

Semantic and Context-aware Linguistic Model for Bias Detection

Sicong Kuang Brian D. Davison

Lehigh University, Bethlehem PA

sik211@lehigh.edu, davison@cse.lehigh.edu

Abstract

Prior work on bias detection has predominantly relied on pre-compiled word lists. However, the effectiveness of pre-compiled word lists is challenged when the detection of bias not only depends on the word itself but also depends on the context in which the word resides. In this work, we train neural language models to generate vector space representation to capture the semantic and contextual information of the words as features in bias detection. We also use word vector representations produced by the GloVe algorithm as semantic features. We feed the semantic and contextual features to train a linguistic model for bias detection. We evaluate the linguistic model’s performance on a Wikipedia-derived bias detection dataset and on a focused set of ambiguous terms. Our results show a relative F1 score improvement of up to 26.5% versus an existing approach, and a relative F1 score improvement of up to 14.7% on ambiguous terms.

1 Introduction

Bias in reference works affects people’s thoughts [Noam, 2008]. It is the editor’s job to correct those biased points of view and keep the reference work as neutral as possible. But when the bias is subtle or appears in a large corpus, it is worth building computational models for automatic detection. Most prior work on bias detection rely on pre-compiled word lists [Recasens *et al.*, 2013; Iyyer *et al.*, 2014; Yano *et al.*, 2010]. This approach is good at detecting simple biases that depend merely on the word. Such methods are appropriate when the word itself indicates strong subjectivity polarity or the author’s stance intuitively and straightforwardly. In Examples 1a and 2a shown below¹, both “terribly” and “disastrous” are subjective words indicating the author’s negative emotion; the word “terrorist” in Example 3a clearly identifies the author’s stance on the event. Use of a pre-compiled word list is sufficient to detect such words.

1. (a) The series started **terribly** for the Red Sox.
(b) The series started very **poorly** for the Red Sox.

¹All examples in this work are extracted from the dataset derived from Wikipedia 2013 [Recasens *et al.*, 2013].

2. (a) Several notable allegations of lip-synching have been recently targeted at her due to her **disastrous** performances on Saturday Night Live.
(b) Several notable allegations of lip-synching have been recently targeted at her due to her **poor** performances on Saturday Night Live.
3. (a) **Terrorists** threw hand grenades and opened fire on a crowd at a wedding in the farming community of Patish, in the Negev.
(b) **Gunmen** threw hand grenades and opened fire on a crowd at a wedding in the farming community of Patish, in the Negev.

However, using a pre-compiled word list also has significant drawbacks. It is inflexible in the sense that only words appearing in the list can be detected. Words with similar meanings but not collected in the list would not be detected. Thus this method only focuses on the surface form of the word while neglecting its semantic meaning. Focusing on the word itself also means neglecting the context in which the word resides. But some bias can only be detected when contextual information is considered. Words associated with this kind of bias, such as “white” in Example 4a, are often ambiguous and hard to detect using only a pre-compiled word list. The meaning of such words can only be clarified by interpreting the context of the word. The modified sentence in each example is the correct version supplied by Wikipedia editors.

4. (a) By bidding up the price of housing, many **white** neighborhoods again effectively shut out blacks, because blacks are unwilling, or unable, to pay the premium to buy entry into white neighborhoods.
(b) By bidding up the price of housing, many **more expensive** neighborhoods again effectively shut out blacks, because blacks are unwilling, or unable, to pay the premium to buy entry into white neighborhoods.

Recent years have seen progress in learning vector space representations for both words and variable-length paragraphs [Pennington *et al.*, 2014; Mikolov *et al.*, 2013b; Le and Mikolov, 2014a; Mikolov *et al.*, 2013a]. In this work, we use and build models to generate semantic and contextual vector space representations. Equipped with semantic and contextual information, we then build a semantic and context-aware linguistic model for bias detection.

