CIKM AnalytiCup 2020: COVID-19 Retweet Prediction with Personalized Attention

T Vinayaka Raj
Rakuten Inc.
vinayaka.raj@rakuten.com

ABSTRACT

This paper describes the first place winning solution for the CIKM AnalytiCup 2020 COVID-19 retweet prediction challenge. The objective of the challenge is to predict the popularity of COVID-19 related tweets in terms of the number of retweets, and the submitted solutions of the challenge are ranked based on Mean Squared Logarithmic Error(MSLE) on the leaderboard. The proposed deep learning model to predict retweet counts uses minimal hand-engineered features and learns to predict retweet count based on a personalized attention mechanism. As a tweet keyword may have different informativeness for different users, the personalized attention mechanism helps the deep learning model to weigh the importance of tweet keywords based on a user's interest to retweet. Additional techniques such as adding external data sets to training and pseudo-labeling are also experimented with to further improve the MSLE score. The final solution comprises of an ensemble of different personalized attention-based deep learning models, and the source code for the solution can be found at https://github.com/vinayakaraj-t/CIKM2020.

KEYWORDS

Deep Learning, Personalized attention, COVID-19, Pseudo-labelling

1 INTRODUCTION

Understanding information diffusion in social networks is imperative as it helps to comprehend social interactions among users in a better way. Information spread on a large scale in social networks enables marketers, advertisers to design their campaigns more effectively to target potential customers. In addition to that, identifying influential users [8] in social media is also significant as these users contribute immensely to information diffusion during viral marketing campaigns. Relationships between users on social networks heavily affect the amount of information exchanged among themselves. Furthermore, understanding how fake news spreads in social networks is also crucial to prevent the propagation of misinformation during global pandemics such as COVID-19.

Modeling information diffusion in social networks is a hot research topic that has garnered more interest in research communities of late. In CIKM AnalytiCup 2020, the competition objective is to model the information spreading mechanism during COVID-19 by predicting the retweet count of tweets on Twitter. Retweeting a tweet is one of the functions of Twitter that helps users to quickly share their tweets or tweets of other users to all of their followers. Retweets can be seen as one of the ways information spreads on Twitter and are very crucial to understand the information diffusion mechanism on Twitter. Some of the practical applications of information diffusion using tweets are political audience design [9, 15], fake news spreading and tracking [10, 17] and health promotion [3].

In this paper, all the techniques used to predict the retweet count of tweets are discussed in detail. The first section provides a summary of the dataset presented to solve the problem. Hand-engineering new features and their pre-processing techniques are also explained in this section. Model architecture and the personalized attention mechanism are described in the next section, and finally, in the last section, all the experiments carried out to improve the model score on the test leaderboard are explained in detail.

2 DATASET

TweetsCOV19 [5] dataset provided in the competition is a large collection of COVID-19 related tweets that are extracted using a seed list of 268 COVID-19 related keywords [4] from a large corpus of anonymized and annotated TweetsKB [6] corpus. TweetsCOV19 contains all the COVID-19 related tweets from October 2019 to April 2020 and the total number of tweets in the dataset is around 8 million that are posted by 3.7 million users. For each tweet, the user who tweeted that, time of the tweet, metadata information such as #followers, #favorites and #Friends and text information of tweets are provided in the dataset. Text information of tweets is split into entities, hashtags, mentions, and URLs. Entities of each tweet are created using Fast Entity Linker [1, 11]. The sentiment of each tweet is also provided and is extracted using SentiStrength [16] which scores each tweet between -4(very negative) to 4(very positive).

In addition to the given metadata features, few more features are derived from the given tweet metadata information and used to predict the retweet count. A full list of features and their preprocessing techniques is provided in Table 1.

Both original tweet keywords and their respective annotated entities are extracted and considered for analysis. Numbers and special characters are removed from hashtags and mentions, and duplicate keywords are removed from entities, hashtags, and mentions. URLs are split into two parts. The hostname of the tweet URL is extracted as URL-1, and the path of the URL is considered as URL-2. Besides that, numbers and special characters are removed from URL-2.

3 MODEL ARCHITECTURE

Figure 1 shows the network architecture of the retweet prediction model and the attention mechanism. High cardinal feature such

In: Dimitar Dimitrov, Xiaofei Zhu (eds.): Proceedings of the CIKM AnalytiCup 2020, 22 October, 2020, Gawlay (Virtual Event), Ireland, 2020, published at http://ceur-ws.org.

