An extensive study on how ConTEXT Matters

Jaskirat Singh Bhatia
University Of Waterloo
jaskiratsinghbhatia4@gmail.com

Abstract

The last decade has seen a surge of research in the area of Natural Language Processing due to the unprecedented success of deep learning. It is not very well known that NLP classification tasks can often be based on causal relations between certain preprocessing techniques. In our project, we do an intensive literature survey on Causal Inference on NLP and investigate how various preprocessing techniques can have a confounding effect on different types of Natural Language classification tasks. The two classification tasks we investigate are Fake news detection and sentiment analysis. The main idea behind the project is to study how preprocessing can remove the confounding effects in a text. We reach a conclusion that this can be harmful to a few classification tasks in Natural Language Processing like in the case of Fake News Detection. In our study, we work on two different models namely BERT and Bidirectional LSTM (Baseline), and see similar causal effects on both the models.

1 Introduction

Text Classification is the procedure of assigning pre-determined labels to text, and it plays a crucial and significant task in many NLP applications [1]. NLP refers to the processing of spoken and written form of texts which acts as a model of communication among humans with the utilization of computational methods [2]. In the past decade, there has been a steep rise in the use of NLP to solve various Data Science and Artificial Intelligence problems. The need for sophisticated and efficient information handling tools that can handle a huge amount of data at a high level lead to the development of information extraction and retrieval technologies [3]. Many NLP applications aim to infer causal conclusions from non-experimental data. Such observational data often contains confounders, which can be defined as the variables that influence both causes and effects [4]. Taking care of these confounders becomes really important when it comes to NLP tasks and that is where causal inference is of great help. Causal inference aims to understand how intervening on one variable affects another variable [5].

Text data, in general, offers a lot of challenges due to its high-dimensional nature. However, text data differs from other high-dimensional data in a way that the confounding in text data can be evaluated by humans. But the same cannot be done by a machine learning model. Therefore, it becomes important to reduce the dimensionality of this data and convert it into a form that could be passed to a machine or a machine learning model.

One such technique that helps in removing confounding from NLP data and reduces the dimensionality of data is preprocessing. Preprocessing is one of the steps towards information extraction. It transforms the original data from one format to another. This processed format is suitable for employing different types of feature extraction methods. It has been observed in a study that machine learning models tend to learn spurious patterns and associations, especially in NLP tasks [6]. The whole idea behind preprocessing in NLP is to make it easier for the machine learning model to learn the associations in the dataset in a better way. There are certain ways to do it, some of the common ways are removing stopwords, lemmatization, and stemming.

StopWords: Typically, in order to identify and extract all the important features, it is necessary to filter out the words or terms, which occur frequently, does not play a significant role in the document text, and does not add much meaning to a sentence [7]. Removal of such words saves a lot of processing time and memory space and it does not have any adverse effect on the retrieval process. Additionally, it also helps the training model to learn the features that are actually important in the text and not some spurious associations. It is generally thought to be a good idea to remove the stopwords in text categorization task but in this paper, the goal is to argue the above statement and prove that the removal of such words is only useful when the context of the sentence is to be ignored. On the other hand, removing stop words from the documents may not be of use when we actually care about the context. Consider a sentence 'This movie is not good'. Since 'not' is one of the many stopwords, its removal changes the context of the whole sentence and therefore leads to confounding and which further leads to spurious association in the learning of the model. Spurious associations are due to confounding, but not direct or indirect causal effects [6].

Stemming and Lemmatization: Stemming is an information retrieval procedure to reduce the word to its root form that is achieved by trimming off the stem from the word. The stemming algorithm can sometimes go wrong while processing words in past tense like "ran"; after performing the stemming algorithm, it remains "ran" [8]. Lemmatization is very similar to stemming, but the difference here is that instead of just removing the suffix from the word, we determine what the base word will be for the given form of that word. [9]. Performing preprocessing techniques like lemmatization actually change the entire meaning of the sentence; which in some cases is irrelevant but, it might eliminate any confounding which the tense or the verb has with other words in the sequence. Again, it can have a positive impact on the model by removing confounding, or it can cause a model to form spurious patterns.