2 Background

Current research in bias detection often uses both pre-compiled word lists and machine learning algorithms [Recasens *et al.*, 2013; Iyyer *et al.*, 2014; Yano *et al.*, 2010]. Most define the bias detection problem as a binary classification problem. Gentzkow and Shapiro [2010] select 1,000 phrases based on the frequency that these phrases appear in the text of the 2005 *Congressional Record*. They form a political word list that can separate Republican representatives from Democratic representatives as the initial step in detecting the political leaning of the media. Greenstein and Zhu [2012] applied Gentzkow and Shapiro’s method to Wikipedia articles to estimate Wikipedia’s political bias. Their result shows many Wikipedia articles contain political bias and the polarity of the bias evolves over time.

Sentiment analysis in bias detection is often used to detect a negative tone or a positive tone of a sentence or a document which should have been neutral [Kahn *et al.*, 2007; Saif *et al.*, 2012]. This kind of bias in reference works is easier to detect due to the emotional identifier it uses, usually an adjective. Recasens *et al.* [2013] use a pre-compiled word list from Liu *et al.* [2005] to detect non-neutral tone in reference works. Yano *et al.* [2010] evaluated the feasibility of automatically detecting such biases using Pennebaker *et al.*’s LIWC dictionary [2015] compared to human judgments using Amazon Mechanical Turk in the politics domain.

We learn word and document vector representations from two neural language models [Le and Mikolov, 2014b] and GloVe algorithms [Pennington *et al.*, 2014]. The word vectors and document vectors are used as semantic and contextual features to build a linguistic model. Below we introduce the models and algorithm we use to learn the features.

Neural Language Model

Neural language models are trained using neural networks to obtain vector space representations [Bengio *et al.*, 2006]. Although the vector space representations of the words in a neural language model are initialized randomly, they will eventually learn the semantic meaning of the words through the prediction task of the next word in a sentence. [Mikolov *et al.*, 2013b; Le and Mikolov, 2014b]. Using the same idea, we treat every document also as a unique vector. And the document vector will eventually learn the semantics through the same prediction task as we do for word vector.

We use stochastic gradient descent optimization algorithm via backpropagation algorithm to train document vector representations and word vector representations. The model that considers the document vector as the topic of the document or the contextual information when predicting the next word, is called the Distributed Memory Model (dm). Since in the process of building a dm model, word vectors in the corpus will capture the semantic meanings; in our work, besides using the dm model to learn document vectors as contextual features, we also use the dm model to learn word vectors as semantic features. The Distributed Bag of Words model (dbow) only learns document vector representations and it is trained by predicting words randomly sampled from the document [Le and Mikolov, 2014b]. In this work, we also use dbow model to learn document vectors as contextual features.

GloVe Algorithm

In both the dm and dbow models, text is trained from a local context window. By utilizing global word-word co-occurrence counts, the ratio of co-occurrence probabilities are able to capture the relevance between words. Pennington *et al.* [2014] use this idea to construct a word-word co-occurrence matrix, and reduce the dimensionality by factorization. The resulting matrix contains vector space representations for each word. In this work, we use GloVe’s pre-trained word vectors learned from Wikipedia in 2014² as semantic features to train a linguistic model.

3 Approach

Our work extends the work of Recasens *et al.* [2013], who use eight pre-compiled word lists to generate boolean features to train a logistic regression model to detect biased words. In Recasens’s work, 32 manually crafted features for each word being considered are utilized to build a logistic regression model. Among the features, about two thirds of their features (20/32) are boolean features derived from the pre-compiled word lists. Other features include the word itself, lemma, part of speech (POS) and grammatical relation.

By using pre-compiled word lists, their method neglects semantic and contextual information. Moreover, in their evaluation, they evaluate their model’s performance as the ratio of sentences with the correctly predicted biased word. This metric has two flaws: first using a word-feature matrix as input, the linguistic model is a word-based classification model and thus word-based evaluation metrics are needed; second, to calculate the sentence-based metric, the authors obtain the predicted probabilities for all words in the sentence—the word with the highest probability is predicted as the biased word. The authors’ implicit assumption is that there must exist a biased word in every sentence, which is not the case in real-world text. Since the dataset is derived from Wikipedia, non-biased words form the majority class and so accuracy is not an effective metric. In contrast, we focus on the model’s quality on detection of biased words. To address the above problems, we use word-based evaluation metrics—precision, recall and F1 score—to evaluate performance.

