Detecting Fake News Spreaders on Twitter Using Universal Sentence Encoder

Notebook for PAN at CLEF 2020

Soumayan Bandhu Majumder, Dipankar Das
Department of Computer Science & Engineering
Jadavpur University
{soumayanmajumder,dipankar.dipnil2005}@gmail.com

Abstract. In the present attempt, we have developed a framework to detect the fake news spreaders on twitter by utilizing their tweets. Here, we have employed the pre-trained sentence embedding of Google and fed this embedding to a Long Short Term Memory (LSTM) based deep learning framework. Finally, the embedding is passed through an attention layer and predicts whether an author is prone to spread fake news or not. We have built models for two languages – English and Spanish. We have achieved 72% accuracy in this fake news spreader detection task.

1 Introduction

In the present work, we have developed techniques to detect profiles of fake news spreaders. Fake news detection is becoming one of the challenging tasks of recent years. However, in this present task, instead of detecting fake news, we are concentrating on detecting it at user level. Thus, in order to accomplish our goals, we handle tweets at chunk level instead of handling each tweet, because here we have to detect the author of fake news, not the news type.

We participated in the profiling fake news spreader shared task [14] under PAN workshop and the organizers provided us the dataset. We have used a pre-trained word embedding and fed it into the LSTM with attention based deep learning framework to achieve the desired output. The systems were hosted and evaluated on TIRA [15], a web service that aims to facilitate software submissions and evaluations for shared tasks. Here we first use the universal sentence encoder of Google for converting texts to embedding then pass each authors tweet one by one into the LSTM network, here we will send tweets of each author in different time stamps one by one. After that send this output to attention layer to know the importance of each tweet and at last pass it through the sigmoid activation layer.

The rest of the paper is organized as follows. Related work on this particular topic is discussed in Section 2 whereas Section 3 briefly shows the insights of the datasets. Section 4 describes the method we used to detect the fake author and also describes our models and proposed architecture in depth. Section 5 is dedicated to experiments and results. Finally, in Section 6, we present the conclusions and briefly discuss about future work.

2 Related Work

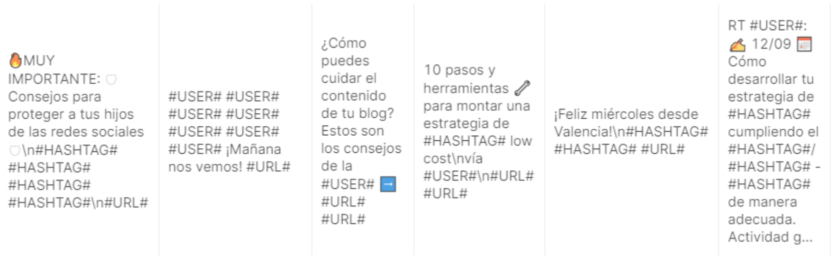
Fake news is currently one of the hottest topics of last four to five years and many researches are being conducted in this field. Some of the researchers suggest solving the fake news detection problem using the content of the news and some suggest detecting it based on the social context. Therefore, we can detect fake news in two ways – a) news content model, and b) social context model. In case of news content model, we can detect fake news in two ways – a) knowledge based detection, and b) style based detection. In knowledge based detection, we generally check the news content or extract the knowledge from the news or other repositories and compare it with the authentic news sites whereas in case of style based detection, we focus on the writing styles instead of the news content or knowledge. Here, we mainly focus on the linguistic features and readability features of the news. In social content model, we can detect fake news also in two ways – a) propagation based techniques, and b) credibility based techniques. In propagation based techniques, we find various news propagated on the social media and track the original news based on such propagated news. In credibility based technique, we have to find the relationship between news article and users, publishers, posts, comments etc.

George et. al. [1] analyse the influence of linguistic properties and contextual features in detecting fake news by using different type of techniques like Naïve Bayes, SVM, KNN etc. Perez-Rosas et. al. [2] here cover seven different types of news domains and analyse linguistic differences in fake and real news and also compare different domain characteristics. Bedi et. al. [4] use authorized news database to verify fake news and real news. Dey et. al. [5] do feature extraction and analyse the linguistic patterns and then apply KNN algorithm to classify news. Uppal et. al. [7] propose discourse level analysis for deception detection of news documents.

However, one problem with these above mentioned techniques is that they detect the fake news after it spreads on social media. But, if we can detect it from the source, this problem can be avoided. So, one of our aims should be to detect the author of the fake news. Already, there are some works on fake news spreader detection [10]. But, it is applied only in English language. So, in this paper, we are going to implement it in both English and Spanish languages and we have used some different embedding techniques instead of GLOVE embedding.

