Towards easy-to-implement misinformation automatic detection for online social media

Julio Amador Díaz López
Data Science Institute
Imperial College London, UK
j.amador@imperial.ac.uk

Miguel Molina-Solana
Data Science Institute
Imperial College London, UK
mmolinas@ic.ac.uk

Juan Gómez-Romero
Dept. Computer Science and AI
Universidad de Granada, Spain
jgomez@decsai.ugr.es

Abstract

The introduction of social media technologies has fostered the diffusion of misinformation. The speed and format with which misinformation is created and the pace at which it diffuses poses new challenges for practitioners and policy-makers alike. In this paper, we show that a simple network using character-level inputs given to a CNN is well suited to detect misinformation. We believe the effectiveness of such a simple architecture is because the CNN can exploit morphological differences between misinformation and other types of information. To test our intuition, we compare the character CNN inputs to different others like word embeddings. We find that the network using character CNN outperforms models that do not take into consideration morphologies and matches the performance of others that consider it to some extent. We argue that the nature of misinformation, the low availability of training data and the multidisciplinary background of individuals implementing misinformation detection algorithms make simple, easy-to-implement networks such as the character CNN good alternatives to other models.

1 Introduction

In 2016, the world witnessed the climax of an epidemic of misinformation. Democracies from around the world were suddenly confronted to deceivers that, regardless of their goals, looked to influence people's beliefs and perceptions; and even if such worries had been present every time new technological breakthrough appears (Pennycook and Rand, 2017), the magnitude and scale of the problem facilitated through social media overwhelmed governments from Britain to the United States. So profound were the stakes and consequences that countries like France and the United Kingdom have sparked national discussions on how to rein in misinformation campaigns.

Even if the problem of misinformation and deception detection is not new, its format (e.g., the way it can be easily crafted and then spread through social media) poses new challenges for academics and practitioners alike. In fact, there has been little research in the field of automatic deception detection in social media[1] (Rubin, 2017) as many argue that studying textual elements within social media posts is challenging and contend that a better way to study the problem is by analysing its patterns of diffusion (Monti et al., 2019).

However, long-established research in automatic deception detection together with newer studies in psychological research suggests there is a case to be made in using textual features of social media posts to detect misinformation. The former argue that deceivers carefully select their language to avoid being caught (Feng and Hirst, 2013). But research has shown that monitoring linguistic patterns is difficult for deceivers and can be used to detect lies[2]. The latter suggests polarisation makes it more likely for individuals to rationalise beliefs and raise the possibility deceivers may engineer their posts to maximise influence (Nyhan and Reifler, 2010; Polage, 2012; Pennycook et al., 2017; Swire et al., 2017). Recent research looking into misinformation in the 2016 presidential campaign in the United States (Amador et al., 2017; Vosoughi et al., 2018) finds that deceivers are using both language leakages and polarisation. Language leakages were detected by statistically significant differences in the use of capitalisation and special characters (Amador et al., 2017). Dif-

[1] Notable exceptions are: Mukherjee et al. (2013); Toma and Hancock (2012); Ranabothu Nithin and Kumar (2012).

[2] A well-known example of language leakages can be found in the AIDS disinformation campaign in the 80s where deceivers claiming to be English native speakers wrote an article with the text 'virus flu' instead of 'flu virus'.

ferences in polarisation were identified by differences in the usage and variations of tone and sentiment (Vosoughi et al., 2018). By exploiting such traits, one could potentially build a language-based model to detect misinformation.

