We construct the social information graph involving news articles and apply GCN to aggregate local neighborhood information when generating graph representations. A joint text and graph model is then used to propagate information from both directions. Experimental results show even little social signals can lead to significant improvement. Last but not least, we explore the situation where we cannot obtain context information for test articles. In this case, we designed pre-training strategies that can inject external knowledge about entities, frames, and sharing users into the text model so that it can better identify relevant text spans for bias classification. We also show larger performance gains can be achieved when the supervision is limited, demonstrating the advantage of our model in such cases. Empirical results demonstrate that our models

significantly outperform competitive baseline methods, by more accurately regularize the text representation given additional signals available in the social context and by identifying the portion of the text where stance and bias are most readily perceptible.

1. INTRODUCTION

The last decade has witnessed a tremendous advance in the way information is generated and disseminated. Instead of a few dedicated sources that collect and publish content for the mass to consume, social platforms now provide the means for any user to distribute their content, resulting in a sharp increase in the number of information outlets and articles covering news events and controversial issues. As a direct result of this process, the information provided by various sources is often shaped by their underlying perspectives, interests, and ideologies. Understanding the stances, or the underlying ideologies, expressed in these articles is a highly challenging but important task. It can help provide insight into current political discourse and help gauge public sentiment on policy issues on a large scale. Meanwhile, identifying the perspective difference and making it explicit can help strengthen trust in the newly-formed information landscape and ensure that all perspectives are well represented. Moreover, It can help lay the foundation for the automatic detection of false content and rumors so that social platforms can take actions to prevent the spread of such information.

Among all kinds of online content, we focus on the politically related text, specifically debates in online forums and online news articles. Unlike traditional document classification tasks, documents on a controversial issue or a news event usually utilize highly similar vocabulary. Moreover, texts from the opposite sides may even agree with each other to some extent although different aspects of a topic may be emphasized [1], [2]. As a result, despite the increasing capability of the current text model, existing approaches modeling textual information alone still get limited performance in either stance or bias prediction [3]–[6]. However, it is interesting to see that there is rich context information available in these online environments. For example, CreateDebate[1] is an online debate forum. Figure 1.1 shows an example of debates on this website. Users express their opinion on a certain topic ("Legalization of Marijuana" in this example) and reply to others to support or dispute their arguments. The information of authors and conversational structures between posts can definitely contribute to the prediction of the stance.

[1]↑https://createdebate.com

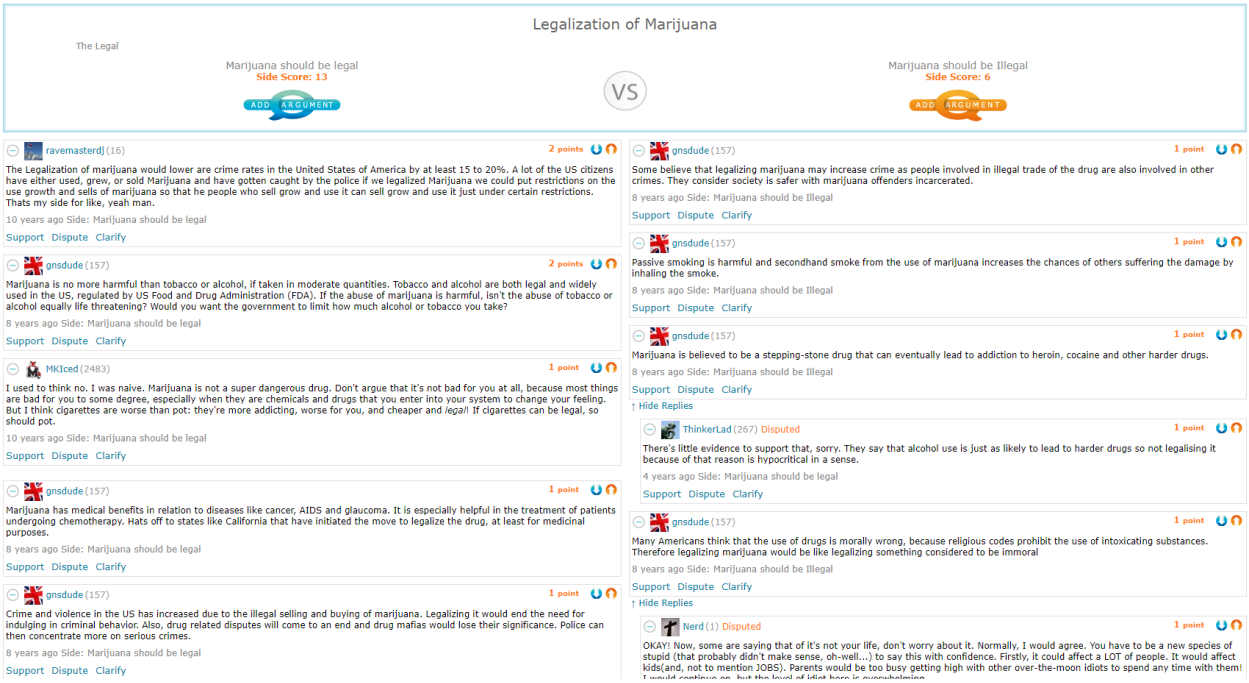


Figure 1.1. Snapshot of a Debate about Marijuana Legalization on CreateDebate Forum.

Another case involves the propagation of news articles on Twitter. Many people nowadays use Twitter as their major source of the news feed and retweet to share what they are interested in. Fig. 1.2 shows two news tweets by Fox News and CNN respectively. We also see the interaction statistics for these tweets at the bottom, like retweets. In fact, the sharing pattern of news articles on Twitter provides knowledge in understanding the relationship between different users and articles, thus can also help to determine their biases.

Based on these findings, we propose to model both the textual content and social context of documents of interest. Every piece of text comes with its own context. Intuitively, we may infer the label of text based on who is the author, how people with different backgrounds react to it, and what entities and relations are mentioned, and how they are described. We design models that take advantage of each of these social contexts and show empirically that how they can help in understanding the stance and bias in the text better. The intuition is to encourage information flow from relevant social context to text by representing them in the corresponding embedding spaces. We briefly describe our proposed models in the following paragraphs.

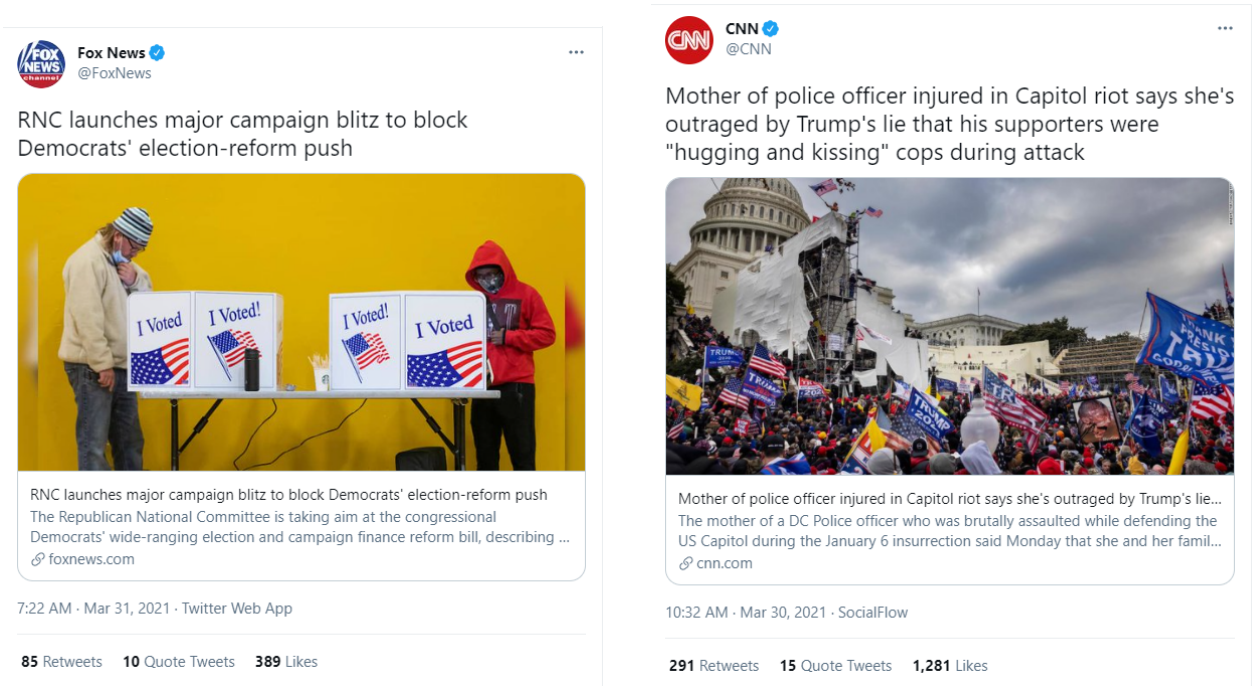


Figure 1.2. Snapshots of News Articles on Twitter.

The first context we consider is naturally the author. Usually, the content in the text reflects the author's opinion on a certain topic. We aim at predicting the stance of text on controversial issues and evaluate our model over the Internet Argumentation Corpus [7]. This is a dataset crawled from online debate forums where users voice their opinions and engage with other users holding different views. We suggest viewing the stance prediction task as a representation learning problem, and embed the text, authors, and attributes jointly based on their interactions. This joint embedding strategy not only helps to regulate the text vectors given the profile and interaction of authors but also makes inference possible as we can also predict the stance of authors directly. Two sets of constraints are applied, namely author constraints and consecutive constraints. Author constraints force agreement between all the posts by the same author. Consecutive constraints are soft ones that encourage disagreement in stances between neighboring posts in a debate thread, based on our findings that people tend to disagree with the post they reply to. Global inference with the above two constraints is utilized at both test and training time to further regulate the learned

representations. Experimental results show that our model can achieve significantly better results compared to previous competitive collective classification approaches.

While it is great to make use of author profile and interaction to enhance the model's ability to predict preference, most of the time we have little information about the author, especially for many online news outlets where we may not even know the author. However, people nowadays often propagate what they like on social platforms, such as Twitter or Facebook. This provides another source of supervision for text classification as people are more likely to share articles whose opinions or stances they agree with. At the same time, people may also follow or connect with notable political figures they are interested in on social medias. To capture how information is disseminated in social networks. We use Graph Convolutional Networks (GCN, [8], [9]), a recently proposed neural architecture for representing relational information, to capture the documents' sharing (retweet) pattern. We show that social information can be used effectively as a source of distant supervision, and when direct supervision is available, even little social information can significantly improve performance.

The last factor we consider is the linguistic information, including entity mentions and frame usage in the documents. News articles often cover real-world events that are related to famous political figures. The same event can also be viewed from different angles and thus completely opposite conclusions can be reached. In order to detect bias in news articles, we believe it is important and effective to examine how entities and their relations are described and which aspects of an event are emphasized. Given that the cost of obtaining supervision is hard, we propose to utilize the social and linguistic information available as self-supervision signals to pre-train the text models so that we can inject knowledge about entities, frames, and biases into them. In the end, we can aggregate the text models pre-trained with various information to make the prediction of the underlying ideologies of a news article through an ensemble. Extensive experiments have been conducted to show that our proposed model even outperforms more advanced textual baselines. Moreover, our proposed method is not model-dependent and can be readily applied to other newly developed models.

The remaining of the dissertation is organized as follows. Chapter 2 presents preliminaries and related technical background. Chapter 3 introduces a structured representation

learning method for stance detection in the online debate setting. Chapter 4 covers text and graph joint model for incorporating sharing pattern on social media for news bias detection. Chapter 5 discusses the pre-training framework proposed to injecting knowledge about entities, frames, and biases into the text model to generate better representation to identify the perspective of news articles. Chapter 6 concludes this dissertation and discusses possible directions for future work.

2. BACKGROUND

In this chapter, we present some background knowledge that is useful in understanding the models and frameworks proposed in this dissertation. This is intended to be a brief review of relevant topics so that readers can quickly recall given that they have previous experience. For further details about each topic, please refer to the corresponding materials that are dedicated to them.

2.1 Representation Learning

Representation learning (or feature learning, we use them interchangeably in this dissertation) refers to a set of techniques that automatically discover representations from raw data for downstream tasks, including classification and clustering. This is in contrast with traditional manual feature engineering where people design features based on domain knowledge of the target task. Due to its efficiency and quality, Feature learning has now been widely used in almost every domain in machine learning. People can feed raw data directly to the models without putting effort into the tedious steps of feature engineering.

2.1.1 Unsupervised Representation Learning

To goal of unsupervised representation learning is often to discover low dimensional features that capture the structures of the high dimensional raw data. Some of the classic unsupervised learning approaches fall into this category as they try to identify the similarity and differences between the features of all examples. K-means clustering [10] partition the observations into k clusters in which each observation belongs to the cluster with the closest mean (known as the cluster centroid). A one-hot indicator function for the nearest cluster or the distance from a data point to all centroids can be used to generate new features. Principal component analysis (PCA) [11], a widely used linear dimension reduction method, provides another example. It finds the principal components of a collection of data points and uses them to determine the orthonormal basis in which all individual dimensions are linearly uncorrelated. The principal components are selected based on the amount of variation in the

data captured in descending order. So we can project the data to a new space determined by the first a few principal components to obtain lower-dimensional data with minimal loss of information. A nonlinear dimensionality reduction technique, t-SNE, is useful to convert high-dimensional data into a space of two or three dimensions for visualization. It is often used to visualize representations learned by an artificial neural network, for example, the hidden representations for classification tasks. In the field of natural language processing, it is very popular to train word representation in an unsupervised way on a huge corpus. This includes the traditional context-free representations, like word2vec[12] and glove [13], and the recently proposed contextualized word representations, like ELMo [14] or BERT [15]. These pre-trained representations are then used in various tasks ranging from text classification to text generation, sometimes even get updated during the supervised training phase to adapt to the specific task.

2.1.2 Supervised Representation Learning

Supervised representation learning can use the data label to provide feedback to the feature learning process. It tries to reduce the error which captures how well the learned representation can be used to produce the label. Artificial neural networks are the most popular method in this category. The famous example comes from the computer vision model for the face recognition task. A deep neural network model is shown to learn hierarchical feature representations as shown in Fig., where each layer learns a distinct set of features based on the previous layer's output. The raw face image is fed as input to the network. While the following hidden layers may learn to identify edges, corners, and contours, parts, until the identity of the person. In natural language processing, we see the same trend where deep models, like BERT, also encode a rich hierarchy of linguistic information, with surface features at the bottom, syntactic features in the middle, and semantic features at the top [16]. Deeper layers may be required when the long-distance dependency is needed for the task, e.g. to track subject-verb agreement.

2.1.3 Representation Learning for Graph Data

With the growing popularity of the word2vec model among text, the same technique is borrowed to learn representations for graph data. One of the first attempts is DeepWalk [17], which generates random walks starting from each node in the graph and considers them as "sentences" in text. They are then fed to the word2vec model to obtain node representations that capture the similarity between them based on the graph structure. Node2vec [18] generalized prior work and designed a biased random walk procedure where users can control the way to explore neighborhoods. This added flexibility leads to the learning of richer node representations. More recently, a number of graph neural network models are proposed to further push the capability to capture rich signals covered by a graph structure. Graph Convolutional Network (GCN) is one of the most popular among them due to its effectiveness and efficiency in message passing through neighborhood aggregation.

2.2 Integer Linear Programming

Linear programming (LP) is a special case of mathematical optimization which has linear objective functions and linear constraints. The feasible set is a polytope, a convex, connected set defined as the intersection of finitely many half-spaces. It is widely used to obtain the best outcome in a variety of domains, such as transportation, manufacturing, and engineering. In matrix-vector notation, a linear program in standard form [19] will be written as

$$\text{minimize} \quad z = c^T x \tag{2.1}$$

$$\text{subject to} \quad Ax = b \tag{2.2}$$

$$x \geq 0 \tag{2.3}$$

with $b \geq 0$. Here x and c are vectors of length n, b is a vector of length m, and A is an $m \times n$ matrix called the constraint matrix. Note that although there are only equality

constraints in the standard form, it is possible to have inequality constraints in a linear program. For instance, given the constraint

$$ax \leq b \tag{2.4}$$

we can convert it to an equality constraint by including a slack variable s:

$$ax + s = b \tag{2.5}$$

together with the constraint $s \geq 0$.

For linear programming, all variables are continuous. However, in some situations, it only makes sense for them to take on integer values. For example, the number of people assigned to a job. We refer to the case in which all of the variables in a linear program are restricted to be integers as integer linear programming (ILP). If only some of the variables in the problem are restricted to be integers, they are called mixed integer programming (MIP).

Discrete problems are often harder to solve because the behavior of the objective and constraints may change dramatically when we move from one feasible point to another, even if these two points are very close in the space. In fact, ILP is NP-Complete. So it does mean that we often need to use approximate algorithms for ILP to solve the problem within a reasonable time. For instances with a small number of variables, exact algorithms can be used.

2.3 Graph Neural Networks

Graphs are a kind of data structure that models a set of objects and their relationships. Many real-world data can be modeled using graphs, including social networks, chemical structures, and knowledge graphs. Due to its popularity and distinct characteristics involved in the way information is presented, researches of graph analysis with machine learning have been receiving more and more attention recently [20]. Among them, graph neural networks (GNNs), deep learning based methods that operate on graphs are the most popular.

GNNs work on a principle called message passing. At each time step, information is propagated through the graph structure. This phase is defined in terms of message function M_t and vertex update function U_t. At time t, each vertex v aggregate the messages received to generate message representation

$$m_v^{t+1} = \sum_{w \in \mathcal{N}_v} M_t(h_v^t, h_w^t, \mathrm{e}_{vw}) \tag{2.6}$$

where h_v^t and h_w^t are the hidden states for vertex v and w respectively at time step t, e_{vw} is the features on the edge from node v to w, and $\mathcal{N}_v$ is the set of local neighbors of node v. The hidden states are updated given the new messages

$$h_v^{t+1} = U_t(h_v^t, m_v^{t+1}) \tag{2.7}$$

The final representations for nodes in the graph are the hidden states obtained after T time steps, i.e. h_v^T.

A graph convolutional network (GCN) is a special instance of the above abstraction. The message function M_t is an average of hidden states of nodes in the local neighborhood

$$m_v^{t+1} = \frac{\sum_{w \in \mathcal{N}_v} h_w^t}{|\mathcal{N}_v|} \tag{2.8}$$

While the vertex update function W_t is a linear transformation followed by an activation layer

$$h_v^{t+1} = \sigma(m_v^{t+1} W_t) \tag{2.9}$$

where W_t is the weight matrix for linear transformation and σ is the activation function, e.g. ReLU.

2.4 Hierarchical Attention Networks

A hierarchical attention network (HAN, [21]) is a text representation model that is designed for documents. It has two distinctive characteristics: (1) it has a hierarchical structure

that mirrors the structure of documents; (2) it has two levels of attention mechanisms at the word-level and sentence-level, enabling it to assign different weight to content when generating the sentence and document representations. We introduce the components in the design of HAN in this section.

2.4.1 Recurrent Neural Network

Recurrent neural networks (RNNs) are a class of artificial neural network models that uses sequential data or time-series data. RNNs can process variable-length sequences of inputs with their internal state updated at each position to represent the sub-sequence already processed. Therefore the computation takes into account historical information. It is very suitable to process text data given that it has a sequence structure.

Recurrent neural networks leverage backpropagation through time (BPTT) algorithm to determine the gradients and update their parameters. As the length of the sequence increase, the problem of exploding gradients and vanishing gradients may occur since they are computed as the multiplication of many values. When the gradient is too small, it continues to become smaller through the sequence until they reach zero and become insignificant. The model will stop learning at that time. Exploding gradients occur when the gradient is too large. The model weights will grow too large and eventually be represented as NaN.

Long short-term memory (LSTM) networks are a popular RNN variant as a solution to the vanishing gradient problem. They designed three gates, namely input gate, output gate and forget gate to control the flow of information that is useful for the prediction. These additions can help to capture long-term dependencies and also alleviate the vanishing gradient problem. The original LSTM process the text in one direction, e.g. from left to right. In order to capture the context from both directions, a bidirectional LSTM (BiLSTM) is proposed by aggregating the hidden states from LSTM networks applied in opposite directions.