In our experiments, we will go through in detail about the models we used (Baseline and BERT) and the result of each model with different combinations of pre-processing on each dataset (Twitter Sentiment 140 and Fake News).

2 Related Work

Many researchers in the past decade have worked on the NLP classification tasks and reached the conclusion that the difference in treatment of the document content can lead to different results. Several papers demonstrate cases where NLP systems appear not to learn what humans would expect them to learn [6]. This might be due to confounding in the documents. In a study performed by Glockner et al. in 2018 [10], they demonstrate that simply replacing words by synonyms or hypernyms, which should not alter the applicable label and is supposed to preserve the context of the text, nevertheless breaks ML-based NLI systems. In a similar study, the authors replaced some of the words to test the behavior of sentiment analysis algorithms in the presence of stylistic variation and as a result, they found that that similar word pairs produce significant differences in sentiment score [11].

In another study performed by Lu et al in 2018 [12], the authors alter the text of the documents programmatically to invert gender bias. The original documents were later combined with the manipulated documents resulting in a gender-balanced dataset for learning word embeddings. This was basically done to remove the confounding from the text if any exists so that the spurious patterns are not formed. The approach seemed to work better than just the original dataset. Furthermore, there are other researches being done on language as confounders. In one such study, the authors took the work done by Lu et al [12], and described a data augmentation approach for alleviating gender stereotypes associated with animate nouns for morphologically-rich languages like Spanish and Hebrew [13].

In the task to identify fake or truth behavior, it is important to understand the positive interactions between a normal user and his day to day traits, to identify which user tends to spread fake news or the "fake news sharing behavior" [14]. Research along the field of Fake news detection shows that an individual's online behavior is highly related to his/her personality in real life, the cultural norms where the person was bought up, and also the gender, age, etc [15]. On the other hand how a user behaves online on various platforms also plays a major part in interpreting the "dark side of social media" [14]. Some online traits a user displays can be self-promotions, promoting hate, deceiving/scamming people, or displaying emotional coldness. The psychological studies show

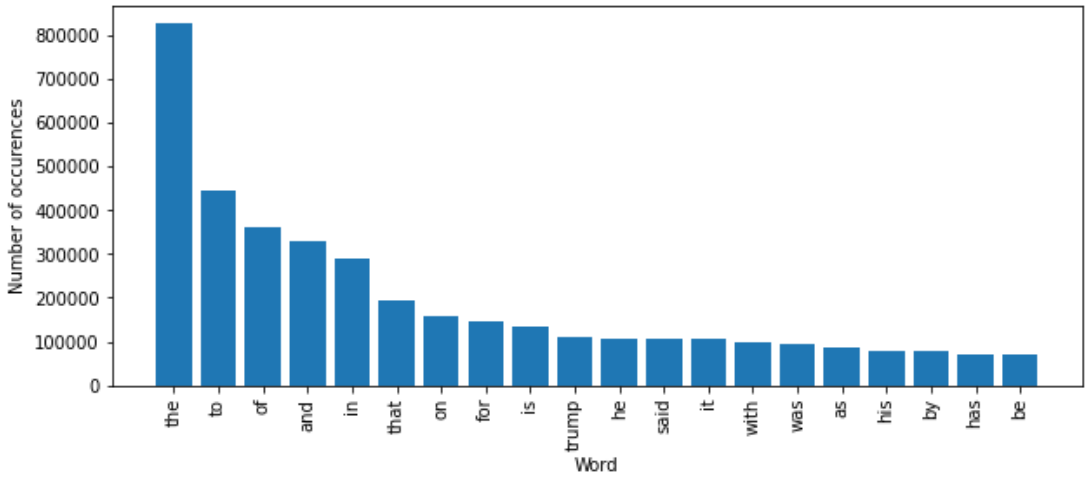


Figure 1: Top words in news dataset (including stop words)

that this information can be used to get unbiased information on the user and act as a surrogate confounder.