Nevertheless, the before-mentioned characteristics specific to misinformation in online social media pose essential challenges. First, deceivers use of special characters in social media raises the possibility of having plenty of out-of-vocabulary (OOV) tokens. Next, difficulties in annotating social media together with the pace it is being generated, makes it challenging to have a large amount of data that can easily be used in different contexts. Together, these two characteristics may hinder the effectiveness of popular models and even state-of-the-art neural language techniques. The former would struggle to learn in the presence of a large amount of OOV tokens. The latter may not be as useful as pre-trained, context-aware embeddings may tussle in the presence of rapidly changing contexts. As such, we argue that character level inputs passed to a CNN are particularly well-suited for the task due to its ability to better handle morphologies and syntactic differences related to language leakages in misinformation. To test this hypothesis, we compare the performance of character CNN inputs to different others that (1) do not take into consideration morphologies, (2) "partially" take into considerations morphologies, and (3) take into consideration morphologies. Specifically, we test the following hypotheses:

A bi-LSTM using character-level inputs passed through a CNN achieves higher accuracy and F1 scores than:

1. A bi-LSTM using randomly initialised word embeddings.
2. A bi-LSTM using fasttext embeddings.
3. A bi-LSTM using character-level embeddings randomly initialised.

For our experiments, we use a set of tweets related to the presidential election in the United States in 2016. Tweets were annotated as misinformation or regular information.

The main contributions of this article are to underscore the fact that specific characteristics of misinformation in online social media, together with the difficulty of assessing what constitutes misinformation make models that exploit morphological differences particularly well-suited for the task of detecting misinformation, to assess the effectiveness of different types of inputs to handle morphologies and to put forward a dataset for assessing misinformation.

The next two sections describe the data and experiments; Section 4 discusses our results, and Section 5 concludes the work summarising the main findings and pointing towards future work.

2 Materials and methods

2.1 Data

We used a dataset containing tweets related to the 2016 presidential election in the United States. The sample was collected between the months of November 2016 and March 2017 using Twitter's publicly available API and the following search terms: `#MyVote2016`, `#ElectionDay`, `#electionnight`, `@realDonaldTrump` and `@HillaryClinton`. This collection yielded 57,379,672 tweets. To make the data manageable, we removed retweets and kept only original tweets that achieved more than a 1000 retweets. These transformations reduced the data to 9001 tweets, which were manually labelled as *misinformation* if its text could be considered within any of the categories described in Rubin et al. (2015) and *regular information* otherwise. This final dataset contained 1729 tweets labelled as misinformation and 7272 others labelled as regular information[3]. We considered only the textual field of each of the tweets for our study.

Previous analyses of this dataset found there are differences in the way misinformation and regular information are crafted (Amador et al., 2017). As an example, Table 1 from Amador et al. (2017) presents differences in the way strings are built for each category for the textual field of the tweets. Specifically, Table 1 shows differences in usage of digits, capitalization, exclamations and non-standard characters.

Statistical significant differences between misinformation and others underscore the presence of language leakages within the category misinformation. In other words, if language usage within the category misinformation did not present leakages, one would expect morphologies between the two categories to be the same. This result should allow models that consider morphologies more able to identify misinformation. The choice of

[3]The dataset can be found at `https://zenodo.org/record/1048820.XT9abJNKjOQ`

	p-value	t-stat	Mean Misinformation	Mean Other
Digits	8.02e-09	0.084	2.676%	2.104%
Caps	1.41e-08	0.082	14.016%	12.665%
Exclamations	4.40e-03	0.047	0.222%	0.309%
Non-standard	1.00e+00	0.008	0.458%	0.447%

Table 1: Features about the text of tweets. Rows are ordered by statistical significance; significant variables are above the line.

character-level inputs passed through a CNN was based on results in Table 1 as we expect convolutional layers within the bi-LSTM character CNN operate on each of the characters to achieve invariance, thus making the architecture robust to different morphologies.

The dataset was split into training (70%), development (20%) and test (10%) sets. Moreover, the data was balanced using random undersampling. Out of the balanced training set, two other sets were created. The first one contained indices for every word in every tweet in the training set; we call this the *word-level set*. The second one contained indices for every character in every tweet in the training set, and we will refer to it as the *character-level set*. For the word-level training set, we considered only the first 30 words within a tweet (as this was the average number of words within the training set) and padded with zeros. For the character-level training set, we considered 120 characters within the tweet and padded with zeros[4].