Table 1: Feature Information and Preprocessing Techniques

Feature	Description	Preprocessing Technique
week	Week info extracted from timestamp	One-hot-encoded
time	Time info extracted from timestamp	Log transformed and then standardized
year	Year info extracted from timestamp	Log transformed and then standardized
no_entities	No. of entities in a tweet	Log transformed and then standardized
keyword_entities	No. of COVID related entities in a tweet	Log transformed and then standardized
no_hashtags	No. of hashtags in a tweet	Log transformed and then standardized
keyword_hashtags	No. of COVID related hashtags in a tweet	Log transformed and then standardized
no_mentions	No. of mentions in a tweet	Log transformed and then standardized
no_urls	No. of urls in a tweet	Log transformed and then standardized
Sentiment	Sentiment score from SentiStrength (-4 to 4)	One-hot-encoded
#Favorites	Tweet favorites	Log transformed and then standardized
#Followers	No. of followers of an user	Log transformed and then standardized
#Friends	No. of friends of an user	Log transformed and then standardized
#Followers/#Friends	No. of followers and friends ratio	Log transformed and then standardized
#Friends/#Favorites	No. of friends and favorites ratio	Log transformed and then standardized
#Favorites/#Followers	No. of favorites and followers ratio	Log transformed and then standardized
username	Encrypted username	Label encoded

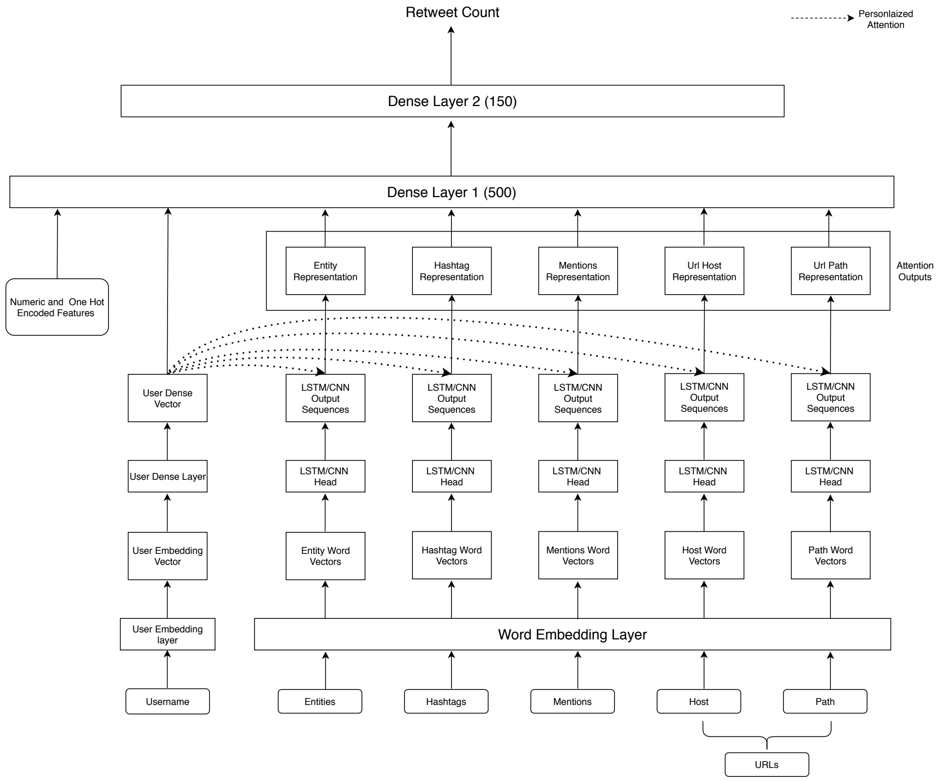

Figure 1: Architecture of the deep learning model to predict retweet counts.

as username is embedded as a fixed-length vector using user embedding layer, which is then passed through a series of user dense layers to get the final representation of users.