3 Proposed Method

Our solution relies on comparing the results of preprocessing of documents versus no preprocessing before passing it on to the model. The idea is to have a more clear understanding of how preprocessing techniques should be used in NLP classification tasks can be harmful and remove the confounders in the text. Therefore, the approach that we follow in this paper is to try different techniques like stopwords removal, stemming, and lemmatization on different datasets and train our model on the processed dataset. On the contrary, we train the same model on the original dataset with no preprocessing. We then compare the results of the models defined above and concentrate on the causal effect that preprocessing has on the results. We used two datasets in our solution. The first corresponds to the fake news detection dataset and the second is about Twitter sentiment analysis. We show that, for fake news detection, the classifiers trained on the original dataset performed better than those trained on the processed dataset. This is because the stopwords play an important part when it comes to formal documents like news, and removing those words can lead to the removal of the confounders and change the context of the documents. On the other hand, when the same approach was followed for the Twitter sentiment analysis dataset, the preprocessing plays a crucial role in training the classifiers. This is because of the model picking up spurious correlations in those datasets when the preprocessing was not performed, and hence the less accurate results.

4 Experiments

We performed a total of 16 experiments which included training 2 different classifiers on 2 different datasets. To ensure the reliability of the results, we treat the variables in the models as control variables, i.e. we did not change the variables like learning rate, optimizer, and loss function for different experiments.

4.1 Data

We used 2 different datasets for our study. (1) Fake news detection dataset and (2) Twitter Sentiment Analysis dataset.

Fake news detection dataset: The fake news detection dataset consists of news articles from different sources. The training dataset contains information like the news title, the text of the news article, the date the article was published, and the label specifying whether the article is fake or real. On the other hand, the test dataset consists of the same information as the training set without the labels.

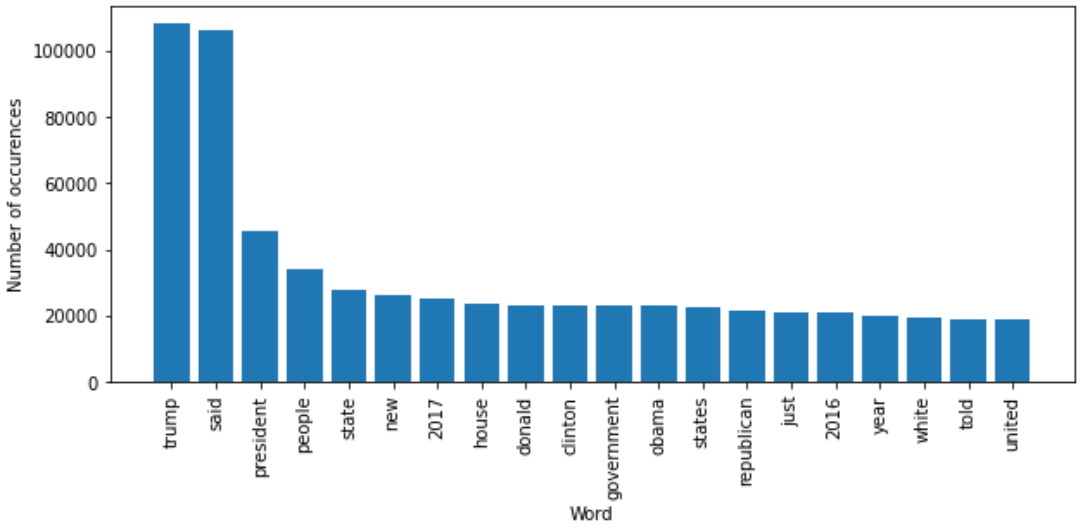


Figure 2: Top words in news dataset (excluding stop words)

The idea is to train a machine learning model that could detect whether a news article is real or fake. We perform Exploratory Data Analysis of the dataset prior to moving over to the implementation phase. Now the main idea of the paper is to compare the results of the model before and after the preprocessing is performed, keeping the model parameters the same. Figure 1 and Figure 2 denotes the most recurring words of the dataset before and after removing the stopwords respectively.