2.2 Models

Character Convolutional Neural Networks (character CNNs) (Kim et al., 2016) are networks consisting of input, output and multiple hidden layers. Specifically, the character CNN relies on character-level inputs and employs a convolutional neural network and pooling over characters that are then passed to a Recurrent Neural Network language model. Outputs can vary depending on the task at hand. Kim et al. (2016) showed that character CNNs are particularly well-suited at building representations for morphologies and syntax.

[4]Considering the first 30 words for word embeddings produces a vocabulary of 26147 tokens, whereas considering all characters produces 584 tokens. Using every word in the case of word embeddings would overly-increase the vocabulary and also could potentially lead to overfitting in the case of the model using word embeddings randomly initialised.

Recurrent Neural Networks (RNNs) are networks aimed at processing sequential data by computing a result from an input sample *and* previous states. Usually, RNNs process and input sequence to produce an output sequence. Within RNNs, Long Short Term Memory (LSTM) networks (Horchreiter and Schmidhuber, 1997) are particularly suited to deal with sequences. Formally, an LSTM unit is composed of input, output and forget gate. The cell acts as memory, capable of remembering values over arbitrary time intervals. The LSTM cell combines values from the input gate and its memory, through its forget gate, to calculate output values.

2.3 Implementation

We used Keras (Chollet et al., 2015) with Tensorflow (Abadi et al., 2015) backend to perform our experiments. Finally, to calculate test statistics, we used SKLearn (Pedregosa et al., 2011).

3 Experiments

Our main objective is to test whether a model using character-level inputs passed through a CNN similar to the one presented by Kim et al. (2016) achieves higher accuracies and F1-scores than models that: (1) do not consider morphologies, (2) consider them partially and (3) consider them in full. Specifically, we compare the performance of the following model:

bi-LSTM using character CNN inputs: Character embeddings randomly initialised + 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 + 2 bi-LSTM layers with 64 hidden units + dense layer using sigmoid activation

to the following models:

1. bi-LSTM using word embeddings: Word embeddings randomly initialised + 2 bi-LSTM layers with 64 hidden units + dense layer using sigmoid activation

2. bi-LSTM using word embeddings: Word embeddings initialised with fasttext + 2 bi-LSTM layers with 64 hidden units + dense layer using sigmoid activation

3. bi-LSTM using character embeddings: Character embeddings randomly initialised + 2 bi-LSTM layers with 64 hidden units + dense layer using sigmoid activation

For networks using character-level embeddings –character CNN and network (3)– we used embeddings of size 4. For networks using word-embeddings – networks (1) and (2), we used embeddings of size 300. Network (2) was initialised with the use of fasttext (Bojanowski et al., 2016) but, given the small size of the dataset, the embedding layers were not trained. All networks were trained using RMSprop, but networks using character-level embeddings used gradient clipping as they were learning longer sequences. All networks were trained for 20 epochs and hyperparameters together with regularisation using dropout were tuned using the development set.

It is essential to notice that the network using character CNN should be considered as belonging to the group of networks that fully considers morphologies. Convolutional layers within this architecture can be used as embeddings (Kim et al., 2016), making it similar to the network (3).

4 Discussion

Table 2 confirms our hypothesis; i.e., an LSTM character CNN out-performs other architectures when considering the best performing run. Moreover, when taking the mean of all 20 runs for each of the models, it is possible to see that the network using character CNN inputs performs at least as well as networks (1) and (2). This result is particularly remarkable when considering that networks (1) and (2) have around 60x more parameters than the character CNN.

Cao and Rei (2016); Cherry et al. (2018) suggest that character CNNs are better equipped than the other networks to cope with morphologies. We put forward this characteristic would be useful when dealing with the particularities of misinformation in online social media. A simple, yet not definitive, way of checking our intuitions would be to compare recalls for these models. If networks were struggling with data labelled as misinformation, we would expect sensitivity for the category *misinformation* be lower than that for the category *regular information*. Table 4 presents this information.

