A Focused Analysis of Twitter-based Disinformation from Foreign Influence Operations

Julio Amador Díaz López
Pranava Madhyastha
j.amador@imperial.ac.uk
pranava@imperial.ac.uk
Imperial College London
London, UK

ABSTRACT

Detection of foreign political influence operations is an important problem in the current era of high-information transaction. In this paper, we present a focused study on disinformation from a foreign influence campaign over twitter during the 2016 US presidential election. We introduce a new dataset of political disinformation related to a foreign influence operation on Twitter during the 2016 presidential campaign in the United States. We further analyze the differences between information pushed forward by foreign agents and legitimate information concerning word usage. We also investigate the utility of subword level information for classification. Contrary to popular belief we observe that considering only subword level information may lead to sub-optimal results.

KEYWORDS

Disinformation, Twitter

1 INTRODUCTION

As the spread and diffusion of *fake news* has reached the mainstream, the detection of all kinds of disinformation, which are understood as pieces of information purposefully crafted to deceive, has attracted significant interest from the NLP community. The detection of an important type of disinformation campaign – foreign influence operations – occupies academics and practitioners alike, particularly so in times of an election. However, this is a very challenging task, as the detection of any type of disinformation is particularly difficult even for humans. There is an urgent need to build automated systems for detecting disinformation and stem its spread.

Research in automated deception detection has made extensive use of textual features to detect disinformation. This research is grounded on psychological and social scientific insights showing that deceivers' usage of language is often flawed. That is, when deceivers try to craft their messages to imitate non-deceivers, frequently it is found to contain language "leakages". Such information is extremely relevant for the detection of disinformation [Feng and Hirst 2013; Rubin 2017]. A well-known example of such leakages can be found in the AIDS disinformation campaign, where deceivers used syntax that would not be otherwise used by native speakers; e.g., "virus flu" against "flu virus" [Ellick and Westbrook 2018]. More recently, misspellings in political disinformation were found to be particularly useful in flagging specific social media posts as originating from a bad actor spreading disinformation [Alba 2020].

Different research strands have focused on studying diverse aspects of disinformation; e.g. [Barrón-Cedeño et al. 2019; Monti et al. 2019; Vlachos and Riedel 2014; Wang 2017; Zubiaga et al. 2016]. Specifically to the study of morphologies is the work of [Kapusta and Obonya 2020] and [Zervopoulos et al. 2020]. [Kapusta and Obonya 2020] use a corpus in Slovak and conclude that pre-processing morphologies helps in classification performance. [Zervopoulos et al. 2020] study disinformation content related to the protests in Hong-Kong and find significant differences in morphological variance between disinformation and other types of information and that such differences can be exploited to improve classifier performance.

In this paper, we focus on the 2016 US presidential election. We are particularly interested in understanding word usage and relevance of subword information for detection. Towards this end, we present a new dataset of political disinformation on Twitter[1]. We analyze distributional representations to uncover the patterns associated with disinformation. We also compare the contribution of word-level and character-level information in the context of more complex machine learning models for detection. Our primary contributions in this paper are: a) we release a curated dataset aimed at detecting disinformation (Section 2); b) we present an analysis of the word-usage in the context of disinformation during the 2016 US elections (Section 3); c) our analysis reveals the potential limitations of sub-word units for deception detection (Section 3).

2 DATA

Our dataset is made up of two different parts: set (1) was collected between November 9th 2016 and March 31st 2017 using the following keywords: `#MyVote2016`, `#ElectionDay`, `#electionnight`, `@realDonaldTrump`, `@HillaryClinton` to tweets related to the election campaign. This collection yielded a total of 57, 379, 672 tweets. Set (2) was retrieved from [Linvill and Warren 2020] and consists of 2, 946, 220 tweets ranging from June 19th, 2015 to December 31st 2017. To ensure tweets corresponded only to the presidential campaign in the United States, we restricted tweets in set (2) to those before March 31st 2017, yielding a total of 1, 244, 480. Of these, we only retain original tweets (i.e., we purge 'retweets' or duplicate mentions). It is important to note that the set (2) corresponds to accounts identified by the FBI as belonging to a foreign influence campaign. For more details see [Linvill and Warren 2020].