The word embedding layer is shared by the preprocessed keywords of tweet entities, mentions, hashtags, and URLs and is initialized by any pre-trained word vectors. For a tweet, entity word vectors are a sequence of word vectors queried from the embedding

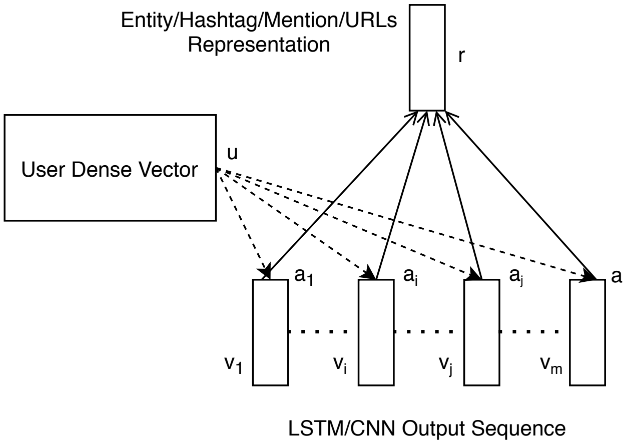

Figure 2: Personalized attention mechanism

layer and the length of the sequence is equivalent to the number of entity keywords extracted from that tweet. The length of the sequence is fixed for the entire dataset and is a hyper-parameter in the model. Entity keywords smaller than the sequence length are padded with zeros on the left, and the larger ones are trimmed on the right. Word vectors of hashtags, mentions, and URLs are created the same way as the entity word vectors. These extracted word vectors are then inputted to an LSTM/CNN layer, which is then used to learn the representation of entities, mentions, hashtags, and URLs from their respective word vector sequence.

User vector representation u and the LSTM/CNN output vectors v are used to create personalized attention mechanism [18]. Attention weight a is formulated as:

$$e_i = v_i^T * tanh(W_u * u + b_u)$$

$$a_i = \frac{exp(e_i)}{\sum_{j=1}^{m} exp(e_j)}$$

where W_u and b_u are user projection parameters and m is the sequence length. The final representation r of entities/hashtags/mentions/URLs is given by:

$$r_i = \sum_{j=1}^{m} a_j * v_j$$

The final representation vectors of entities, hashtags, mentions, and URLs are then concatenated together with the user vector and other features such as tweet metadata and time-based features. The final concatenated feature vector is then passed through a series of dense layers to estimate the retweet count.

4 EXPERIMENTS

4.1 Experiment Setting

Dataset provided for the competition comprises of all COVID 19 related tweets from 2019-09-30 to 2020-05-31. Of which, the entire month of May 2020 is considered for testing and is split into two testing sets - testing set 1 & 2. Testing set 1 is used for validating the model on the leader board and testing set 2 is used to rank the final winners of the competition. The rest of the data set from 2019-09-30 to 2020-04-30 is used for training the model. The training data set is sorted in chronological order and the very recent 5% tweets of the training data set are filtered out and used as the validation set. Information about training, validation, and testing splits are provided in Table 2.

Table 2: Dataset Splits

Data Split	Start Date	End Date
Training	2019-09-30	2020-04-25
Validation	2020-04-26	2020-04-30
Testing Set 1	2020-05-01	2020-05-15
Testing Set 2	2020-05-16	2020-05-31

Table 3: Model Setup

Optimizer	Adam
Learning Rate	0.0001
Batch Size	2048
Entity Sequence Length	10
Hashtag Sequence Length	5
Mentions Sequence Length	5
URL-1 Sequence Length	3
URL-2 Sequence Length	15
User Embedding Size	64
User Dense Layer	150
Word embedding size	150
LSTM units	150
CNN units	150
Dense layer 1	500
Dense Layer 2	150

Mean Square Logarithmic Error (MSLE) is the evaluation metric used in this competition. MSLE is given by:

$$MSLE = \frac{1}{N} \sum_{i=0}^{N} ((log(a_i + 1) - log(p_i + 1))^2$$

where a_i and p_i are the actual and predicted retweet counts respectively. MSLE penalizes under estimations more than over estimations.

The model described in Figure 1 is trained on a Tesla V100 GPU machine. The optimal hyper-parameter settings are selected based on the model with the best MSLE score on the validation set, and the tuned hyperparameters of the model setup are provided in Table 3.