Results from Table 4 shows that sensitivities for the category *regular information* are considerably higher than for *misinformation*. These results, together with the fact that topic distribution between categories is similar by the construction of the dataset (see: Amador et al. (2017)) strongly suggest that all networks struggle more with morphologies from data labelled as misinformation than that of data labelled as regular information.

Notice that, even if all of the models find more difficult the category *misinformation*, the bi-LSTM that uses character CNN inputs performs better than both the bi-LSTM using fasttext and word embeddings. However, these differences are not statistically significant. To understand whether the bi-LSTM using character CNN inputs was performing better, we carried out error analysis. In specific, we used the union between the development and test sets to calculate the number of tweets that were incorrectly classified by each of the networks. Then, we manually inspected 50 tweets for each of the models to see whether there appear to be any systematic errors made by the networks[5].

We found that the character CNN was incorrectly classifying 734 tweets, network (1) 1156 tweets, network (2) 1013, and network (3) 1690 tweets. The character CNN appeared to struggle the most with URLs, 32 tweets, with hashtag `#ElectionNight` 17 tweets, and with unusual characters 19 tweets. Network (1), struggles the most with tweets containing URLs, 34 tweets, followed by `@realDonaldTrump`, 24 tweets, unusual characters 22 tweets and hashtag `#ElectionNight` 7 tweets. Network (2) struggled with tweets containing URLs, 33 tweets, `@realDonaldTrump` 29 tweets, unusual characters 18 tweets, `#ElectionNight` 10 tweets and 3 tweets in other languages. Finally, network (3) struggled with tweets containing URLs, 37 tweets, `@realDonaldTrump` 31 tweets, unusual characters 9 tweets, and hashtag

[5]Time constraints did not allow us to check a more substantial sample.

Network	Train accuracy	Test accuracy	F1-score
bi-LSTM character CNN	**0.6261**	**0.65**	**0.60**
(1) bi-LSTM word-embeddings	0.5872	0.64	0.58
(2) bi-LSTM fasttext-embeddings	0.6249	0.63	0.58
(3) bi-LSTM character embeddings	0.5027	0.59	0.45

Table 2: Best of 20 runs considering test accuracy for 'character CNN' , 'LSTM word-embeddings', 'LSTM fasttext-embeddings', 'LSTM character embeddings'. In boldface, the best accuracy for 20 epochs of training.

Network	Train accuracy	Test accuracy	F1-score
bi-LSTM character CNN	**0.6231**	**0.6155**	**0.5580**
(1) bi-LSTM word-embeddings	0.5949	**0.6205**	**0.5620**
(2) bi-LSTM fasttext-embeddings	**0.6255**	**0.6200**	**0.5575**
(3) bi-LSTM character embeddings	0.5195	0.5320	0.4360

Table 3: Average accuracy (over 20 runs) for 'character CNN' , 'LSTM word-embeddings', 'LSTM fasttext-embeddings', 'LSTM character embeddings'. Differences between each and every one of the values in boldface are **not** statistically significant at the 10% level of accuracy.

	Sensitivity	
Network	Misinformation	Regular information
bi-LSTM character CNN	**0.298**	**0.873**
(1) bi-LSTM word-embeddings	**0.289**	**0.873**
(2) bi-LSTM fasttext-embeddings	0.282	**0.872**
(3) bi-LSTM character embeddings	0.177	0.848

Table 4: Recalls for 'LSTM character CNN' , 'LSTM word-embeddings', 'LSTM fasttext-embeddings', 'LSTM character embeddings'. In boldface, the highest recall for each category. Differences between values in boldface are **not** statistically significant at 10% level of accuracy.

`#ElectionNight` 9 tweets.