For the negative samples (samples which are not disinformation), we remove all tweets that have any author level content that corresponds to accounts in (1). We also use tweets only in English. To

[1]While Twitter refers to accounts used by this research as spreading misinformation, we follow [Linvill and Warren 2020] and refer to these accounts as spreading disinformation.

KnOD'21 Workshop - April 14, 2021

ensure tweets in the sample are relevant, we restrict the tweets to those that belonged to the US as the geographical location in the metadata. Specifically, we restricted our sample to tweets that have geolocation coordinates to be within the US. We used Twitter's API to ensure that tweets we considered were coming from users whose accounts have not been suspended by Twitter four years after the events and consider this to be a proxy for valid accounts. In specific, we called Twitter's user API[2] and eliminated accounts that returned errors 50, user not found, and 63 and 64, suspended accounts. This yielded 3, 324 tweets. Finally, we manually checked these tweets to make sure their content was related to the 2016 presidential election.

Next, we used random undersampling to balance the dataset. We thus present a new dataset that consists of 6, 808 unique tweets (i.e., 'retweets' or duplicate mentions are purged) in English that relates to the 2016 presidential election in the United States. The complete dataset has 16,193 tokens. Concerning categories, the dataset contains 3,324 tweets with 8,871 unique tokens labelled as legitimate information and 3,484 tweets with 10,434 unique tokens labelled as disinformation. Finally, we removed strings beginning with the following characters: `#`, `@`, `.@`, and `https://`, and removed emojis. This made the average length of the strings 10.8736 tokens. For our analyses, we normalized the text by converting all the strings into lowercase.

Specific to the analyses in section 3, we partitioned the dataset into training (60%), development (30%) and tests (10%) sets. The training set has 2,083 tweets labelled disinformation, and 2,001 labelled legitimate. The development set has 1,046 tweets labelled disinformation and 997 labelled legitimate. The test set contains 355 tweets labelled disinformation and 326 labelled legitimate.

The dataset is openly available here:
https://zenodo.org/record/4639608#.YF3wxi2ZPOQ.

3 ANALYSIS

In this section, we first present our analysis on the word-usage and then expand on the utility of sub-word information. We further present an analysis of word-based and character-based CNN models on our dataset.

Word usage. Within the context of our research (i.e., foreign disinformation in the 2016 presidential election in the United States), we aim to understand whether: "tweets containing disinformation have different word-usage patterns from those containing legitimate information." To investigate this, we begin by exploring the word co-occurrence space spanned by these tweets.

We then use point-wise mutual information (PMI) to capture collocations and associations. We obtain two co-occurrence matrices with PMI: one for disinformation and the other for legitimate information. Each of the matrices was of size 5000×5000. We further reduce the dimensionality of the matrices with Latent Semantic Analysis (LSA) to 300-dimensions resulting in matrices of size 5000×300 using. We finally measure the cosine-distance between all the 5000 words.

We further use sub-word level representations following Bojanowski2017. For each of the 5000 most frequent words, we obtain the representation of 'words' as a combination of character bi-grams. In this way, any word can be represented by the sum of their character bi-grams. This is one of the predominant ways of increasing coverage and decreasing out-of-vocabulary words in the literature [Liu et al. 2019; Sennrich et al. 2013].

[2]See: https://developer.twitter.com/en/docs/accounts-and-users/follow-search-get-users/api-reference/get-users-lookup

Token	Word	Sub-word
Mean diff	0.879673	0.143676
realdonaldtrump	0.9902	0.6234
clinton	0.9301	0.2890
obama	0.8745	0.0996
fake	0.8428	0.2675
media	0.9011	0.1946

Table 1: Cosine distance between different tokens. Columns indicate word or sub-word distances. Last row represents the mean distance between tokens for word/sub-word.