2.4.2 Hierarchical Structure and Attention Mechanism

In order to obtain representation for a sentence, one of the early methods is to compute the average word embeddings for all words in the sentence. Similarly, one would think about

averaging the sentence representation for a document. That is how the hierarchical structure appears. However, instead of using the word embeddings directly, we can use BiLSTM to model the context for a word or a sentence depending on the level. This enables the model to adjust the meaning of the text given various contexts. Additionally, simple averaging may not be the best choice since not all words or sentences are equally important, especially when different tasks or objectives are considered. The attention mechanism can be a better choice here as it can assign different weights automatically after training with supervision.

3. CASE I: IMPROVING ONLINE DEBATE STANCE PREDICTION WITH STRUCTURED REPRESENTATION LEARNING

In this chapter, we consider the stance prediction problem in an online debate forum setting. We first explain the importance of understanding the opinions expressed on these online discussion platforms. Then the social context available, including users' profile information and interactions, is discussed. We cast the stance prediction task as a structured representation learning problem, in order to take advantage of both collective relational classification and distributed representation learning methods. We report experimental results at the end to show the effectiveness of our proposed model.

3.1 Introduction

In recent years, social media platforms play an increasingly important role in shaping political discourse. Online debate forums allow users to voice their opinions and engage with other users holding different views. Understanding the interactions between the users on these platforms can help provide insight into current political discourse, argumentation strategies, and can help gauge public sentiment on policy issues on a large scale. The importance of understanding debate dialog has motivated significant research efforts [3], [4], [22]–[27].

In this chapter, we focus on stance prediction, automatically identifying the stance expressed in debate posts on various issues. For example, Figure 3.1 describes a short debate dialog about marijuana legalization between three users (denoted a_1, a_2, a_3). The content associated with each user is classified as supportive of legalization (PRO), or not (CON).

Early work took a text classification approach [22], [23], classifying individual posts using a rich feature set. Since debate posts are not written in isolation, but rather express the conversational interactions between users, modeling these interactions can help alleviate some of the difficulty of this task. More recent work takes a collective classification approach [3], [4], which models the dependencies between authors and their content and captures the

Author	Discussion Text	Stance
a_1	*"There are no deaths related to the actual use for marijuana this past year"*	Pro
a_2	*"Whether it kills people or not, it still is harmful to your body."*	Con
a_3	*"There are many things that are harmful. Alcohol is more harmful than marijuana.*	Pro
a_2	*"But that doesn't mean marijuana should be made legal because the other two are."*	Con
a_3	*"If we were to make everything illegal because it was harmful we would be living with nothing."*	Pro

Figure 3.1. Example of Excerpts from a Debate between Three Users about Marijuana Legalization.

debate structure. For example, the interactions between users can express *agreements* (or *disagreements*), which would entail a similar (or different) stance prediction associated with their content. The stance decision can also be considered as a user-level decision, as users tend to maintain the same stance throughout the debate, forcing stance agreement between all of their posts. Unfortunately, despite these efforts, stance classification remains a challenging problem.

In this chapter, we suggest a new approach for representing the structural dependencies of debate dialogs, by taking a *structured representation learning* approach. Intuitively, our system is designed to exploit the advantages of collective relational classification methods (often discussed in the context of graphical models) and distributed representation learning (often discussed in the context of deep learning and embedding). We suggest a method for combining the two approaches in a single framework that can exploit their complementary strengths.

Our key intuition is that the embedding function can be trained to respect the relevant structural dependencies. We jointly embed all the debate objects (i.e., authors, stances, and textual posts), by considering the relationships between these objects. For example, we model stance classification as a relationship between a post and a given stance label, by measuring the similarity between their embedded representations. We can also model the relationships between input objects; the similarity between the representations of two posts would entail an agreement between the labels associated with them, thus allowing us to perform collective classification over all the input instances. Specifically, we define the factor graph corresponding to the dependencies between stance predictions in a debate thread and use the similarity between the embedded representation of objects as a scoring function for the factors. We explain this process in more detail in Section 3.3.

The main strength of distributed representations is in their ability to share information between the represented objects. We exploit this property and show that by adding additional information to the embedding space, the overall performance of the model improves, *even if this information is not directly relevant to the classification task*. We demonstrate this fact by comparing stance prediction performance, when trained over the multiple topics separately or jointly (thus allowing the model to share information between the representations of multiple debate topics).

We evaluate our approach over the Internet Argument Corpus [7], [28], collected from two debate websites,CreateDebate and 4Forums. We conduct several experiments, both using in-domain data and out-of-domain data (when we train and test on different debate topics). Our experiments show that formulating the problem as structured representation learning indeed allows debate entities to share information and generalize better, resulting in even larger improvements when multiple stances (corresponding to different output labels) are trained jointly. Furthermore, we show that by using inference over the relationships between the learned representations we can outperform traditional collective classification methods.

Our contributions include (1) joint relational embedding for debate entities, allowing the model to share information between related topics and underlying ideologies (2) suggest a collective classification approach, defined over the embedding space, and using it to cast

representation learning as a structured prediction problem, and (3) an extensive experimental study in which we evaluate several different modeling choices and information sharing scenarios.

3.2 Related Work

Stance prediction in online debates is an important subjectivity classification task. Early work viewed the problem as a binary classification task and focused on feature representations [22], [23], while later work took a collective approach [3], [4], [25]. Stance prediction is not limited to online debates, as was also studied in the context of congressional speeches [29], [30] and social media outlets, such as Twitter [31]–[33], including a recent SemEval-16 task [26]. While most works view the task as supervised classification tasks, several works suggest exploiting the interactions between users as a form of distant supervision [27], [31]. This task is broadly related to argumentation mining [24] and stance reason classification [34].

Our technical work relies on exploiting distributed representations (i.e., embedding), building on highly influential work on embedding words [12], [13], sentences [35] and even full documents [36]. Our work explores the connections between text, users, and attributes, attempting to create a common representation for them. The closest to our work is [37], which jointly integrates different kinds of cues (text, attribute, graph) into a single latent representation to get user embeddings.

Our work is also broadly related to deep learning methods that capture the structural dependencies between decisions. This can be done either by modeling the dependencies between the hidden representations of connected decisions using RNN/LSTM [38], [39], or by explicitly modeling the structural dependencies between output predictions [40]–[42]. Unlike these work, we formulate our problem as a structured representation learning problem, which to our knowledge is the first work to identify the ties between the two problems.

3.3 Model Overview

In this chapter, we suggest casting stance classification as a structured representation learning task. Our approach revolves around two key ideas.

First, stance classification can be done by embedding both the input objects (i.e., posts) and the output labels in the same space. The representations learned for text, users and their attributes will reflect their semantic closeness. Therefore the actual classification can be performed by comparing the similarity between the embedded representations of an input object and the competing output labels.

Second, we can augment this representation with additional structural constraints, capturing relevant domain information, such as the connection between posts by the same author, and the disagreements between debate participants. These constraints can help to correct some errors in the individual predictions by the model using the knowledge we have on this task and dataset.

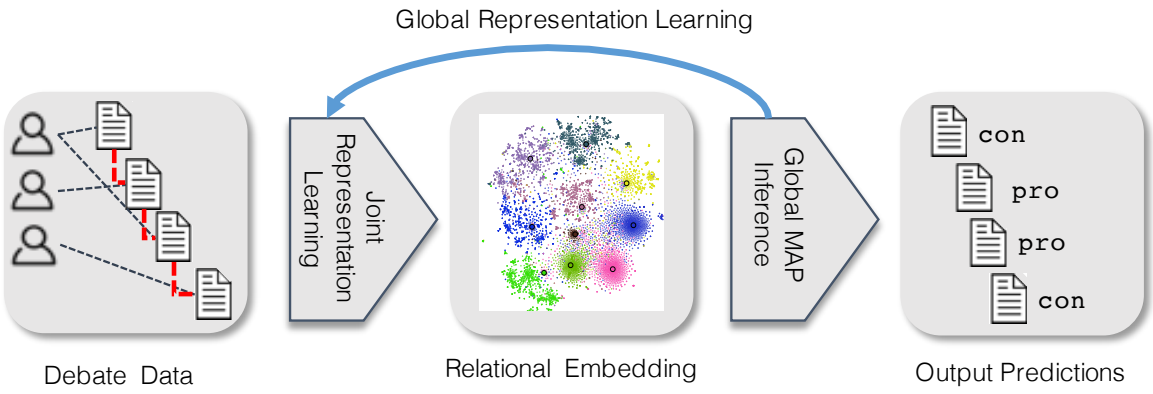


Figure 3.2. Overall Learning and Inference Processes.

To help clarify these ideas intuitively, consider the debate dialog in Figure 3.1. Our learning approach uses the structural and textual information in the dialog in three ways, as shown in the process depicted in Figure 3.2.

1. Joint Representation Learning The embedding learning objective is designed to represent relevant relational information, allowing the representation of different input objects to share information. For example, stances on different topics may share a similar ideology. Figure 3.3 demonstrates the joint embedding space. The relationship between

authors and their posts is preserved by the proximity of their embedded representations. Similar relationships between posts and their corresponding stances and underlying ideologies are also represented. To accomplish this goal we define a joint objective function over different relations.

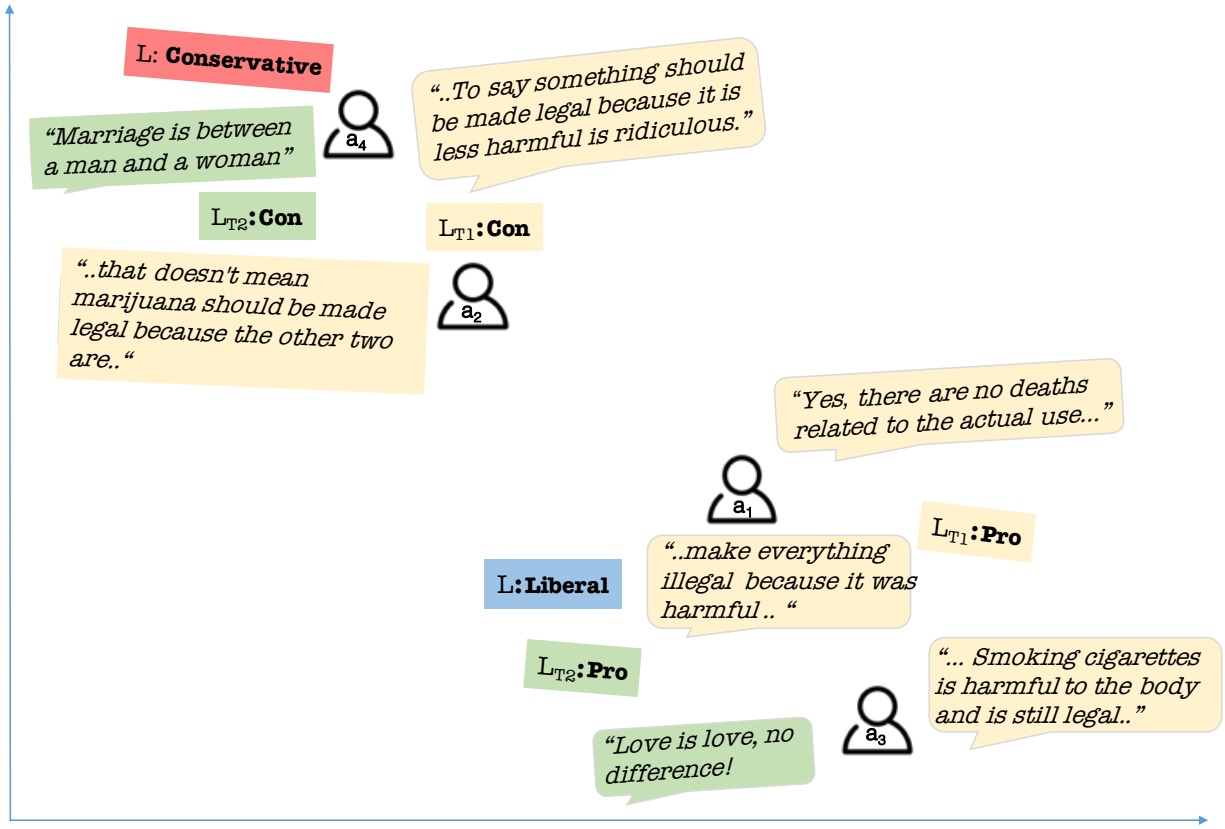


Figure 3.3. Relational Embedding Representing Authors, Stances and Posts on Various Topics in the Same Embedding Space.

The model is trained to maximize the similarity between corresponding entity pairs (positive examples) compared to irrelevant ones (negative examples). We define the positive examples based on relational information and increase the similarity between their vectorized representations during training. We explain this process in detail in Section 3.5.

2. Global MAP Inference (Collective Classification) Representing the input objects and their labels in the same embedding space allows us to reason about the relationships between them. We view the prediction task as a collective classification, in which all the posts in one or more given debate threads are decided together. We model inference required for

the MAP assignment using a factor graph. For example, the graph described in Figure 3.4, contains nodes corresponding to author level stance decisions (denoted L_{a_i}), and their posts levels stance decision (denoted $L_{t(a_i^j)}$). We score these decisions using the learned embeddings. For example, scoring the output assignment PRO to the post corresponding to $L_{t(a_i^j)}$ will be done by observing the similarity (dot product) between their vectorized representations v_{PRO} and $v_{t(a_i^j)}$.

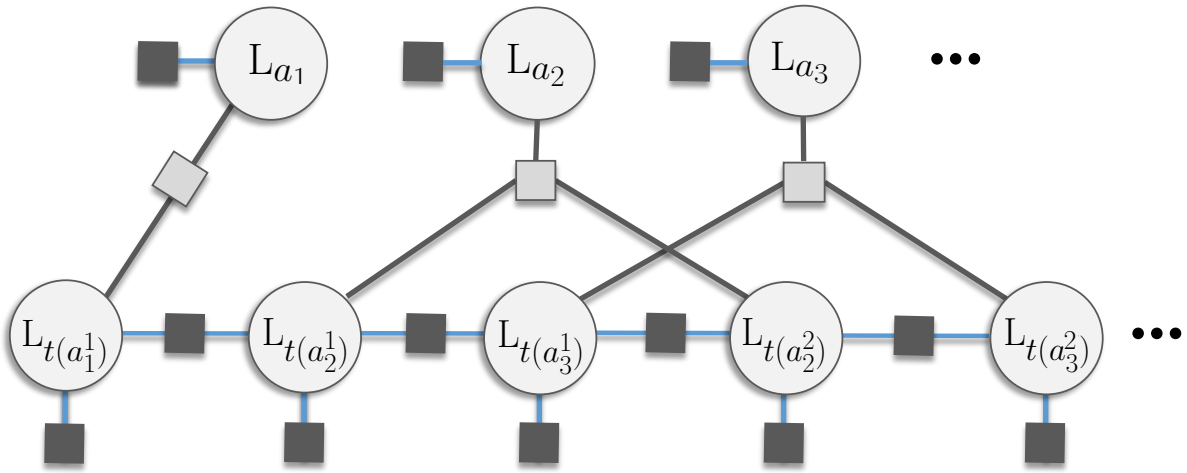


Figure 3.4. Collective Decision over Debate Thread Structure and Authors.

Factor nodes can either have a degree of 1 (e.g. scoring the similarity between an author or post and an output label) or 2 (e.g. scoring the relationship between consecutive posts in a debate discussion thread). We also allow hard constraints (light gray factors in Figure 3.4), which force the model to produce consistent assignments. (for example, an author and the text associated with it should have the same label). We explain this process in detail in Section 3.4.

3. Global Representation Learning A natural extension of the above model is to combine the previous two steps and adopt a global training approach that uses joint prediction during training. In this case, the loss function used when learning the embedding is defined with respect to the structural dependencies imposed by the factor graph. This approach is similar in spirit to deep structured learning approaches [42], however, in this case, the structured learning process is defined directly over the embedding space. This process is explained in Section 3.5.5.

3.4 Collective Classification

Our joint embedding model maps authors, attributes, and text into the same space. Thus it allows us to compute the similarity between any pair of authors, texts, attributes, or their combination. This is a very useful property, as information from all aspects can now be used for predicting the target of interest. For example, more information is available for identifying the stance of a post by using its author and neighboring posts comparing to the post's embedding alone. We exploit this property by defining the classification as a global inference process, enforcing the constraints and preferences on all of the predictions.

ILP Formulation

We exploit the dependencies described above, using joint prediction over the different aspects. We formulate the decision as an Integer Linear Programming (ILP) which allows us to enforce the consistency of preferences between decisions. The ILP objective function is defined over the similarity scores between objects' vector representation in the joint embedding space. Since integer linear constraints over 0-1 variables can represent logical constraints, we define the ILP constraints using both representations to help improve readability.

In the stance prediction task, all the posts from multiple debate threads that potentially share authors form a single ILP instance. The ILP global optimization objective is defined over authors a_{i}, the textual content (posts) $\{t_{\mathrm{i}}^0, ..., t_{\mathrm{i}}^k\}$ associated with a_{i}, and other textual posts $\{t_m^p, ..., t_l^q\}$, responding to or responded by a_{i}'s posts.

We create different types of boolean decision variables corresponding to the decision tasks above. We assign a boolean variable $\texttt{AuthorLabel}(a_{\mathrm{i}}, r_{\mathrm{j}})$ to represent author a_{i} has attribute r_{j} (i.e., its stance), and associate a score $sim(\mathrm{e}_{a_{\mathrm{i}}}, \mathrm{e}_{r_{\mathrm{j}}})$ with that variable. Similarly, we assign a boolean variable $\texttt{TextLabel}(t_{\mathrm{i}}^k, r_{\mathrm{j}})$ to represent that the text t_{i}^k is labeled with an attribute r_{j}, and associate a score $sim(\mathrm{e}_{t_{\mathrm{i}}^k}, \mathrm{e}_{r_{\mathrm{j}}})$ with that variable.

To ensure the consistency of the predicted variables, we define two types of constraints.

1. *Single output value on a debate topic*:

   $\forall \mathrm{i} \quad \sum_{\mathrm{j}} \texttt{AuthorLabel}(a_{\mathrm{i}}, r_{\mathrm{j}}) = 1$

   $\forall \mathrm{i}, k \ \sum_{\mathrm{j}} \texttt{TextLabel}(t_{\mathrm{i}}^k, r_{\mathrm{j}}) = 1$

2. *Output consistency*:

$$\forall \mathrm{i}, \mathrm{j}, k \quad \texttt{AuthorLabel}(a_{\mathrm{i}}, \ r_{\mathrm{j}}) = \texttt{TextLabel}(t_{\mathrm{i}}^{k}, \ r_{\mathrm{j}})$$

Note that in the debate domain, this constraint forces agreement between all the posts by the same author.

We also add variables capturing the dependencies between connected posts. For debate threads, a boolean variable $\texttt{Disagree}(t_{\mathrm{i}}^{p}, t_{l}^{q})$ is created for any two posts $t_{\mathrm{i}}^{p}, t_{l}^{q}$ when t_{l}^{q} is a response to t_{i}^{p}, and associate a score *disagree_parameter* with that variable. This score is a hyper-parameter for local models, capturing the preference towards disagreement between consecutive posts in a debate. It is set according to the training set. When using global learning, it is also included in the training, such that similarity scores of consecutive posts will be adjusted appropriately (similar intuition as a margin constraint).

$$\forall t_{\mathrm{i}}^{p}, t_{l}^{q} \ \ \texttt{Disagree}(t_{\mathrm{i}}^{p}, t_{l}^{q}) \wedge \texttt{TextLabel}(t_{\mathrm{i}}^{p}, r_{\mathrm{j}}) \rightarrow \neg \texttt{TextLabel}(t_{l}^{q}, r_{\mathrm{j}})$$

The set of all possible decisions for the three set of variables are denoted as A for `AuthorLabel`, B for `TextLabel`, Γ for `Disagree`.