In this work, we train two neural language models using stochastic gradient descent and backpropagation, a distributed memory model and a distributed bag of words model, to learn vector space representations to capture the contextual information of each word under consideration. Our assumption is that equipped with contextual information the linguistic model should be better able to detect bias associated with ambiguous words. To tackle the problem that the pre-compiled word list method only focuses on remembering the form of the words in the list, we use recent approaches from Pennington *et al.* [2014] and Mikolov *et al.* [2013a; 2014b] to obtain vector space representations that can capture the fine-grained semantic regularities of the word. We incorporate the semantic features and contextual features when building a logistic regression model for the bias detection task.

²<http://nlp.stanford.edu/projects/glove/>

4 Experiment and Analysis

Since our task comes from Recasens et al. [2013], we aim to build a linguistic model to detect framing bias and epistemological bias. Recasens et al. used multiple boolean features derived from pre-compiled word lists (true if in the list, false otherwise) to describe the target word. Our first expectation is that by using the finer structure of the word vector space using methods by Pennington et al. [2014] and Mikolov et al. [2013a], the finer-grained semantic regularities should become more visible and thus get better bias detection performance because similar words will be classified similarly. Second, by generating document vector space representations to capture the context of each word, we should improve the model’s performance on bias detection associated with ambiguous words, since we can potentially distinguish different uses of the same word.

We use Recasens et al.’s approach as baseline. To better understand the behavior of the semantic features and the contextual features, we design our experiments to be in three scenarios: first we retain all the features in Recasens et al.’s work and only add our semantic features to train a logistic regression model; second we retain all the features in Recasens et al.’s work and add our contextual features to train a logistic regression model; third we add both the semantic and contextual features. In their work, Recasens et al.’s feature space consists (in part) of lexical features (word and POS) and syntactic features (grammatical relationships). A list of all 32 features may be found in Recasens et al. [2013].

To better measure the contextual feature’s behavior in detecting bias associated with ambiguous words, we extract a focused subset of the test cases consisting of ambiguous words (i.e., those in the training set that are inconsistently labeled as biased). We measure the precision, recall and F1 score of the focused set before and after we add the contextual features. The logistic regression model computes each word’s probability to be biased. We derive a threshold probability to decide beyond which the words should be predicted as biased by choosing the threshold when the F1 score is maximized on the training set, examining thresholds across (0, 1) using intervals of 0.001.

4.1 Dataset

Wikipedia endeavors to enforce a neutral point of view (NPOV) policy³. Any violation of this policy in the Wikipedia content will be corrected by Wikipedia editors. As a free online reference, Wikipedia publishes its data dumps once per month (English version Wikipedia). By doing a *diff* operation on the same Wikipedia articles from two different Wikipedia dumps, we are able to extract the “before form” string (the sentence with a single biased word from the old Wikipedia article) and the “after form” string (the same sentence with the biased word corrected by the Wikipedia editors) [2013]. With such a labeled data set from Wikipedia, we are ready to build a linguistic model to automatically detect biased words in a reference work.

We use the raw dataset from Recasens et al. [2013] derived from articles from Wikipedia in 2013. The biased words are

Data	Number of sentences	Number of words
Train	1779	28638
Test	207	3249
Focused set	NA	706

Table 1: Statistics of the dataset

	baseline	dm doc vec	dbow doc vec	dm doc vec + dbow doc vec
# features	32	332	332	632
precision	0.245	0.228	0.228	0.224
recall	0.228	0.335	0.335	0.330
F1 score	0.236	0.271	0.271	0.267

Table 2: Results on test set after adding contextual features

labeled by Wikipedia editors. However, since some details of their data preparation are not included in their paper, our statistics of the dataset after processing and cleaning (shown in Table 1) are slightly different from theirs.

4.2 Baseline

For our baseline, we built a logistic regression model using the approach of Recasens et al. [2013]. To better prepare the data, we also added the following steps in data cleaning which are not specified in their paper: we discard data tuples in both training set and test set if the “before form” string and “after form” string only differ by numbers or contents inside $\langle \rangle$ and $\{ \}$, since contents inside $\langle \rangle$ and $\{ \}$ are not text in Wikipedia and we also ignore the words within $\langle \rangle$ and $\{ \}$ when we generate the word-feature matrix. We also remove tuples from the dataset in which the biased word belongs to the stopwords set. Moreover, we use regex to check and remove those tuples if the biased word of that tuple happens to be in the Wikipedia article’s title. We use the Stanford CoreNLP (version 3.4.1) [Marneffe *et al.*, 2006] to generate grammatical features, such as part of speech, lemma and grammatical relationships. The result of the baseline is shown in the first column of Table 2.