We present our analysis on the dataset in Table 1. Here our premise is that the distributional information captures word usage. So if two words are used in similar ways, they should be very similar across the two classes, i.e., the cosine distance between them should be tending to 0. In Table 1, we notice that the mean difference between similar words indicates that the word-usage between the two classes are significantly different from each other. This is especially true when we consider word-level representations. We further notice that the sub-word level representations are much more closer than word-level information. We hypothesize that sub-word level information controls for morphological and typological variation and thereby does not capture the diversity as well as the word-level representations. We also present a few example words to illustrate the difference. We observe that the distances for 'realdonaldtrump' and 'clinton' are significantly different, indicating the diversity in contexts. However, we observe that the sub-word level representations are generally closer to each other.

Word or sub-word representations? The above findings suggest that models which control for sub-word differences are sub-optimal. We further examine this by using the partitioned dataset (i.e., partitioned into train, development and test sets) and building classifiers with concatenated representations of LSA based word representations. In this case, the representations for the words are obtained directly from LSA, while for the sub-word level, for each word we sum the representations of the character-bigrams. We specifically make use of Naive Bayes, Logistic Regression and SVM based classifiers. We present our results in Table 2.

	Word level		Sub-word level	
Classifier	**Accuracy**	**F1 score**	**Test**	**F1 score**
Logistic	0.6823	0.6824	**0.6334**	**0.6329**
Naive Bayes	0.5862	0.5858	0.6164	0.6136
SVM	**0.7275**	**0.7238**	0.6236	0.6226

Table 2: Accuracy and macro F1 scores for the Logistic, Naive Bayes and SVM classifiers. Classification was done using word and sub-word level representations built section 3.

We observe that compared to sub-word level representations, word-level representations appear to obtain better accuracy and F1-scores. We note that, while these results are not conclusive, they seem to support the overarching theme regarding the utility of sub-word level information being sub-optimal.

We further use best performing SVM based classifier that uses word-level representations and compared it to a frequently-used CNN based based classifier for the task of detecting misinformation in [Kim et al. 2016]. We use the classifier in the standard setup for our experiments[3].

Classifier	Accuracy	F1 score
SVM word-level	**0.7275**	**0.7238**
CNN word-level	0.6986	0.6970
CNN sub-word	0.7095	0.7088

Table 3: Accuracy, and macro F1 scores for the SVM and Character CNN classifiers. The SVM used representations in section 3. The Character CNN used fasttext word and sub-word representations.

We note that the CNN classifier was trained using fasttext embeddings, both, at the word and sub-word levels. We present our results in Table 3.We observe that SVM classifier that uses word-level representations outperforms the more complex CNN based models. We further perform an in depth analysis of the CNN based model and provide details in the appendix.

Bag-of-words/characters representations. Results presented above may be driven by the particular representations built for the preliminary analysis. In order to test the robustness of these findings, we train the Naive Bayes, Logistic and SVM classifiers using bag-of-words and bag-of-character representations. The former is trained using word level uni-grams whereas the latter is trained using character level unigrams, bi-grams and tri-grams in order to consider different sub-word representations. Both of them used TFIDF. To underscore the robustness of our results, we use a 10-fold cross validation and average accuracy scores.

The first three rows in Table 4 present the results. We note that changing the representations from dense to sparse does not change our results.

Morphologies. To understand if morphologies are the contributing factors for the classification performance, we further perform experiments with an SVM based classifier that is trained on bag-of-stemmed-words, i.e, we first stem all the words to remove the morphological inflections and then train the bag of words based classifier.

The last two rows of Table 4 shows that, by removing prefixes and suffixes from the corpus, the classifier using sub-word level representations is affected the most. This suggest merely controlling for morphologies rather than integrating morphological information may result in sub-optimal classification performance.