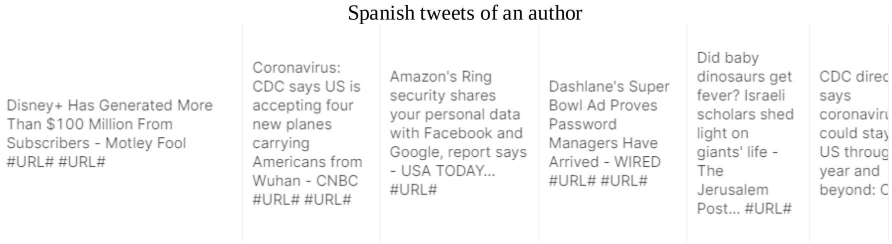
3 Dataset

We participated in the profiling fake news spreader shared task of the PAN workshop at the CLEF (Conference and Labs of the Evaluation Forum) 2020 conference. In this task, the dataset was provided by the shared task organizers. We are given with hundred tweets of each author and a total of 300 authors tweets have been given to us for training. For each English and Spanish language, we are given 300 authors tweet (100 tweets for each author) to train the model. For evaluation, we have to submit our software to the TIRA infrastructure and after that we have to execute it on the test dataset. We have given snapshots of some tweets of both English and Spanish language from my training set.



🔥MUY IMPORTANTE: 🛡 Consejos para proteger a tus hijos de las redes sociales 🛡\n#HASHTAG# #HASHTAG# #HASHTAG# #HASHTAG#\n#URL#	#USER# #USER# #USER# #USER# #USER# #USER# #USER# ¡Mañana nos vemos! #URL#	¿Cómo puedes cuidar el contenido de tu blog? Estos son los consejos de la #USER# ➡ #URL# #URL#	10 pasos y herramientas 🖊 para montar una estrategia de #HASHTAG# low cost\nvía #USER#\n#URL# #URL#	¡Feliz miércoles desde Valencia!\n#HASHTAG# #HASHTAG# #URL#	RT #USER#: ✍ 12/09 📅 Cómo desarrollar tu estrategia de #HASHTAG# cumpliendo el #HASHTAG#/ #HASHTAG# - #HASHTAG# de manera adecuada. Actividad g...

Spanish tweets of an author



Disney+ Has Generated More Than $100 Million From Subscribers - Motley Fool #URL# #URL#	Coronavirus: CDC says US is accepting four new planes carrying Americans from Wuhan - CNBC #URL# #URL#	Amazon's Ring security shares your personal data with Facebook and Google, report says - USA TODAY... #URL#	Dashlane's Super Bowl Ad Proves Password Managers Have Arrived - WIRED #URL# #URL#	Did baby dinosaurs get fever? Israeli scholars shed light on giants' life - The Jerusalem Post... #URL#	CDC direc says coronaviru could stay US throug year and beyond: C

English tweets of an author

4 Methods

***<u>Pre-processing</u>*:** Here, we first tried to conduct some text pre-processing techniques on the tweets by removing everything except letters from '*a'*- '*z'* and '*A'* – '*Z'*. We have also removed all the *urls* and *html* tags or elements present in the tweets. Finally, we removed all types of *emoji* or emoticons present in the tweets. We send this pre-processed data to a universal-sentence-encoder. But, we observed that the universal-sentence-encoder performs better if we send those tweets in raw form rather than pre-processed form. Thus, we choose to use those tweets without pre-processing.

<u>LSTM Framework:</u> Here, we have implemented embedding of our tweets and then passed it through the LSTM layer and then attention layer and predicted the output lastly through the sigmoid activation function. Here, we have used sentence embedding, instead of word embedding. We have employed **universal-sentence-encoder-xling_en_es_1** (This is a cross lingual module and an extension of the normal universal sentence encoder). We also tried with universal-sentence-encoder-multilingual-large-v3, but as this shared task is specifically for English and Spanish language, we choose universal-sentence-encoder-xling_en_es_1 (which is specifically trained for English and Spanish language) which can handle 16 languages including English and Spanish. We named it as Universal Sentence Encoder or U.S.E., interchangeably.

One feature of this universal sentence encoder is that it always gives output of a 512 dimension vector whatever be the input is. Here, we are given with the tweets of an author as the input to the U.S.E. and it produces 512 dimension vector. Thus, for each of the authors, we are getting an output of (100 x 512) dimensional vector output, as we have 100 tweets of each author. As we have 300 such authors in both English and Spanish languages for training our model, we developed a LSTM network with 128 units. The overall framework takes 512 features at a time and contains 100 such timestamps. In order to avoid over fitting, we have also used dropout of 0.3 and recurrent dropout of 0.2. We tried with different hyper-parameters but these two gives the best result. We also tried with different batch-sizes such as 16, 32 and 64 but we choose 64 among these.