It is possible to see that the network (3) was better able to cope with unusual characters. However, sensitivities in Table 4 lead us to think the bi-LSTM finds it challenging to learn the sequence of characters for both categories, thus suggesting the problem with network (3) is related to a failure to learn the sequence. The character CNN and network (2) cope similarly with unusual characters. This result suggests that fasttext embeddings serve their purpose of dealing with different morphologies. Nevertheless, it is essential to underscore that the model using character CNN obtains similar results than fasttext with 60x fewer parameters. Finally, network (1) struggles the most with unusual characters. This is most likely because tokens containing such characters are OOV. Concerning `@realDonaldTrump` it is possible to see that all networks but the one using character CNN struggled with this mention. However, the real effect of the latter is more difficult to interpret as the failure to categorise these tweets maybe since `@realDonaldTrump` is an OOV token or may be due to sentiment. Even so, it is essential to underline that the model using character CNN did not struggle with `@realDonaldTrump`. Finally, incorrect classification of tweets containing hashtag `#ElectionNight` was caused by sentiment ambiguities in all of the cases.

Even if there is a case to be made for the results to be driven by morphologies, error analysis suggests that existing differences in sentiment may be behind the result as well. We believe this may not be the case as if the sentiment was the main driver behind the performance, we would expect network (1) to perform better than the network using character CNN. Despite this, exploring representations of the character CNN and network (1) should be useful to understand what drives this result entirely and is left as further research. In the case of the network (3), it is easy to see that this network struggles with the data regardless of its category.

The poor performance of the network (3) is particularly puzzling as we would expect the model to be able to cope with sub-strings of text as long

as the string is not entirely random; e.g., the case of regular information. It is possible that the bi-LSTM is not able to cope with the length of character strings and is, hence, unable to build good representations even for regular information. Attention mechanisms may help in alleviating this problem. However, implementation of attention may render moot the use of a bi-LSTM coupled with character-level embeddings and, hence, not further explored.

Even if it is out of the scope of this paper, it is important to discuss how SOTA models such as BERT or GPT would perform. We expect these models to struggle with misinformation from online social media because their upper layers would find it challenging to deal with syntax or meaning[6]. However, lower-level layers dealing with morphologies could prove useful in detecting differences in the usage of characters between misinformation and any other type of information. Many SOTA language models tap into new research related to character-level embeddings (Kim et al., 2015; Zhai et al., 2018) and CNNs in the context of NLP to deal with morphologies. Moreover, both character-level embeddings and CNNs have been proven useful in cases where not many data is available (Zhai et al., 2018). Furthermore, even if we expect SOTA models to struggle with fast-changing contexts of misinformation, the limitations of our dataset would not allow us to make a fair comparison between models. This is because the dataset is drawn from the 2016 presidential election in the US and, most likely, has already assimilated in the context of these models, thus rendering our criticism invalid.

All-in-all, the network using character CNN inputs can produce robust results for the task of identifying misinformation in online social media. As discussed above, we believe its performance is due to its ability to cope with morphologies. Networks using character CNNs have proven a robust alternative when dealing with morphologies in different tasks (Cao and Rei, 2016; Cherry et al., 2018), and we believe this ability is of particular importance for the task at hand. Our analysis showed that networks considering morphologies were better able to categorise misinformation. Furthermore, our results put forward marginal advantages in performance for the networks using character CNN inputs. It is important to notice that such advantages were not statistically significant. Nevertheless, our results suggest further studying and using networks with character CNN inputs for detecting misinformation is a promising avenue for future research.

[6]It suffices to remember "COVFEFE".

5 Conclusion

We studied the effectiveness of a character CNN architecture for detecting misinformation in online social media. There has been a long line of research studying and exploiting language leakages deceivers fall into to detect misinformations (Feng and Hirst, 2013). For the dataset used, it has been shown there are syntactic differences between misinformation and other types of information (Amador et al., 2017), which raises the possibility this could be present more broadly in online social media. We argue that particularities of the way misinformation is created in online social networks make character CNN inputs good candidates to be used for identifying misinformation in online social media.

