Title

Detecting political biases of named entities and hashtags on Twitter

Journal
EPJ Data Science, 12(1)

ISSN
2193-1127

Authors
Xiao, Zhiping
Zhu, Jeffrey
Wang, Yining
et al.

Publication Date
2023

DOI
10.1140/epjds/s13688-023-00386-6

Xiao et al. *EPJ Data Science* _###################_
https://doi.org/10.1140/epjds/s13688-023-00386-6

EPJ.org

EPJ Data Science
a SpringerOpen Journal

REGULAR ARTICLE Open Access

Detecting political biases of named entities and hashtags on Twitter

Zhiping Xiao[1*], Jeffrey Zhu[1], Yining Wang[1], Pei Zhou[2], Wen Hong Lam[1], Mason A. Porter[3,4] and Yizhou Sun[1]

*Correspondence: patricia.xiao@cs.ucla.edu
[1]Department of Computer Science, University of California, Los Angeles, 580 Portola Plaza, 90095, Los Angeles, California, United States of America
Full list of author information is available at the end of the article

Abstract

Ideological divisions in the United States have become increasingly prominent in daily communication. Accordingly, there has been much research on political polarization, including many recent efforts that take a computational perspective. By detecting political biases in a corpus of text, one can attempt to describe and discern the polarity of that text. Intuitively, the named entities (i.e., the nouns and the phrases that act as nouns) and hashtags in text often carry information about political views. For example, people who use the term "pro-choice" are likely to be liberal, whereas people who use the term "pro-life" are likely to be conservative. In this paper, we seek to reveal political polarities in social-media text data and to quantify these polarities by explicitly assigning a polarity score to entities and hashtags. Although this idea is straightforward, it is difficult to perform such inference in a trustworthy quantitative way. Key challenges include the small number of known labels, the continuous spectrum of political views, and the preservation of both a polarity score and a polarity-neutral semantic meaning in an embedding vector of words. To attempt to overcome these challenges, we propose the **P**olarity-aware **E**mbedding **M**ulti-task learning (**PEM**) model. This model consists of (1) a self-supervised context-preservation task, (2) an attention-based tweet-level polarity-inference task, and (3) an adversarial learning task that promotes independence between an embedding's polarity dimension and its semantic dimensions. Our experimental results demonstrate that our **PEM** model can successfully learn polarity-aware embeddings that perform well at tweet-level and account-level classification tasks. We examine a variety of applications—including spatial and temporal distributions of polarities and a comparison between tweets from Twitter and posts from Parler—and we thereby demonstrate the effectiveness of our **PEM** model. We also discuss important limitations of our work and encourage caution when applying the **PEM** model to real-world scenarios.

Keywords: Data sets; Word embeddings; Multi-task learning; Adversarial training

1 Introduction

In the United States, discourse has seemingly become very polarized politically and it often seems to be divided along ideological lines [1, 2]. This ideological division has become increasingly prominent, and it influences daily communication.

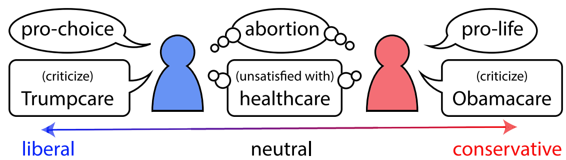


Figure 1 Illustration of inferring political polarities from text

The analysis of data from social media is important for studying human discourse [3, 4]. To study the polarization of social opinions in online communication, we attempt to detect polarity biases of entities and hashtags. There are a variety of ways to model political biases; see, e.g., VoteView (see https://voteview.com/) [5]. A space of political opinions can include axes for social views (e.g., ranging from "conservative" to "progressive"), economic views (e.g., ranging from "socialist" to "capitalist"), views on government involvement (e.g., ranging from "libertarian" to "authoritarian"), and many others. The simplest model of a political spectrum, which we use in the present paper, is to consider a one-dimensional (1D) political space with views that range from "liberal" to "conservative".

By glancing at a corpus of text (such as a newspaper article or a tweet), humans can often readily recognize particular views in it without the need to analyze every word in the corpus. Many items (including named entities and hashtags) in a corpus of text are helpful for inferring political views [6], and people can quickly discern political views even in small corpora of text or in short speeches.

On Twitter, political biases are often reflected in the entities and hashtags in tweets. The entities that we use are nouns and noun phrases (i.e., phrases that act as nouns), which we identify from text corpora by using existing natural-language-processing (NLP) tools. For instance, as we illustrate in Fig. 1, if somebody uses the term "pro-choice" to describe abortion, they may have a liberal-leaning stance on a liberal–conservative axis of political views [7]. By contrast, if somebody uses the term "pro-life", perhaps they have a conservative-leaning stance. We propose to automate this process in an interpretable way by detecting the political biases of entities and hashtags, inferring their attention weights in tweets, and then inferring the political polarities of tweets.

The problem of inferring political polarities from text is somewhat reminiscent of "fairness-representation" problems [8, 9]. This analogy is not perfect, and these problems have different objectives. We aim to reveal polarities, whereas fairness studies are typically interested in removing polarities. The notion of fairness entails that outputs are unaffected by personal characteristics such as gender, age, and place of birth. In recent studies, Zhao et al. [8] examined how to detect and split gender bias from word embeddings and Bose and Hamilton [9] developed models to hide personal information (such as gender and age) from the embeddings of nodes in graph neural networks (GNNs). Political bias can be more subtle and change faster than other types of biases. A key challenge is the labeling of political ideologies. Unlike the inference of gender bias, where it is typically reasonable to use discrete (and well-aligned) word pairs such as "he"/"she" and "waiter"/"waitress" as a form of ground truth, political polarity includes many ambiguities [10]. Political ideology exists on a continuous spectrum, with unclear extremes, so it is very hard to determine either ground-truth polarity scores or well-aligned word pairs (e.g., "he" versus "she" is aligned with "waiter" versus "waitress") [11].

To infer polarities, we seek to learn an embedding that can help reveal both the semantic meaning and the political biases of entities and hashtags. We propose a model, which we call the **P**olarity-Aware **E**mbedding **M**ulti-task learning (**PEM**) model, that involves three tasks: (1) preservation of the context of words; (2) preservation of corpus-level polarity information; and (3) an adversarial task to try to ensure that the semantic and polarity components of an embedding are as independent of each other as possible.

Our paper makes the following contributions:

(1) We raise the important and practical problem of studying political bias in a corpus of text, and we assemble a data set from Twitter to study this problem. Our code, the data sets of the politicians, and the embedding results of our models are available at https://bitbucket.org/PatriciaXiao/pem/src/master/.

(2) We propose the **PEM** model to simultaneously capture both semantic and political-polarity meanings.

(3) Our **PEM** model does not rely on word pairs to determine political polarities. Consequently, it is flexible enough to adapt to other types of biases and to use in other context-preservation strategies.

(4) Our data, source code, and embedding results are helpful for tasks such as revealing potential political polarities in a text corpus.

2 Related work and preliminary discussions

2.1 Political-polarity detection

There are a variety of ways to formally define the notion of political polarity [5]. We consider a 1D axis of political views that range from "liberal" to "conservative". In the United States, members of the Democratic party tend to be liberal and members of the Republican party tend to be conservative [1, 12]. This prior knowledge is helpful for acquiring high-quality labeled data [13], but such data are restricted in both amount and granularity.

The detection of political polarity has been a topic of considerable interest for many years [14, 15]. Additionally, for more than a decade, social-media platforms like Twitter have simultaneously been an important source of political opinion data and have themselves impacted political opinions in various ways [16, 17]. Some researchers have attempted to infer the political views of Twitter accounts from network relationships (such as following relationships) [13, 18, 19]. Other researchers have attempted to infer polarity from tweet text [20, 21].

We seek to infer the political polarities of entities and hashtags in tweets. Gordon et al. [22] illustrated recently that word embeddings can capture information about political polarity, but their approach does not separate polarity scores from embeddings and thus cannot explicitly tell which words are biased. Most prior research has focused on tweet-level or account-level polarities [23, 24] or on case studies of specific "representative" hashtags [25]. By contrast, our **PEM** model focuses on biases at a finer granularity (specifically, entities and hashtags).

2.2 Neural word embeddings

We use the term *neural word embeddings* to describe approaches to represent tokens (e.g., words) using vectors to make them understandable by neural networks [26–28]. Words can have very different meanings under different tokenizations. In our paper, we tokenize text into entities (including nouns and noun phrases), hashtags, emoji, Twitter handles, and other words (including verbs, adjectives, and so on). One way to obtain a neural word embedding is the SKIP-GRAM version of WORD2VEC approaches [29], which are based on the assumption that similar words have similar local textual contexts. Another approach, which is called GLOVE [30], relies on a global co-occurrence matrix of words. Other methods, such as transformers [31, 32], generate contextualized embeddings (in which a word can have different embeddings in different contexts). These models encode words, which initially take the form of a sequence of characters, into a vector space. Therefore, these models are also often called "encoders".

In contrast to all of the above studies, our **PEM** model learns an embedding that captures both the semantic meanings and the political polarities of words. Our framework is not limited to any specific embedding strategy. If desired, one can replace the embedding part (namely, Task #1) of our **PEM** model by other encoders.

2.3 Fairness of representations

Many researchers have observed that word embeddings often include unwanted biases [33]. In studies of fairness, a model is considered to be "fair" if its outputs are unaffected by personal characteristics, such as gender and age; it is "biased" (i.e., "unfair") if such features influence the outputs. Models often inherit biases from training data sets, and they can exacerbate such biases [34]. Researchers have undertaken efforts to reveal biases and mitigate them [9]. For example, Zhao et al. revealed gender-bias problems using their WINOBIAS model [35] and attempted to generate gender-neutral representations using their GN-GLOVE model [8].

Such representation-learning algorithms motivate us to separate politically-biased and politically-neutral components in embeddings (see [8]) and to use an adversarial training framework to enhance the quality of the captured polarities (see [9]). However, our work has a different focus than [8] and [9]. These works were concerned with reducing biases, whereas we seek to reveal differences between polarized groups.

2.4 Sentiment analysis

Sentiment analysis aims to determine the attitude (negative, positive, or neutral) of a corpus of text [36, 37]. The use of neural word embeddings is common in statistical approaches to sentiment analysis [38, 39]. Some of these approaches account for the importance levels of entities [40, 41].