[3]Further details regarding the setup are provided in the appendix

	Word level		Sub-word level	
Classifier	**Accuracy**	**Std Dev**	**Accuracy**	**Std Dev**
Logistic	0.8735	0.0185	0.8501	0.0110
Naive Bayes	0.8499	0.0068	0.7980	0.0105
SVM	0.9173	0.00487	0.8341	0.0071
SVM stemmed	0.9071	0.0054	0.8165	0.0099

Table 4: Mean accuracy and standard deviations for the Logistic, Naive Bayes and SVM classifiers using bag-of-words and bag-of-characters representations.

4 CONCLUSION

This paper is a focused study on the disinformation campaign from a foreign influence operation in Twitter during the 2016 US presidency election. We introduce a new dataset of political disinformation to explore differences between disinformation and legitimate information. Our analysis of the dataset indicates divergent word-usage patterns between disinformation and legitimate information. We also study the effect of sub-word patterns and its utility for classification. Our results indicate that classifiers that only rely on sub-word based information may have better coverage, but may control for morphological features. This may result in sub-optimal performance. We hope that our dataset can help inform novel insights relating to disinformation and propaganda and leads to development of better detection algorithms.

REFERENCES

Davey Alba. 2020. How Russia's Troll Farm Is Changing Tactics Before the Fall Election. https://www.nytimes.com/2020/03/29/technology/russia-troll-farm-election.html

Alberto Barrón-Cedeño, Israa Jaradat, Giovanni Da San Martino, and Preslav Nakov. 2019. Proppy: Organizing the news based on their propagandistic content. *Information Processing and Management* 56, 5 (sep 2019), 1849–1864. https://doi.org/10.1016/j.ipm.2019.03.005

Adam B Ellick and Adam Westbrook. 2018. Opinion | Operation Infektion: A three-part video series on Russian disinformation. https://www.nytimes.com/2018/11/12/opinion/russia-meddling-disinformation-fake-news-elections.html

Vanessa Wei Feng and Graeme Hirst. 2013. *Detecting deceptive opinions with profile compatibility.* Technical Report. 14–18 pages. http://tripadvisor.com.

Jozef Kapusta and Juraj Obonya. 2020. Improvement of misleading and fake news classification for flective languages by morphological group analysis. *Informatics* 7, 1 (feb 2020), 4. https://doi.org/10.3390/informatics7010004

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016. Character-Aware Neural Language Models. In *Thirtieth AAAI Conference on Artificial Intelligence*. www.aaai.org

Darren L. Linvill and Patrick L. Warren. 2020. Troll Factories: Manufacturing Specialized Disinformation on Twitter. *Political Communication* (2020). https://doi.org/10.1080/10584609.2020.1718257

Zihan Liu, Yan Xu, Genta Indra Winata, and Pascale Fung. 2019. Incorporating word and subword units in unsupervised machine translation using language model rescoring. *arXiv preprint arXiv:1908.05925* (2019).

Federico Monti, Fabrizio Frasca, Davide Eynard, Damon Mannion, and Michael M. Bronstein. 2019. Fake News Detection on Social Media using Geometric Deep Learning. (feb 2019). arXiv:1902.06673 http://arxiv.org/abs/1902.06673

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

Victoria Rubin. 2017. Deception Detection and Rumor Debunking for Social Media. *FIMS Publications* (jan 2017). https://ir.lib.uwo.ca/fimspub/92

Rico Sennrich, Martin Volk, and Gerold Schneider. 2013. Exploiting Synergies Between Open Resources for German Dependency Parsing, POS-tagging, and Morphological Analysis. In *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*. INCOMA Ltd. Shoumen, BULGARIA, Hissar, Bulgaria, 601–609. https://www.aclweb.org/anthology/R13-1079

Andreas Vlachos and Sebastian Riedel. 2014. Fact Checking: Task definition and dataset construction. (2014), 18–22. http://www.politifact.com/

William Yang Wang. 2017. Liar, Liar Pants on Fire: A New Benchmark Dataset for Fake News Detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 422–426. https://doi.org/10.18653/v1/P17-2067