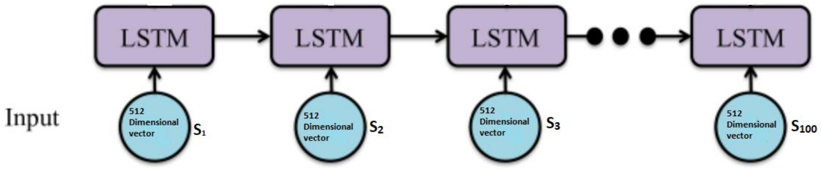


Figure 1: How we send data to LSTM network

In the present framework, we tried to train both the languages differently and then used to train both the language jointly. At first, we used any of these two sentence embedding – 1) U.S.E. multilingual large-v3 (version 3) and 2) U.S.E. xling_en_es-1. After that, we feed those outputs to the LSTM network. Because, we aimed to capture long term dependency through LSTM. However, in some cases, it is forgetful and does not know which input should be given more or less importance.

For example (from twitter) **-** "*Breaking News: Tom Hanks and his wife Rita Wilson announced Wednesday that they have tested positive for the coronavirus*". Here, to predict the word "*coronavirus*", we should give more emphasis to "*tested*" and "*positive*" instead of "*Tom Hanks*" or "*Rita Wilson*". So, here we have to give relative importance to each of the words instead of giving them same importance. Therefore, we have applied the attention model which is presented in the Bahdanu's paper [8]. But here we are applying this in sentence level instead of word level. Because here some news or tweets of an author are fake and some are real. Finally, we feed that output of the attention layer to the sigmoid activation unit. We then compile our model using *adam* optimizer and used *binary_crossentropy* as loss function. As both the classes are balanced, we considered accuracy as our evaluation metric to measure the performances of our models.

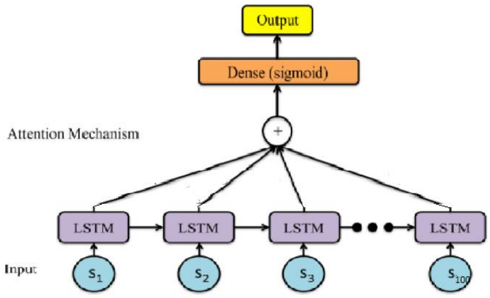


Figure 2. Architecture of our model, Here each $\mathbf{S}_i$ is a 512 dimensional vector.

5 Experiments and Results

We implemented the above mentioned models with universal sentence encoder multilingual large and universal sentence encoder-xling-en_es-1. We also tried to train both the languages separately and jointly with different batch sizes. But, as we mentioned earlier, we don't have access of test set and we have to submit our model to TIRA infrastructure. Then it will automatically do evaluation of our model on blinded test set. The performance of our system is measured by accuracy as the two classes are balanced. Here, we have to measure individual accuracies of each language and then finally average the accuracy values of each language to obtain the final accuracy.

Encoder	**Batch size**	**Number of Train data**	**Number of validation data**	**Accuracy (in %)**
U.S.E. multilingual	16	480	120	77.50
U.S.E. multilingual	32	480	120	74.90
U.S.E. multilingual	64	480	120	74.75
U.S.E. xling en_es-1	16	480	120	80.83
U.S.E. xling en_es-1	32	480	120	77.50
U.S.E. xling en_es-1	64	480	120	81.50

Table 1: Accuracy of each model with different encoder and different batch size trained on both languages

Therefore, for experimenting with different models or architecture, we split our dataset into training set and evaluation set. It was observed that when we train both the language jointly, we split data in the ratio of 8:2. So, 480 authors feeds (each author has 100 tweets) were used for training and 120 authors data were used for validation (check the above mentioned table). From the above table we are getting maximum 81.50% accuracy. In case we train each language differently, we have used 270 authors data for training and 30 authors data for validation (check the below mentioned table).

Encoder	**Language**	**Batch size**	**Number of train data**	**Number of validation data**	**Accuracy (in %)**
U.S.E. multilingual	English	16	270	30	63.33
U.S.E. multilingual	Spanish	16	270	30	80.00

Table 2: Accuracy of each model trained on two languages separately

On the other hand, if we do average of the above two validation accuracies, we achieved around 71% accuracy. Thus, finally, we choose universal sentence encoder xling-en_es-1 trained on both the languages with batch size of 64 over the other models as its validation accuracy is 81.50%. When we submit our final model which uses Universal Sentence Encoder xling-en_es-1 and trained on both the languages jointly with the batch size of 64 to TIRA for evaluation on hidden test set, we get an accuracy of 64% in English language and 80% accuracy on Spanish language, respectively.

6 Conclusions

We achieved 64% accuracy in English language fake author detection task, which says that there is a scope of lot to improve and there is some issue of overfitting too though we used dropout and recurrent dropout at the time of training the model. Thus, in future, we should take this note to improve this model. If we want to use this software in real life then how will it perform much depends on how the test dataset of shared task of PAN reflects the real world dataset. We achieve 80% accuracy in Spanish language author detection task, which is quite satisfactory, but again if we want to implement this software in real world, then its performance totally depends on how much test dataset of PAN shared task reflects the real world dataset.