4.3 Experiment on Contextual Features

For each word in the data set, we generate fixed length vector representations of the Wikipedia articles in which the word resides as the contextual features by training two neural language models. This fixed length document vector of the article, together with the original 32 features from Recasens et al.’s paper [Recasens *et al.*, 2013] will be the input to train a logistic regression model to perform bias detection.

To generate the contextual features for each word in the dataset, we use all 7,464 Wikipedia articles and altogether 1.76 million words as input to train two neural language models, a distributed memory model (dm) and a distributed bag of words model (dbow), using the open source package gensim on a 128GB memory machine with 16 3.3 Ghz cores. The training process took approximately 5 hours using 16 workers (cores). For each model, we iterate over 10 epochs. For each Wikipedia article, we split and clean it using the same procedures as we process the “before form” strings [Recasens *et al.*, 2013]. For each article, we use the Wikipedia article name as the label to train the neural language model. For both models, we use a window size of 10 and vector dimension of 300

³https://en.wikipedia.org/wiki/Wikipedia:Neutral_point_of_view

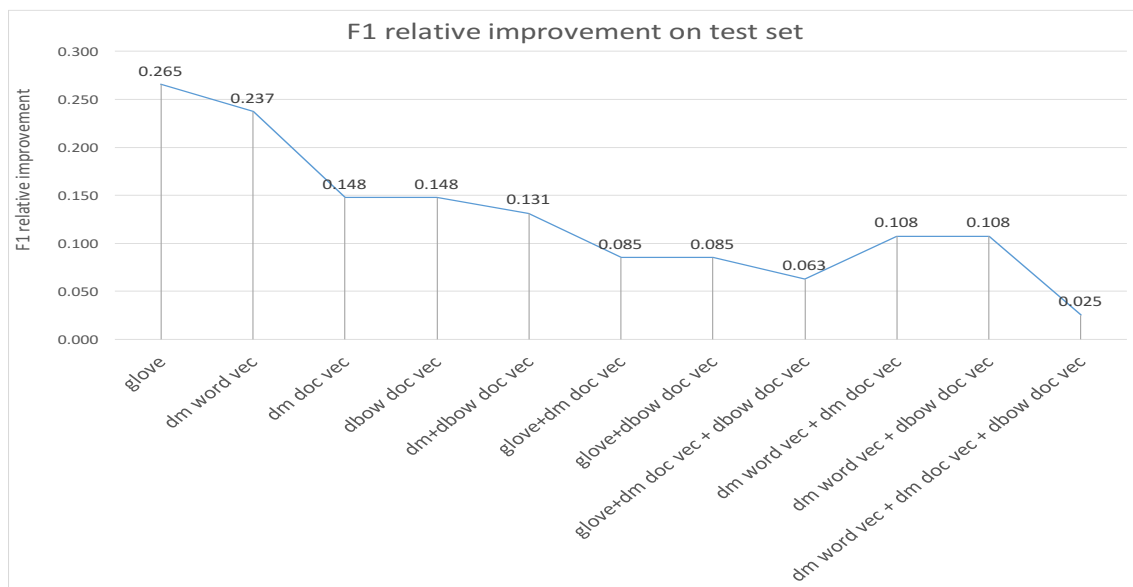


Figure 1: F1 relative improvement on test set

for the vector representations. As suggested by Mikolov and Le [2013b], we also experiment on the combination of dm and dbow vectors as contextual features.