Dataset link

Twitter Sentiment Analysis dataset: The Twitter Sentiment Analysis dataset consists of tweets from people around the globe extracted using the Twitter API. The dataset contains over 1.6 million records which includes the information of the tweet and the user who tweeted it. Each record comprises of tweet id, date of the tweet, user name who tweeted it, the contents of the tweet, and the target label. The target label in this case is the sentiment of the contents of a particular tweet. However, the label is in the form of a numeral where 0 corresponds to negative, 2 corresponds to neutral, and 4 signifies a positive sentiment. Unlike the news dataset, where we had a separate test set, there is only a single set available for the Twitter data. The main idea is to train classifiers on the data which would predict the sentiment of the tweets by learning from the training data. We compare the results from the model trained using preprocessing and the one without using preprocessing. Figure 3 and Figure 4 denotes the most recurring words of the dataset before and after removing the stopwords respectively.

Dataset link

4.2 Models

Our experiments rely on the following two models: Bidirectional Long Short-Term Memory Networks [16] and fine-tuned BERT model [17]. For brevity, we discuss only the implementation details necessary for reproducibility.

Bi-LSTM: When using Bidirectional LSTMs for training the model for both the datasets, the vocabulary is restricted to the most frequent 3000 tokens, replacing out-of-vocabulary tokens by UNK. The maximum input length is fixed at 512 tokens and the smaller documents are padded. Each token is represented by a randomly-initialized 100-dimensional embedding. We have used bidirectional LSTM following an Embedding layer, which is further followed by 2 Dense layers in the case of the news dataset and 4 Dense layers for the Twitter dataset. We have also used Batch Normalization and Dropout between the dense layers. To generate output, we feed this (fixed-length) representation through a fully-connected hidden layer with ReLU activation [18], and then a fully-connected output layer with sigmoid and softmax activations for news and Twitter datasets respectively. Every model uses Adam classifier [19] with a learning rate of 0.015 and is trained for 10 epochs. We also

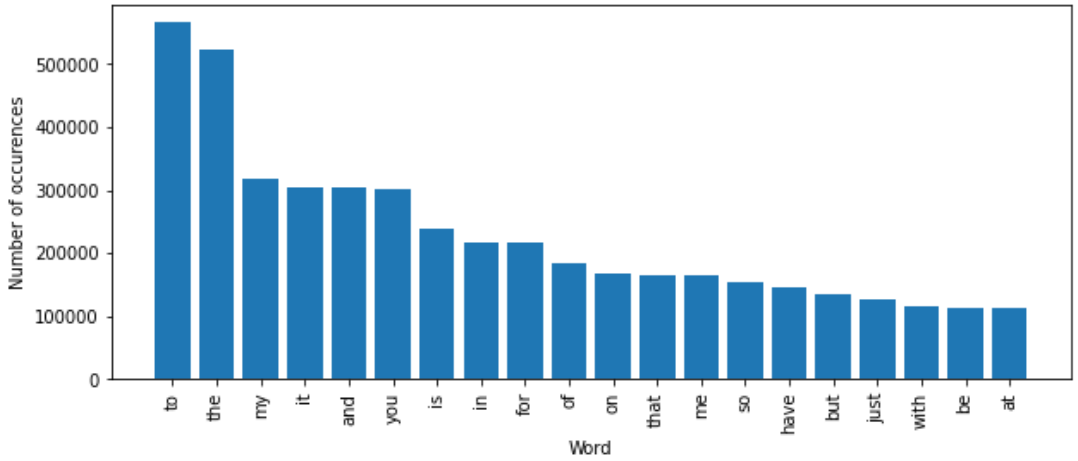


Figure 3: Top words in Twitter dataset (including stop words)