In many applications, sentiment analysis has relied on much richer labeled data sets than those that are available in political contexts [37, 42], where it is rare to find high-quality anchor words (such as good, bad, like, and dislike) [38]. In our paper, we seek to reveal polarities from textual data. Polarity is different from sentiment. For example, most entities have neutral sentiments, but these same entities can still have biased polarities.

2.5 Recognition of named entities

We focus on learning polarity scores for named entities (specifically, nouns and noun phrases) and hashtags. The terminology "named entity", which comes from NLP, refers to a noun or a noun phrase that is associated with an entity. For example, the *United States Congress* is a named entity. We use a named-entity recognition (NER) tool [43, 44] to identify the entities in our training corpus. In an NER information-extraction task, one seeks to discern and classify entities in a text corpus into predefined categories, such as person names, organizations, and locations. We use the popular tools TagMe [45] and AutoPhrase [46] for our tasks.

3 Problem definition

We use "tokens" to denote the smallest word units that we obtain through tokenization of tweets. We tokenize entities, hashtags, emoji, mentioned accounts, and other words. We represent each tweet as a sequence of such tokens. We study the problem of detecting the political biases of entities and hashtags in tweets. To do this, we seek to learn (1) semantic embeddings for each token and (2) the political polarities of each entity and hashtag. We then obtain tweet-level polarity scores by calculating a weighted average of token-level polarity scores.

Definition 1 (Two-Component Polarity-Aware Embeddings) *We design a two-component polarity-aware embedding* $\mathbf{z} \in \mathbb{R}^{d_1+d_2}$ *of each token* $\mathbf{w}$. *Because we seek to learn* 1*D polarity scores, we set* $d_2 = 1$. *We decompose* $\mathbf{z}$ *as follows*:

$$\mathbf{z} = \left[\mathbf{z}^{(s)}, \mathbf{z}^{(p)}\right], \qquad \mathbf{z}^{(s)} \in \mathbb{R}^{d_1}, \mathbf{z}^{(p)} \in \mathbb{R}^{d_2}.$$

The two components of the embedding $\mathbf{z}$ *are*

(1) *the* ***polarity-neutral*** *semantic component* $\mathbf{z}^{(s)}$ *and*
(2) *the* ***polarity-aware*** *political-polarity component* $\mathbf{z}^{(p)}$.

By forcing $\mathbf{z}^{(s)}$ to be polarity-neutral, we seek to enhance the quality of the political polarities that we capture in $\mathbf{z}^{(p)}$. We set $d_1 = d$ and $d_2 = 1$, and we use $f(\mathbf{z}^{(p)}) = z_{d+1}$ as the "polarity score" of a token. When determining tweet-level polarities, we ignore $\mathbf{z}^{(p)}$ for tokens that are neither entities nor hashtags. We expect that $z_{d+1} < 0$ when a word is liberal-leaning and that $z_{d+1} > 0$ when a word is conservative-leaning. The absolute value $|z_{d+1}|$ indicates the magnitude of a political leaning. Using our approach, we are able to infer the political polarity of a token in $\mathcal{O}(1)$ time. We are interested in the polarity scores of tokens that are either entities or hashtags. It is very common to use a 1D polarity score [5], so we do so in the present paper. However, it is straightforward to extend our **PEM** model to incorporate more polarity dimensions.

4 Methodology

4.1 General design

To generate our proposed embeddings, we infer semantic meanings, infer political polarities, and use $\mathbf{z}^{(p)}$ to capture as much political polarity as possible.

We show a schematic illustration of our model in Fig. 2. To capture the meanings of tokens, we learn embeddings from the context of text. We thus propose Task #1 to help preserve contextual information. To infer political polarities from tokens, we propose Task #2, in which we use a weighted average of the entities' and hashtags' polarity component $\mathbf{z}^{(p)}$ to calculate a polarity score of each tweet. To further enhance the quality of the polarity component, we propose Task #3, in which we use an adversarial framework to ensure that the two components, $\mathbf{z}^{(s)}$ and $\mathbf{z}^{(p)}$, are as independent as possible.

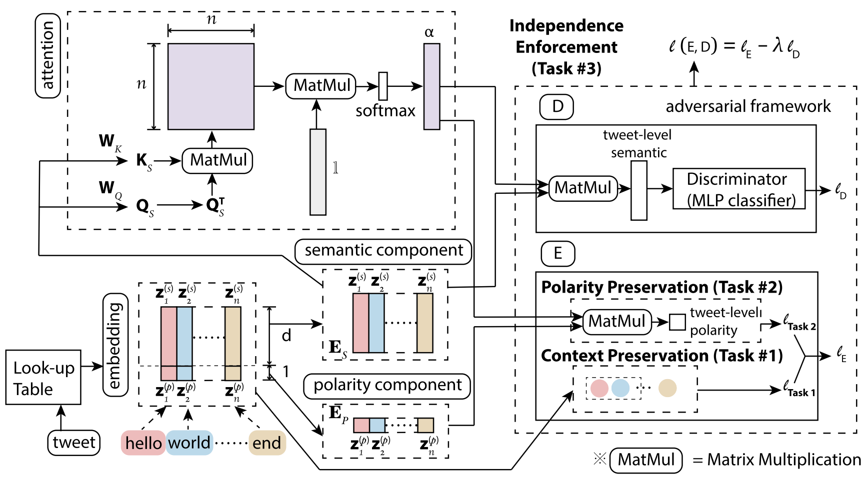


Figure 2 Schematic illustration of our **PEM** model. In this illustration, we consider a tweet with n tokens

4.2 Task #1: context preservation

We want our token-level embeddings to preserve contextual information, which has both semantic information and polarity information. A simple approach is to use SKIP-GRAM [29]. Given a document with tokens $w_1, w_2, \ldots, w_n$, we seek to maximize the mean log probability to observe tokens in a local context. Specifically, we maximize

$$\frac{1}{n}\sum_{t=1}^{n}\sum_{j\in\{-c,\ldots,c\},j\neq 0}\ln p(w_{t+j}|w_t), \tag{1}$$

where c indicates the size of a sliding window and

$$p(w_{t+j}|w_t) = \frac{\exp(\mathbf{z}_t^T\mathbf{z}'_{t+j})}{\sum_{i=1}^{|\mathbf{W}|}\exp(\mathbf{z}_t^T\mathbf{z}'_i)}, \tag{2}$$

where w_i is the ith token in the document, the set $\mathbf{W}$ is the vocabulary set of all tokens, $\mathbf{z}_i$ is the target embedding of token w_i, and $\mathbf{z}'_i$ is the context embedding. When the index $t+j \notin \{1,\ldots,n\}$, we ignore it in (2). In Task# 1, we need both $\mathbf{z}_i$ and $\mathbf{z}'_i$ to be able to distinguish between the target and context roles of the same token [29]. In Task #2 (see Sect. **??**) and Task #3 (see Sect. 4.4), we use only the context embedding $\mathbf{z}'_i$.

The loss function $\ell_{\text{Task 1}}$ for Task #1 is the negative-sampling objective function

$$\ell_{\text{Task 1}} = -\frac{1}{k+1}\left(\ln\left(\sigma\left(\mathbf{z}_t^T\mathbf{z}'_{t+j}\right)\right) + \sum_{i=1}^{k}\mathbb{E}_{w_i\sim P_{\text{noise}}(w)}\left[\ln\left(\sigma\left(-\mathbf{z}_t^T\mathbf{z}'_i\right)\right)\right]\right), \tag{3}$$

where k is the number of negative samples (i.e., token pairs that consist of a target token and a token from a noise distribution) per positive sample (i.e., token pairs that occur in the same sliding window), the sigmoid function σ is $\sigma(x) = \frac{1}{1+\exp(-x)}$, and $P_{\text{noise}}(\cdot)$ is a noise distribution. We obtain negative samples of word pairs from the noise distribution [29], whose name comes from the idea of noise-contrastive estimation (NCE) [47]. A good model should distinguish between data and noise. We use the same noise distribution as in SKIP-GRAM [29]:

$$P_{\text{noise}}(w) = \left(\frac{U(w)}{\sum_{i\in\mathbf{W}} U(i)}\right)^{3/4}, \tag{4}$$

where $U(w)$ denotes the number of appearances of a token w in the training corpus. Minimizing $\ell_{\text{Task 1}}$ approximates the maximization of the mean log probability (1).

In practice, when discussing political affairs, they are usually described by multiple words, namely, phrases. We use AUTOPHRASE [46] to detect phrases in our data sets, and treat them as tokens as well.

We refer to Task #1 as our **Baseline PEM** model, and we call it the "SKIP-GRAM model" when we use it on its own. We use the same hyperparameter settings as in the default settings in the original SKIP-GRAM model [29].

4.3 Task #2: polarity preservation

In Task #2, our goal is for the polarity component of our embeddings to capture reasonable polarity information. The finest granularity of the polarity labels that we can automatically and reliably obtain in large enough numbers are at the level of social-media accounts. We assume that every politician has consistent political views during our observation time (the years 2019 and 2020), and we assign polarity labels to their tweets based on their self-identified party affiliations. We thereby use account-level labels to guide the polarity-score learning of entities and hashtags.

A simple approach is to use the mean polarity score of all entities to estimate the polarity score of a text corpus. However, this approach does not consider the heterogeneous importance levels of entities. When considering political tendencies, some entities (e.g., "pro-choice") are more informative than others (e.g., "plan"). Therefore, we calculate a weighted average of entity polarities in each tweet, with weights that come from attention.

Suppose that we are given a sentence with n tokens (i.e., words, phrases, hashtags, mentions, emoji, and so on) that are embedded as $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_n$, where m of the n tokens are entities or hashtags. The set of indices of the m tokens is $\mathbf{I} = \{i_1, \dots, i_m\}$ (with $m \le n$). The polarity dimensions of the embeddings are

$$\mathbf{E}_P = \left[\mathbf{z}_{i_1}^{(p)}; \mathbf{z}_{i_2}^{(p)}; \dots; \mathbf{z}_{i_m}^{(p)}\right] \in \mathbb{R}^{m\times 1}.$$

We use a standard self-attention mechanism [48], which proceeds as follows. We represent keys, values, and queries in a vector space. Each key has a corresponding value. Upon receiving a query, we evaluate similarities between the queries and the keys. We then estimate the value of a query as a weighted average of the values that correspond to the keys [31].