Given these variables, our prediction function can be define as follows -

$$\arg\max_{\alpha,\beta,\gamma} \sum_{\alpha \in A} \alpha \cdot score(\alpha) + \sum_{\beta \in B} \beta \cdot score(\beta) + \sum_{\gamma \in \Gamma} \gamma \cdot score(\gamma)$$

$$\text{Subject To } \mathbf{C}$$

Where $\mathbf{C}$ is a set of constraints defined above.

Constraints Demonstration

We demonstrate the impact of the two types of constraints introduced above.

The first type of constraint, regarded as *author constraints*, is a hard one that enforces the labels assigned to an author and all posts by him are the same. We regard this Based on the analysis of the dataset, we found this assumption holds true most of the time. It is also intuitive since we know that people tend to keep the same attitude toward an issue and seldom change that unless some life-changing event occurs. Looking at the example in Fig. 3.5, $a_{\mathrm{i}}^{\mathrm{j}}$ stands for post j by author i. The orange line is the decision boundary of the local model. The prediction for post a_{1}^{2} is not correct in this case. However, our inference module

would likely correct this error by considering the decision of posts by the author 1 together. If the aggregated score for all three posts associated with PRO is higher than CON, the final prediction for all of them would be correct. It is also possible that the inference module would misclassify more posts when the aggregated scores cannot reflect reality. However, as long as the local model has a reasonably good performance itself, the inference module would almost always lead to improvement.

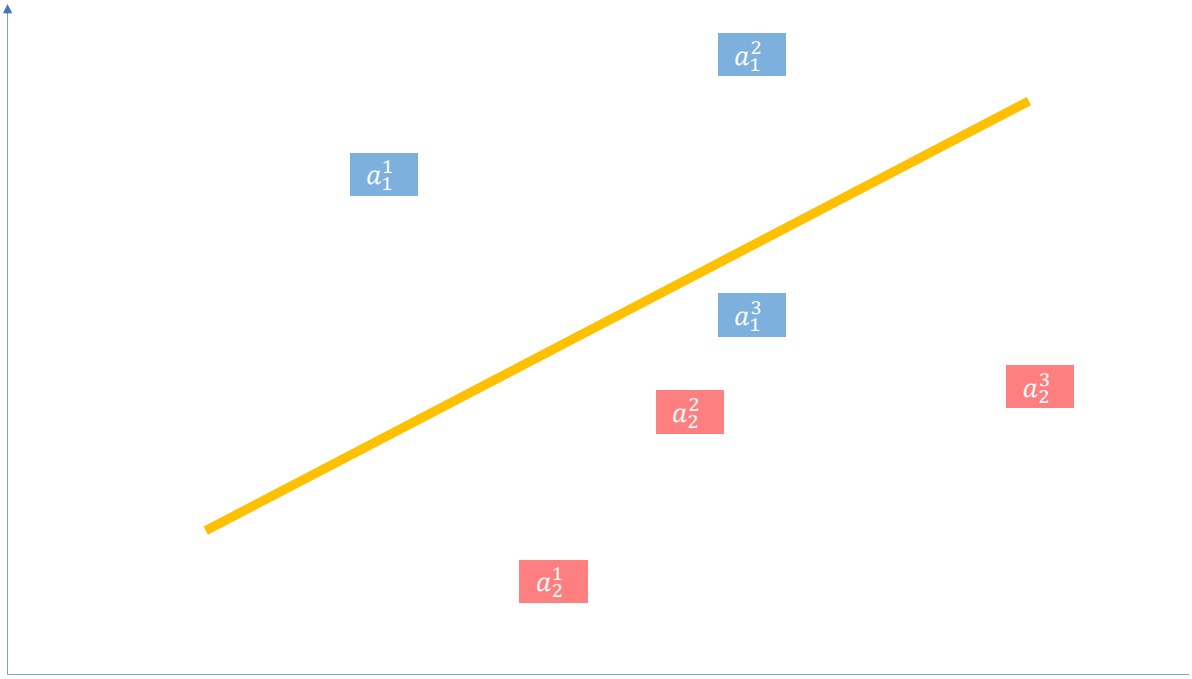


Figure 3.5. Example of Author Constraints.

Another type of constraint is called *consecutive constraints*. It follows the intuition that the stance for a post is usually different from the post it replies to. This assumption holds true to a lesser extent than the previous one, which is why we do not enforce it. Instead, we learn a score from the data to decide how much weight we want to assign to this rule. We illustrate the effect of consecutive constraints with the example in Fig. 3.6. Assume the

five posts in this figure form a debate thread as shown in Table 3.1. Again, the local model makes a mistake for post a_3^1. However, given the position of this post in the debate thread, our inference module is likely to adjust the prediction to *Pro* since the posts it replies to and replied to by both have the stance *Con*.

Table 3.1.
Debate Thread Example.

Post	Stance
a_1^1	Pro
a_2^1	Con
a_3^1	Pro
a_2^2	Con
a_1^2	Pro

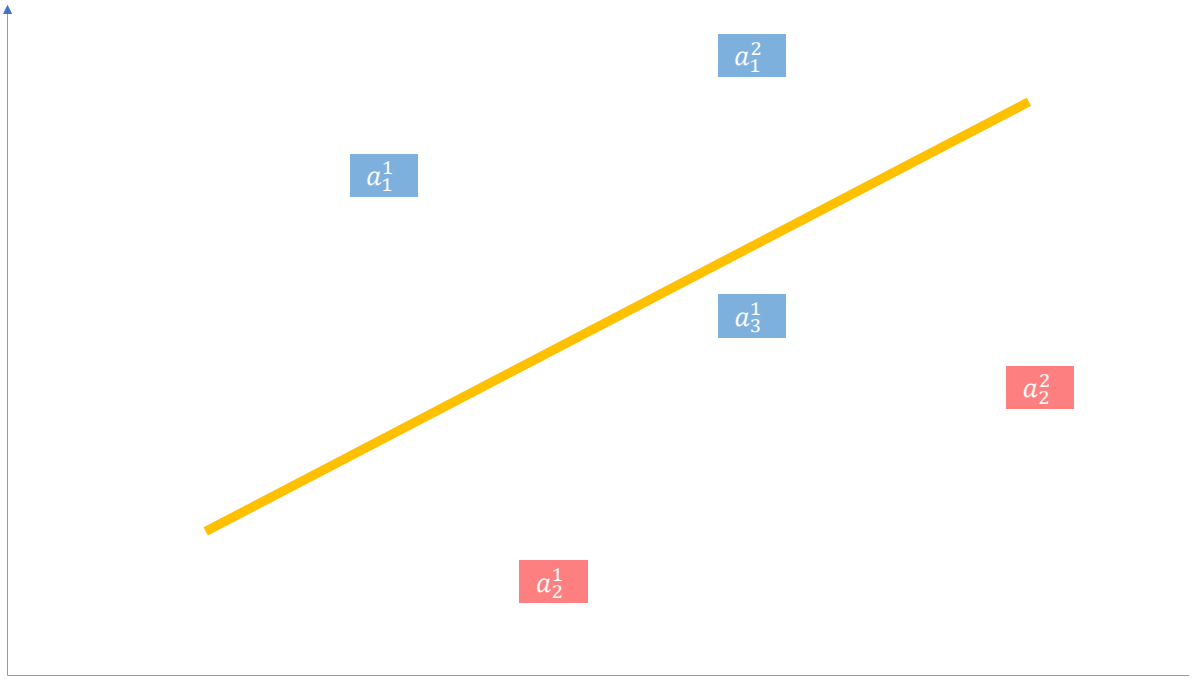


Figure 3.6. Example of Consecutive Constraints.

When these two types of constraints are used together, they complement each other and enable the inference module to fix mistakes made by the local model better, leading to much more consistent and meaningful predictions across debates.

3.5 Representation Learning

3.5.1 Embedding Perspectives

Let A and T denote the set of all authors and text respectively, let R denote the set of all attributes for those authors and text. Stances on various topics are the major attributes considered in this paper. For each topic, we have an embedding vector for the Pro stance and another vector for the Con stance, such as $Pro_{abortion}$ and $Con_{abortion}$. We train our embedding over multiple views of the data, each view connecting users and their content.

Author vs. Text: This objective is to predict text t_{j} linked with author a_{i} given the author representation. Each post is a text unit in our experiments.

$$L_{AT} = -\sum_{\mathrm{i}=1}^{n} \sum_{\mathrm{j}=1}^{text_{a_{\mathrm{i}}}} logP(t_{\mathrm{j}}|a_{\mathrm{i}}) \tag{3.1}$$

Author vs. Attribute: This objective is to predict attribute r_{j} linked with author a_{i} given the author representation. Stance on different topics and user profile information form the attributes set in debate datasets. Each user attribute value (e.g. male or female in gender attribute) is represented by a vector.

$$L_{AR} = -\sum_{\mathrm{i}=1}^{n} \sum_{\mathrm{j}=1}^{attri_{a_{\mathrm{i}}}} logP(r_{\mathrm{j}}|a_{\mathrm{i}}) \tag{3.2}$$

Text vs. Attribute: This objective is to predict attribute r_{j} of the text given text t_{i}. In our experiments, we only used the stance label as attributes of text. However, it may also be possible to inherit attributes from the author of the text.

$$L_{TR} = -\sum_{\mathrm{i}=1}^{m} \sum_{\mathrm{j}=1}^{attri_{t_{\mathrm{i}}}} logP(r_{\mathrm{j}}|t_{\mathrm{i}}) \tag{3.3}$$

Text vs. Text: This objective is to predict text t_{j} given the text t_{i} that share the same attribute. It is used to promote similarity between posts sharing the same stance on a certain topic.

$$L_{TT} = -\sum_{\mathrm{i}=1}^{m} \sum_{\mathrm{j}=1}^{text_{t_{\mathrm{i}}}} logP(t_{\mathrm{j}}|t_{\mathrm{i}}) \tag{3.4}$$

All the conditional probabilities can be computed using a softmax function. Taking $P(t_{\mathrm{j}}|a_{\mathrm{i}})$ as an example:

$$P(t_{\mathrm{j}}|a_{\mathrm{i}}) = \frac{\mathrm{e}xp(\mathrm{e}_{a_{\mathrm{i}}}^{T}\mathrm{e}_{t_{\mathrm{j}}})}{\sum_{k\in T}\mathrm{e}xp(\mathrm{e}_{a_{\mathrm{i}}}^{T}\mathrm{e}_{t_k})} \tag{3.5}$$

3.5.2 Embedding Initialization

In our model, the embedding for each author and attribute can be randomly initialized. The text is a special case since there are complex structures involved. One way to capture this is to use a pre-trained text embedding model to get an initial representation, and then learn a neural network to map it to one in the shared space. Note that this also allows our model to generate embedding for unseen text in the new space.

Specifically, for a text input x, we can compute its embedding e using M hidden layers $l_{\mathrm{i}}, \mathrm{i} = [0, M-1]$. The first hidden layer l_0 is computed from the input x:

$$l_0 = f(W_0 x + b_0) \tag{3.6}$$

Subsequent layers are computed recursively:

$$l_{\mathrm{i}} = f(W_{\mathrm{i}} l_{\mathrm{i}-1} + b_{\mathrm{i}}), \mathrm{i} = 1, ..., M-1 \tag{3.7}$$

Then the output from the final layer produces the embedding:

$$\mathrm{e} = l_{M-1} \tag{3.8}$$

f is the non-linear activation function. We used hyperbolic tangent (tanh) in our experiments.

Note that our model offers the flexibility to use more complex neural network structures, including CNN and RNN, to learn a mapping from the initial word embedding sequences of the text to an embedding in the joint space.

3.5.3 Joint Embedding Learning

Our objective is to learn a semantic embedding for authors, text, and attributes associated with them so that they are close in the embedding space if they are semantically close to each other.

Joint Embedding Loss Function: We can combine these embedding losses from Equations 3.1-3.4 into a joint training objective:

$$L_{Joint}(A, T, R) = \sum_{\mathrm{i} \in (AT, AR, TR, TT)} \lambda_{\mathrm{i}} L_{\mathrm{i}} \tag{3.9}$$

where λ_{i} is the coefficient for each view, indicating the relative importance in the loss function. We set all λ_{i} to the default value 1 in all our experiments.

This is the general framework. Additional views may be added or removed for a certain dataset. For example, we can add a term representing the author vs. author view in the loss function if links between them are available.

3.5.4 Model Optimization

We train our model using a mini-batch Adam optimizer to minimize the loss in Eq. 3.9. However, computing gradient for Eq. 3.1, and Eq. 3.4 is expensive due to the size of the authors or text. To address this problem, we refer to the popular negative sampling approach [43], which reduces the time complexity to be proportional to the number of positive example pairs.

3.5.5 Global Embedding Learning

Although different views of the data are captured in our joint loss function, it does not ensure that the information they provide at inference time will be "cooperative", i.e., it will result in consistent global prediction over all the debate outputs. One potential problem is that examples associated with one view will dominate the training and skew the prediction when constraints are applied. To handle this issue, we included the inference procedure during training. Instead of making sure the loss for each local view is minimized, the global objective promotes the rank of all the gold predictions *jointly*. For instance, at training, posts, together with their author and neighboring posts (if available) are used to infer their stance based on the inference procedure described in section 3.4. Then structured hinge loss can be used to define the prediction loss as in Eq. 3.10.

$$L_{pred} = \sum_{\mathrm{i} \in instances} max(0, max_{y \in Y}(\Delta(y, t_{\mathrm{i}}) + score(x_{\mathrm{i}}, y)) - score(x_{\mathrm{i}}, t_{\mathrm{i}})) \tag{3.10}$$

where x_{i} and t_{i} are the problem instances and corresponding gold predictions, Y denotes all possible predictions, and score$(\cdot)$ is the inference score function. $\Delta(\cdot, \cdot)$ is the hamming loss. It measures the difference between two predictions and is used to create a margin between gold and other predictions.

The loss function used for updating the parameters in the global model is defined as follows.

$$L_{Global} = \lambda_{pred} L_{pred} + \lambda_{AT} L_{AT} + \lambda_{TT} L_{TT} \tag{3.11}$$

The coefficients for different terms in the global loss function can adjust the contribution of the prediction objective (L_{pred}) versus the information sharing objective (L_{AT} and L_{TT}). Again, we set all λ to the default value 1 in the experiments of this paper. We leave the exploration of different coefficient settings to future work. Since the inference is used during global training, the scores for text stance and author stance are part of the inference scores. So they will get updated according to L_{pred}. Therefore we do not include L_{AR} and L_{TR} explicitly in the global loss function.

In our experiments, a mini-batch of debate threads is regarded as an instance during training. To reduce the computational cost, we used the parameters learned with the joint embedding loss function as the starting point for the global training.

3.6 Experiments

To evaluate the different properties of our model and demonstrate their advantages, we evaluate the quality of our structured embedding model on two datasets 4FORUMS and CREATEDEBATE for stance classification tasks at the post level, consisting of eight topics in total. The datasets are taken from the Internet Argumentation Corpus [7]. Table 3.2 shows statistics about these datasets.

Table 3.2.
Data Statistics for 4FORUMS and CREATEDEBATE.

Dataset	Topic	Posts	Users
CREATEDEBATE	Abortion	1741	340
	Gay Rights	1376	370
	Marijuana	626	258
	Obama	985	278
4FORUMS	Abortion	7937	342
	Evolution	6069	311
	Gay Marriage	6897	296
	Gun Control	3755	281

Experimental Design Our experiments are designed to evaluate the different properties of our model. To accomplish that, we compare different variations of the model, corresponding to (1) only the joint embedding (denoted *Joint*), (2) using inference at test time, over the joint embedding (denoted *Inference*), and (3) using global training (denoted *Global*), which also uses inference during training. We report the results of these experiments in the two datasets in Tables 3.3 and 3.4.

Our second set of experiments are designed to evaluate the joint embedding model's ability to share information between the representations of different objects. In this case, we compare the performance of the *Joint* and *Inference* models when additional information is available. We compare three settings (1) In-Domain, when the available training and testing

data are from the same domain (2) In+Out Domain, where we augment the In-Domain training data with additional debate threads from other topics. In this case, the model can represent the relationships between stances on different topics and potentially generalize better. Finally, (3) User-Attribute, where we augment the author attributes with profile information extracted from the debate website. We conduct all of these experiments over the CreateDebate dataset, and report the results in Table **??**.

3.6.1 Experimental Settings:

We used PyTorch [44] to implement the embedding model and Gurobi [45] as our ILP solver[1]. Each debate post is initially represented using the Skip-Thought Vectors [35], and then mapped to an embedding in the shared space through one hidden layer. We do not add more layers as both datasets are relatively small. Hyperbolic tangent (tanh) is used as a non-linear activation function. All other embeddings are randomly initialized following a normal distribution with variance $1/\sqrt{dim}$. The embedding size dim for all experiments is 300. For the training of the neural network, we used a mini-batch Adam optimizer to update parameters. Dropout with a probability of 0.7 is used as regularization. The termination criteria are convergence on training loss. Five epochs of non-improvement on loss are considered as a convergence for joint models, and one epoch for global models. Other parameters in our model include negative sample size k=5, mini-batch size b=10.

3.6.2 Results

Our results on stance classification are described in Table **??** and Table **??**. The results are computed using 5-fold cross-validation. For the CreateDebate dataset, We used the same five data folds as in [3] to ensure our results are directly comparable with theirs. For the 4Forum dataset, we randomly divided debate threads into five folds since the data split is not available in [4]. We regarded the same user in the training and test folds as different ones to avoid leaking label information from training to test. *NB* and *CRF* stands for the best local and collective models in [3]. Note that their system also uses the author constraints,

[1]↑Please refer to https://github.com/BillMcGrady/StancePrediction for data and source code.

as well as a highly engineered feature set and additional weakly-supervised data that we did not use. Despite that fact, our global model significantly outperforms their model with the exception of the Obama domain. In the 4FORUM experiments we compare our models to result with a *PSL* based model [4] which performs similar collective classification defined over a feature-rich representation. In this case, our Global model achieves the best overall performance as well.

Table 3.3.
Average Accuracy on CREATEDEBATE dataset.

Model	Abortion	Gay Rights	Marijuana	Obama	Average
Majority	56.2	64.5	72.0	56.1	62.2
NB [3]	73.3	67.0	72.4	67.0	70.0
CRF [3]	74.7	69.9	75.4	**71.1**	72.8
PSL [4]	66.8	72.7	69.1	63.7	68.1
Joint	62.1	63.1	69.2	57.4	63.0
Inference(AC)	70.4	62.7	66.3	62.2	65.4
Inference(Consecutive)	67.2	65.0	66.8	61.0	65.0
Inference(Both)	**81.1**	75.6	75.0	64.7	74.1
Global	81.0	**77.2**	**77.6**	64.8	**75.2**

Table 3.4.
Average Accuracy on 4FORUMS dataset.

Model	Abortion	Evolution	Gay Marriage	Gun Control	Average
Majority	56.8	65.8	66.0	67.8	64.1
PSL [4]	77.0	80.3	80.5	69.1	76.7
Joint	64.1	67.2	68.5	66.5	66.6
Inference(AC)	72.9	66.8	68.6	68.4	69.2
Inference(Consecutive)	67.5	67.7	72.3	69.6	69.3
Inference(Both)	85.9	80.9	**88.1**	81.6	84.1
Global	**86.5**	**82.2**	87.6	**83.1**	**84.9**

We evaluate the contribution of two constraint sets, Author constraints (AC) enforce author and their posts share the same stance. *Consecutive* will encourage disagreement in stances between neighboring posts as introduced in section 3.3. The addition of these two constraints leads to a significant increase in performance when used together. This is

because AC and Consecutive add agreement and disagreement constraints between test posts, grouping them into clusters and making it easier to be predicted correctly. For instance, given multiple posts from the same author, the model can make correct decisions on all of them even if the prediction based on some individual posts may be wrong. Finally, we observe that structured representation learning (i.e., Global) leads to a performance improvement compared to inference over the joint embedding objective (Inference). This shows the effectiveness of global learning.