We compared the character CNN inputs to inputs that disregard morphologies and to others that consider them through the use of embeddings such as fasttext. We show that the best performing run of the character CNN out-performs the best performing runs of networks using fasttext. We try to understand if character CNNs are better able to cope with morphologies. Even if we can find some evidence supporting this hypothesis, our findings are not conclusive and deserve further exploration. Given the contentious nature of misinformation, the speed and format at which misinformation is created, and the lack of availability of training data, exploiting language leakages of the deceiver and further studying morphologies appears as a promising avenue of research.

Moreover, it is essential to emphasize that finding easy-to-implement, light networks such as character CNN is essential. As interest in detecting misinformation outgrows computer science, having effective alternatives that can be implemented with off-the-shelf tools becomes more critical.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp,

Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Julio Amador, Axel Oehmichen, and Miguel Molina-Solana. 2017. Characterizing Political Fake News in Twitter by its Meta-Data.

P. Bojanowski, E. Grave, A. Jouin, and T. Mikolov. 2016. Enriching Word Vectors with Subword Information.

K. Cao and M. Rei. 2016. A Joint Model for Word Embedding and Word Morphology.

K. Cherry, G. Foster, A. Bapna, O. Firat, and W. Macherey. 2018. Revisiting Character-Based Neural Machine Translation with Capacity and Compression.

François Chollet et al. 2015. Keras. `https://keras.io`.

Vanessa Wei Feng and Graeme Hirst. 2013. Detecting deceptive opinions with profile compatibility. In *Proc. 6th International Joint Conference on Natural Language Processing (IJCNLP)*, pages 14–18.

S. Horchreiter and J. Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Y. Kim, Y. Jernite, D. Sontag, and A. Rush. 2016. Character-Aware Neural Language Models.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. 2015. Character-Aware Neural Language Models.

Federico Monti, Fabrizio Frasca, Davide Eynard, Damon Mannion, and Michael M. Bronstein. 2019. Fake News Detection on Social Media using Geometric Deep Learning.

Arjun Mukherjee, Vivek Venkataraman, Bing Liu, and Natalie Glance. 2013. Fake Review Detection: Classification and Analysis of Real and Pseudo Reviews. Technical Report UIC-CS-2013-03, University of Illinois at Chicago.

Brendan Nyhan and Jason Reifler. 2010. When Corrections Fail: The Persistence of Political Misperceptions. *Political Behavior*, 32(2):303–330.

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*, 12:2825–2830.

Gordon Pennycook, Tyrone D. Cannon, and David G. Rand. 2017. Prior Exposure Increases Perceived Accuracy of Fake News.

Gordon Pennycook and David G. Rand. 2017. Who Falls for Fake News? The Roles of Analytic Thinking, Motivated Reasoning, Political Ideology, and Bullshit Receptivity.

Danielle C. Polage. 2012. Making up History: False Memories of Fake News Stories. *Europe's Journal of Psychology*, 8(2):245–250.

Reddy Ranabothu Nithin and Nitesh Kumar. 2012. Automatic Detection of Fake Profiles in Online Social Networks. Technical report, National Institute of Technology Rourkela.

Victoria Rubin. 2017. Deception Detection and Rumor Debunking for Social Media. *FIMS Publications*.

Victoria L. Rubin, Yimin Chen, and Niall J. Conroy. 2015. Deception detection for news: Three types of fakes. *Proc. Association for Information Science and Technology*, 52(1):1–4.

Briony Swire, Adam J. Berinsky, Stephan Lewandowsky, and Ullrich K. H. Ecker. 2017. Processing political misinformation: comprehending the Trump phenomenon. *Royal Society Open Science*, 4(3):160802.

Catalina L. Toma and Jeffrey T. Hancock. 2012. What Lies Beneath: The Linguistic Traces of Deception in Online Dating Profiles. *Journal of Communication*, 62(1):78–97.

Soroush Vosoughi, Deb Roy, and Sinan Aral. 2018. The spread of true and false news online. *Science*, 359(6380):1146–1151.

Zenan Zhai, Dat Quoc Nguyen, and Karin Verspoor. 2018. Comparing CNN and LSTM character-level embeddings in BiLSTM-CRF models for chemical and disease named entity recognition.