We vertically concatenate the sequence of the semantic (i.e., polarity-neutral) components of the entities' and hashtags' embeddings and write

$$\mathbf{E}_S = \left[\mathbf{z}_{i_1}^{(s)}; \mathbf{z}_{i_2}^{(s)}; \dots; \mathbf{z}_{i_m}^{(s)}\right] \in \mathbb{R}^{m\times d},$$

where the key $\mathbf{K}$ and the query $\mathbf{Q}$ are different linear transformations of $\mathbf{E}_S$. That is,

$$\mathbf{K} = \text{stopgrad}(\mathbf{E}_S)\mathbf{W}_K, \mathbf{Q} = \text{stopgrad}(\mathbf{E}_S)\mathbf{W}_Q,$$

where stopgrad is a stop gradient (so $\mathbf{E}_S$ is not updated by back-propagation of the attention component) and $\mathbf{W}_K, \mathbf{W}_Q \in \mathbb{R}^{d\times d}$ are weight matrices. We calculate the attention vector $\boldsymbol{\alpha} \in \mathbb{R}^{m\times 1}$, which includes an attention score for each entity in a tweet, using the standard softmax function:

$$\boldsymbol{\alpha} = \mathrm{Att}(\mathbf{Q},\mathbf{K}) = \mathrm{softmax}\left(\left(\frac{\mathbf{QK}^T}{\sqrt{m}}\right)\cdot \mathbf{1}_{m\times 1}\right), \tag{5}$$

where the ith component of the softmax function is

$$\mathrm{softmax}(x_i) = \frac{e^{x_i}}{\sum_{k=1}^{m} e^{x_k}}$$

and $\mathbf{1}_{m\times 1}$ is a vector of 1 entries.

Each tweet's polarity score $\tilde{\mathbf{z}}^{(p)}$ is then

$$\tilde{\mathbf{z}}^{(p)} = \boldsymbol{\alpha}^T \mathbf{E}_P \in \mathbb{R}^{1\times 1}. \tag{6}$$

Suppose that there are N tweets in total and that tweet j has the associated label $l_j \in \{-1, 1\}$, where -1 signifies that the tweet is by a politician from the Democratic party and 1 signifies that the tweet is by a politician from the Republican party. (We only consider politicians with a party affiliation.) We infer polarity scores $\{\tilde{\mathbf{z}}_1^{(p)}, \tilde{\mathbf{z}}_2^{(p)}, \ldots, \tilde{\mathbf{z}}_N^{(p)}\}$ for each tweet and then use a hinge loss with the margin parameter $\gamma > 0$ as our objective function. Specifically, we set $\gamma = 1$ and write the loss for Task #2 as

$$\ell_{\text{Task 2}} = \frac{1}{N}\sum_{j=1}^{N}\left(\max\left\{0, \gamma - l_j \tilde{\mathbf{z}}_j^{(p)}\right\}\right). \tag{7}$$

When we use Task #1 and Task #2, we say that we are using our **Polarized PEM** model.

4.4 Task #3: independence enforcement

In Task #3, we encourage the semantic component $\mathbf{z}^{(s)}$ to be polarity-neutral, and we thereby force the political-polarity component $\mathbf{z}^{(p)}$ to capture polarity more accurately. We use an adversarial framework to achieve this goal. We alternately train two competing objectives: (1) learn a high-quality embedding $\mathbf{z}$ that preserves both context and polarity; and (2) learn a semantic embedding $\mathbf{z}^{(s)}$ that is not able to infer a tweet's polarity. Let E denote the first objective, which combines Task #1 and Task #2 and controls the quality of our embedding. The loss function ℓ_E of the first objective is

$$\ell_E = \ell_{\text{Task 1}} + \ell_{\text{Task 2}}. \tag{8}$$

Let D denote the second objective, which is a discriminator that attempts to use a semantic embedding for polarity classification. We start training by running the objective E because our discriminator makes sense only if our embedding is meaningful.

We apply the attention mechanism that we used in Task #2 (for aggregate token-level semantic embeddings) to a tweet-level semantic embedding. We use the weighted average $\tilde{\mathbf{z}}^{(s)} = \boldsymbol{\alpha}^T \mathbf{E}_S \in \mathbb{R}^d$ of the semantic dimensions of a tweet's tokens as our tweet-level semantic embedding. The $\mathbf{W}_K$ and $\mathbf{W}_Q$ functions in Task #3 are different than those in Task #2. We use the discriminator D to discern political-party labels from $\tilde{\mathbf{z}}^{(s)}$. The discriminator is a standard two-layer multilayer perceptron (MLP) classifier that infers a class label 0 for liberal-leaning tokens and a class label 1 for conservative-leaning tokens. Between these two layers, we set the number of elements in the output of each hidden layer to $d_{\mathrm{MLP}} = 100$. We use a binary cross-entropy loss ℓ_D. The ground-truth labels of the tweets are $\mathbf{Y} = \{y_1, \dots, y_N\} \in \{0,1\}^N$ and the inferred polarity scores are $\hat{\mathbf{Y}} = \{\hat{y}_1, \dots, \hat{y}_N\}$. The output label of tweet i is

$$\hat{y}_i = D\big(\tilde{z}^{(s)}\big) = \sigma\left(\mathrm{MLP}\big(\tilde{z}^{(s)}\big)\right) \in [0,1], \tag{9}$$

where σ is the sigmoid function. The discriminator loss is the binary cross entropy

$$\ell_D = -\frac{1}{N}\sum_{i=1}^{N}\big(y_i \ln(\hat{y}_i) + (1-y_i)\ln(1-\hat{y}_i)\big). \tag{10}$$

The encoder E seeks to make ℓ_D large enough so that $\mathbf{z}^{(s)}$ tends to ignore political polarity. The discriminator D seeks to make ℓ_D small enough to be a stronger discriminator. To balance these goals, we use an adversarial framework. The training objective for all tasks together is

$$\ell_{\text{Task 3}} = \min_E \max_D \big(\ell(E,D)\big) = \min_E \max_D (\ell_E - \lambda \ell_D). \tag{11}$$

We always train Task #3 together with Tasks #1 and #2. When we train all three tasks together, it is referred as the **Complete PEM** model.

4.5 Joint training

In Algorithm 1, we present our adversarial framework for our **Complete PEM** model. An adversarial framework trains two neural networks together so that they counteract each other [49, 50]. The quantity θ_E denotes all of the parameters in Tasks #1 and #2, including all of the embedding weights $\mathbf{Z}$, the attention weights, and so on. The quantity θ_D, which we use only in Task #3, denotes the set of discriminator parameters. Each batch that we input into our **PEM** model has data from 16 tweets.

We learn all parameters in θ_E and θ_D during training, but we need to determine the hyperparameter λ. In our experiments, we examined $\lambda = 0.01$, $\lambda = 0.1$, $\lambda = 1$, and $\lambda = 10$. Of these values, our **Complete PEM** model performs the best for $\lambda = 0.1$, so we use $\lambda = 0.1$. When applying the **PEM** model to another data set, one should carefully select a suitable value of λ.

Algorithm 1

```
        Complete PEM: Learning algorithm
  procedure LearnEmbedding(Iter)
      Z ← initialize the embeddings
      Initialize the parameter λ > 0
      for i = 1, ..., Iter do
          while not converged do                         ▷ train θ_E, fix θ_D
```

sample from tweets
$\ell_E \leftarrow \ell_{Task\,1} + \ell_{Task\,2}$
$\ell(E,D) \leftarrow \ell_E - \lambda\ell_D$
update θ_E *to minimize* $\ell(E,D)$
end while
while** not converged **do ▷ *train* θ_D, *fix* θ_E
sample from tweets
$\ell_D \leftarrow$ *Discriminator loss*
update θ_D *to minimize* ℓ_D
end while
end for
return Z ▷ *the learned embedding*
end procedure

In each phase (i.e., either training θ_D or training θ_E), we stop training right after we first observe a drop in the F_1 score (which is the harmonic mean of precision and recall) in the validation set. (Such a performance drop can be an indication of overfitting [51].) We then use the parameter values from just before the performance drop and proceed to the next phase.

5 Experiments

5.1 Data sets

We start by collecting a list of Twitter accounts, including 585 accounts of legislators in the 115$^{\text{th}}$ and 116$^{\text{th}}$ Congresses,[1] the accounts of 8 well-known news outlets (see Table 1), and the accounts of President Barack Obama, President Donald Trump, and their Cabinet members at the time (3 March 2019) that we first collected the data. Our data set consists of (1) the most recent 3,200 tweets of each account that we collected on 3 March 2019 and (2) the tweets of these accounts that were posted between 1 January 2020 and 25 November 2020.

We select the news outlets from those with the most voters (i.e., participants who label the political polarity of news outlets on the AllSides Media Bias Ratings (see https://www.allsides.com/media-bias/media-bias-ratings). Previous studies have inferred the political polarities of news outlets from their content [4, 52], and we seek to examine whether or not our model can also reveal political polarities. The available political labels in the AllSides Media Bias Ratings are "liberal", "somewhat liberal", "neutral", "somewhat conservative", and "conservative". We use the three liberal news outlets with the most votes, the three conservative news outlets with the most votes, and the neutral news outlet with the most votes. We checked manually that the polarities of the Twitter accounts of these news outlets are consistent with the labels that we obtained from the AllSides Media Bias Ratings. When a news outlet has multiple Twitter accounts (e.g., `@cnn` and `@cnnpolitics`), we use the account with the most followers in early February 2020. On 10 February 2020, we finished collecting and sorting the media data.

We split the politicians' tweets (of which there are more than 1,000,000 in total) into training, validation, and testing sets in the ratio 8:1:1. We also use the tweets of the news outlets and those of the unobserved accounts as testing sets.

---

[1] See https://www.congress.gov/members.

Table 1 The selected news outlets and their political polarities. The label "L" denotes a liberal-leaning outlet, "C" denotes a conservative-leaning outlet, and "N" denotes a neutral outlet. These labels come from the AllSides Media Bias Ratings (see https://www.allsides.com/media-bias/media-bias-ratings)

Twitter Account	News Outlet	Polarity
@nytimes	*The New York Times*	L
@guardiannews	*Guardian News*	L
@cnn	*CNN*	L
@csmonitor	*The Christian Science Monitor*	N
@amspectator	*The American Spectator*	C
@foxnewsopinion	*Fox News Opinion*	C
@nro	*National Review*	C

We also test our embedding on three existing data sets: the ELECTION2020 data set [53], which has 965,620,919 tweets that were collected hourly between March 2020 and December 2020; a PARLER data set from 6 Jan 2021 that has 1,384,579 posts;[2] and the TIMME data set [13], which includes 2,975 Twitter accounts with location information and self-identified political-polarity labels (either Democratic or Republican). These Twitter accounts are not run by politicians and are never in a training data set. We thus refer to them as "unobserved accounts". We have access to the most recent 3,200 tweets in each Twitter account's timeline; we keep the tweets that they posted in 2020.