Table 3.5 shows the result on CreateDebate when additional information is available. We extracted user profile information (User-Attribute) from the website[2], consisting of five attributes (Gender, Marital Status, Political Party, Religion, and Education). Clearly richer user information results in a better representation, both for users and as a result, also for the text they author, leading to improved performance. A similar trend occurs when out-of-domain data is available (In+Out Domain). Stances over different debate topics impact the text and author representations. Interestingly, when this data is available, our model is able to outperform the collective approach of [3] in all debate topics, showing that our model can indeed exploit the information shared by the underlying ideologies.

3.7 Chapter Summary

In this chapter, we study the problem of stance prediction, a challenging text classification problem, which requires taking into account textual content, conversational interactions, and author information. People have tried to model it using a graphical model over a fixed feature representation. We follow the observation that all of these problems are connected and allow the model to capture these dependencies by learning representations for all these aspects jointly. We show that by formulating the decision problem over the representation directly and requiring the representation to respect the global dependencies between these aspects, our model can generalize better and exploit additional information even when it is not directly relevant.

[2]↑www.createdebate.com

Table 3.5.
Average Accuracy on CreateDebate dataset with additional information.

	Model	Abortion	Gay Rights	Marijuana	Obama	Average
	CRF	74.7	69.9	75.4	71.1	72.8
In-Domain	Joint	62.1	63.1	69.2	57.4	63.0
	Inference	**81.1**	75.6	75.0	64.7	74.1
In-Domain + User-Attribute	Joint	63.9	63.6	69.5	59.9	64.2
	Inference	81.0	80.0	75.6	65.7	75.6
In+Out Domain	Joint	63.0	62.9	70.3	61.8	64.5
	Inference	79.0	74.5	**77.1**	76.2	76.7
In+Out Domain + User-Attribute	Joint	63.3	65.3	71.0	57.7	64.3
	Inference	79.9	**80.2**	75.1	**77.4**	**78.2**

In the future, we can explore additional domains and evaluate whether including additional aspects, can help provide better generalization. Providing sufficient supervision is one of the main bottlenecks of NLP, we intend to apply our approach in weakly and distantly supervised settings, to help alleviate this difficulty.

4. CASE II: IMPROVING BIAS DETECTION IN NEWS ARTICLES BY ENCODING SHARING STRUCTURE ON SOCIAL MEDIA

In this chapter, we consider the bias prediction problem for news articles. We first emphasize the importance of identifying the underlying political perspective of news articles and the challenge we are facing. Then we construct the social information graph by extracting political and sharing Twitter users for the news articles in our dataset. We propose a joint text and graph model that learn corresponding representations for news article based on both textual and graph information, in order to propagate bias signal to flow between the two sides. We report experimental results for both full supervision and distant supervision to demonstrate the performance gain resulted from the sharing pattern information.

4.1 Introduction

Over the last decade, we witness a dramatic change in the way information is generated and disseminated. Instead of a few dedicated sources that employ reporters and fact-checkers to ensure the validity of the information they provide, social platforms now provide the means for any user to distribute their content, resulting in a sharp increase in the number of information outlets and articles covering news events. As a direct result of this process, the information provided is often shaped by their underlying perspectives, interests, and ideologies. For example, consider the following two snippets discussing the comments made by a Democratic Senator regarding the recent U.S. government shutdown.

thehill.com (*Center*)

Sen. Mark Warner (D-Va.) on Sunday blasted President Trump for his "inept negotiation" to bring an end to the ongoing partial government shutdown. Warner, the ranking member of the Senate Intelligence Committee, lamented the effect the shutdown has had on hundreds of thousands of federal workers who have been furloughed or forced to work without pay.

infowars.com (*Right*)

Senator Mark Warner (D-Va.) is being called out on social media for his statement on the partial government shutdown. Warner blamed the "suffering" of federal workers and contractors on President Trump in a Sunday tweet framing Trump as an "inept negotiator". Twitter users pointed out that Democrats are attending a Puerto Rican retreat with over 100 lobbyists and corporate executives.

Despite the fact that both articles discuss the same event, they take very different perspectives. The first reporting directly about the comments made, while the second one focuses on negative reactions to these comments. Identifying the perspective difference and making it explicit can help strengthen trust in the newly-formed information landscape and ensure that all perspectives are represented. It can also help lay the foundation for the automatic detection of false content and rumors and help identify information echo chambers in which only a single perspective is highlighted.

Traditionally, identifying the author's perspective is studied as a text-categorization problem [31], [46]–[49], focusing on linguistic indicators of bias or issue-framing phrases indicating their authors' bias. These indicators can effectively capture bias in ideologically charged texts, such as policy documents or political debates, which do not try to hide their political leaning and use a topic-focused vocabulary. Identifying the authors' bias in news narratives can be more challenging. News articles, by their nature, cover a very large number of topics resulting in a diverse and dynamic vocabulary that is continuously updated as new events unfold. Furthermore, unlike purely political texts, news narratives attempt to maintain credibility and seem impartial. As a result, bias is introduced in subtle ways, usually by emphasizing different aspects of the story.

Our main insight in this chapter is that the social context through which the information is propagated can be leveraged to alleviate the problem, by providing both a better representation for it, and when direct supervision is not available, a distant-supervision source based on information about users who endorse the textual content and spread it. Several recent works dealing with information dissemination analysis on social networks focused on analyzing the interactions between news sources and users in social networks [50]–[52]. How-

ever, given the dynamic, and often adversarial setting of this domain, the true source of the news article might be hidden, unknown, or masked by taking a different identity. Instead of analyzing the documents' sources, our focus is to use social information, capturing how information is shared in the network, to help guide the text representation and provide additional support when making decisions over textual content.

We construct a *socially-infused textual representation*, by embedding in a single space the news articles and the social circles in which these articles are shared so that the political biases associated with them can be predicted. Figure 4.1 describes these settings. The graph connects article nodes via activity-links to users nodes (*share*), and these users, in turn, are connected via social links (*follow*) to politically affiliated users (e.g., the Republican or Democratic parties twitter accounts). We define an embedding objective capturing this information, by aligning the document representations, based on content, with the representation of users who share these documents, based on their social relations. We use a recently proposed graph embedding framework, Graph Convolutional Networks (GCN) [8], [9] to capture these relationships. GCNs are neural nets operating on graphs, creating node embeddings based on the graph neighborhood of a given node. In the context of our problem, the embedding of a document takes into account the textual content, but also the social context of users who share it, and their relationships with other users with known political affiliations. We compare this powerful approach with traditional graph embedding methods that only capture local relationships between nodes.

Given the difficulty of providing direct supervision in this highly dynamic domain, we study this problem both when direct supervision over the documents is available, and when using *distant-supervision*, in which the document level classification depends on propagating political tendencies through the social network, which is often incomplete and provides conflicting information.

To study these settings we focus on U.S. news coverage. Our corpus consists of over 10,000 articles, covering more than 2,000 different news events, about 94 different topics, taking place over a period of 8 years. We remove any information about the source of the article (both meta-data and in the text) and rely only on the text and the reactions to it on social media. To capture this information, we collected a set of 1,600 users who share the

news articles on Twitter and a handful of politically affiliated users followed by the sharing users, which provide the distant supervision. We cast the problem as a 3-class prediction problem, capturing left-leaning bias, right-leaning bias, or no bias (center).

Our experimental results demonstrate the strength of our approach. We compare direct text classification or node classification methods to our embedding-based approach in both the fully supervised and distant supervised settings, showing the importance of socially infused representations.

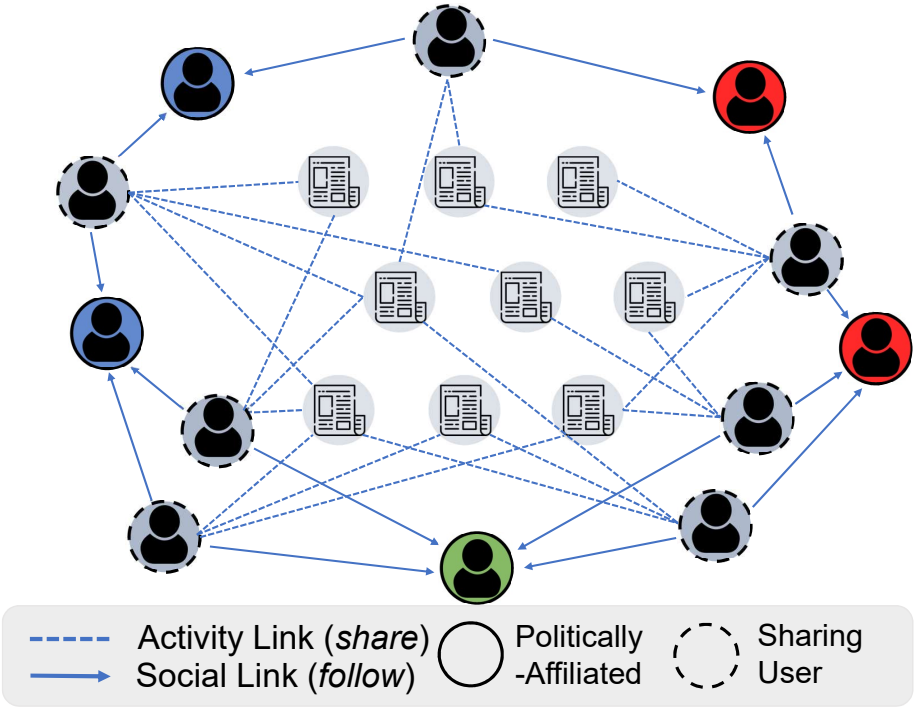


Figure 4.1. Information Flow Graph.

4.2 Related Work

The problem of perspective identification is typically studied as a supervised learning task [46], [53], in which a classifier is trained to differentiate between two specific perspectives.

For example, the *bitter-lemons* dataset consisting of 594 documents describing the Israeli and Palestinian perspectives. More recently, in SemEval-2019, a hyperpartisan news article detection task was suggested[1]. The current reported results on their dataset are comparable to ours when using text information alone, demonstrating that it is indeed a challenging task. Other works use linguistic indicators of bias and expressions of implicit sentiment [46], [48], [54], [55]. In recent years several works looked at indications of framing bias in news articles [56]–[60]. We build on these works to help shape our text representation approach.

Recent works looked at false content identification [50], [61], including a recent challenge[2] identifying the relationship between an article's title and its body. Unlike these, we do not assume the content is false, instead, we ask if it reflects a different perspective.

Using social information when learning text representations was studied in the context of graph embedding [62], extending traditional approaches that rely on graph relations alone [17], [18], [63] and information extraction and sentiment tasks [64], [65]. In this work, we focus on GCNs [9], [66], a recent framework for representing relational data, that adapts the idea of convolutional networks to graphs. Distant supervision for NLP tasks typically relies on using knowledge-bases [67], unlike our setting that uses social information. Using user activity and *known* user biases was explored in [68], our settings are far more challenging as we do not have access to this information.

4.3 Dataset Description

We collected 10,385 news articles from two news aggregation websites[3] on 2,020 different events discussing 94 event types, such as elections, terrorism, etc. The websites provide news coverage from multiple perspectives, indicating the bias of each article using crowdsourced and editorial reviewed approaches[4]. We preprocessed all the documents to remove any information about the source of the article.

We collected social information consisting of Twitter users who share links to the collected articles. We focused on Twitter users who follow political users and share news articles

[1]↑https://webis.de/events/semeval-19/
[2]↑http://www.fakenewschallenge.org
[3]↑Memeorandum.com and Allsides.com
[4]↑https://www.allsides.com/media-bias/media-bias-rating-methods

frequently (100 articles minimum). We found 1,604 such Twitter users. The list of political users was created by collecting information about active politically affiliated users. It consists of 135 Twitter users who are mainly politicians, political journalists, and political organizations. The set of political users and Twitter users are disjoint. The summary of the dataset is shown in Table 4.1.

Table 4.1.
Dataset Statistics for Allsides.

Articles	10,385	Twitter Users	1,604
-Left	3,931	Pol. Users	135
-Right	2,290	Left Pol. Users	49
-Center	4,164	Right Pol. Users	51
Sources	86	Center Pol. Users	35
Types	94	Avg # shared per Article	23.29
Events	2,020	Avg # pol. users followed	20.36

Data Folds We created several data splits to evaluate our model in the supervised settings, based on three criteria: *randomly* separated, *event* separated and *time* separated splits. In the event-separated case, we divide the news articles such that all articles covering the same news event will appear in a single fold. For the time-separated case, we sort the publication dates (from oldest to latest) and divide them into three folds. Each time one fold is used as training data (33%) and the other two combined as test data (66%). We use the same folds throughout the experiment of supervised classification for evaluation purposes.

Constructing the Social Information Graph We represent the relevant relationships as an information graph, similar to the one depicted in Figure 4.1. The social information graph $G = \{V, E\}$, consisting of several different types of vertices and edges, is defined as follows:

- Let $P \subset V$ denote the set of the *political users*. These are Twitter users with a clear, self-reported, political bias. They may be the accounts of politicians (e.g., Sarah Palin, Nancy Pelosi), political writers in leading newspapers (e.g., Anderson Cooper), or political organizations (e.g., GOP, House Democrats). Note that even political users that share a general political ideology can differ significantly in the type of issues and agenda they would pursue, which would be reflected in their followers.

- Let $U \subset V$ denote the set of *Twitter users* that actively spread content by sharing news articles. The political bias of these users is not directly known, only indicated indirectly through the *political users* they follow and news articles they share on Twitter.
- Let $A \subset V$ denote the set of news articles shared by the *Twitter users* (U).

The graph vertices are connected via a set of edges described hierarchically, as follows:

- $E_{UP} \subset E$: All the Twitter users are connected to the political users whom they follow. Note that a Twitter user may be connected to many different political users.
- $E_{AU} \subset E$: All the articles are connected to the Twitter users who share them. Note that an article may be shared by many different Twitter users.

4.4 Text and Graph Model

Our goal is to classify news articles into three classes corresponding to their bias. Since we have both the textual and social information for the news articles, we can obtain representations for them using either the text or graph models. In this section, we briefly go through the text representation methods and then move to describe the graph-based models we considered in this paper.

4.4.1 Text Representations and Linguistic Bias Indicators

To predict the bias of the news articles, we can consider it as a document classification task. We use the textual content of a news article to generate a feature representation. Deciding on the appropriate representation for this content is one of the key design choices. Previous works either use traditional, manually engineered representations for capturing bias [48] or use latent representations learned using deep learning methods [49]. We experimented with several different choices of the two alternatives and compared them by training a classifier for bias prediction over the document directly. The results of these experiments are summarized in Table 4.2. We provide a brief overview of these alternatives and point to the full description in the relevant papers.

Linear BoW Unigram features were used. The articles consist of 77,772 unique tokens. We used TFIDF vectors as unigram features obtained by using scikit-learn [69].

Bias Features These are content-based features drawn from a wide range of approaches described in the literature on political bias, persuasion, and misinformation, capturing structure, sentiment, topic, complexity, bias, and morality in the text. We used the resources in [70] to generate 141 features based on the news article text, which were shown to work well for the binary hyper-partisan task [71].

Averaged Word Embedding (WE) The simplest approach for using pre-trained word embeddings. An averaged vector of all the document's words using the pre-trained GloVe word embeddings **Pennington2014** is used to represent the entire article.

Skip-Thought Embedding Unlike the Averaged word vector that does not capture context, we also used a sentence level encoder, Skip-Thought [72], to generate text representations. We regard each document as a long sentence and map it directly to a 4800-dimension vector.

Hierarchical LSTM over tokens and sentences We used a simplified version of the Hierarchical LSTM model [21]. In this case, documents are first tokenized into sentences, then each sentence was tokenized into words. We used a word-level LSTM to construct a vector representation for each sentence, by taking the average of all the hidden states. Then, we ran another single layer unidirectional LSTM over the sentence representations to get the document representation by taking an average of all the hidden states.

4.4.2 Graph-Based Representations

In addition to the textual information, the news articles are also part of the information network defined in Section 4.3. Intuitively, news articles shared by the same Twitter users are likely to have the same bias, and users who share a lot of news in common are close in their political preferences. A similar intuition connects users who follow similar politically affiliated users. Capturing this information allows us to predict the bias of a news article, given its social context. We design our embedding function to map all graph nodes into a low dimensional vector space, such that the graph relationships are preserved in the embedding

space. In the shared embedding space, nodes that are connected (or close) in the graph should have higher similarity scores between their vector representations.

Directly Observed Relationships in Graph (DOR)

Our first embedding approach aims to preserve the local pairwise proximity between two vertices directly. This is similar to first-order graph embedding methods [63]. There are two different relations observed in the graph: Twitter user to political user (follow) and news article to Twitter user (share). We construct our embedding over multiple views of the data, each view w corresponds to a specific type of graph relation. We can then define an loss function L_w for each view w as follows:

- Twitter User to Political User (UP): This objective maximizes the similarity of a Twitter user, u and all the political users in the set $P_u \subset P$, where P_u is the set of political users that u follows.

$$L_{UP} = -\sum_{u \in U} \sum_{p \in P_u} logP(p|u) \tag{4.1}$$

- News Article to Twitter User (AU): This objective maximizes the similarity of a news articles, a and all the Twitter users in the set $U_a \subset U$, where U_a is the set of Twitter users who shared news article a on Twitter.

$$L_{AU} = -\sum_{a \in A} \sum_{u \in U_a} logP(u|a) \tag{4.2}$$

All the conditional probabilities can be computed using a softmax function. Taking $P(p|u)$ as an example:

$$P(p|u) = \frac{exp(\mathrm{e}_u^T \mathrm{e}_p)}{\sum_{q \in P} exp(\mathrm{e}_u^T \mathrm{e}_q)} \tag{4.3}$$

where e_u and e_p are embeddings of twitter user u and political user p respectively.

Computing Eq. 4.1 and Eq. 4.2 can be expensive due to the size of the network. To address this problem, we refer to the popular negative sampling approach [73], which reduces

the time complexity to be proportional to the number of positive example pairs (i.e. number of edges in our case).

The losses defined for the two views are summed with the classification loss defined in Eq. 4.9 as the final loss function to be optimized in the DOR embedding model.

$$L_{DOR} = L_{clf} + L_{UP} + L_{AU} \tag{4.4}$$

Graph Convolutional Networks (GCN)

Graph Convolutional Networks is an efficient variant of convolutional neural networks which operate directly on graphs. It can be regarded as special cases of a simple differentiable message-passing framework [74]:

$$h_{\mathrm{i}}^{(l+1)} = \sigma\left(\sum_{\mathrm{j}\in N(\mathrm{i})} M^{(l)}(h_{\mathrm{i}}^{(l)}, h_{\mathrm{j}}^{(l)})\right) \tag{4.5}$$

where $h_{\mathrm{i}}^{(l)} \in \mathbb{R}^{d^{(l)}}$ is the hidden state of node v_{i} in the l-th layer of the neural network, with $d^{(l)}$ as the dimensionality of representation at layer l. $N(\mathrm{i})$ is the set of direct neighbors of node v_{i} (usually also include itself). Incoming messages from the local neighborhood are aggregated together and passed through the activation function $\sigma(\cdot)$, such as $\tanh(\cdot)$. $M^{(l)}$ is typically chosen to be a (layer-specific) neural network function. Kipf and Welling [9] used a simple linear transformation $M^{(l)}(h_{\mathrm{i}}^{t}, h_{\mathrm{j}}^{t}) = W^{(l)}h_{\mathrm{j}}$ where $W^{(l)}$ is a layer-specific weight matrix.

This linear transformation has been shown to propagate information effectively on graphs. It leads to significant improvements in node classification [9], link prediction [8], and graph classification [75].

One GCN layer can be expressed as follows:

$$H^{(l+1)} = \sigma(\hat{A}H^{(l)}W^{(l)}) \tag{4.6}$$

Here, $\hat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$ is the normalized adjacency matrix. $\tilde{A} = A + I_N$ is the adjacency matrix of the undirected graph with added self-connections. I_N is the identity matrix. $\tilde{D}$

is a diagonal matrix with $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$. $W^{(l)}$ is the layer-specific trainable weight matrix. $H^{(l)} \in \mathbb{R}^{N \times D^{(l)}}$ is the matrix of hidden states in the l-th layer. $H^{(0)} = X$ is the input vectors. It can either be one-hot representations of nodes or features of the nodes if available. $\sigma(\cdot)$ is the activation function.