For metrics, precision is defined as

$$\frac{\# \text{ words predicted to be biased and labeled as biased}}{\# \text{ words predicted to be biased}} \quad (1)$$

Recall is defined as

$$\frac{\# \text{ words predicted to be biased and labeled as biased}}{\# \text{ words labeled as biased}} \quad (2)$$

F1 score is defined as the harmonic mean of precision and recall

$$\frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (3)$$

We use F1 score to measure the overall performance of the linguistic model of the baseline. The result is shown in Table 2. We can see a decrease in the precision and an increase in the recall, which result in an overall increase of F1. This indicates a significant rise in false positives. Compared to the baseline, the precision of the contextual-aware model slightly drops. But we should point out that contextual features are only helpful when detecting bias associated with ambiguous words. There are relatively few ambiguous words (706 out of 3249) in the test set. For non-ambiguous words, the contextual features are not helping but increase the feature dimensionality.

4.4 Experiment on Semantic Features

To capture fine-grained semantic regularities of words, we use pre-trained word vectors of size 300 from the GloVe algorithm [Pennington *et al.*, 2014] trained on articles from Wikipedia 2014. Since the dm model can also learn the word vector representation inside its input documents, we also use the dm model to generate word vectors of size 300 as semantic features. The learned semantic features are used as input

	baseline	GloVe	dm word vec
# features	32	332	332
precision	0.245	0.284	0.304
recall	0.228	0.316	0.282
F1 score	0.236	0.299	0.292

Table 3: Results on test set after adding semantic features

to train a logistic regression model to classify bias, with the result presented in Table 3. The result shows that compared to contextual features, semantic features generally performs better in this task. Semantic features trained by the GloVe algorithm give the best F1 score. This suggests that semantic features trained either by GloVe or the dm model could significantly improve a linguistic model’s performance on bias detection.

4.5 Combination of Semantic and Contextual Features

To see if the two types of features together can strengthen the logistic regression model’s power in detecting bias, we try different combinations of semantic and contextual features to build linguistic models. The relative improvement of F1 score of different combinations against baseline is shown in Figure 1. The result shows in general semantic features alone perform better than both contextual features and the combinations of those two. The result shows by adding the GloVe as semantic features alone can reach a relative improvement of up to 26.5%. The group of results after adding contextual features alone gives second tier best result showing the model can learn from contextual features along. However, the performance drop significantly when combining semantic and contextual features. After adding contextual features, the relative ratio of F1 drops. However, we cannot conclude that contextual features do not help, since they are only helpful

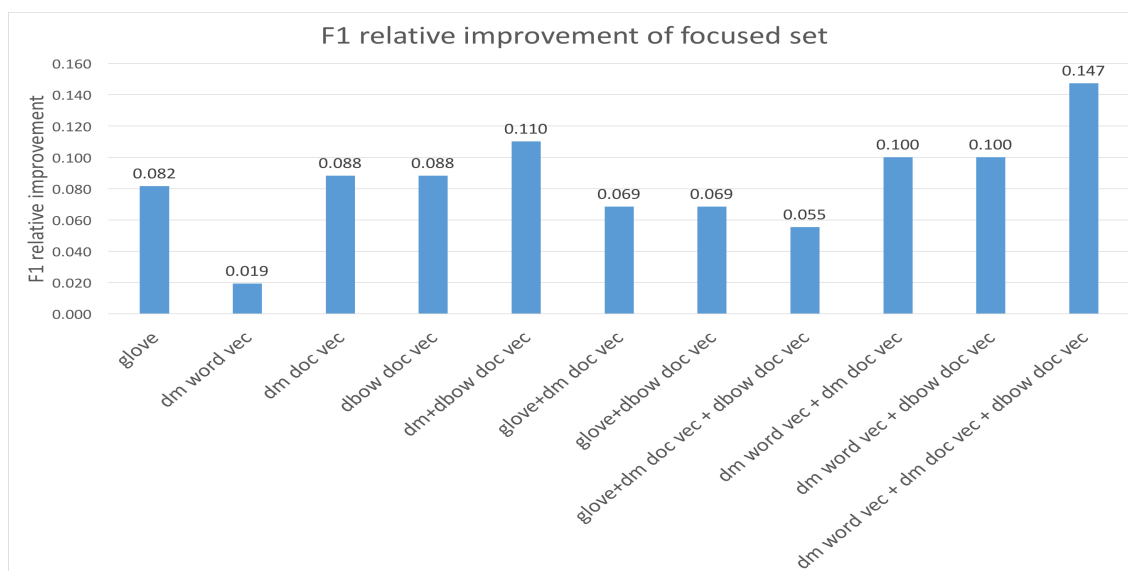


Figure 2: F1 relative improvement on focused set

	baseline	glove	dm word vec	dm doc vec	dbow doc vec	dm doc vec+dbow doc vec
precision	0.239	0.286	0.254	0.267	0.267	0.271
recall	0.484	0.438	0.453	0.500	0.500	0.516
F1	0.320	0.346	0.326	0.348	0.348	0.355