Alexandros Zervopoulos, Aikaterini Georgia Alvanou, Konstantinos Bezas, Asterios Papamichail, Manolis Maragoudakis, and Katia Kermanidis. 2020. Hong Kong Protests: Using Natural Language Processing for Fake News Detection on Twitter. In *IFIP Advances in Information and Communication Technology*, Vol. 584 IFIP. Springer, 408–419. https://doi.org/10.1007/978-3-030-49186-4_34

Arkaitz Zubiaga, Elena Kochkina, Maria Liakata, Rob Procter, and Michal Lukasik. 2016. Stance classification in rumours as a sequential task exploiting the tree structure of social media conversations. In *COLING 2016 - 26th International Conference on Computational Linguistics, Proceedings of COLING 2016: Technical Papers.*

A APPENDIX

A.1 Word usage.

We detail here how the representations used in section **Word usage** are built. We first consider 5,000 most frequent words in all the corpus out of 16,193 words and then calculate the co-occurrence of these words *within* each of the two categories separately (i.e., we calculate the co-occurrence of the 5000 most frequent words in all the corpus within tweets labelled *disinformation* and then within those tweets labelled *legitimate information*)[4]. We consider a word w_j to co-occur with word w_i if w_j is within a window of 5 tokens to the left or right of w_i. We consider a wide window to capture the differences in word-usage. Figure 1 shows the vector spaces for disinformation, legitimate information and the complete dataset.

A.2 Classifiers

To perform experiments with the Naive Bayes, Logistic and SVM classifiers we made use of the `sklearn` [Pedregosa et al. 2011] package. Furthermore, hyperparameters of all of the classifiers were chosen according to the development set. In particular, the parameters `C`, `penalty`, and `fit_intercept` were tuned for the logistic classifier, the parameters, `C`, `gamma`, and `kernel` were tuned for the SVM classifier, and the parameters `alpha` and `fit_prior` were tuned for the Naive Bayes classifier.

A.3 Character CNN

Here we detail the Character CNN used in section 3:

- Character CNN inputs: Embeddings initialised with fasttext
  + 1D Convolution with 3 filters of size 2 and a tanh activation
  + 1D Convolution with 4 filters of size 3 and a tanh activation
  + 1D Convolution with 5 filters of size 5 and a tanh activation
  + Max-pooling over time layer + dense layer using sigmoid activation.

The number of hidden units, dropout, and the learning rate were tuned using uniform random sampling. Moreover, the CharCNN was trained for 100 epochs. Details related to the hyperparameters can be found in the code. Furthermore, Figure 3 plots loss, training and validation accuracy. It is important to underscore that the Char-CNN is far more complex with just over 11K parameters, whereas the SVM used only 3.6K parameters (average tweet length of 12 times 300 dimensional representations flattened).

[4]Further explorations were done with 1000, 2000, and 8000 most frequent tokens. We note that it did not lead to any significant differences.

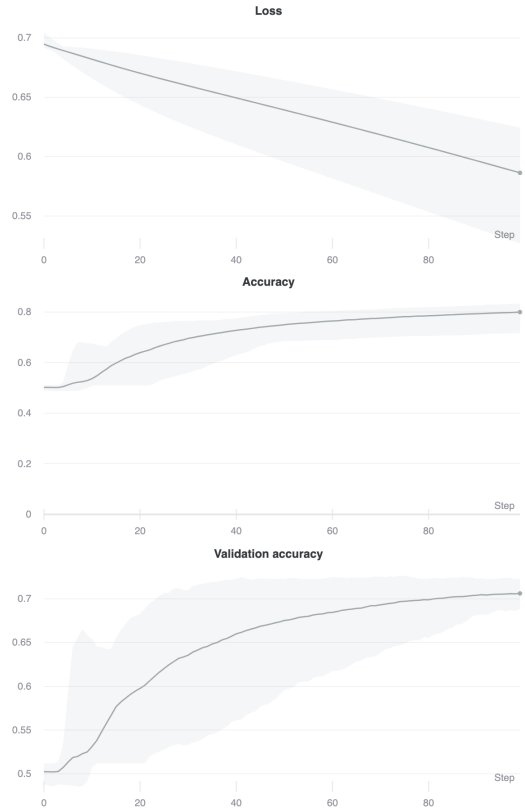

Figure 2: Average loss, train and validation accuracy for the Character CNN using word level embeddings as inputs.