Multiple GCN layers can be stacked in order to capture high-order relations in the graph. We consider a two-layer GCN in this paper for semi-supervised node classification. Our forward model takes the form:

$$V = \tanh\left(\hat{A}\tanh\left(\hat{A}XW^{(0)}\right)W^{(1)}\right) \tag{4.7}$$

where X is the input matrix with one-hot representations and V is the representation matrix for all nodes in the graph.

Figure 4.2 shows an example of how our GCN model aggregates information from a node's local neighborhood. The orange document is the node of interest. Blue edges link to first-order neighbors and green edges link to second-order neighbors.

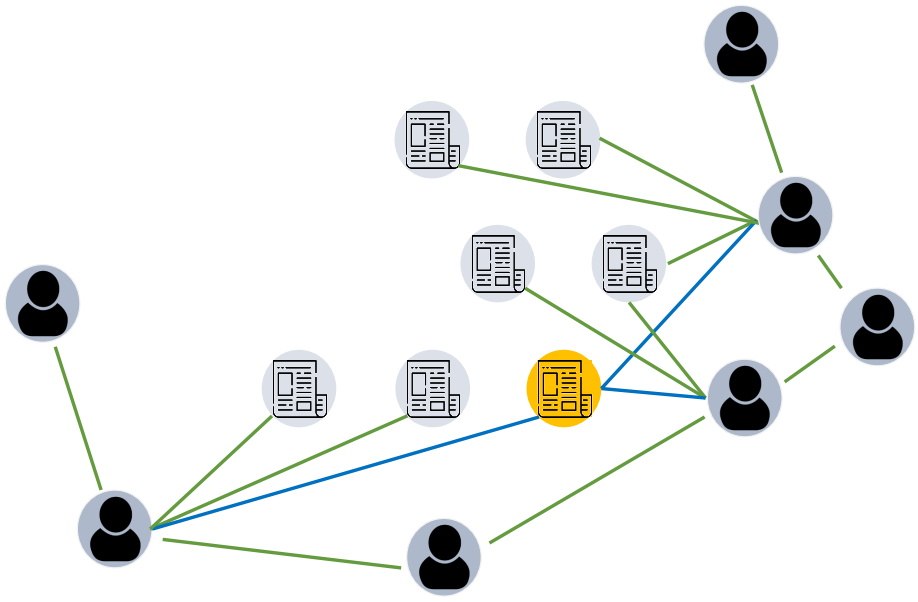


Figure 4.2. Example of Unfolding of GCN Computational Graph.

4.4.3 Document Classification

The representation v of a news article (obtained with text models or graph models) captures the high level information of the document. It can be used as features for predicting the bias label with a feed-forward network.

$$p = \text{softmax}(W_c v + b_c) \tag{4.8}$$

We use the negative log likelihood of the correct labels as classification training loss:

$$L_{clf} = -\sum_{a} \log p_{a_j} \tag{4.9}$$

where j is the bias label of news article a.

4.5 Joint Model

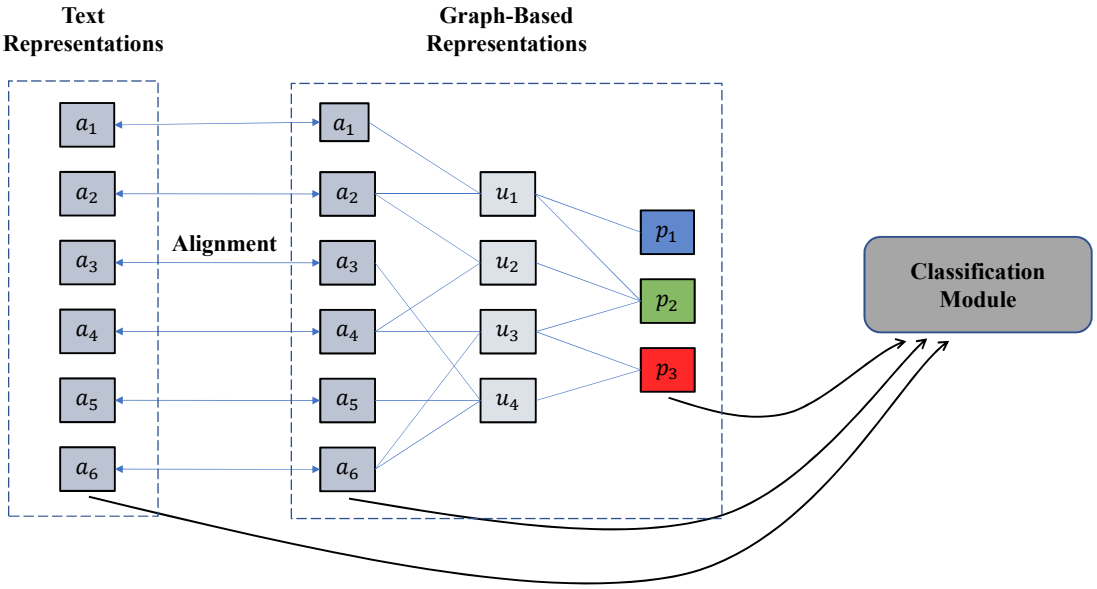


Figure 4.3. Overall Architecture: Representations are learned for news articles based on textual information and graph structure; these two representations are aligned in our joint model; only labels of political users are available during training in distant supervision case.

Given that we have two representations available for news articles, namely the textual one and social one, it is natural to make the prediction combining both of them. We propose to align the representations of the same document from graph and text models in a joint training fashion as shown in Figure 4.3. The objective function for the alignment is:

$$L_{align} = -\sum_{a \in A} \log P(\mathrm{e}_a^G | \mathrm{e}_a^T) \tag{4.10}$$

where e_a^T is the embedding for document a based on its content, and e_a^G is the embedding for document a based on graph structures. $P(\mathrm{e}_a^G | \mathrm{e}_a^T)$ is defined the same way as in Eq. 4.3.

$$P(\mathrm{e}_a^G | \mathrm{e}_a^T) = \frac{exp(\mathrm{e}_a^G \mathrm{e}_a^T)}{\sum_{d \in A} exp(\mathrm{e}_a^T \mathrm{e}_d^G)} \tag{4.11}$$

Negative sampling is again utilized to reduce time complexity.

Connecting the text and graph embedding of the same news articles, allows the bias signal to flow between the two sides. Therefore the text model may learn from the social signal and the graph model may use textual content to adjust its representation as well. This is especially beneficial for examples where information from one source (text or graph) is ambiguous or even misleading, which is common in real-world datasets where the examples are influenced by countless different factors such that the model is not possible to capture all of them. For example, an article that is leaning left may actually be shared by more center users than left users due to the event type or source of origin. We describe the loss function for the joint model in two settings - full supervision (i.e., labels associated with *documents* directly) and distant supervision, when bias information is only provided for a handful of *political users*, which do not actively share documents.

Full Supervision In the full supervision case, the loss consists of three parts, namely the classification loss of text model (L_{clf}^T), the classification loss of graph model (L_{clf}^G), and the loss for aligning the embeddings of the text and the graph models (L_{align}).

$$L_{\text{joint}} = \alpha L_{clf}^T + \beta L_{clf}^G + \gamma L_{align} \tag{4.12}$$

Here α, β, and γ are hyper-parameters to adjust the contribution of the three parts. We set all of them to default value 1 in experiments in this paper.

Distant Supervision Unlike the full supervision case where we have training labels for documents, we only have access to the labels of political users. However, since the text and social representation use the same space, user bias information can be propagated to the document representation, acting as a distant supervision source. Additionally, we can make use of the predicted labels for documents obtained after the distant supervised training as a guide for another round of training which compute loss for all documents as well. This process can potentially be repeated in an EM style with the predicted labels updated after each round. In practice, we found the model achieved the best performance on our dataset after one round of such training with predicted labels serve as supervision for documents.

Inference Given the graph representation, decisions can be made in multiple ways. Each document has a dual representation, as a text node and a social node. Also, given the social context of a document, the decision can be defined over the users that share it (assuming that users tend to share the information which agrees with their biases). To take advantage of that fact, we define a simplified inference process. At test time, we can predict the bias of a news article with the embeddings from the text model (Text), the embeddings from the graph model (Graph), and the embeddings of sharing users who shared this article (User). The last method (User) works by averaging bias prediction scores s_u^b for all Twitter users that shared an article a. The bias prediction score is computed in Eq. 4.8 before the softmax$(\cdot)$ applied.

$$\arg\max_b \frac{\sum_{u \in U_a} s_u^b}{|U_a|} \tag{4.13}$$

Finally, two or three of the scores listed above can be combined to make the decision.

4.6 Experiments

We designed our experiments to evaluate the contribution of social information in both the fully supervised setting, and when only distant supervision is available through the social graph. We begin by evaluating several text classification models that help contextualize the

social information. Finally, we evaluate our model's ability to make predictions when very little social information is available at test time.

4.6.1 Implementation Details

We used the spaCy toolkit for preprocessing the documents. All models are implemented with PyTorch [76][5]. Hyperbolic tangent (tanh) is used as a non-linear activation function. We use a feed-forward neural network with one hidden layer for the bias prediction task given textual or social representation. The sizes of LSTM hidden states for both word level and sentence level are 64. The sizes of hidden states for both GCN layers are 16. For the training of the neural network, we used the Adam optimizer [77] to update the parameters. We use 5% of the training data as the validation set. We run the training for 200 epochs (50 epochs for HLSTM models) and select the best model based on performance on the validation set. Other parameters in our model includes negative sample size k=5, mini-batch size b=30 (mini-batch update only used for HLSTM models). The learning rate is 0.001 for HLSTM models and 0.01 otherwise.

4.6.2 Experimental Results

Text Classification Results The result of supervised text classification is summarized in Table 4.2. We report the accuracy of bias prediction. Results clearly show that HLSTM outperforms the other methods in the supervised text classification setting. Also, adding the hand-engineered bias features with HLSTM representation does not help to improve performance.

Network Classification Results We show the results of predicting bias using graph information alone, without text, in Table 4.3. The GCN model outperforms DOR significantly in each of the four settings. Similar to the text classification results, performance on random and event splits are comparable. However, there is a sharp drop in performance for the time split. This can be explained by the fact that temporally separated news events will discuss different entities and world events and as a result will have very different word distributions.

[5]↑Please refer to https://github.com/BillMcGrady/NewsBiasPrediction for data and source code.

Table 4.2.
Supervised Classification Using Textual Features.

Model	Split	Text
Majority	Rand	40.10
	Event	40.10
	Time	40.50
Linear BoW	Rand	58.47
	Event	59.88
	Time	55.41
Bias Feat.	Rand	54.06
	Event	53.51
	Time	52.96
Avg WE	Rand	59.37
	Event	59.37
	Time	53.46
SkipThought	Rand	68.67
	Event	66.35
	Time	60.89
HLSTM	Rand	**74.59**
	Event	**73.55**
	Time	**66.98**
HLSTM + Bias Feat.	Rand	69.32
	Event	69.87
	Time	66.79

Event-separated splits are less susceptible to this problem, as similar figures and topics are likely to be discussed in different events.

Table 4.3.
Classification Results Using Social Relations in Full Supervised and Distant Supervised Setting.

Model	Split	Graph	User	G+U
DOR	Rand	74.74	72.02	74.57
	Event	74.87	72.74	75.18
	Time	65.65	65.07	65.36
	Dist	56.45	56.95	56.54
GCN	Rand	88.65	78.83	**88.89**
	Event	**88.78**	76.11	88.70
	Time	81.14	71.31	**82.00**
	Dist	63.72	40.08	**67.03**

Table 4.4.
Results of Joint Model Combining Text and Graph Relations.

Model	Split	Graph	User	G+U	Text	G+T	G+U+T
GCN + SkipThought	Rand	89.95	81.49	89.75	70.61	90.34	91.02
	Event	89.40	79.06	89.64	69.16	90.15	90.78
	Time	84.95	76.59	85.30	64.12	84.09	86.25
	Dist	67.78	45.30	70.03	58.68	69.82	70.66
GCN + HLSTM	Rand	89.03	83.66	88.57	86.84	91.48	**91.74**
	Event	89.34	80.22	88.62	88.39	91.69	**91.72**
	Time	84.83	74.50	85.09	81.36	85.57	**86.21**
	Dist	71.74	69.39	71.16	61.13	**72.16**	71.85

Table 4.5.
Results of Joint Model with Reduced Links for Test Documents.

Model	Split	Graph	User	G+U	Text	G+T	G+U+T
GCN + HLSTM (50%)	Rand	86.73	78.62	86.24	85.62	89.31	89.35
	Event	86.55	78.34	85.89	84.52	89.21	89.51
	Time	82.25	70.93	81.45	80.05	85.57	85.48
GCN + HLSTM (10%)	Rand	76.13	57.76	75.55	78.61	81.35	81.49
	Event	76.58	57.10	75.75	77.60	80.55	80.93
	Time	73.24	54.09	72.48	72.92	76.52	76.75

Joint Model Results

Table 4.4 shows the results of our joint model. When aligning the text and graph embeddings using joint training, both show improvement, and prediction with text or graph representations alone is better than those listed in Table 4.2 and 4.3, especially for text. Note that the increase in accuracy is much greater for the more expressive HLSTM model. Making predictions with the aggregation of multiple scores usually leads to better accuracy.

Interestingly, the model's distant supervision performance is almost comparable with fully supervised text classification results. This demonstrates the strength of our joint model, and its ability to effectively propagate label information from users down to documents.

We also evaluated our model when less social information was available at test time. We tested our joint model with only 50% and 10% of the links for test articles kept. The results are summarized in Table 4.5. Clearly, the performance improves as more social links are available. However, even with little social links provided in the latter case, our joint model

Table 4.6.
Examples of Bias Prediction by Text and Joint Model.

Text	Joint	Gold	Title
Right	Right	Right	Hacked Powell email reveals Hillary 'hates' Obama for 2008
Right	Right	Right	Donald Trump will let James Comey testify
Center	Center	Center	Clinton: I am done with being a candidate
Center	Center	Center	Senate confirms Sessions as attorney general
Left	Left	Left	Clinton: Trump Doesn't See President Obama as an American Video
Left	Left	Left	Trump uses Twitter to promote leaked intelligence on North Korea
Center	Left	Left	Hillary Clinton's Campaign Says It Will Participate In Wisconsin Recount
Left	Center	Center	Supreme Court justices hint at striking Voting Rights Act provision
Left	Center	Right	Boston Marathon bombs: how investigators use technology to identify suspects
Right	Right	Left	Israel risks becoming apartheid state if peace talks fail, says John Kerry

propagates information effectively and results in an increase in performance compared to text classification.

Qualitative Analysis In Table 4.6, we compared the bias prediction by our text and joint model on several news articles (only titles shown in the table). These examples demonstrate the subtlety of bias expression in the text, which helps motivate social representations to support the decision.

4.7 Chapter Summary

In this chapter, we follow the intuition that the political perspectives expressed in news articles will also be reflected in the way the documents spread and the identity of the users who endorse them. We suggest a GCN-based model capturing this social information, and show that it provides a distant supervision signal, resulting in a model performing comparably to supervised text classification models. We also study this approach in the supervised setting and show that it can significantly enhance a text-only classification model.

Modeling the broader context in which text is consumed is a vital step towards getting a better understanding of its perspective. We intend to explore other context information available, especially content-related ones like how different entities are described and events are framed.

5. CASE III: IMPROVING BIAS DETECTION IN NEWS ARTICLES BY PRE-TRAINING WITH SOCIAL AND LINGUISTIC INFORMATION

In the previous chapter, the sharing pattern of a news article on social media is utilized to improve bias prediction. However, it requires a lot of human effort to collect the sharing and following networks for a large set of news articles. Moreover, such information may not be available to all articles, in which case the joint model will fail to work for those articles since the connections are missing. Therefore it is natural to look for ways to leverage such social context signals through pre-training of the textual models. We can also explore other rich signals available in the news text itself to enhance our model's ability to detect bias. To this end, we propose to pre-train text models to inject knowledge we have about various social contexts, such as entities, sharing patterns, and frame usage. This information can be considered as indirect supervision that can help the model to learn what kind of text is related to certain entities, frames, or biases such that semantically close text will be embedded closer. Take frame information as an example, news articles covering the same real-world event often narrate the story with different frames, which can usually reflect the underlying perspectives it has and the agenda it would like to push. The same event can then be viewed from different angles and thus completely opposite conclusions can be reached. For example, when covering a comment related to the US government shutdown, one article states that "Senate Mark Warner blasted President Trump", while the second one emphasized the reaction of this comment, i.e. "Twitter users pointed out that Democrats are attending a retreat", implying they are not actively working toward a solution. This example clearly shows that it is important and effective to examine how an event is described when trying to identify the bias.

5.1 Introduction

The perspectives underlying the way information is conveyed to readers can prime them to take similar stances and shape their worldview [78], [79]. Given the highly polarized

coverage of news events, recognizing these perspectives can help ensure that all points of view are represented by news aggregation services, and help avoid "information echo chambers" in which only a single viewpoint is represented. It may also help to prevent the spread of false information online by showing people news with different perspectives.

Past work studying the expression of bias in the text has focused on lexical and syntactic representations of bias [46], [48], [55]. Expressions of bias can include the use of the passive voice (e.g., *"mistakes were made"*), or references to known ideological talking points [56]–[60] (e.g., *"pro-life"* vs. *"pro-choice"*). However, bias in news media is often nuanced and very difficult to detect. Journalists often strive to appear impartial and use language that does not reveal their opinions directly. Also, by their nature, news articles describing the same real-world event will share many similar details of the event, regardless of their political perspectives. Instead, bias is often expressed through informational choices [6], which highlight different aspects of the news story and frame facts shared by all articles in different ways. For example, the following articles capture different perspectives (Top *left*, Bottom *right*), while discussing the same news event– the 2021 storming of the U.S. Capitol [1].

Adapted from **NYTimes** (*Left*)

How Republicans Are Warping Reality Around the Capitol Attack ... Jim Hoft, did not reply to questions but did send along several of his own news articles related to claims of antifa involvement in the Capitol attack — citing the case of a man named **John Sullivan**, whom the right-wing media has dubbed an **"antifa leader"** in efforts to prove its theory of infiltration.

Adapted from **Fox News** (*Right*)

BLM activist inside Capitol claims he was 'documenting' riots, once said 'burn it all down'. **John Sullivan** has previously called for 'revolution' and to 'rip Trump' out of his office. An anti-Trump activist who once said he wanted to "rip" the president out of office entered the Capitol Building Wednesday alongside a mob of pro-Trump protesters, but he said he was just there to "document" it.

The two articles discuss the presentation of *John Sullivan* as an Antifa member[2] who participated in the Capitol storming. However the story is framed in very different ways -

[1] ↑https://en.wikipedia.org/wiki/2021_storming_of_the_United_States_Capitol
[2] ↑https://en.wikipedia.org/wiki/Antifa_(United_States)

while the bottom article frames the story directly as a discussion of Antifa involvement, the top discusses it in the context of political messaging and journalism. Furthermore, we notice that the difference is focused on a specific entity - John Sullivan.

Despite the fact that these distinctions are easily detectable by a human reader familiar with the political divisions in the U.S., they are very difficult to detect automatically. Recent success stories using large-scale pre-training for constructing highly expressive language models [15] are designed to capture co-occurrence patterns, likely to miss these subtle differences.

In this chapter, we suggest that bias detection requires a different set of self-supervised pre-training objectives that can help provide a better starting point for training downstream biased detection tasks. Specifically, we design three learning objectives. The first, captures *political knowledge*, focusing on the embedding of political entities discussed in the text. The second one captures *external social context*. Following the intuition that different social groups would engage with documents expressing a different bias (e.g., left-leaning users are more likely to read the NYTimes article compared to the Fox News article), we collect social information contextualizing news articles and learn to predict the social context of each article, based on its content, thus aligning the two representations. Finally, the third is based on linguistic knowledge, focusing on the *issue framing* decisions made by the authors. Framing decisions have been repeatedly shown to capture political bias [48], [80], and we argue that infusing a language model with this information can help capture relevant information. Note that this information is only used for pre-training. Other works using social information to analyze political bias [81], [82] augment the text with social information, however since this information can be difficult to obtain in real-time, we decided to investigate if it can be used as a distant supervision source for pre-training a language model.