Table 4: Result on focused set when one type of feature is added

when detecting bias associated with ambiguous words. There are only a few ambiguous words in the test set. For non-ambiguous words, the contextual features are not helping but increase the feature dimensionality. It shows that in general cases, the logistic regression model does not learn well when adding the combination of semantic and contextual features.

4.6 Experiment on Focused Set

To better measure the performance of the contextual features in detecting bias associated with ambiguous words, we extracted a focused set of ambiguous words within the test set. We put the word in the focused set if the word is in the training set, labeled as biased at least once, and it is also labeled as not biased at least once. We found words such as “white”, “Arabs”, “faced”, “nationalist” and “black” to be in this focused set. We test our contextual features: dm vector, dbow vector and the combination of the two vectors on the focused set. We also test using the semantic features and the combination of semantic features and contextual features. The result is shown in Tables 4 and 5; the relative improvement of F1 score against the baseline is shown in Figure 2. In the focused set, the maximum F1 score relative improvement of 14.7% is obtained when adding both the dm document vector and dbow document vector combined with dm word vectors.

In the focused set, the advantage of the GloVe feature is not as obvious as in the full test set. Our result shows contex-

tual features (dm document vector + dbow document vector) do help in detecting bias associated with ambiguous words. The model’s performance reaches a maximum when the dm document vector and dbow document vector are combined with dm word vector. GloVe features alone behave consistently well in general cases. The result shows the linguistic model behaves better in detecting bias associated with ambiguous words when the contextual information in which the word resides is given. But when we combine GloVe features and contextual features together, the performance gets worse. The performance of the model when GloVe features are combined with contextual features is consistent in both test set and focused set. The result suggests that in bias detection for reference works, we should train two linguistic models: one with added semantic features from either GloVe or the dm model to determine non-ambiguous words’ bias detection; one with adding semantic and contextual features learned from dm and dbow models to determine bias associated with ambiguous words. Example 5a was found in the focused set, where it was not predicted correctly by baseline but predicted correctly after dm document vector and dbow document vector are added to train the logistic regression model:

5. (a) According to eyewitnesses, when one of the occupants went to alert the **Israelis** that people were inside, **Israelis** began to shoot at the house.
- (b) According to eyewitnesses, when one of the occupants went to alert the **Israeli soldiers** that people were inside, the **soldiers** began to shoot at the house.

The example was extracted from the Wikipedia article “Zeitoun incident”. After we learn the document vector representation of the article “Zeitoun incident” and add it as context when training the linguistic model, the ambiguous word “Israelis” is now recognized as a biased word.

	baseline	GloVe + dm doc vec	GloVe + dbow doc vec	GloVe + dm doc vec + dbow doc vec	dm word vec + dm doc vec	dm word vec + dbow doc vec	dm word vec + dm doc vec + dbow doc vec
precision	0.239	0.280	0.280	0.275	0.271	0.271	0.285
recall	0.484	0.438	0.438	0.438	0.500	0.500	0.516
F1 score	0.320	0.342	0.342	0.337	0.352	0.352	0.367

Table 5: Result on focused set when the combination of two types of features are added

5 Future Work

In this work, we consider vector space representations of text in the bias detection task. Traditional bias detection is usually conducted through manually crafted features as input in a machine learning algorithm such as SVM or logistic regression. After words have been successfully represented as vectors via word analogy, these vectors could be understood by complex language models such as deep neural networks. Future work can consider a deep learning solution for the bias detection task. The solution will be in two phases. Without manually crafted features, in the first phase text in which the target word resides will be input in the neural network model to train vector representations; next the vector representations will be treated as features to train a classifier for bias detection task.