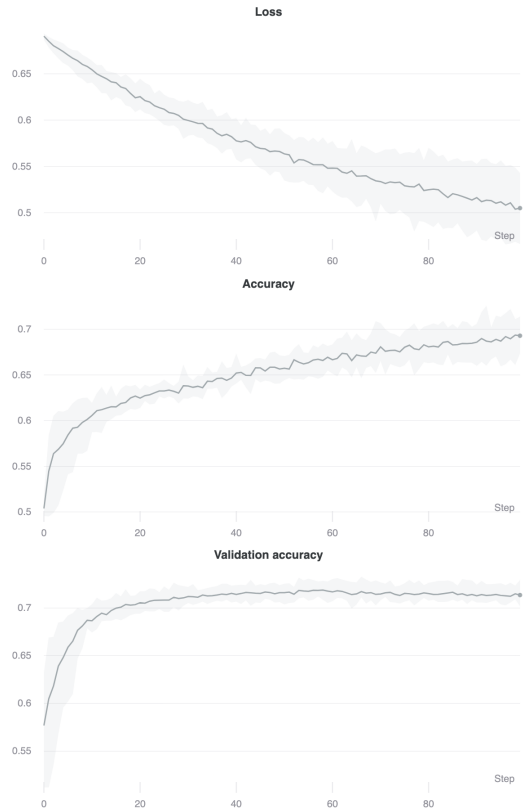

Figure 3: Average loss, train and validation accuracy for the Character CNN using subword level embeddings as inputs.

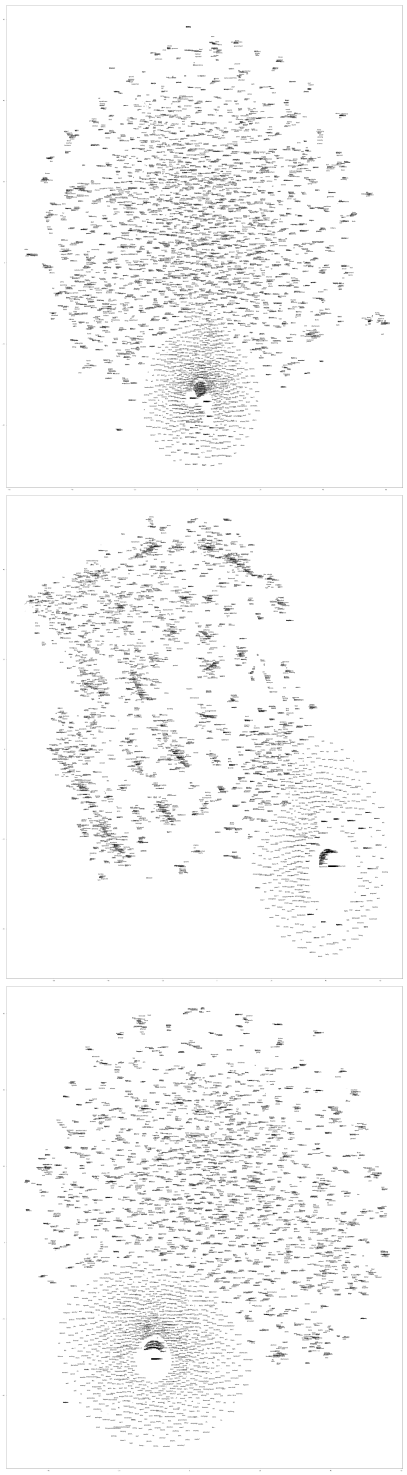

Figure 1: Visual representation of word vector spaces. Figure in the upper section contains the visual representation of every token in the corpus. Figures in the middle and lower sections contain the visual representation of the tokens in the legitimate and disinformation subsets of the dataset respectively.