These pre-training tasks are then used for training a **M**ulti-head **A**ttention **N**etwork (MAN) which creates a bias-aware representation of the text.

We conducted our experiments over two datasets, Allsides [81] and SemEval Hyperpartisan news detection [83]. We compared our approach to several competitive text classification models and conducted a careful ablation study designed to evaluate the individual contribu-

tion of pre-training through knowledge from various contexts. Our results demonstrate the importance of all aspects, each contributing to the model's performance.

5.2 Related Work

The problem of perspective identification is originally studied as a text classification task [46], [49], [53], in which a classifier is trained to differentiate between specific perspectives. Other works use linguistic indicators of bias and expressions of implicit sentiment [48], [56], [59].

Recent work by [6] aims to characterize content relevant for bias detection. Unlike their work which relies on annotated spans of text, we aim to characterize this content without explicit supervision.

In the recent SemEval-2019, a hyperpartisan news article detection task was suggested[3]. Many works attempt to solve this problem with deep learning models [84], [85]. We build on these works to help shape our text representation approach.

Several recent works also started to make use of concepts or entities appearing in the text to get a better representation. [86] treats the extracted concepts as pseudo words and appends them to the original word sequence which is then fed to a CNN. The KCNN model by [87], used for news recommendation, concatenates entity embeddings with the respective word embeddings at each word position to enhance the input. We take a different approach and instead try to inject knowledge of entities into the text model through the masked entity training. [88] also uses entity-level masking for training. However, they predict the tokens for the masked entity instead of relying on meaningful representations for entities like ours.

Political framing, due to its relation with ideology and perspective, is studied in the NLP communities [59], [89], [90]. There is also growing interest in utilizing framing differences to identify bias in news articles [80].

Pre-trained models are widely used in numerous NLP tasks, from the early word2vec representation [91] to the generic language models like ELMo [14] and BERT [15]. Recently, people also started to work on task-specific pre-training that tries to bring task and domain-

[3]↑https://pan.webis.de/semeval19/semeval19-web/

related knowledge into the model. [92] is similar to our work as it proposes to enhance the BERT model through training on review data and sentiment classification tasks so that it can obtain better performance across multiple review-based tasks.

5.3 Political Perspective Identification Task

The problem of political perspective identification in news media can be formalised as follows. Given a news article d, where d consists of sentences s_{i}, $\mathrm{i} \in [1, L]$, and each sentence s_{i} consists of words $w_{\mathrm{i}t}$, $t \in [1, T]$. L and T are the number of sentences in d and number of words in s_{i} respectively. The goal of this task is to predict the political perspective y of the document. Given different datasets, this can either be a binary classification task, where $y \in \{0, 1\}$ (hyperpartisan or not), or a multi-class classification problem, where $y \in \{0, 1, 2\}$ (left, center, right).

The overall architecture of our model is shown in Figure 5.1. It includes two sequence encoders, one for word level and another for sentence level. The hidden states from an encoder are combined through a multi-head self-attention mechanism. With pre-training on various social and linguistic information, the generated sentence and document vectors will consider not only the context within the text but also the knowledge about the entities (e.g. their political affiliation, or stance on controversial issues), sharing users, and frame indicators. We explain the structure of our model and the rich social and linguistic context we consider in detail below. Note that our pre-training strategies proposed in Section 5.4 is not tied with any specific model structure and can be easily applied to other text models.

5.3.1 Multi-Head Attention Network

The basic component of our model is the Hierarchical LSTM model [21]. The hierarchical structure can enable us to get both sentence and document representations. With the addition of the attention mechanism at both levels that will be described in detail below, it can provide meaningful explainability for the model behavior, where the most attended to text span contributed more to the model prediction. What's more, this model is lightweight compared to other recent textual representations models, like BERT, with much fewer pa-

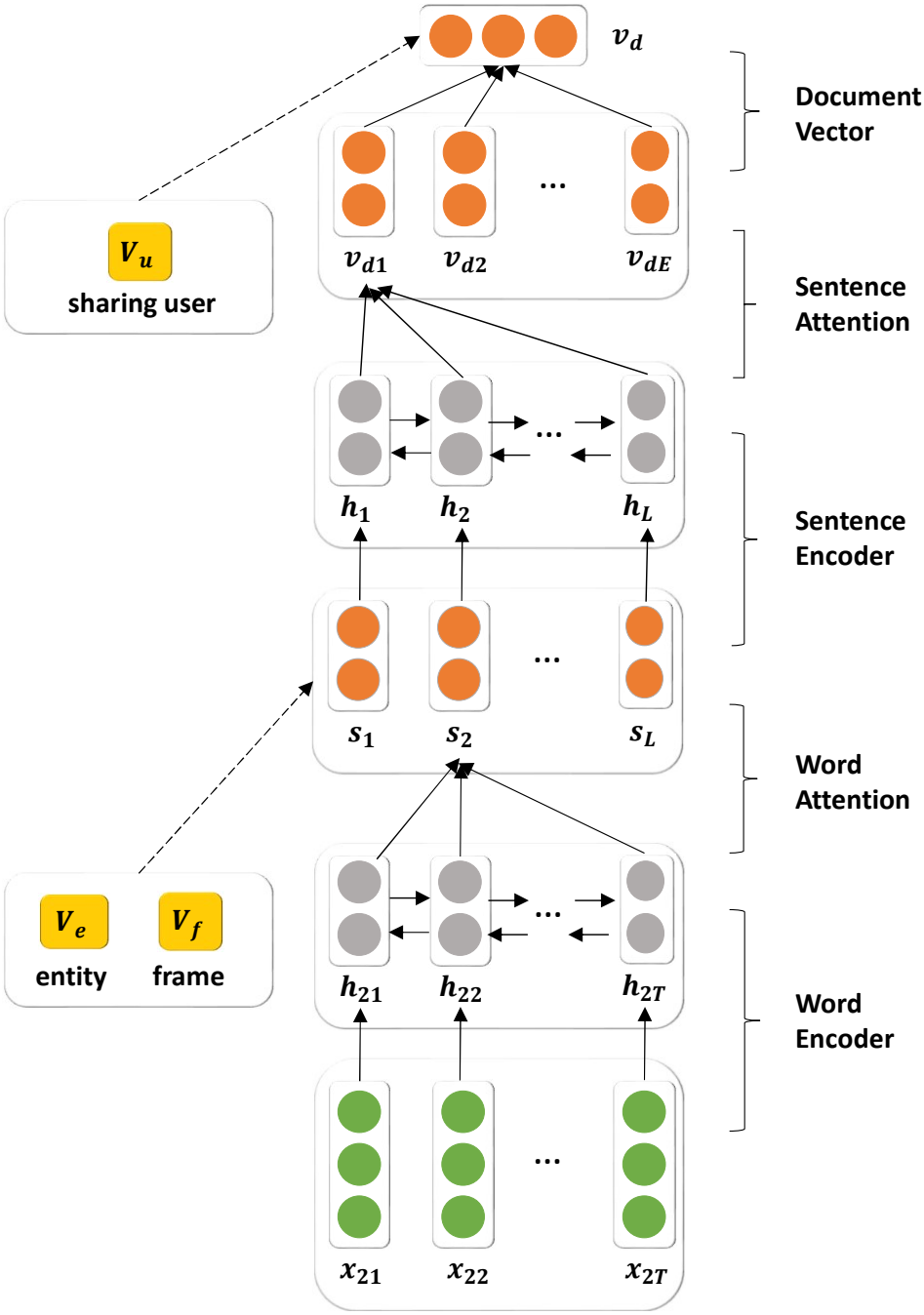


Figure 5.1. Overall Architecture of MAN Model.

rameters that need to be trained. This is suitable in our situation where the computation resource is limited. The goal of our model is to learn document representation v_d for political

perspective prediction. It consists of several parts: a word sequence encoder, a word-level attention layer, a sentence sequence encoder, and a sentence-level attention layer. We describe the details of these components in this section.

LSTM Networks Long Short Term Memory networks (LSTMs) [93] are a special kind of RNN, capable of learning long-term dependencies. Many recent works have demonstrated their ability to generate meaningful text representations. For each element in the input sequence, the hidden state h is computed by a LSTM cell with the following functions:

$$\mathrm{i}_t = \sigma(W_\mathrm{i} x_t + U_\mathrm{i} h_{t-1} + b_\mathrm{i}) \tag{5.1}$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{5.2}$$

$$g_t = \tanh(W_g x_t + U_g h_{t-1} + b_g) \tag{5.3}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{5.4}$$

$$c_t = f_t \cdot c_{t-1} + \mathrm{i}_t \cdot g_t \tag{5.5}$$

$$h_t = o_t \cdot \tanh(c_t) \tag{5.6}$$

where h_t is the hidden state at time t, c_t is the cell state at time t, x_t is the input at time t, h_{t-1} is the hidden state at time $t-1$ or the initial hidden state at time 1, i_t, f_t, g_t, o_t are the input, forget, cell and output gates, respectively. σ is the sigmoid function, and $\cdot$ is the Hadamard product (element-wise product).

To capture the context in both directions, we use bidirectional LSTM in this work. The final hidden state h at each position of a sequence is a concatenation of the forward hidden state $\overrightarrow{h}$ and backward hidden state $\overleftarrow{h}$ computed by the respective LSTM cells.

Word Sequence Encoder Given a sentence with words $w_{\mathrm{i}t}$, $t \in [1, T]$, each word is first converted to its embedding vector $x_{\mathrm{i}t}$. We can adopt pre-trained Glove [13] word embeddings or deep contextualized word representation ELMo [94] for this step. The word vectors are then fed into a word-level bidirectional LSTM network to incorporate contextual information within the sentence. The hidden states $h_{\mathrm{i}t}$ from the bidirectional LSTM network are passed to the next layer.

Word Level Attention In [21], a self-attention mechanism is introduced to identify words that are important to the meaning of the sentence, and therefore higher weights are given to them when forming the aggregated sentence vector.

$$p_{itw} = tanh(W_w h_{it} + b_w) \tag{5.7}$$

$$\alpha_{itw} = \frac{exp(p_{itw}^T p_w)}{\sum_t exp(p_{itw}^T p_w)} \tag{5.8}$$

$$s_{iw} = \sum_t \alpha_{itw} h_{it} \tag{5.9}$$

p_{itw} encodes the importance of a specific word according to its context, which is compared with the word level preference vector p_w to compute a similarity score. The scores are then normalized to get the attention weight α_{itw} through a softmax function. A weighted sum of the word hidden states is computed based on the attention weight as the sentence vector s_{iw}.

Inspired by the multi-head attention scheme in [95], we propose multi-head attention in our model to extend its ability to jointly attend to information at different positions. The sentence vector s_i is computed as an average of s_{iw} obtained from different attention heads. Note that we learn a separate copy of the parameters W_w, b_w and p_w for each attention head.

$$s_i = \frac{\sum_w s_{iw}}{NH_W} \tag{5.10}$$

where NH_W is the number of word-level attention heads.

Sentence Sequence Encoder and Sentence Level Attention Given the sentence vectors s_i, $i \in [1, L]$, we can generate the document vector v_d in a similar way. Sentence vectors s_i are fed into a sentence level bidirectional LSTM network to propagate context information along sentences. The hidden states h_i from the sentence bidirectional LSTM are passed to the sentence level attention layer. Bias is usually not expressed in every sentence in a document, especially for news articles that generally cover real-world events and try to seem impartial. To this end, we again use the attention mechanism to highlight sentences

that are useful to determine the bias of an article. Similar to word-level attention, the hidden states h_{i} are used to compute the attention weight for each sentence. After that, the document vector v_{ds} is obtained as a weighted average of hidden states h_{i}. v_{ds} obtained from different attention heads are averaged to generate entity oriented document representation v_d.

$$p_{\mathrm{is}} = tanh(W_s h_{\mathrm{i}} + U_s v_{\mathrm{e}} + b_s) \tag{5.11}$$

$$\alpha_{\mathrm{is}} = \frac{\mathrm{e}xp(p_{\mathrm{is}}^T p_s)}{\sum_t \mathrm{e}xp(p_{\mathrm{is}}^T p_s)} \tag{5.12}$$

$$v_{ds} = \sum_t \alpha_{\mathrm{is}} h_{\mathrm{i}} \tag{5.13}$$

$$v_d = \frac{\sum_s v_{ds}}{NH_S} \tag{5.14}$$

where NH_S is the number of attention heads at the sentence level.

Document Classification The document representations v_d captures the bias-related information in news article d. They can be used as features for predicting the document bias label.

$$f_d = W_c v_d + b_c \tag{5.15}$$

$$p_d = \mathrm{softmax}(f_d) \tag{5.16}$$

We use the negative log likelihood of the correct labels as classification training loss:

$$L = -\sum_d \log p_{d\mathrm{j}} \tag{5.17}$$

where j is the bias label of d.

5.3.2 Political Entities

News articles, especially the ones we are interested in in this work, are mainly covering real-world events involving political entities and their relations. To better understand the stance over controversial issues and the underlying ideology reflected in the text, it is very important to have extensive world knowledge about these entities, including their traits, opinions, and relevant events. We obtain the entity knowledge representations through learning on Wikipedia data.

Wikipedia2Vec [96] is a model that learns entity embeddings from Wikipedia. It learns embeddings of words and entities by iterating over the entire Wikipedia pages and maps similar words and entities close to one another in a continuous vector space. It jointly optimizes the following three submodels:

1. Wikipedia link graph model, which learns entity embeddings by predicting neighboring entities in Wikipedia's link graph, an undirected graph whose nodes are entities, and edges represent links between entities in their Wikipedia pages.

2. Word-based skip-gram model, which learns word embeddings by predicting neighboring words given each word on a Wikipedia page.

3. Anchor context model, which aims to place similar words and entities near one another in the vector space. The objective here is to predict neighboring words given each entity referred to on a Wikipedia page.

The learned entity embeddings encode the background knowledge about these entities in Wikipedia, such as gender, ideology, among others. We use them to initialize our entity embeddings in Section 5.4.1 which enables us to inject background knowledge of entities to the text model through pre-training.

5.3.3 Social Information Graph

With the great popularity of social media platforms, many people nowadays tend to share their personal interests and opinions and exchange ideas about social events with

others online. This also applies to the sharing of news articles on social media. Intuitively, news articles shared by the same user are likely to have the same bias, and users who share a lot of news in common are close in their political preferences as well. Hence, we can use this information to guide the pre-training of our text model.

We follow the work in [81] to learn the embeddings through the structure of the social information graph for users who share articles. The graph consists of three types of vertices, namely political users, sharing users, and news articles. Political users are famous politicians or journalists with a clear, self-reported political bias. Sharing users are Twitter users who shared news articles in the dataset. There are two types of edges: 1) following edge between a sharing user to a political user and 2) sharing edge between a sharing user to a news article). Graph Convolutional Networks (GCN) is used to model the graph structure to predict the bias of political users. It aggregates information from the local neighborhood for each node in the graph. Therefore the training of GCN helps to propagate political preference information from political users to sharing users. We use the learned embeddings to guide the pre-training in Section 5.4.2 so that our text model can use this as distant supervision to map the representation of news articles shared by the same user to be close in the vector space since they are more likely to have the same perspective.

5.3.4 Frame Indicators

Political framing, studied by political scientists, provides a useful way to study different political perspectives. The frames surrounding an issue can change the reader's perception without having to alter the actual facts as the same information is used as a base. It is a political strategy that used to bias the discussion on an issue toward a specific stance. For example, regarding the topic of abortion, the liberal side will highlight the freedom of choice for women to decide whether to terminate a pregnancy while the conservative side may emphasize the morality aspect instead, arguing the right of the fetus.

Previous work [80] shows that frame indicators can be used to identify the political perspectives effectively for different topics. These are words that have high pointwise mutual information with a specific frame. They can be considered to represent a more detailed point

within a frame. Therefore we propose to use these frame indicators to guide the pre-training of the text model so that it can learn to distinguish the nuance between different frames and talking points.

5.4 Pre-training

As discussed in the introduction, the supervision on news bias requires a lot of human effort to get. Moreover, the text model trained only on the political perspective labels cannot benefit from the rich knowledge we have from the various social and linguistic contexts presented in the previous section. To enhance the performance of political perspective identification, we may need to bring external knowledge and signals from the aforementioned context to enable the text model to take them into account when processing the news article. Eventually, we want to show that the model works best by exploiting all different kinds of knowledge and signals related to the task.

5.4.1 Entity Guided Pre-training

The goal of entity-guided pre-training is to inject knowledge about entities into our text model to help solve the political perspective identification problem. We first extract entities from the data corpus and then learn knowledge representations for them using Wikipedia2Vec introduced in 5.3.2. We then use the learned entity representations to pre-train the text model such that it is able to predict the masked entity given the context in a sentence.

We utilize the entity linking system DBpedia Spotlight [97] to recognize and disambiguate the entities in news articles. We use the default configuration of DBpedia Spotlight, including the confidence threshold of 0.35, which helps to exclude uncertain or wrong entity annotations. We keep only entities with Person or Organization types that appear in the corpus since they are usually the main agents in the news articles and the characteristics involved in the discussion of them can often reveal the perspective of those articles.

Inspired by the masked language modeling objective used in BERT [15], we propose an entity-level masking task for injecting background knowledge of entities into the text model based on the news articles in which they are mentioned. The objective is to predict the

masked entity based on the context provided by the other words in a sentence. Specifically, the entity mentions (regardless of the number of tokens in text) are replaced with a special token "[MASK]" during preprocessing. We use a bidirectional LSTM (sentence level encoder described in 5.3.1) to encode the sentence, and the hidden state of the mask token will be used for prediction. We use negative sampling to randomly generate negative entity candidates from all entities in our dictionary uniformly. The prediction can be done by comparing the similarity score between the hidden state and the embedding of candidate entities mapped to the same space through a hidden layer.

$$h_{it}^{T} \cdot (W_{\mathrm{e}} v_{\mathrm{e}} + b_{\mathrm{e}}) \tag{5.18}$$

where h_{it} is the hidden state for the masked token, v_{e} the embedding of entity e, W_{e} and b_{e} the parameters for the mapping hidden layer. We use the multi-class cross-entropy loss for all pre-training tasks.

The learned sentence encoder will then be able to highlight the context in the news articles that is more related to the properties and traits of the mentioned entities.

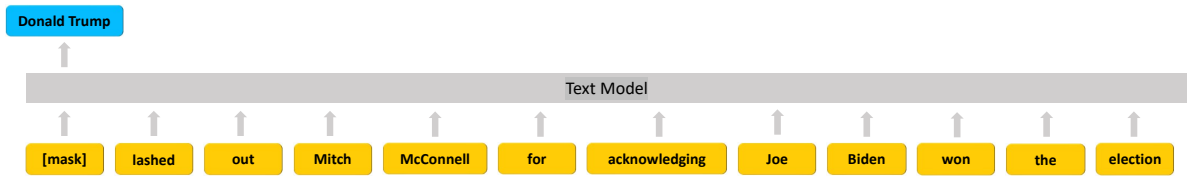


Figure 5.2. Example of Entity Guided Pre-training.

5.4.2 Sharing User Guided Pre-training

As we discussed in Section 5.3.3, the sharing behavior by Twitter users can be regarded as signals to guide the pre-training of our text model. In order to benefit from the social information available, we propose to predict the sharing user given a news article. Similar

to the previous part, we use negative sampling to generate negative sharing user candidates uniformly. The prediction is based on similarity scores defined below

$$v_{v_d}^T \cdot (W_s v_s + b_s) \tag{5.19}$$

where v_d is the document vector for d, v_s the embedding of sharing user s, W_s and b_s the parameters for the hidden layer.