6 Conclusion

In this work, we have noted some drawbacks of using pre-compiled word lists to detect bias. We use recent research progress in vector space representations of words and documents as semantic features and contextual features to train a logistic regression model for the bias detection task. Our experiment shows that semantic features learned from the GloVe algorithm reach a F1 relative improvement of 26.5% against baseline. In the experiment on a focused set of ambiguously labeled words, the linguistic model reaches the highest gain in F1 score when adding the combination of contextual features learned from the dm and dbow models combined with semantic features learned from the dm model. Semantic features learned from the GloVe algorithm behave consistently well in all experiments. The linguistic model behaves better in detecting bias associated with ambiguous words when the context in which the word resides is given.

References

- [Bengio *et al.*, 2006] Yoshua Bengio, Holger Schwenk, Jean-Sébastien Senécal, Frédéric Morin, and Jean-Luc Gauvain. Neural probabilistic language models. In *Innovations in Machine Learning*, pages 137–186. Springer, 2006.
- [Gentzkow and Shapiro, 2010] Matthew Gentzkow and Jesse M Shapiro. What drives media slant? Evidence from US daily newspapers. *Econometrica*, 78(1):35–71, 2010.
- [Greenstein and Zhu, 2012] Shane Greenstein and Feng Zhu. Collective intelligence and neutral point of view: the case of Wikipedia. NBER Working Paper 18167, National Bureau of Economic Research, June 2012.
- [Iyyer *et al.*, 2014] Mohit Iyyer, Peter Enns, Jordan L Boyd-Graber, and Philip Resnik. Political ideology detection using recursive neural networks. In *Proceedings of the Association for Computational Linguistics*, pages 1113–1122, 2014.
- [Kahn *et al.*, 2007] Jeffrey H. Kahn, Renee M. Tobin, Audra E. Massey, and Jennifer A. Anderson. Measuring emotional expression with the linguistic inquiry and word count. *The American Journal of Psychology*, pages 263–286, 2007.
- [Le and Mikolov, 2014a] Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proc. 31st Int’l Conf. on Machine Learning (ICML)*, pages 1188–1196, June 2014.
- [Le and Mikolov, 2014b] Quoc V. Le and Tomas Mikolov. Distributed Representations of Sentences and Documents. *ArXiv e-prints*, May 2014.
- [Liu *et al.*, 2005] Bing Liu, Mingqing Hu, and Junsheng Cheng. Opinion observer: Analyzing and comparing opinions on the web. In *Proc. 14th Int’l Conf. on World Wide Web (WWW)*, pages 342–351, 2005.
- [Marneffe *et al.*, 2006] M. Marneffe, B. Maccartney, and C. Manning. Generating typed dependency parses from phrase structure parses. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC-2006)*, Genoa, Italy, May 2006. European Language Resources Association (ELRA). ACL Anthology Identifier: L06-1260.
- [Mikolov *et al.*, 2013a] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient Estimation of Word Representations in Vector Space. *ArXiv e-prints*, January 2013.
- [Mikolov *et al.*, 2013b] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Inf. Processing Systems (NIPS)*, pages 3111–3119, 2013.
- [Noam, 2008] Cohen Noam. Dont like Palin’s Wikipedia story? Change it. *The New York Times*, September 2008.
- [Pennebaker *et al.*, 2015] James W. Pennebaker, Ryan L. Boyd, Kayla Jordan, and Kate Blackburn. The development and psychometric properties of LIWC2015. *UT Faculty/Researcher Works*, 2015.
- [Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. GloVe: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.
- [Recasens *et al.*, 2013] Marta Recasens, Cristian Danescu-Niculescu-Mizil, and Dan Jurafsky. Linguistic models for analyzing and detecting biased language. In *ACL (1)*, pages 1650–1659, 2013.
- [Saif *et al.*, 2012] Hassan Saif, Yulan He, and Harith Alani. Semantic sentiment analysis of twitter. In *Proc. 11th Int’l Semantic Web Conf. (ISWC)*, pages 508–524. Springer, 2012.
- [Yano *et al.*, 2010] Tae Yano, Philip Resnik, and Noah A. Smith. Shedding (a thousand points of) light on biased language. In *Proc. NAACL HLT Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk*, pages 152–158, 2010.