We can also use different types of embedding in future for this task like BERT, but main disadvantage of BERT is that it can take maximum 512 words at a time, so there is a constraint. However, we will plan some reliable frameworks to handle the issues.

7 References

[1] George, J., Skariah, S., & Aleena Xavier, T. (2020). Role of Contextual Features in Fake News Detection: A Review. *2020 International Conference On Innovative Trends In Information Technology (ICITIIT)*. doi: 10.1109/icitiit49094.2020.9071524

[2] P´erez-Rosas, V., Kleinberg, B., Lefevre, A., & Mihalcea, R. (2018). Automatic Detection of Fake News. In *Proceedings of the 27th International Conference on Computational Linguistics* (pp. 3391–3401). Santa Fe, New Mexico, USA: Association for Computational Linguistics.

[3] Qi, P., Cao, J., Yang, T., Guo, J., & Li, J. (2019). Exploiting Multi-domain Visual Information for Fake News Detection. *2019 IEEE International Conference On Data Mining (ICDM)*. doi: 10.1109/icdm.2019.00062

[4] Bedi, A., Pandey, N., & Khatri, S. (2019). A Framework to Identify and secure the Issues of Fake News and Rumours in Social Networking. *2019 2Nd International Conference On Power Energy, Environment And Intelligent Control (PEEIC)*. doi: 10.1109/peeic47157.2019.8976800

[5] Dey, A., Rafi, R., Hasan Parash, S., Arko, S., & Chakrabarty, A. (2018). Fake News Pattern Recognition using Linguistic Analysis. *2018 Joint 7Th International Conference On Informatics, Electronics & Vision (ICIEV) And 2018 2Nd International Conference On Imaging, Vision & Pattern Recognition (Icivpr)*. doi: 10.1109/iciev.2018.8641018

[6] Rajesh, K., Kumar, A., & Kadu, R. (2019). Fraudulent News Detection using Machine Learning Approaches. *2019 Global Conference For Advancement In Technology (GCAT)*. doi: 10.1109/gcat47503.2019.8978436

[7] Uppal, A., Sachdeva, V., & Sharma, S. (2020). Fake news detection using discourse segment structure analysis. *2020 10Th International Conference On Cloud Computing, Data Science & Engineering (Confluence)*. doi: 10.1109/confluence47617.2020.9058106

[8] Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural Machine Translation by Jointly Learning to Align and Translate.

[9] Karimi, H., Roy, P., Saba-Sadiya, S., & Tang, J. (2018). Multi-Source Multi-Class Fake News Detection. In *Proceedings of the 27th International Conference on Computational Linguistics* (pp. 1546–1557). Santa Fe, New Mexico, USA: Association for Computational Linguistics.

[10] Giachanou, A., Ríssola, E. A., Ghanem, B., Crestani, F., & Rosso, P. (2020). The Role of Personality and Linguistic Patterns in Discriminating Between Fake News Spreaders and Fact Checkers. *Natural Language Processing and Information Systems: 25th International Conference on Applications of Natural Language to Information Systems, NLDB 2020, Saarbrücken, Germany, June 24–26, 2020, Proceedings*, *12089*, 181–192. https://doi.org/10.1007/978-3-030-51310-8_17.

[11] Zhang, G., Davoodnia, V., Sepas-Moghaddam, A., Zhang, Y., & Etemad, A. (2020). Classification of Hand Movements From EEG Using a Deep Attention-Based LSTM Network. *IEEE Sensors Journal*, *20*(6), 3113-3122. doi: 10.1109/jsen.2019.2956998

[12] Kosmajac, D., & Keselj, V. (2019). Twitter User Profiling: Bot and Gender Identification. In *CLEF 2019 Labs and Workshops, Notebook Papers*. CEUR-WS.org.

[13] Przybyła, P. (2019). Detecting Bot Accounts on Twitter by Measuring Message Predictability. In *CLEF 2019 Labs and Workshops, Notebook Papers*. CEUR-WS.org.

[14] Rangel F., Giachanou A., Ghanem B., & Rosso P. (2020). Overview of the 8th Author Profiling Task at PAN 2020: Profiling Fake News Spreaders on Twitter. In: L. Cappellato, C. Eickhoff, N. Ferro, and A. Névéol (eds*.) CLEF 2020 Labs and Workshops, Notebook Papers.* CEUR Workshop Proceedings.CEUR-WS.org.

[15] Potthast, M., Gollub, T., Wiegmann, M., & Stein, B. (2019). TIRA Integrated Research Architecture. In: Ferro, N., Peters, C. (eds.) *Information Retrieval Evaluation in a Changing World - Lessons Learned from 20 Years of CLEF.* Springer.