5.2 Entity identification

We use the union of the set of entities from three main sources to identify potential entities while training.

To detect nouns, we consider all nouns and proper nouns from parts-of-speech (POS) tagging[3] to be reasonable entities.

To detect phrases that act as nouns, we use AUTOPHRASE (version 1.7) [46] to learn a set of phrases from all politicians' tweets in our data. We then use this set of phrases when tokenizing all employed data sets. AUTOPHRASE assigns a score in the interval $[0, 1]$ to each potential phrase, where a higher score indicates a greater likelihood to be a reasonable phrase. After looking at the results, we manually choose a threshold of 0.8, and we deem all multi-word noun phrases whose scores are at least this threshold to be of sufficiently high quality.

To detect special terms that represent entities that may not yet be part of standard English, we apply TAGME (version 0.1.3) [45] to our training set to include named entities that we are able to link to a Wikipedia page.

5.3 Results

5.3.1 Polarity component

We compute token-level polarity scores by examining the polarity component $\mathbf{z}^{(p)}$ of each embedding. We transform all tokens except mentions into lower-case versions. We do this because Twitter handles (i.e., user names) are case-sensitive, but upper-case and lower-case letters have the same meaning (and thus can be used as alternatives to each other) for other entities (including hashtags).

[2] This data set is available at the repository https://gist.github.com/wfellis/94e5695eb514bd3ad372d6bc56d6c3c8.

[3] See https://www.nltk.org/api/nltk.tag.html.

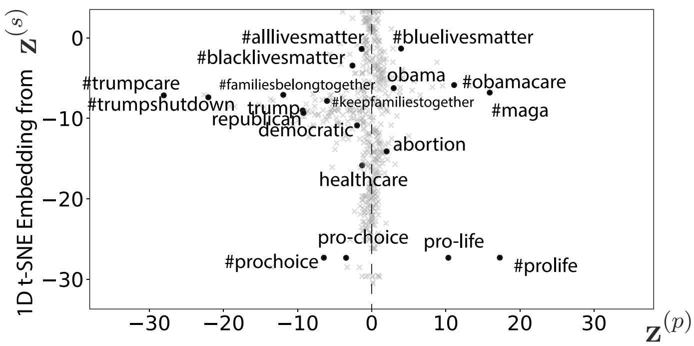


Figure 3 Visualization of the political polarities in our embedding results. The horizontal axis gives the values of the polarity score $\mathbf{z}^{(p)}$, and the vertical axis is a 1D t-SNE value (which we use to facilitate visualization) that we calculate from the semantic embedding $\mathbf{z}^{(s)}$

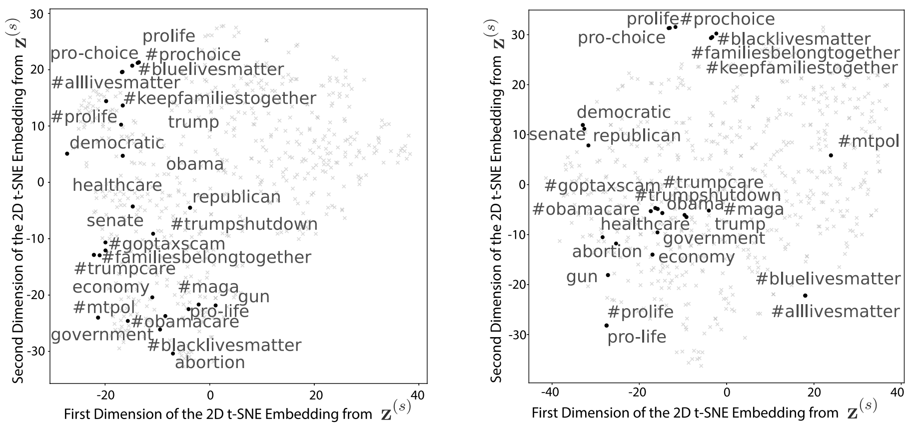
According to our results, of the entities and hashtags that politicians used in our data (which we collected in 2019 and 2020), the ones with the strongest liberal polarities are **#trumpcare**, **#actonclimate**, **#forthepeople**, **#getcovered**, and **#goptaxscam**. The entities and hashtags with the strongest conservative polarities are **#va10**, **#utpol**, **#ia03**, **#tcot**, and **#wa04**.

Our results illustrate that hashtags that refer to electoral districts can be strongly conservative-leaning. Politicians with different political leanings may use hashtags in different ways, and examining a hashtag that is associated with an electoral district is a good way to illustrate this. Additionally, conservative politicians may use a particular non-germane hashtag for certain content more often than liberal politicians. For example, some tweets that used **#va10** contributed to a discussion of a #VA10 forum that was hosted by the Republican party in Fauquier County (`@fauquiergop`).

In Fig. 3, we show our embedding results for the 1,000 most-frequent entities and hashtags and for a few highlighted ones that we select manually. To facilitate visualization, the vertical axis is a 1D t-distributed stochastic neighbor embedding (t-SNE) values [54]. In theory, words with particularly close semantic meanings are near each other along this axis. In our embedding results, hashtags are more likely than other tokens to capture a clear political polarity.

Some of our observations are unsurprising. For example, terms that are related to "**pro-life**" are typically conservative-leaning, whereas terms that are related to "**pro-choice**" are typically liberal-leaning.

Other observations are more nuanced. For example, liberal-leaning Twitter accounts sometimes use text that one is likely to associate more with conservative-leaning views, and vice versa. The embeddings of "**trump**" and "**obama**" give one pair of examples, and the hashtags **#trumpcare** and **#obamacare** give another. Hashtags without semantic context can also appear in tweets. Another interesting observation is that **#blacklivesmatter** and **#alllivesmatter** are both liberal-leaning. In [55], it was pointed out that **#alllivesmatter** was used as a counterprotest hashtag between August 2014 and August 2015. This observation helps illustrate that the polarities of tokens can change with time. Nowadays, **#bluelivesmatter** is used more than **#alllivesmatter** as an antonym of **#blacklivesmatter** in practice (in the sense of having a similar semantic meaning but opposite political



(a) **Complete PEM** semantic components.

(b) **Polarized PEM** semantic components.

Figure 4 Visualization of the semantic components of our (**a**) **Complete** and (**b**) **Polarized PEM** embeddings. We project these components onto a plane by calculating t-SNE values. Both results are reasonable, but the **Polarized PEM** results tend to encourage semantically-related words to be closer to each other. For example, **#familiesbelongtogether** and **#keepfamiliestogether** are used similarly in practice and they are close to each other in the embedding from our **Polarized PEM** model

polarity). Additionally, **#alllivesmatter** now appears commonly in topics such as animal rights.

5.3.2 Semantic components

To demonstrate the quality of the semantic components $\mathbf{z}^{(s)}$, we calculate the cosine similarity of the embedding vectors of the tokens. Our results appear to be reasonable. For example, we observe that the closest token to "**gun**" is "**firearm**" and that the closest token to "**healthcare**" is "**care**". The t-SNE values from our **Polarized PEM** model and **Complete PEM** model also suggest that these semantic components have reasonable quality.

In Fig. 4(a), we plot the results of calculating t-SNE values to project the semantic dimensions of the most-frequent 600 tokens and several manually-selected tokens from our **Complete PEM** embeddings onto a plane. In Fig. 4(b), we show the t-SNE values for our **Polarized PEM** embeddings. These plots illustrate similarities in the semantic meanings of these tokens. For example, we observe that **#AllLivesMatter** and **#BlueLivesMatter** have similar meanings. By comparing Figs. 4(a) and 4(b), it seems that the semantic components of our **Polarized PEM** embeddings may be slightly more reasonable than those of our **Complete PEM** embeddings.

5.3.3 Account-level case studies

We compute a Twitter account's political polarity by calculating the mean of the polarity scores of all of its tweets. Suppose that an account posted N tweets. The i^{th} tweet consists of n tokens, with embeddings $\{\mathbf{z}_1, \dots, \mathbf{z}_n\}$ and polarity scores $\{\mathbf{z}_1^{(p)}, \dots, \mathbf{z}_n^{(p)}\}$. The tweet-level polarity score of this tweet is $b_i = (\sum_{j=1}^{n} \mathbf{z}_j^{(p)})/n$. We estimate the overall polarity score of the account to be $b = (\sum_{i=1}^{N} b_i)/N$. If $b_i < 0$, we regard account i as liberal-leaning; if $b_i > 0$, we regard it as conservative-leaning; if $b_i = 0$, we regard it as neutral. We show our

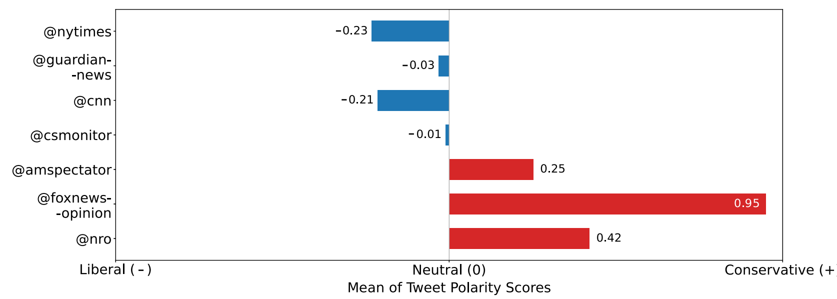


Figure 5 Our estimates of the political polarities of news outlets based on their most recent 3,200 tweets. We collected these tweets starting on 3 March 2019

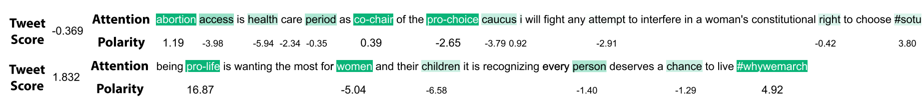


Figure 6 Illustrations of estimating tweet polarities using an attention mechanism. We show the weights from our **Complete PEM** model in green, where darker shades signify greater importance levels. We show the polarity scores underneath the entities and hashtags

results (which seem reasonable) in Fig. 5. We plot liberal-leaning accounts in blue and conservative-leaning accounts in red.