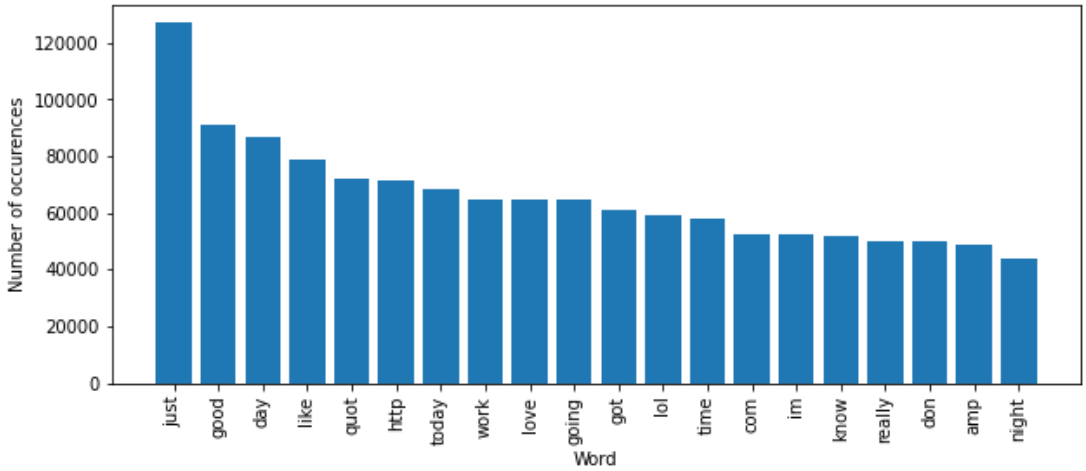


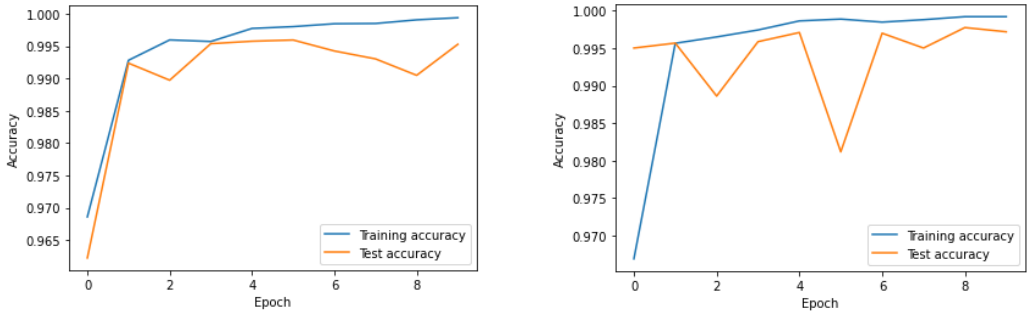
Figure 4: Top words in Twitter dataset (excluding stop words)

applied early stopping when validation loss does not decrease for 3 epochs. Figure 5 shows training and testing accuracy of the model with preprocessing (stopwords removal, lemmatization, and stemming) and without any preprocessing respectively.

Fine-tuned BERT: We fine-tuned BERT to compare the results with our baseline model. We have used the maximum token length as 256 for the news dataset and 512 for Twitter dataset, to account for BERT's sub-word tokenization. We trained the model for 5 epochs and used Early Stopping when validation loss does not decrease for 3 epochs. Figure 6 shows training and testing accuracy of the model with preprocessing (stopwords removal, lemmatization, and stemming) and without any preprocessing respectively.