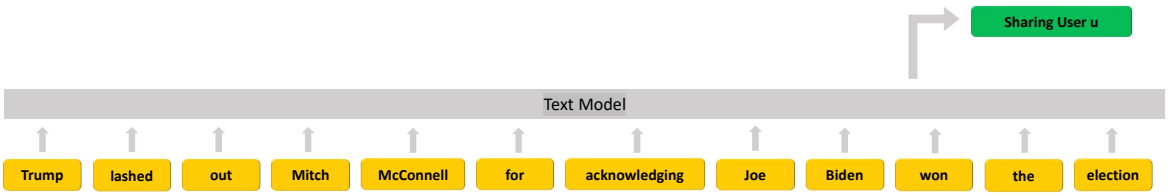


Figure 5.3. Example of Sharing User Guided Pre-training.

5.4.3 Frame Indicator Guided Pre-training

The frame indicator guided pre-training is almost identical to the entity-guided one except that the masked tokens are frame indicators instead of entity mentions.

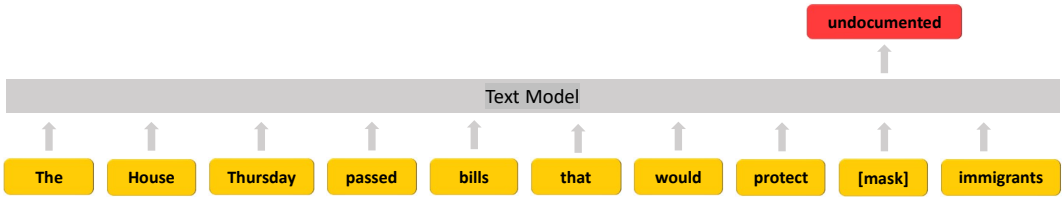


Figure 5.4. Example of Frame Indicator Guided Pre-training.

5.4.4 Ensemble of Multiple Models

Given the entity and user embeddings are not in the same space, we use them to pre-train separate models. All pre-trained models are then trained with the supervision of political

perspective labels in the same way. We also explore an ensemble of the three models, which makes predictions based on a weighted sum of unnormalized scores f_d in equation 5.15 from these models at test time.

$$\sum_m f_{dm} * \beta_m \tag{5.20}$$

where m denotes a trained prediction model, f_{dm} the unnormalized scores for document d by model m and β_m the weight given to model m which can be tuned based on the data.

5.5 Experiments

We aim to answer the following research questions (RQs) in the experiment:

RQ1: what is the performance gain of pre-training the text model with each social and linguistic information, with respect to the baseline models?

RQ2: what is the respective contribution by the individual pre-trained models to the full ensemble model?

RQ3: how will the performance gain change given the different amount of labeled data available for training?

5.5.1 Datasets and Evaluation

We run experiments on two news article datasets: Allsides and SemEval. The statistics of both datasets are shown in Table 5.1.

Allsides This dataset [81] is collected from two news aggregation websites[4] on 2020 different events discussing 94 event types. The websites provide news coverage from multiple perspectives, indicating the bias of each article using crowdsourced and editorial reviewed approaches. Each article has a political perspective label left, center, or right. We used the same randomly separated splits in [81] for evaluation in this chapter so that our results are directly comparable with theirs.

[4]↑Allsides.com and Memeorandum.com

SemEval This is the official training dataset from SemEval 2019 Task 4: Hyperpartisan News Detection [83]. The task is to decide whether a given news article follows a hyperpartisan argumentation. There are 645 articles in this dataset and each is labeled manually with a binary label to indicate whether it is hyperpartisan or not. Since the test set is not available at this time. We conducted 10-fold cross-validation on the training set with the exact same splits used in [84] so that we can compare with the system that ranked in the first place in the competition.

Table 5.1.
Datasets Statistics.

Dataset	Center	Left	Right	Avg # Sent.	Avg # Words
Allsides	4164	3931	2290	49.96	1040.05
		Hyperpartisan			
SemEval	407	238		27.11	494.29

5.5.2 Baselines

We compare our model with several competitive baseline methods.

BERT is a language representation model based on deep bidirectional Transformer architectures [95]. It was pre-trained with the masked language model and next sentence prediction tasks on a huge corpus. As a result, it can achieve state-of-the-art results on a wide range of tasks by fine-tuning with just one additional output layer.

CNN_Glove (CNN_ELMo) is the model from the team that ranked first in hyperpartisan news detection task in SemEval 2019 [84]. It uses the pre-trained Glove (ELMo) word vectors, which are then averaged as sentence representations. The sentence vectors are fed into 5 convolutional layers of different kernel sizes. The outputs for all convolution layers are combined to form the input to a fully connected layer, which maps to the final text representation. Some extra improvements include batch normalization and ensemble of multiple models.

5.5.3 Implementation Details

We use the spaCy toolkit for preprocessing the documents. All models are implemented with PyTorch [76][5]. The 300d Glove word vectors [13] trained on 6 billion tokens are used to convert words to word embeddings. The ELMo model we used is the medium one with output size 512. They are not updated during training. The sizes of LSTM hidden states for both word level and sentence level are 300 for both Allsides and SemEval dataset. The number of attention heads at both word and sentence levels is set to 4 for the Allsides dataset, while it is set to 1 for the SemEval dataset due to its size. For the training of the neural network, we used the Adam optimizer [77] to update parameters. On Allsides dataset, 5% of the training data is used as the validation set. We perform early stopping using the validation set. However, same as [84], we use the evaluation part of each fold for early stopping and model selection due to the limited size of the SemEval dataset. The patience for early stopping p is equal to 10, meaning that the training stops when there is no improvement in validation performance for ten consecutive epochs. The learning rate lr is set to 0.001 for all models except BERT for which $2e-5$ is used. The mini-batch size $b = 10$ for bias prediction.

Regarding pre-training data sources, we use the training set for Allsides, and extract 100,000 news articles for SemEval from the large dataset provided by SemEval 2019 Task 4 respectively. The entity and user embeddings used for pre-training are obtained through external resources described in Section 5.3.2 and 5.3.3. The embeddings for frame indicators are randomly initialized. All of them were updated during the pre-training to better adapt to the text model. The optimizer and most hyper-parameters stay the same as the training of bias prediction. The mini-batch size is set to 2000 and 300 for models using Glove and ELMo respectively since the training is at the sentence level. The number of examples for each type of pre-training and dataset is shown in Table 5.2. Note that the entity-guided and frame-guided pre-training is at the sentence level and sharing guided pre-training is at the document level. Therefore there is an order of magnitude higher number of examples for the first two pre-training strategies.

[5]↑Please refer to https://github.com/BillMcGrady/NewsBiasPretraining for data and source code.

Table 5.2.
Pre-training Statistics.

Dataset	Entity-Guided	Sharing-Guided	Frame-Guided
Allsides	122665	75488	1038864
SemEval	1561823	-	21628494

5.5.4 Results

Results on Allsides

We report the average accuracy and macro F1 scores on test sets for Allsides dataset in Table 5.3. The results are divided into two groups based on whether contextualized word representations are used. To answer RQ1, we observed that, in most cases, models with pre-training outperform the MAN baseline. It demonstrates our pre-training step can effectively utilize signals in social and linguistic context to enhance the text model to identify bias expressed in more subtle ways. Therefore it generates high-quality document representation for political perspective prediction. The sharing guided pre-training did not lead to much improvement by itself. This is mainly because the sharing users in our dataset often share news articles with various perspectives. Our ensemble model achieves the best result in terms of both accuracy and macro F1 scores no matter whether contextualized word embeddings are used or not. It shows the signals from various sources are complementary with each other such that even a simple combination of prediction scores can lead to significant improvement. The gaps between our model and baselines decrease when contextualized word representations are used since local context is better captured in this setting.

Results on SemEval

The performance of various models on the SemEval dataset can be found in Table 5.4. Note that there is no sharing user guided result in this table since we do not have social graph information available in this dataset. Again the results are grouped based on word representation used. CNN_Glove and CNN_ELMo are results reported by the winning team in the SemEval competition. They proposed an ensemble of multiple CNN models where

Table 5.3.
Test Results on Allsides Dataset.

Model	Accuracy	Macro F1
MAN_Glove	78.29	76.96
+ Entity-Guided	80.50	79.50
+ Sharing-Guided	78.93	77.84
+ Frame-Guided	81.26	80.15
Ensemble	**83.74**	**82.84**
BERT	81.55	80.13
MAN_ELMO	81.41	80.44
+ Entity-Guided	82.27	81.23
+ Sharing-Guided	81.37	80.48
+ Frame-Guided	82.56	81.66
Ensemble	**85.00**	**84.25**

each CNN takes sentence representation generated by average ELMo embedding as input. It is worth noting that our model with Glove as word representation is comparable with the winning team's model with ELMo, showing the advantages of pre-training. The other trends hold as well in the SemEval dataset. In both datasets, our pre-trained models beat BERT easily since they are tuned specifically for the task.

Table 5.4.
Test Results on SemEval Dataset. † indicates results reported in [84].

Model	Accuracy	Macro F1
CNN_Glove †	79.63	-
MAN_Glove	81.58	79.29
+ Entity-Guided	82.65	80.75
+ Frame-Guided	83.27	81.73
Ensemble	**84.03**	**82.42**
CNN_ELMO †	84.04	-
BERT	84.03	82.60
MAN_ELMO	84.66	83.09
+ Entity-Guided	85.59	84.15
+ Frame-Guided	85.27	83.32
Ensemble	**86.21**	**84.33**

Ablation Study

To answer RQ2, we show the results for ablations of our ensemble model based on MAN_Glove in Table 5.5. The performance drops when removing each one of the pre-trained models from the ensemble, showing that the information obtained from different sources is complementary with each other. To make a fair comparison with the baseline model, we also report the performance of an ensemble of multiple baseline models (denoted as -Pre-training) with different seeds from random initialization. This shows the absolute gain through pre-training to adapt the text representations for political perspective identification.

Table 5.5.
Ablation Study on Allsides Dataset.

Model	Accuracy	Macro F1
Ensemble	83.74	82.84
- Entity-Guided	82.57	81.65
- Sharing-Guided	82.78	81.78
- Frame-Guided	82.39	81.40
- Pre-training	81.54	80.40

Results with Limited Training Data

One of the obstacles in obtaining good performance in political perspective identification tasks is the lack of supervision data. We compare the performance of the MAN_Glove model with and without entity-guided pre-training with different levels of training examples available in Figure 5.5. These results can help to answer RQ3. It shows that the performance gain obtained from our pre-training strategy increases as the size of the training set decreases. This is a very useful property as it can greatly improve model performance when there is limited training data. It is worth noting that the Sharing-Guided Pre-training achieves much higher performance when supervision is limited. This is because the signals from the sharing users can be considered as noisy bias labels and it is trained at document level instead of sentence level like the other two. However, since the other two pre-training methods

introduce extra knowledge to the text model, they can lead to better performance when the supervision is abundant to provide enough bias information for training.

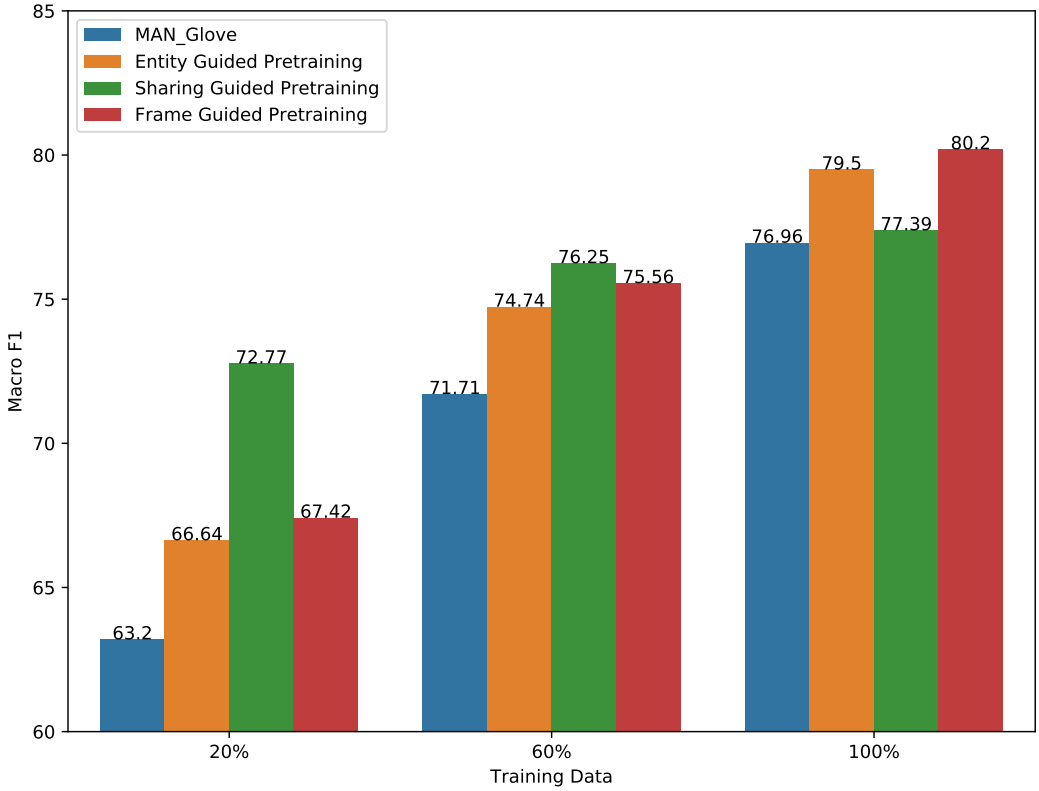


Figure 5.5. Test Results with Different Number of Training Examples.

Qualitative Results

Human Annotation Comparison

The BASIL dataset [6] has human annotations of bias spans. It contains 300 articles on 100 events with 1727 bias spans annotated. On the sentence level, spans of lexical and informational bias are identified by annotators by analyzing whether the text tends to affect a reader's feeling towards one of the main entities. We compute the average attention assigned by our model to the annotated bias spans. Table 5.6 shows the results of the baseline model (MAN) and the same model pre-trained with entity information (+Entity). The attention

scores assigned to the human annotation spans are higher across training, validation, and test sets.

Table 5.6.
Average Attention Scores on Basil Annotations.

Model	Training	Validation	Test
MAN	0.706	0.701	0.652
+ Entity	0.737	0.728	0.666
Improvement	4.36%	3.76%	2.13%

5.6 Chapter Summary

In this chapter, we propose a pre-training framework to adapt text representation for political perspective identification. Empirical experiments on two recent news article datasets show that an ensemble of pre-trained models achieves significantly better performance in bias detection compared to competitive text baselines. It is also shown that our pre-training model can achieve even larger performance gain when the supervision is limited.

In fact, these various context information are not independent. We intend to extend this work to pre-train better text models by incorporating information from various sources together.

6. CONCLUSION

In this dissertation we study the problem of stance and bias detection in text, a challenging classification problem, which requires connecting textual content analysis with relevant social contexts. In particular, we consider three social contexts that can be helpful in understanding the stance and bias expressed in text.

- Firstly the author profile information and conversational structure between text are used to predict stance of posts on online debate forums. We design structured representation learning model to embed all items in the shared space over which inference can be applied to constrain the decisions.

- Secondly the sharing patterns of text on social media is considered to analyze the political perspectives of news articles. We design joint text and graph model to propagate information between text and graph models.

- Thirdly we propose to pre-train the text representation model using linguistic and social information available. We believe, by injecting these related knowledge about entities, frames, and biases into the text model, the model can learn to distinguish different perspectives better. Models pre-trained with various information can also complement each other when combined.

We summarize our findings here, and discuss potential continuations of this research.

Stance prediction of online debate posts requires connecting textual content analysis with conversational interactions and author information. Traditionally, this is done using a graphical model, which learns a scoring function for each aspect, over a fixed feature representation. We follow the observation that all of these problems are connected, and allow the model to capture these dependencies by allowing it to *learn* a representation for all these aspects jointly, rather than using a fixed representation. We show that by formulating the decision problem over the representation directly, and requiring the representation to respect the global dependencies between these aspects, our model can generalize better and exploit additional information even when it is not directly relevant. To the best of our

knowledge, this work is the first to cast representation learning as a structured prediction problem, we believe that this approach is applicable to many other domains where the input has complex inter-connected structure. Such domains include other conversation analysis tasks, shared representations of text and images and information networks such as citations graphs and social network analysis.

In the case of detecting the political perspectives expressed in news articles, we realize that biases will also be reflected in the way the documents spread and the identity of the users who endorse them. We suggest a GCN-based model capturing this social information, and show that it provides a distant supervision signal, resulting in a model performing comparably to supervised text classification models. We also study this approach in the supervised setting and show that it can significantly enhance a text-only classification model. The use of GCN here enable us to remove inference at both test and training time. This is because the embedding procedure for each node in the graph already takes the neighborhood information into consideration, similar to what will be done at inference time. Empirical study demonstrates the power of our joint text and graph model. The prediction performance of document representation from text and graph model both excels that of trained separately. Moreover, by adding the prediction scores of two representations, we can obtain the best overall performance.

Although the sharing pattern on social media reveal much information about biases of news articles, it still requires time and human effort to collect, and sometimes no such information is available at all. Therefore, we want to consider some contexts that is readily available, which leads to narrative of entities and events in news articles. News articles, unlike purely political texts, usually attempt to maintain credibility and seem impartial. As a result, bias is introduced in subtle ways, usually by emphasizing different aspects of the story. Therefore, if we can bring knowledge about entities and frames into the model, either implicitly or explicitly, there is a higher chance we can correctly identify the bias of an article. With that in mind, we propose to pre-train the text model with rich social and linguistic information available, including entity mentions, sharing pattern, and frame usages. During the pre-training, the text model would incorporate external knowledge and learn to embed sentences and documents which discuss the same topic similarly close to each

other. This enable the text model to better associate parts of text with the respective entities or frames. Empirical Results show that our pre-training approach achieves significant better performance than existing text models. The improvement increases when the supervision is reduced, which is a great advantage since the supervision is expensive to get for this task.

With models designed to make use of the various social contexts available in different settings for the stance and bias detection tasks, one future direction is naturally to come up with a general framework that can integrate these models and social contexts. Such integration has the potential to provide a better background for the understanding of the text and ideas reflected in it. This is not trivial though since we need to be careful about controlling the impact of different information when they are aggregated during the learning the text representations. Another interesting direction is to consider the combination of task-specific pre-training and language model based pre-training (e.g. BERT). The latter has been used in many tasks with great success. However, we have shown that task-specific pre-training can provide signals that are not captured in the more general LM-based approach. The LM-based pre-training on huge corpus augmented with task-specific pre-training with social context information may lead to a better text model, which takes advantage of both worlds, for identifying stance and bias.

REFERENCES

[1] R. M. Entman, "Framing: Toward clarification of a fractured paradigm," *Journal of Communication*, vol. 43(4), pp. 51–58, 1993.

[2] D. Chong and J. N. Druckman, "Framing theory," *Annu. Rev. Polit. Sci.*, vol. 10, pp. 103–126, 2007.

[3] K. S. Hasan and V. Ng, "Stance classification of ideological debates: Data, models, features, and constraints," in *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, 2013, pp. 1348–1356.

[4] D. Sridhar, J. Foulds, B. Huang, L. Getoor, and M. Walker, "Joint models of disagreement and stance in online debate," in *Proc. of the Annual Meeting of the Association Computational Linguistics (ACL)*, vol. 1, 2015, pp. 116–125.

[5] R. Baly, G. Karadzhov, D. Alexandrov, J. Glass, and P. Nakov, "Predicting factuality of reporting and bias of news media sources," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 3528–3539. DOI: 10.18653/v1/D18-1389. [Online]. Available: https://www.aclweb.org/anthology/D18-1389.

[6] L. Fan, M. White, E. Sharma, R. Su, P. K. Choubey, R. Huang, and L. Wang, "In plain sight: Media bias through the lens of factual reporting," 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. 6343–6349. DOI: 10.18653/v1/D19-1664. [Online]. Available: https://www.aclweb.org/anthology/D19-1664.

[7] R. Abbott, B. Ecker, P. Anand, and M. A. Walker, "Internet argument corpus 2.0: An sql schema for dialogic social media and the corpora to go with it.," in *Proc. of the International Conference on Language Resources and Evaluation (LREC)*, 2016.

[8] T. N. Kipf and M. Welling, "Variational graph auto-encoders," *CoRR*, vol. abs/1611.07308, 2016. arXiv: 1611.07308. [Online]. Available: http://arxiv.org/abs/1611.07308.