Some previous research [18, 19] on relationships (e.g., following and retweeting relationships) between Twitter accounts has inferred clearer polarities in news outlets than what we obtain using our approach. This suggests that interactions may be more helpful than text itself at identifying the political polarities of Twitter accounts.

5.3.4 Illustrations of estimating tweet polarities with the attention mechanism

See Fig. 6 for examples of our **Complete PEM** model's attention weights and polarity scores. Both the attention weights and the polarity scores appear to be reasonable.

5.3.5 An ablation study of the attention mechanism

We summarize the performance of the three versions of our **PEM** model in Table 2. The left column gives our classification results when we use an attention mechanism. Recall that our **Baseline** model does not use an attention mechanism. In models with an attention mechanism, we use the score that we infer from Task #2, which calculates a weighted average of the tokens' political-polarity component $\{\mathbf{z}^{(p)}\}$. In the right column, we show the accuracy and F_1 scores when we use the mean value of the elements of $\{\mathbf{z}^{(p)}\}$. Recall that we interpret tweets with negative scores as liberal and tweets with positive scores as conservative.

The results in Table 2 suggest that Task #2 alone can successfully capture polarity information, but introducing Task #3 to enhance the independence of the semantic and polarity components can improve inference of the political-polarity component $\mathbf{z}^{(p)}$. However, forcing $\mathbf{z}^{(s)}$ to be polarity-neutral makes it harder to preserve accurate semantic information. (See Figs. 4(a) and 4(b).) This illustrates why our **Complete PEM** model does not always outperform our **Polarized PEM** model.

Table 2 The classification performance on the withheld tweets of politicians and on the Twitter accounts of politicians. The subscript "no attn" signifies that we use the mean value of $\{\mathbf{z}^{(p)}\}$ directly (i.e., without applying an attention mechanism). SKIP-GRAM (i.e., the **Baseline PEM** model) and GLOVE use a pretrained embedding with the same MLP binary classifier as in our discriminator. (To train this classifier, we use a training set that includes 80% of the politicians' tweets.) In each entry, we show the accuracy followed by the F_1 score. We show the best results for each column in bold. The names of our models are also in bold

Model	Tweet-Level Results (accuracy; F_1)	Account-Level Results (accuracy; F_1)
SKIP-GRAM	0.7705; 0.7736	0.8769; 0.8797
GLOVE	0.7438; 0.7453	0.8578; 0.8620
$\text{BERT}_{\text{base}}$	**0.8595**; **0.8603**	**0.9965**; **0.9968**
BERTWEET	0.8399; 0.8435	0.9844; 0.9853
Polarized PEM$_{\text{no attn}}$	0.7681; 0.7682	0.9757; 0.9758
Complete PEM$_{\text{no attn}}$	0.7991; 0.7994	0.9827; 0.9827
Polarized PEM	0.8339; 0.8337	0.9861; 0.9872
Complete PEM	0.8338; 0.8330	0.9931; 0.9936

Table 3 The classification performance on the unobserved accounts. We never include tweets from these accounts in a training data set. In each entry, we show the accuracy followed by the F_1 score. We show the best results for each column in bold. The names of our models are also in bold

Model	Tweet-Level Results (accuracy; F_1)	Account-Level Results (accuracy; F_1)
SKIP-GRAM	0.5822; 0.5636	0.6660; 0.6604
GLOVE	0.5680; 0.5491	0.6486; 0.6372
$\text{BERT}_{\text{base}}$	**0.6541**; 0.6280	0.7234; 0.7218
BERTWEET	0.6284; 0.6486	0.7836; 0.7778
Polarized PEM$_{\text{no attn}}$	0.6066; 0.6244	0.8157; 0.8196
Complete PEM$_{\text{no attn}}$	0.6061; 0.6258	0.8494; 0.8475
Polarized PEM	0.6308; 0.6965	0.8493; 0.8758
Complete PEM	0.6479; **0.6987**	**0.8612**; **0.8870**

5.4 Results on a few downstream tasks

We illustrate that our embeddings are reliable and useful for several downstream tasks.

5.4.1 Classification results

First, we discuss the classification results of our **Polarized** and **Complete PEM** models.

We select 10% of the politicians' tweets (there are 127,143 such tweets) uniformly at random and withhold these tweets as the testing set for Table 2. We select another 10% of the tweets, which we also choose uniformly at random, as a validation set. We use the remaining 80% of the tweets (i.e., 1,017,137 tweets) as our training set. We train all models (see Table 2 and Table 3) on the same training set.

In Table 2, we show the performance of the models on the testing set. We perform the tweet-level classification task on the withheld tweets of the politicians. We never include these tweets in the training set. We perform the account-level classification task on the accounts of all politicians with tweets in the testing set. For a given account, we use its tweets in the testing set to infer its political score by calculating the mean polarity score of all of its tweets.

In Table 3, we show the tweet-level and account-level classification performance levels for the unobserved accounts. (See Sect. 5.1 for a description of these accounts.)

We use the SKIP-GRAM and GLOVE embeddings as baselines. For each of these embeddings (which we do not adjust), we use the same MLP classifier that we use as a discriminator in Task #3 and train the MLP classifiers on our training set until they converge.

We fine-tune the transformer classifiers $\text{BERT}_{\text{base}}$ [32] and BERTWEET [56] (which uses the $\text{BERT}_{\text{base}}$ model configuration and is trained using RoBERTa-style pretraining) on our training set as baselines. We use the uncased (i.e., ignoring capitalization) version of $\text{BERT}_{\text{base}}$; the classifier BERTWEET separates lower-case and upper-case letters. We use the fine-tuned transformers to classify the tweets of politicians (see Table 2) and the tweets of the unobserved accounts (see Table 3).

For the model variants that do not incorporate attention, we calculate each polarity score by computing the mean values of the polarity components $\mathbf{z}^{(p)}$ of the entities and hashtags. We compute the polarities of accounts in the same way as in our examples with news outlets (see Sect. 5.3.3).

By comparing Table 2 and Table 3, we conclude that our models perform better than the transformers ($\text{BERT}_{\text{base}}$ and BERTWEET) on the unobserved accounts. Possible reasons include the following:

1. Our polarity score can take any real value, so it can highlight extremists and exploit extreme tweets that help expose an account's polarity. $\text{BERT}_{\text{base}}$ only allows polarity values between 0 and 1.
2. Models, such as the transformers, with many parameters can suffer from severe overfitting problems, especially when a training set is too small. In Sect. 6, we discuss potential drawbacks of a training data set that includes tweets only by politicians.

5.4.2 *Classification results using only semantic components*

To demonstrate that including Task #3 allows the polarity component $\mathbf{z}^{(p)}$ to capture more political information and makes the semantic components $\mathbf{z}^{(s)}$ more politically neutral, we conduct an experiment in which we use only the semantic components of the tokens for a classification task. Specifically, we examine account-level classification of the politicians' withheld tweets (see Table 4).

In the left column of Table 4, we show our account-level classification results using only $\mathbf{z}^{(s)}$. We obtain these results by training a discriminator with the same architecture as in **Task #3**. We train it on our training set (which has 80% of the politicians' tweets) until the classifier converges on our validation set (which has 10% of politicians' tweets). We then use it to classify tweets in the testing set (which has 10% of politicians' tweets).

Of our classification tasks in Sect. 5.4.1, doing account-level classification based on the politicians' tweets in the testing set is the least challenging one. For more challenging classification tasks, such as the classification of the tweets of the unobserved accounts, the accuracies that we obtain by using SKIP-GRAM (i.e., the **Baseline PEM** model), the**Polarized PEM** model, and the **Complete PEM** model are 0.5701, 0.5809, and 0.5756, respectively. Their accuracies for classifying the unobserved accounts are 0.6450, 0.6624, and 0.6551, respectively. These numerical values suggest that their performance levels are similar on these tasks.

The results in Table 4 suggest that the design of our **Complete PEM** model helps encourage political information to be in the polarity component $\mathbf{z}^{(p)}$, rather than in the semantic components $\mathbf{z}^{(s)}$.

5.4.3 *Polarity distribution of politicians*

We use the same approach as in Sect. 5.4.1 to estimate the polarity scores of the Twitter accounts of politicians. We plot the associated probability densities for both Democrats and Republicans in Fig. 7, and we observe stark polarization.

Table 4 The account-level classification performance on the politicians' withheld tweets in our testing set. We never include these tweets in our training data set, but our training set does include other tweets by the accounts that posted these tweets. In each entry, we show the accuracy followed by the F_1 score. We show the best results for each column in bold. The names of our models are also in bold. The SKIP-GRAM row indicates our **Baseline PEM** results

Model	Results Based on $\mathbf{z}^{(s)}$ (accuracy; F_1)	Results Based on $\mathbf{z}^{(p)}$ (accuracy; F_1)
SKIP-GRAM	0.8394; 0.8451	0.8457; 0.8503
Polarized PEM	**0.8994**; **0.9008**	0.9861; 0.9872
Complete PEM	0.8111; 0.8204	**0.9931**; **0.9936**

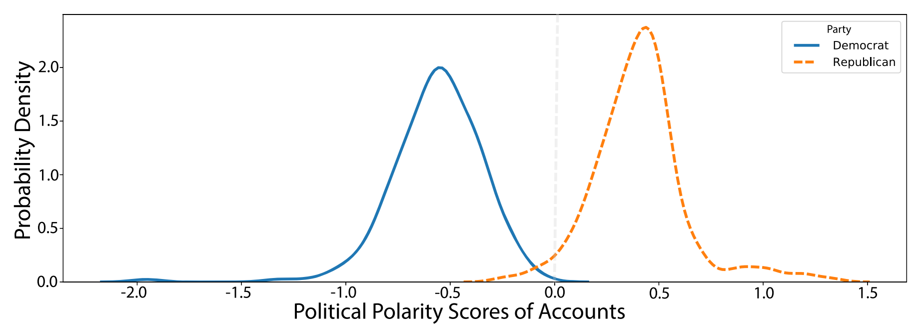


Figure 7 Probability densities of the polarity scores of the Twitter accounts of politicians