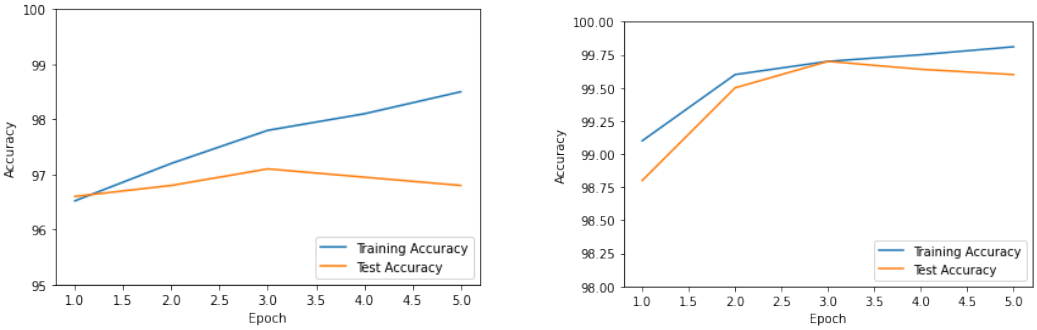
4.3 Results

Fake News detection: When models were trained on the news dataset, it was observed that both the models (Bi-LSTM and BERT) performed better with the original data, i.e. when no preprocessing was done. Table 1 shows the testing accuracy of Bi-LSTM for different approaches. The last column corresponds to the fake news detection task. The model achieved an accuracy of 99.7% in the case when there was no processing involved. On the contrary, the accuracy was just below 99% when some sort of preprocessing was done.



(a) With preprocessing (b) Without preprocessing

Figure 5: Bi-LSTM Training vs Test accuracy on news dataset



(a) With preprocessing (b) Without preprocessing

Figure 6: BERT Training vs Test accuracy on news dataset

Similar results were obtained when the predictions were made using BERT. Table 2 shows the testing accuracy of BERT for different approaches and the maximum accuracy was achieved when no preprocessing was done on the data.

The above results might be due to the fact that the news articles are a sort of formal documents and every word has some context linked to it. Therefore it becomes important to take into consideration the context and not remove stopwords in such cases since there seems to be a causal relation between stopwords and other words in the form of context.

Twitter Sentiment Analysis: When models were trained on the Twitter dataset, both the models (Bi-LSTM and BERT) performed better when preprocessing was performed. Table 1 shows the testing accuracy of Bi-LSTM for different approaches. The second column corresponds to the sentiment analysis task. The model achieved an accuracy of 78.8% in the case when the stopwords were removed from the data and lemmatization was done. On the contrary, the accuracy was just over 74% when preprocessing was not at all done.

Similar results were obtained when the predictions were made using BERT. Table 2 shows the testing accuracy of BERT for different approaches and the maximum accuracy was achieved when preprocessing was done on the data. We managed to achieve an accuracy of over 80% when stopwords were removed and lemmatization was done, as compared to around 75% when no preprocessing was done.

The above results might be due to the fact that the tweets are a sort of informal documents and a lot of the words written in a tweet might not have any context linked to it. Therefore it becomes really important to skip those commonly occurring words and focus on the words which actually seem to contribute towards having a causal relation with the document.

	Testing Accuracy(%)	
Data Preprocessing	Twitter Sentiment Analysis	Fake news detection
Removed stopwords	77.2	98.8
Lemmatize only	76.1	98.4
Lemmatize and removed stopwords	78.8	99.3
No preprocessing	74.2	99.7

Table 1: Bi-LSTM testing accuracy for different approaches on different datasets

	Testing Accuracy(%)	
Data Preprocessing	Twitter Sentiment Analysis	Fake news detection
Removed stopwords	78.2	97.1
Lemmatize only	76.9	97.8
Lemmatize and removed stopwords	80.4	98.9
No preprocessing	75.6	99.7

Table 2: BERT testing accuracy for different approaches on different datasets

5 Conclusion

As proposed earlier, confounding within a text can have a significant impact on NLP classification tasks. We discussed various preprocessing techniques and their confounding effects on a text sequence. Moreover, we applied those techniques on two different datasets (Twitter Sentiment-140 and Fake News Dataset) to elucidate that context plays a crucial role when it comes to the classification tasks in Natural Language Processing and that preprocessing not always leads to better models. In fact, it should be avoided in case the context of the documents needs to be preserved. On the contrary, performing preprocessing is quite helpful in other cases. Therefore, it becomes really important to have an idea of when to perform preprocessing and when not.