[9] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017. [Online]. Available: https://openreview.net/forum?id=SJU4ayYgl.

[10] S. Lloyd, "Least squares quantization in pcm," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982. DOI: 10.1109/TIT.1982.1056489.

[11] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philosophical Magazine*, vol. 2, pp. 559–572, 6 1901.

[12] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[13] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[14] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 2227–2237. DOI: 10.18653/v1/N18-1202. [Online]. Available: https://www.aclweb.org/anthology/N18-1202.

[15] J. Devlin, M.-W. Chang, K. Lee, and K. 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, Jun. 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. [Online]. Available: https://www.aclweb.org/anthology/N19-1423.

[16] G. Jawahar, B. Sagot, and D. Seddah, "What does BERT learn about the structure of language?" In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 3651–3657. DOI: 10.18653/v1/P19-1356. [Online]. Available: https://www.aclweb.org/anthology/P19-1356.

[17] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *KDD*, 2014.

[18] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 855–864.

[19] D. Bertsimas and J. Tsitsiklis, *Introduction to linear optimization*. Athena Scientific, 1997.

[20] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, and M. Sun, *Graph neural networks: A review of methods and applications*, cite arxiv:1812.08434, 2018. [Online]. Available: http://arxiv.org/abs/1812.08434.

[21] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document 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, Jun. 2016, pp. 1480–1489. DOI: 10.18653/v1/N16-1174. [Online]. Available: https://www.aclweb.org/anthology/N16-1174.

[22] S. Somasundaran and J. 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*, Association for Computational Linguistics, 2010, pp. 116–124.

[23] P. Anand, M. Walker, R. Abbott, J. E. F. Tree, R. Bowmani, and M. 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*, Association for Computational Linguistics, 2011, pp. 1–9.

[24] D. Ghosh, S. Muresan, N. Wacholder, M. Aakhus, and M. Mitsui, "Analyzing argumentative discourse units in online interactions," in *Proceedings of the First Workshop on Argumentation Mining*, 2014, pp. 39–48.

[25] M. A. Walker, P. Anand, R. Abbott, and R. Grant, "Stance classification using dialogic properties of persuasion," in *Proc. of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*, Association for Computational Linguistics, 2012, pp. 592–596.

[26] S. Mohammad, S. Kiritchenko, P. Sobhani, X. Zhu, and C. Cherry, "Semeval-2016 task 6: Detecting stance in tweets," in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, 2016, pp. 31–41.

[27] R. Dong, Y. Sun, L. Wang, Y. Gu, and Y. Zhong, "Weakly-guided user stance prediction via joint modeling of content and social interaction," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, ACM, 2017, pp. 1249–1258.

[28] M. A. Walker, R. Abbott, and J. King, "A corpus for research on deliberation and debate.," in *Proc. of the International Conference on Language Resources and Evaluation (LREC)*, 2012.

[29] M. Bansal, C. Cardie, and L. Lee, "The power of negative thinking: Exploiting label disagreement in the min-cut classification framework," *COLING 2008: Companion Volume: Posters*, pp. 15–18, 2008.

[30] C. Burfoot, S. Bird, and T. Baldwin, "Collective classification of congressional floor-debate transcripts," in *Proc. of the Annual Meeting of the Association Computational Linguistics (ACL)*, Association for Computational Linguistics, 2011, pp. 1506–1515.

[31] K. Johnson and D. Goldwasser, "" all i know about politics is what i read in twitter": Weakly supervised models for extracting politicians' stances from twitter," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 2016, pp. 2966–2977.

[32] I. Augenstein, T. Rocktäschel, A. Vlachos, and K. Bontcheva, "Stance detection with bidirectional conditional encoding," in *Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 876–885.

[33] J. Ebrahimi, D. Dou, and D. Lowd, "Weakly supervised tweet stance classification by relational bootstrapping.," in *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, 2016.

[34] K. S. Hasan and V. Ng, "Why are you taking this stance? identifying and classifying reasons in ideological debates," in *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, 2014, pp. 751–762.

[35] R. Kiros, Y. Zhu, R. Salakhutdinov, R. S. Zemel, A. Torralba, R. Urtasun, and S. Fidler, "Skip-thought vectors," in *The Conference on Advances in Neural Information Processing Systems (NIPS)*, 2015, pp. 3294–3302.

[36] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *International Conference on Machine Learning*, 2014, pp. 1188–1196.

[37] J. Li, A. Ritter, and D. Jurafsky, "Learning multi-faceted representations of individuals from heterogeneous evidence using neural networks," *CoRR*, 2015.

[38] A. Vaswani, Y. Bisk, K. Sagae, and R. Musa, "Supertagging with lstms," in *Proc. of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*, 2016, pp. 232–237.

[39] A. Katiyar and C. Cardie, "Investigating lstms for joint extraction of opinion entities and relations," in *Proc. of the Annual Meeting of the Association Computational Linguistics (ACL)*, Berlin, Germany, Aug. 2016, pp. 919–929.

[40] G. Durrett and D. Klein, "Neural crf parsing," in *Proc. of the Annual Meeting of the Association Computational Linguistics (ACL)*, 2015, pp. 302–312.

[41] G. Lample, M. Ballesteros, K. Kawakami, S. Subramanian, and C. Dyer, "Neural architectures for named entity recognition," in *Proc. of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*, 2016, pp. 1–10.

[42] D. Andor, C. Alberti, D. Weiss, A. Severyn, A. Presta, K. Ganchev, S. Petrov, and M. Collins, "Globally normalized transition-based neural networks," in *Proc. of the Annual Meeting of the Association Computational Linguistics (ACL)*, 2016, pp. 2442–2452.

[43] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'13, Lake Tahoe, Nevada: Curran Associates Inc., 2013, pp. 3111–3119. [Online]. Available: http://dl.acm.org/citation.cfm?id=2999792.2999959.

[44] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[45] I. Gurobi Optimization, *Gurobi optimizer reference manual*, 2016. [Online]. Available: http://www.gurobi.com.

[46] S. Greene and P. Resnik, "More than words: Syntactic packaging and implicit sentiment," in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Boulder, Colorado: Association for Computational Linguistics, Jun. 2009, pp. 503–511. [Online]. Available: https://www.aclweb.org/anthology/N09-1057.

[47] B. Beigman Klebanov, E. Beigman, and D. Diermeier, "Vocabulary choice as an indicator of perspective," in *Proceedings of the ACL 2010 Conference Short Papers*, Uppsala, Sweden: Association for Computational Linguistics, Jul. 2010, pp. 253–257. [Online]. Available: https://www.aclweb.org/anthology/P10-2047.

[48] M. Recasens, C. Danescu-Niculescu-Mizil, and D. Jurafsky, "Linguistic models for analyzing and detecting biased language," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Sofia, Bulgaria: Association for Computational Linguistics, Aug. 2013, pp. 1650–1659. [Online]. Available: https://www.aclweb.org/anthology/P13-1162.

[49] M. Iyyer, P. Enns, J. Boyd-Graber, and P. Resnik, "Political ideology detection using recursive neural networks," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Baltimore, Maryland: Association for Computational Linguistics, Jun. 2014, pp. 1113–1122. DOI: 10.3115/v1/P14-1105. [Online]. Available: https://www.aclweb.org/anthology/P14-1105.

[50] S. Volkova, K. Shaffer, J. Y. Jang, and N. Hodas, "Separating facts from fiction: Linguistic models to classify suspicious and trusted news posts on twitter," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 647–653. DOI: 10.18653/v1/P17-2102. [Online]. Available: https://www.aclweb.org/anthology/P17-2102.

[51] M. Glenski, T. Weninger, and S. Volkova, "Identifying and understanding user reactions to deceptive and trusted social news sources," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 176–181. [Online]. Available: https://www.aclweb.org/anthology/P18-2029.

[52] F. N. Ribeiro, L. Henrique, F. Benevenuto, A. Chakraborty, J. Kulshrestha, M. Babaei, and K. P. Gummadi, "Media bias monitor: Quantifying biases of social media news outlets at large-scale," in *Proceedings of the Twelfth International Conference on Web and Social Media, ICWSM 2018, Stanford, California, USA, June 25-28, 2018.*, 2018, pp. 290–299. [Online]. Available: https://aaai.org/ocs/index.php/ICWSM/ICWSM18/paper/view/17878.

[53] W.-H. Lin, T. Wilson, J. Wiebe, and A. 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*, ser. CoNLL-X '06, New York City, New York: Association for Computational Linguistics, 2006, pp. 109–116. [Online]. Available: http://dl.acm.org/citation.cfm?id=1596276.1596297.

[54] Y. Choi and J. Wiebe, "+/-EffectWordNet: Sense-level lexicon acquisition for opinion inference," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1181–1191. DOI: 10.3115/v1/D14-1125. [Online]. Available: https://www.aclweb.org/anthology/D14-1125.

[55] H. Elfardy, M. Diab, and C. Callison-Burch, "Ideological perspective detection using semantic features," in *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, Denver, Colorado: Association for Computational Linguistics, Jun. 2015, pp. 137–146. DOI: 10.18653/v1/S15-1015. [Online]. Available: https://www.aclweb.org/anthology/S15-1015.

[56] E. Baumer, E. Elovic, Y. Qin, F. Polletta, and G. Gay, "Testing and comparing computational approaches for identifying the language of framing in political news," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Denver, Colorado: Association for Computational Linguistics, May 2015, pp. 1472–1482. DOI: 10.3115/v1/N15-1171. [Online]. Available: https://www.aclweb.org/anthology/N15-1171.

[57] C. Budak, S. Goel, and J. M. Rao, "Fair and balanced? quantifying media bias through crowdsourced content analysis," *Public Opinion Quarterly*, vol. 80, no. S1, pp. 250–271, 2016. DOI: 10.1093/poq/nfw007.

[58] D. Card, J. Gross, A. Boydstun, and N. A. Smith, "Analyzing framing through the casts of characters in the news," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 1410–1420. DOI: 10.18653/v1/D16-1148. [Online]. Available: https://www.aclweb.org/anthology/D16-1148.

[59] A. Field, D. Kliger, S. Wintner, J. Pan, D. Jurafsky, and Y. Tsvetkov, "Framing and agenda-setting in Russian news: A computational analysis of intricate political strategies," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 3570–3580. [Online]. Available: https://www.aclweb.org/anthology/D18-1393.

[60] F. Morstatter, L. Wu, U. Yavanoglu, S. R. Corman, and H. Liu, "Identifying framing bias in online news," *Trans. Soc. Comput.*, vol. 1, no. 2, 5:1–5:18, Jun. 2018, ISSN: 2469-7818. DOI: 10.1145/3204948. [Online]. Available: http://doi.acm.org/10.1145/3204948.

[61] A. Patwari, D. Goldwasser, and S. Bagchi, "TATHYA: A multi-classifier system for detecting check-worthy statements in political debates," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 06 - 10, 2017*, 2017, pp. 2259–2262. DOI: 10.1145/3132847.3133150. [Online]. Available: https://doi.org/10.1145/3132847.3133150.

[62] S. Pan, J. Wu, X. Zhu, C. Zhang, and Y. Wang, "Tri-party deep network representation," in *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2016.

[63] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *The International World Wide Web Conference*, 2015.

[64] Y. Yang, M.-W. Chang, and J. Eisenstein, "Toward socially-infused information extraction: Embedding authors, mentions, and entities," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 1452–1461. DOI: 10.18653/v1/D16-1152. [Online]. Available: https://www.aclweb.org/anthology/D16-1152.

[65] R. West, H. S. Paskov, J. Leskovec, and C. Potts, "Exploiting social network structure for person-to-person sentiment analysis," *TACL*, 2014.

[66] M. S. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, 2018, pp. 593–607. DOI: 10.1007/978-3-319-93417-4_38. [Online]. Available: https://doi.org/10.1007/978-3-319-93417-4%5C_38.

[67] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, "Distant supervision for relation extraction without labeled data," 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. 1003–1011. [Online]. Available: https://www.aclweb.org/anthology/P09-1113.

[68] D. X. Zhou, P. Resnick, and Q. Mei, "Classifying the political leaning of news articles and users from user votes," in *Proceedings of the Fifth International Conference on Weblogs and Social Media, Barcelona, Catalonia, Spain, July 17-21, 2011*, 2011. [Online]. Available: http://www.aaai.org/ocs/index.php/ICWSM/ICWSM11/paper/view/2782.

[69] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[70] B. D. Horne, S. Khedr, and S. Adali, "Sampling the news producers: A large news and feature data set for the study of the complex media landscape," in *Proceedings of the Twelfth International Conference on Web and Social Media, ICWSM 2018, Stanford, California, USA, June 25-28, 2018.*, 2018, pp. 518–527. [Online]. Available: https://aaai.org/ocs/index.php/ICWSM/ICWSM18/paper/view/17796.

[71] B. D. Horne, W. Dron, S. Khedr, and S. Adali, "Assessing the news landscape: A multi-module toolkit for evaluating the credibility of news," in *Companion Proceedings of the The Web Conference 2018*, ser. WWW '18, Lyon, France: International World Wide Web Conferences Steering Committee, 2018, pp. 235–238, ISBN: 978-1-4503-5640-4. DOI: 10.1145/3184558.3186987. [Online]. Available: https://doi.org/10.1145/3184558.3186987.

[72] R. Kiros, Y. Zhu, R. Salakhutdinov, R. S. Zemel, R. Urtasun, A. Torralba, and S. Fidler, "Skip-thought vectors," in *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, 2015, pp. 3294–3302. [Online]. Available: http://papers.nips.cc/paper/5950-skip-thought-vectors.

[73] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'13, Lake Tahoe, Nevada: Curran Associates Inc., 2013, pp. 3111–3119. [Online]. Available: http://dl.acm.org/citation.cfm?id=2999792.2999959.

[74] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proceedings of the 34th International Conference on Machine Learning*, D. Precup and Y. W. Teh, Eds., ser. Proceedings of Machine Learning Research, vol. 70, International Convention Centre, Sydney, Australia: PMLR, Jun. 2017, pp. 1263–1272. [Online]. Available: http://proceedings.mlr.press/v70/gilmer17a.html.

[75] D. Duvenaud, D. Maclaurin, J. Aguilera-Iparraguirre, R. Gómez-Bombarelli, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints," in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'15, Montreal, Canada: MIT Press, 2015, pp. 2224–2232. [Online]. Available: http://dl.acm.org/citation.cfm?id=2969442.2969488.

[76] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[77] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[78] M. Gentzkow and J. M. Shapiro, "What drives media slant? evidence from us daily newspapers," *Econometrica*, vol. 78, no. 1, pp. 35–71, 2010.

[79] M. Gentzkow and J. M. Shapiro, "Ideological segregation online and offline," *The Quarterly Journal of Economics*, vol. 126, no. 4, pp. 1799–1839, 2011.

[80] S. Roy and D. Goldwasser, "Weakly supervised learning of nuanced frames for analyzing polarization in news media," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online: Association for Computational Linguistics, Nov. 2020, pp. 7698–7716. DOI: 10.18653/v1/2020.emnlp-main.620. [Online]. Available: https://www.aclweb.org/anthology/2020.emnlp-main.620.

[81] C. Li and D. Goldwasser, "Encoding social information with gcn for political perspective detection in news media," in *ACL*, Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 2594–2604. [Online]. Available: https://www.aclweb.org/anthology/P19-1247.

[82] V.-H. Nguyen, K. Sugiyama, P. Nakov, and M.-Y. Kan, "Fang: Leveraging social context for fake news detection using graph representation," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 1165–1174.

[83] J. Kiesel, M. Mestre, R. Shukla, E. Vincent, P. Adineh, D. Corney, B. Stein, and M. Potthast, "SemEval-2019 task 4:hyperpartisan news detection," Minneapolis, Minnesota, USA: Association for Computational Linguistics, Jun. 2019, pp. 829–839. DOI: 10.18653/v1/S19-2145. [Online]. Available: https://www.aclweb.org/anthology/S19-2145.

[84] Y. Jiang, J. Petrak, X. Song, K. Bontcheva, and D. Maynard, "Hyperpartisan news detection using ELMo sentence representation convolutional network," in *SemEval*, Minneapolis, Minnesota, USA: Association for Computational Linguistics, Jun. 2019, pp. 840–844. DOI: 10.18653/v1/S19-2146. [Online]. Available: https://www.aclweb.org/anthology/S19-2146.

[85] K. Hanawa, S. Sasaki, H. Ouchi, J. Suzuki, and K. Inui, "The sally smedley hyperpartisan news detector at SemEval-2019 task 4," in *SemEval*, Minneapolis, Minnesota, USA: Association for Computational Linguistics, Jun. 2019, pp. 1057–1061. DOI: 10.18653/v1/S19-2185. [Online]. Available: https://www.aclweb.org/anthology/S19-2185.

[86] J. Wang, Z. Wang, D. Zhang, and J. Yan, "Combining knowledge with deep convolutional neural networks for short text classification," in *IJCAI*, ser. IJCAI, Melbourne, Australia: AAAI Press, 2017, pp. 2915–2921, ISBN: 978-0-9992411-0-3. [Online]. Available: http://dl.acm.org/citation.cfm?id=3172077.3172295.

[87] H. Wang, F. Zhang, X. Xie, and M. Guo, "Dkn: Deep knowledge-aware network for news recommendation," in *WWW*, ser. WWW' 18, Lyon, France: International World Wide Web Conferences Steering Committee, 2018, pp. 1835–1844, ISBN: 978-1-4503-5639-8. DOI: 10.1145/3178876.3186175. [Online]. Available: https://doi.org/10.1145/3178876.3186175.

[88] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu, "ERNIE: Enhanced language representation with informative entities," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 1441–1451. DOI: 10.18653/v1/P19-1139. [Online]. Available: https://www.aclweb.org/anthology/P19-1139.

[89] K. Johnson, D. Jin, and D. Goldwasser, "Modeling of political discourse framing on twitter," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 11, 2017.

[90] C. Shurafa, K. Darwish, and W. Zaghouani, "Political framing: Us covid19 blame game," in *International Conference on Social Informatics*, Springer, 2020, pp. 333–351.

[91] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[92] H. Xu, B. Liu, L. Shu, and P. Yu, "BERT post-training for review reading comprehension and aspect-based sentiment analysis," 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, Jun. 2019, pp. 2324–2335. DOI: 10.18653/v1/N19-1242. [Online]. Available: https://www.aclweb.org/anthology/N19-1242.

[93] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comp.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. [Online]. Available: http://dx.doi.org/10.1162/neco.1997.9.8.1735.

[94] M. Gardner, J. Grus, M. Neuman, O. Tafjord, P. Dasigi, N. F. Liu, M. Peters, M. Schmitz, and L. Zettlemoyer, "Allennlp: A deep semantic natural language processing platform," 2017. eprint: arXiv:1803.07640.

[95] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., Curran Associates, Inc., 2017, pp. 5998–6008. [Online]. Available: http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

[96] I. Yamada, A. Asai, H. Shindo, H. Takeda, and Y. Takefuji, "Wikipedia2vec: An optimized tool for learning embeddings of words and entities from wikipedia," *arXiv*, 2018.

[97] J. Daiber, M. Jakob, C. Hokamp, and P. N. Mendes, "Improving efficiency and accuracy in multilingual entity extraction," in *I-SEMANTICS*, ser. I-SEMANTICS '13, Graz, Austria: ACM, 2013, pp. 121–124, ISBN: 978-1-4503-1972-0. DOI: 10.1145/2506182.2506198. [Online]. Available: http://doi.acm.org/10.1145/2506182.2506198.

VITA

Chang Li was born in Hengyang, Hunan, China. He received his Ph.D. degree in Computer Science from Purdue University under the supervision of Dr. Dan Goldwasser. His research interests span natural language processing and machine learning. Specifically, his Ph.D. thesis is about improving stance and bias detection in text by incorporating social context. He has extensive experience in text classification, sentiment analysis, and representation learning. During his Ph.D. study, he interned at Twitter and Google. Before coming to Purdue, he received his B.S. degree in Computer Science from Nanjing University in China. He enrolled in the Elite Program at Nanjing University.