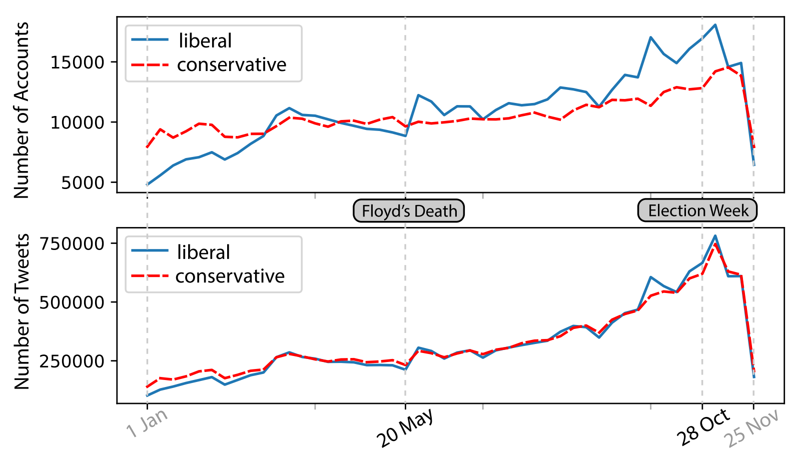


Figure 8 Weekly trends of liberal and conservative tweets on Twitter in 2020. We plot these trends at both (top) the account level and (bottom) the tweet level. The week of George Floyd's murder began on 20 May 2020. The week of the 2020 United States presidential election began on 28 October

5.4.4 *Temporal variation of political polarities*

We now examine temporal changes in the inferred political polarities of the 49,428 Twitter accounts in the TIMME data set [13] that tweeted in 2020. To examine such temporal variation, we chunk the tweets from 2020 of each of these accounts in 7-day intervals starting from 1 January and examine trends over time. (The final interval is cut off and is hence shorter.)

We use the same approach as in Sect. 5.4.1 to infer tweet-level and account-level polarities. As we can see in Fig. 8, our embedding results illustrate plausible trends on Twitter.

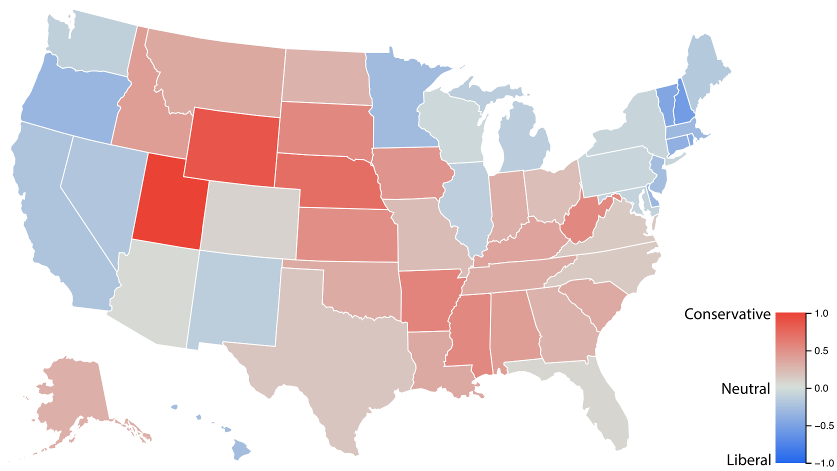


Figure 9 The mean polarity score of the Twitter accounts in each state (and the geographic regions Washington, D.C., Puerto Rico, and Guam) in the United States. We normalize the polarity scores to [−1, 1]

Many liberal-leaning accounts were active starting in the week of the murder of George Floyd. As the week of the U.S. presidential election approached, people were using Twitter more actively, and then discussions of the election seemed to recede after it was over. Based on our results, we also suspect that there may be more liberal-leaning accounts than conservative-leaning accounts on Twitter.

5.4.5 Geographic distribution of political polarities

The TIMME data set [13] has 51,060 accounts with self-reported geographic locations in the United States. Using these locations, we examine the liberal versus conservative tendencies of tweets across the U.S. in 2020. We calculate the polarity of each Twitter account using the mean of the polarities of the tokens in its tweets; we show these account polarities geographically in Fig. 9. We use the mean polarity of all accounts in a state (and the geographic regions Washington, D.C., Puerto Rico, and Guam) to calculate the state's polarity, and we then normalize the states' polarity scores $\mathbf{q} = \{q_1, \dots, q_{53}\}$ to the interval $[-1, 1]$ by calculating $\hat{q}_i = (q_i - \frac{\sum_{j=1}^{53} q_j}{53}) / \max\{|q_1|, \dots, |q_{53}|\}$. After this normalization, -1 is the most liberal score and $+1$ is the most conservative score. Our results are consistent with the tendencies that were reported in national polls for the 2020 U.S. election.[4]

5.4.6 Revealing biases in data sets

We use the embedding results of our **Complete PEM** model to examine biases in data sets. In practice, using these results entails assuming that we can trust the polarities that we learn from the coarse-grained labels of the politicians' parties. Under this assumption, we find that the TIMME data set is politically neutral and that the ELECTION2020 data set [53] is somewhat liberal-leaning. In the ELECTION2020 data set, the mean polarity of the tweets in each week is liberal-leaning. Of the 119 keywords that were provided in Version 1 of this data set, there are 78 liberal-leaning keywords and 41 conservative-leaning keywords. Our embedding also suggests that posts on Parler tend to be more conservative

[4] See https://www.realclearpolitics.com/epolls/2020/president/National.html.

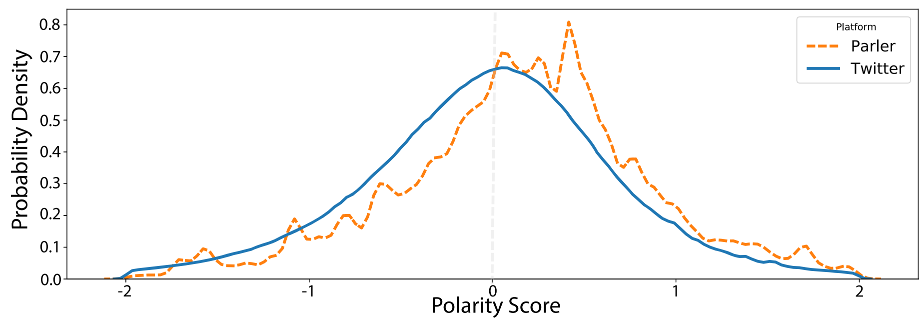


Figure 10 Distributions of polarity scores of Twitter tweets and Parler posts. The Twitter curve is smoother because the Twitter data set is much larger than the Parler data set

than tweets on Twitter. In Fig. 10, we plot the distributions of the polarities of the Twitter tweets and Parler posts. We compute these empirical probability densities using kernel density estimation (KDE) with a Gaussian kernel (i.e., the default setting) in the SEABORN library [57].

5.5 Performance robustness

In Table 2 and Table 3, we reported our best performance levels (from six different random seeds). We also want to examine the robustness of these performance levels. We use the same hyperparameter settings as before, but now we use 5-fold cross validation and different random seeds to initialize the models.

We still train the models on the politicians' tweets. However, instead of randomly using 80% of them as our training set, we now do a 5-fold cross validation. That is, we split the politicians' tweets evenly and uniformly at random into 5 sets that we select uniformly at random, and we withhold one set at a time as our validation and testing sets (with 10% each, with the tweets in them selected uniformly at random). None of the training sets are identical to the one that we used previously.

After training a model on the training set, we evaluate it on the testing data set of politicians. We then use the trained models to infer the polarities of the tweets from the unobserved accounts using the approaches in Table 3.

In Table 5, we report the means and standard deviations from our 5-fold cross validation. The results illustrate that the models' performance levels are robust, although the tweet-level performance levels are more robust than the account-level performance levels.

5.6 Bot analysis

Our investigation does not account for the activity of automated accounts (i.e., bots). We use the verified Twitter accounts of politicians, so we assume that these are not bot accounts. However, bots are widespread on Twitter and other social media [58], We check for potential bots in our Twitter accounts and compare the inferred bot probabilities of these accounts with our inferred political polarities. We find that the probability that an account is a bot has little correlation with its political polarity.

To evaluate the probability that a Twitter account is a bot, we use Botometer (version 4) [59]. It has two options—universal and English—for the language that it employs for bot detection. The universal bot score is evaluated in a language-independent way, but

Table 5 The mean values and standard deviations for 5-fold cross validation of different models, which we initialize with different random seeds. We show the best results for each column in bold. The names of our models are also in bold

Model	Tweet-Level Results (accuracy; F_1)	Account-Level Results (accuracy; F_1)
Politicians' Accounts (Mean Value ± Standard Deviation)		
SKIP-GRAM	0.7700 ± 0.0026; 0.7707 ± 0.0029	0.8833 ± 0.0113; 0.8996 ± 0.0100
GLOVE	0.7231 ± 0.0039; 0.7319 ± 0.0035	0.8575 ± 0.0205; 0.8798 ± 0.0161
$\text{BERT}_{\text{base}}$	**0.8586** ± 0.0006; **0.8587** ± 0.0006	**0.9963** ± 0.0034; **0.9963** ± 0.0034
BERTWEET	0.8337 ± 0.0010; 0.8327 ± 0.0010	0.9828 ± 0.0077; 0.9826 ± 0.0077
Polarized PEM$_{\text{no attn}}$	0.7691 ± 0.0011; 0.7665 ± 0.0011	0.9721 ± 0.0244; 0.9723 ± 0.0243
Complete PEM$_{\text{no attn}}$	0.7955 ± 0.0009; 0.7937 ± 0.0009	0.9805 ± 0.0169; 0.9811 ± 0.0167
Polarized PEM	0.8338 ± 0.0007; 0.8336 ± 0.0007	0.9841 ± 0.0030; 0.9845 ± 0.0030
Complete PEM	0.8332 ± 0.0006; 0.8327 ± 0.0006	0.9915 ± 0.0026; 0.9927 ± 0.0026
Unobserved Accounts (Mean Value ± Standard Deviation)		
SKIP-GRAM	0.5822 ± 0.0007; 0.5635 ± 0.0008	0.6561 ± 0.0053; 0.6324 ± 0.0074
GLOVE	0.5764 ± 0.0009; 0.5574 ± 0.0009	0.6387 ± 0.0073; 0.6222 ± 0.0099
$\text{BERT}_{\text{base}}$	0.6348 ± 0.0007; 0.6231 ± 0.0006	0.7182 ± 0.0078; 0.7149 ± 0.0072
BERTWEET	0.6282 ± 0.0006; 0.6280 ± 0.0005	0.7752 ± 0.0176; 0.7695 ± 0.0173
Polarized PEM$_{\text{no attn}}$	0.6245 ± 0.0011; 0.6067 ± 0.0011	0.8062 ± 0.0191; 0.8105 ± 0.0182
Complete PEM$_{\text{no attn}}$	0.6259 ± 0.0014; 0.6063 ± 0.0015	0.8467 ± 0.0177; 0.8450 ± 0.0178
Polarized PEM	0.6284 ± 0.0023; 0.6865 ± 0.0020	0.8463 ± 0.0063; 0.8666 ± 0.0059
Complete PEM	**0.6472** ± 0.0030; **0.6907** ± 0.0028	**0.8550** ± 0.0075; **0.8814** ± 0.0072