Acknowledgement

We would like to thank Google Colab for being such an amazing free resource. We trained all our models on Google Colab on a Tesla T4 GPU. We would also like to thank Kaggle for providing the datasets to perform this study.

References

[1] Qian Li Hao Peng Jianxin Li Congying Xia Renyu Yang Lichao Sun Philip S. Yu and Lifang He. *A Survey on Text Classification: From Shallow to Deep Learning*. IEEE Transactions on Neural Networks and Learning Systems, 2020.

[2] Hisham Assal John Seng Franz Kurfess Emily Schwarz Kym Pohl. *Semantically-Enhanced Information Extraction*. IEEE, 2011.

[3] Paul O' Neil Woojin Paik. *The Chronological Information Ex- traction System (CHESS)*. IEEE, 1998.

[4] Katherine Keith David Jensen and Brendan O'Connor. *Text and Causal Inference: A Review of Using Text to Remove Confounding from Causal Estimates*. Association for Computational Linguistics, 2020.

[5] Judea Pearl. *Causality: Models, Reasoning and Inference*. Springer, 2000.

[6] Divyansh Kaushik Eduard Hovy Zachary C. Lipton. *Learning the difference that makes a difference with counterfactually-augmented data*. International Conference on Learning Representations, 2020.

[7] Sudersan Behera. *Implementation of a finite state automation to recognize and remove stop words in english texts on its retreival*. Proceedings of the 2nd International Conference on Trends in Electronics and Informatics, 2018.

[8] Anjali Ganesh Jivani. *A Comparative Study of Stemming Algorithms*. Int J Comp Tech Appl Vol two, 2011.

[9] Jenna Kanerva Filip Ginter Tapio SalakoskiTurku. *Universal Lemmatizer- A Sequence to Sequence Model for Lemmatizing Universal Dependencies Treebanks*. Journal of Natural Language Engineering, 2020.

[10] Max Glockner Vered Shwartz and Yoav Goldberg. *Breaking nli systems with sentences that require simple lexical inferences*. Association for Computational Linguistics (ACL), 2018.

[11] Judy Hanwen Shen Lauren Fratamico Iyad Rahwan and Alexander M Rush. *Darling or babygirl? investigating stylistic bias in sentiment analysis.* 5th Workshop on Fairness, Accountability, and Transparency in Machine Learning (FATML), 2018.

[12] Kaiji Lu Piotr Mardziel Fangjing Wu Preetam Amancharla and Anupam Datta. *Gender bias in neural natural language processing*. arXiv preprint arXiv:1807.11714, 2018.

[13] Ran Zmigrod Sebastian J. Mielke Hanna Wallach and Ryan Cotterell. *Counterfactual data augmentation for mitigating gender stereotypes in languages with rich morphology*. Association for Computational Linguistics (ACL), 2019.

[14] Shalini Talwar Amandeep Dhir Puneet Kaur Nida Zafar and Melfi Alrasheedy. *Why do people share fake news? Associations between the dark side of social media use and fake news sharing behavior*. Journal of Retailing and Consumer Services, 2019.

[15] Huan Liu Lu Cheng1 Ruocheng Guo1 Kai Shu. *Towards Causal Understanding of Fake News Dissemination*. Illinois Institute of Technology Conference 21, 2020.

[16] Alex Graves and Jurgen Schmidhuber. *Framewise phoneme classification with bidirectional lstm and other neural network architectures*. International Joint Conference on Neural Networks (IJCNN), 2005.

[17] Jacob Devlin Ming-Wei Chang Kenton Lee and Kristina Toutanova. *Bert: Pre-training of deep bidirectional transformers for language understanding*. North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), 2019.

[18] Vinod Nair and Geoffrey E Hinton. *Rectified linear units improve restricted boltzmann machines*. International Conference on Machine Learning (ICML), 2010.

[19] Diederik P Kingma and Jimmy Ba. Adam. *A method for stochastic optimization*. International Conference on Learning Representations (ICLR), 2015.