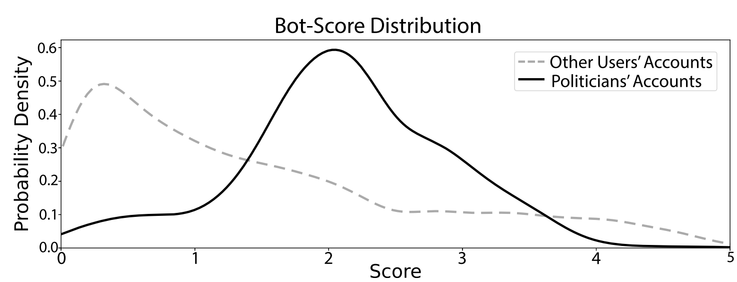


Figure 11 Probability densities of the bot scores of the Twitter accounts of politicians (solid curve) and all other Twitter accounts (dashed curve)

the English bot score is more accurate for accounts that tweet primarily in English, so we use the English option.

There are many different types of Twitter bots (see https://botometer.osome.iu.edu/faq). For simplicity, we use only an overall bot score from Botomer. The score of a bot varies between 0 and 5, with larger scores signifying that an account is more likely to be a bot. In Fig. 11, we show the probability densities of bot scores for politicians and ordinary Twitter accounts.

In Fig. 12, we plot the distributions of the overall bot scores versus the absolute values of polarity scores (i.e., $|\{\mathbf{z}^{(p)}\}|$) for both politicians' Twitter accounts and ordinary Twitter accounts. The absolute values of the polarity scores indicate the extremeness of an account's content according to our PEM model.

5.7 Impact of assigning polarity scores to other tokens

We use tokens other than hashtags and entities in our **PEM** model, but we have not assigned political polarities to them. We feel that this design decision improves the inter-

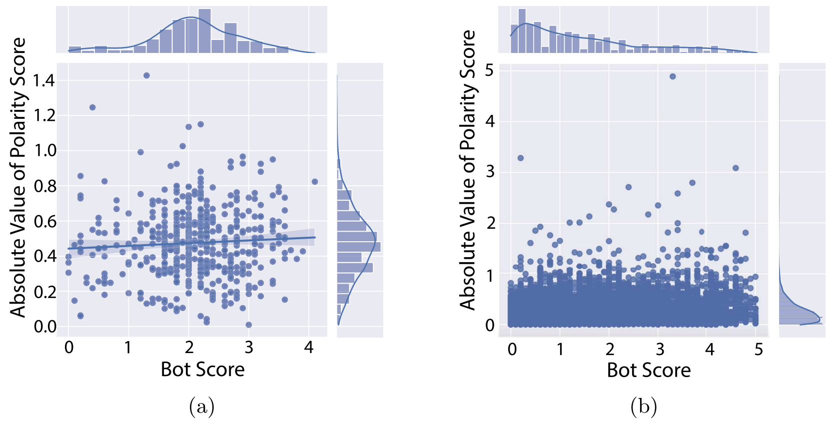


Figure 12 Distribution of the overall bot score versus the absolute values of the polarity scores of the content of (**a**) politicians' Twitter accounts and (**b**) all other Twitter accounts

Table 6 The tweet-level classification performance on the politicians' withheld tweets in our testing set when we assign polarity scores to all tokens versus only assigning polarity scores to hashtags and entities. In each entry, we show the accuracy followed by the F_1 score. We show the best results for each column in bold

Results (accuracy; F_1)	Polarized PEM	Complete PEM
Using $\mathbf{z}^{(p)}$ of All Tokens	**0.8369**; **0.8366**	0.8337; **0.8334**
Using $\mathbf{z}^{(p)}$ of Only Entities and Hashtags	0.8339; 0.8337	**0.8338**; 0.8330

pretability of our model. For some words, such as **"a"** or **"the"**, it definitely does not make sense to assign a political polarity.

As one can see in Table 6, assigning political polarities to tokens other than named entities and hashtags does not seem to harm our classification performance. We show it by comparing the tweet-level classification results of our **Complete PEM** model on the withheld testing set of the politicians' tweets (i.e., the same testing set that we used in Sect. 5.4.1).

6 Limitations

We highlight several important limitations of our work. Naturally, our discussion is not exhaustive, and it is also relevant to think about other limitations.

6.1 Incomplete data

We consider only textual information. Therefore, we overlook images, videos, and other types of information.

6.2 Model limitations

We designed our **PEM** model to infer political polarity scores from entities and hashtags, so it is not helpful for inferring the polarity of tweets that have no entities or hashtags. Additionally, our **PEM** model does not take time stamps into account, so it does not consider the dynamic nature of polarities.

6.3 Training-set biases and other issues

Our design decision of assigning political polarities to items in a training set enables one to automatically assign labels at scale. However, it can be undesirable to make such assignments a priori.

We use the tweets of politicians because their accounts are verified and they have a consistent, unambiguous, and self-identified political affiliation. However, this choice introduces biases and other potential issues. First, the size of our training data set is necessarily limited, and it is easier for models to overfit data when using small data sets than when using large ones. Second, our results may be sensitive to the time window in which we collected tweets. For example, polarization in tweets may be more apparent during elections than at other times. Third, politicians are not necessarily representative of other social-media users. Fourth, we did not train our model to handle bot or cyborg accounts. We used verified Twitter accounts in our training data set, so it presumably does not have any bots or cyborgs. (Our estimation of bot probabilities supports this presumption.) Bot accounts are very common on Twitter [58], so it is necessary to be cautious when applying our model directly to typical Twitter data sets.

The verified Twitter accounts of politicians are very different in nature from the Twitter accounts of other users. We saw ramifications of such differences in our classification results. Using $\text{BERT}_{\text{base}}$ to classify tweets from politicians versus those of other accounts yields an accuracy of 0.7590 and an F_1 score of 0.7595 on the testing set. If we partition the set of non-politician accounts into two groups that each have the tweets of 1,293 accounts (which we assign uniformly at random) and try to classify the group of each tweet, we obtain an accuracy of 0.4600 and an F_1 score of 0.6276.

6.4 Quantifying political polarity

There are many possible ways to quantify political polarity. We chose to assign labels of "liberal" and "conservative", but other dichotomies are also relevant. Moreover, we designed our **PEM** model learn a single type of polarity. It cannot simultaneously reveal multiple types of political polarities.

6.5 Sarcasm and irony

In our work, we did not analyze nuanced situations, such as sarcasm and irony, that depend heavily on context. Sarcasm plays an important role in social media [10], and it is worth generalizing our **PEM** model to be able to handle it successfully in the future.

7 Conclusions

We studied the problem of inferring political polarities in embeddings of entities and hashtags. To capture political-polarity information without using auxiliary word pairs, we proposed **PEM**, a multi-task learning model that employs an adversarial framework.

Our experiments illustrated the effectiveness of our **PEM** model and the usefulness of the embeddings that one can produce from it. In principle, it is possible to extend our approach to extract any type of polarity of an embedding (while attempting to minimize the effects of polarity on other components). One can also extend our **PEM** model to deploy it with a variety of embedding strategies.

8 Ethics statement

There are several ethical points to consider in our work.

First, one needs to consider our data sets. The data that we used comes from publicly available sources, and our training data comes from the verified accounts of politicians. We do not store any sensitive information (such as real-time locations) from Twitter. It is important to be aware of Twitter's privacy policy (see https://twitter.com/en/privacy) when downloading and using data from Twitter.

There are also important ethical considerations when using the results of embeddings like ours. Our **PEM** model yields interesting and occasionally counterintuitive results. One must be cautious when using such results for subsequent tasks (e.g., when drawing conclusions about an individual's political views). Additionally, models inherit biases from training data sets, and they can exacerbate such biases [34].

The conclusions that we obtained from applying our **PEM** model are based on the existing posts of social-media accounts. One must be cautious when subsequently inferring what such accounts may post in the future and especially if one seeks to use any insights from our model to inform behavior, actions, or policy.

Acknowledgements
We thank Yupeng Gu and Zhicheng Ren for helpful discussions.

Funding
This research was supported by the National Science Foundation (through grants III-1705169, NSF 1937599, NSF 2119643, and 1922952), an Okawa Foundation Grant, Amazon Research Awards, Cisco research grant USA000EP280889, Picsart Gifts, and Snapchat Gifts.

Abbreviations
PEM, Polarity-aware Embedding Multi-task learning; t-SNE, t-distributed stochastic neighbor embedding; BERT, Bidirectional Encoder Representations from Transformers; GloVe, Global Vectors for Word Representation; TIMME, Twitter Ideology-detection via Multi-task Multi-relational Embedding; NCE, noise-contrastive estimation; KDE, kernel density estimation.

Availability of data and materials
Our code, the data sets of the politicians, and the embedding results of our models are available at https://bitbucket.org/PatriciaXiao/pem/src/master/.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare no competing interests.

Author contributions
ZX, PZ, MAP, and YS conceived and conceptualized the study. ZX, JZ, YW, and WHL performed the analysis and wrote the initial draft of the paper. ZX, MAP, and YS reviewed and extensively edited the manuscript, determined what additional analysis was necessary, and produced the final version of the manuscript. All authors read and approved the final manuscript.

Author details
[1]Department of Computer Science, University of California, Los Angeles, 580 Portola Plaza, 90095, Los Angeles, California, United States of America. [2]Information Sciences Institute, University of Southern California, Marina del Rey, 90292, Los Angeles, California, United States of America. [3]Department of Mathematics, University California, Los Angeles, 520 Portola Plaza, 90095, Los Angeles, California, United States of America. [4]Santa Fe Institute, 1399 Hyde Park Road, 87501, Santa Fe, New Mexico, United States of America.

Received: 19 September 2022 Accepted: 31 March 2023

References

1. Levendusky M (2009) The partisan sort: How liberals became Democrats and conservatives became Republicans. University of Chicago Press, Chicago
2. Webster SW, Abramowitz AI (2017) The ideological foundations of affective polarization in the US electorate. Am Polit Res 45(4):621–647
3. Schober MF, Pasek J, Guggenheim L, Lampe C, Conrad FG (2016) Social media analyses for social measurement. Public Opin Q 80(1):180–211
4. Chao Z, Molitor D, Needell D, Porter MA (2022) Inference of media bias and content quality using natural-language processing. ArXiv preprint. arXiv:2212.00237
5. Boche A, Lewis JB, Rudkin A, Sonnet L (2018) The new Voteview.com: Preserving and continuing Keith Poole's infrastructure for scholars, students and observers of Congress. Public Choice 176(1–2):17–32
6. Gentzkow M, Shapiro JM (2010) What drives media slant? Evidence from US daily newspapers. Econometrica 78(1):35–71
7. Rye BJ, Underhill A (2020) Pro-choice and pro-life are not enough: an investigation of abortion attitudes as a function of abortion prototypes. Sexual Cult 24:1829–1851
8. Zhao J, Zhou Y, Li Z, Wang W, Chang K-W (2018) Learning gender-neutral word embeddings. In: Proceedings of the 2018 conference on empirical methods in natural language processing, pp 4847–4853
9. Bose AJ, Hamilton WL (2019) Compositional fairness constraints for graph embeddings. In: Proceedings of the 36th international conference on machine learning. PMLR, vol 97
10. Tayal DK, Yadav S, Gupta K, Rajput B, Kumari K (2014) Polarity detection of sarcastic political tweets. In: 2014 international conference on computing for sustainable global development (INDIACom). Institute of Electrical and Electronics Engineering, New Delhi, pp 625–628
11. Pla F, Hurtado L-F (2014) Political tendency identification in Twitter using sentiment analysis techniques. In: Proceedings of COLING 2014, the 25th international conference on computational linguistics: technical papers, pp 183–192
12. Lieberman R, Mettler S, Pepinsky TB, Roberts KM, Valelly R (2017) Trumpism and American democracy: History, comparison, and the predicament of liberal democracy in the United States. Perspective Polit 17(2):470–479
13. Xiao Z, Song W, Xu H, Ren Z, Sun Y (2020) TIMME: Twitter ideology-detection via multi-task multi-relational embedding. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining. KDD '20. Association for Computing Machinery, New York, pp 2258–2268
14. Pierce RJ Jr (1988) Two problems in administrative law: Political polarity on the district of Columbia circuit and judicial deterrence of agency rulemaking. Duke Law J 37:300–328
15. Maynard D, Funk A (2011) Automatic detection of political opinions in Tweets. In: García-Castro R, Fensel D, Antoniou G (eds) Extended semantic web conference, pp 88–99
16. Barberá P (2015) How social media reduces mass political polarization. Evidence from Germany, Spain, and the US. Available at http://pablobarbera.com/static/barbera_polarization_APSA.pdf
17. Bail CA, Argyle LP, Brown TW, Bumpus JP, Chen H, Hunzaker MF, Lee J, Mann M, Merhout F, Volfovsky A (2018) Exposure to opposing views on social media can increase political polarization. Proc Natl Acad Sci USA 115(37):9216–9221
18. Gu Y, Chen T, Sun Y, Wang B (2016) Ideology detection for Twitter users with heterogeneous types of links. arXiv:1612.08207
19. Tien JH, Eisenberg MC, Cherng ST, Porter MA (2020) Online reactions to the 2017 'unite the right' rally in charlottesville: Measuring polarization in Twitter networks using media followership. Appl Netw Sci 5(1):10
20. Iyyer M, Enns P, Boyd-Graber J, Resnik P (2014) Political ideology detection using recursive neural networks. In: Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 1: long papers), pp 1113–1122
21. Lai M, Tambuscio M, Patti V, Ruffo G, Rosso P (2019) Stance polarity in political debates: A diachronic perspective of network homophily and conversations on Twitter. Data Knowl Eng 124:101738
22. Gordon J, Babaeianjelodar M, Matthews J (2020) Studying political bias via word embeddings. In: Companion proceedings of the web conference 2020, pp 760–764
23. Vergeer M (2015) Twitter and political campaigning. Sociol Compass 9(9):745–760
24. Jungherr A (2016) Twitter use in election campaigns: A systematic literature review. J Inf Technol Polit 13(1):72–91
25. Powell M, Kim AD, Smaldino PE (2022) Hashtags as signals of political identity: #BlackLivesMatter and #AllLivesMatter. Available at https://osf.io/preprints/socarxiv/tqs2x/
26. Bengio Y, Ducharme R, Vincent P, Jauvin C (2003) A neural probabilistic language model. J Mach Learn Res 3:1137–1155
27. Levy O, Goldberg Y (2014) Neural word embedding as implicit matrix factorization. In: Advances in neural information processing systems, pp 2177–2185
28. Li Y, Xu L, Tian F, Jiang L, Zhong X, Chen E (2015) Word embedding revisited: A new representation learning and explicit matrix factorization perspective. In: IJCAI'15: proceedings of the 24th international conference on artificial intelligence, pp 3650–3656
29. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems, pp 3111–3119
30. Pennington J, Socher R, Manning CD (2014) GloVe: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1532–1543
31. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: Advances in neural information processing systems, pp 5998–6008
32. Devlin J, Chang M-W, Lee K, Toutanova K (2019) 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). Association for Computational Linguistics, Minneapolis, pp 4171–4186. https://www.aclweb.org/anthology/N19-1423
33. Mehrabi N, Morstatter F, Saxena N, Lerman K, Galstyan A (2019) A survey on bias and fairness in machine learning. arXiv:1908.09635

34. O'Neil C (2016) Weapons of math destruction: how big data increases inequality and threatens democracy. Broadway Books, New York
35. Zhao J, Wang T, Yatskar M, Ordonez V, Chang K-W (2018) Gender bias in coreference resolution: Evaluation and debiasing methods. In: Association for computational linguistics: human language technologies, vol 2, pp 15–20
36. Medhat W, Hassan A, Korashy H (2014) Sentiment analysis algorithms and applications: A survey. Ain Shams Eng J 5(4):1093–1113
37. Astya P et al (2017) Sentiment analysis: Approaches and open issues. In: 2017 international conference on computing, communication and automation (ICCCA). Institute of Electrical and Electronics Engineers, Greater Noida, pp 154–158
38. Yu L-C, Wang J, Lai KR, Zhang X (2017) Refining word embeddings for sentiment analysis. In: Proceedings of the 2017 conference on empirical methods in natural language processing, pp 534–539
39. Fu P, Lin Z, Yuan F, Wang W, Meng D (2018) Learning sentiment-specific word embedding via global sentiment representation. In: Proceedings of the thirty-second AAAI conference on artificial intelligence, vol 32
40. Batra S, Rao D Entity based sentiment analysis on Twitter (2010). Class report, vol 224. Available at https://nlp.stanford.edu/courses/cs224n/2010/reports/drao-sidbatra.pdf
41. Song Y, Jeong S, Kim H (2017) Semi-automatic ccnstruction of a named entity dictionary for entity-based sentiment analysis in social media. Multimed Tools Appl 76(9):11319–11329
42. Tang D, Wei F, Yang N, Zhou M, Liu T, Qin B (2014) Learning sentiment-specific word embedding for Twitter sentiment classification. In: Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 1: long papers), pp 1555–1565
43. Nadeau D, Sekine S (2007) A survey of named entity recognition and classification. Lingvist Investigat 30(1):3–26
44. Li J, Sun A, Han J, Li C (2018) A survey on deep learning for named entity recognition. In: Proceedings of the 27th international conference on computational linguistics, pp 2145–2158
45. Ferragina P, Scaiella U (2010) TagMe: On-the-fly annotation of short text fragments (by Wikipedia entities). In: Proceedings of the 19th ACM international conference on information and knowledge management, pp 1625–1628
46. Shang J, Liu J, Jiang M, Ren X, Voss CR, Han J (2018) Automated phrase mining from massive text corpora. IEEE Trans Knowl Data Eng 30(10):1825–1837
47. Gutmann M, Hyvärinen A (2010) Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics. JMLR workshop and conference proceedings, pp 297–304
48. Hu D (2019) An introductory survey on attention mechanisms in NLP problems. In: Proceedings of SAI intelligent systems conference. Springer, Heidelberg, pp 432–448
49. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: Advances in neural information processing systems, pp 2672–2680
50. Chen X, Duan Y, Houthooft R, Schulman J, Sutskever I, Abbeel P (2016) InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In: Advances in neural information processing systems, pp 2172–2180
51. Cawley GC, Talbot NLC (2010) On over-fitting in model selection and subsequent selection bias in performance evaluation. J Mach Learn Res 11:2079–2107
52. Gruppi M, Smeros P, Adalı S, Castillo C, Aberer K (2022) Scilander: mapping the scientific news landscape. ArXiv preprint. arXiv:2205.07970
53. Chen E, Deb A, Ferrara E (2021) #Election2020: the first public Twitter dataset on the 2020 US presidential election. J Comput Soc Sci. Available at https://doi.org/10.1007/s42001-021-00117-9.
54. van der Maaten L, Hinton G (2008) Visualizing data using t-SNE. J Mach Learn Res 9:2579–2605
55. Gallagher RJ, Reagan AJ, Danforth CM, Dodds PS (2018) Divergent discourse between protests and counter-protests: #BlackLivesMatter and #alllivesmatter. PLoS ONE 13(4):0195644
56. Nguyen DQ, Vu T, Nguyen A-T (2020) BERTweet: a pre-trained language model for English tweets. In: Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations, pp 9–14
57. Waskom ML (2021) Seaborn: Statistical data visualization. J Open Sour Softw 6(60):3021
58. Ferrara E, Varol O, Davis C, Menczer F, Flammini A (2016) The rise of social bots. Commun ACM 59(7):96–104
59. Sayyadiharikandeh M, Varol O, Yang K-C, Flammini A, Menczer F (2020) Detection of novel social bots by ensembles of specialized classifiers. In: Proceedings of the 29th ACM international conference on information & knowledge mMnagement, pp 2725–2732

Publisher's Note