4.2 Results

The performance of the models on the final testing dataset is shown in Table 4. A single personalized attention model with fast text embedding and LSTM head for learning tweet representation provides an MSLE score of 0.12860 on the test dataset. A large collection of annotated tweets for the months of April 2020 and March 2020 from the dataset corpus TweetsKB is added to the training dataset and a deep learning model is trained on the whole dataset. This addition of an external dataset decreases the MSLE score by 3.76%.

Table 4: Model Performance Results

Model Type	MSLE
Fasttext Embedding	0.12860
Fasttext Embedding + External Dataset	0.12376
Ensemble	0.12071
Ensemble + Pseudo-Labelling	0.12055

TweetsCOV19 is the subset of TweetsKB and hence doesn't include all the tweets of users but their COVID related tweets. Including all tweets of a user not only help the personalized mechanism to understand the relation between users and their tweets but also help the model to learn a rich representation of users and tweet keywords. To further improve the score, techniques such as ensembling and pseudo-labeling are also tried.

4.2.1 *Ensembling.* In addition to initialing the Twitter keywords with fasttext embeddings [2, 7], pre-trained word vectors such as glove840 [12], glovetwitter [12], fasttext wiki [2, 7] and LexVec [13, 14] are also used to train the deep learning model. Among the five models trained with different pre-trained vectors, fasttext embedding initialization provides the best score on the testing leaderboard. Furthermore, another set of models is trained by replacing the LSTM head with CNN head and also with all the five pre-trained word vectors. Individual MSLE scores of the models with CNN head are much lower than the models with LSTM heads but ensembling all the models together provide a significant improvement on the testing leaderboard. In total, there are ten personalized attention models, and the final solution is created by ensembling all the ten output predictions with simple averaging. Ensembling decreases the MSLE score by 2.464%.

4.2.2 *Pseudo-Labelling.* Pseudo-labelling is another trick tried to decrease the MSLE score. Best output solutions on the leaderboard of the test set 1 and test set 2 are used as labels for the respective data sets and are then added to the training set for building the deep learning models. Similar to the ensembling technique described above, ten different models with different pre-trained word vectors and LSTM/CNN heads are built with the new dataset and are then averaged. Pseudo-labelling decreased the MSLE score by a very small percentage of 0.132%.

5 CONCLUSION

A methodology to estimate retweet count for COVID related tweets is proposed in this paper. The personalized attention-based deep learning model described in this paper uses less hand-engineered features and learns a rich representation of users and tweet keywords to predict retweet count. To further improve the performance of the model, techniques such as adding external datasets, ensembling, and pseudo-labeling are also tried. The final solution to estimate retweet counts is created by an ensemble of deep learning models which placed the team first on the testing leaderboard.

REFERENCES

[1] Roi Blanco, Giuseppe Ottaviano, and Edgar Meij. 2015. Fast and Space-Efficient Entity Linking in Queries. In *Proceedings of the Eight ACM International Conference on Web Search and Data Mining* (Shanghai, China) *(WSDM 15)*. ACM, New York, NY, USA, 10.

[2] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching Word Vectors with Subword Information. *CoRR* abs/1607.04606 (2016). arXiv:1607.04606 http://arxiv.org/abs/1607.04606

[3] Jae Eun Chung. 2017. Retweeting in health promotion: Analysis of tweets about Breast Cancer Awareness Month. *Computers in Human Behavior* 74 (04 2017). https://doi.org/10.1016/j.chb.2017.04.025

[4] Dimitar Dimitrov. 2020 (accessed August 2020). *COVID-19 related keywords*. https://data.gesis.org/tweetscov19/keywords.txt

[5] Dimitar Dimitrov, Erdal Baran, Pavlos Fafalios, Ran Yu, Xiaofei Zhu, Matthäus Zloch, and Stefan Dietze. 2020. TweetsCOV19 - A Knowledge Base of Semantically Annotated Tweets about the COVID-19 Pandemic. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. Association for Computing Machinery, New York, NY, USA, 2991–2998. https://doi.org/10.1145/3340531.3412765

[6] Pavlos Fafalios, Vasileios Iosifidis, Eirini Ntoutsi, and Stefan Dietze. 2018. TweetsKB: A Public and Large-Scale RDF Corpus of Annotated Tweets. In *European Semantic Web Conference*. Springer, 177–190.

[7] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of Tricks for Efficient Text Classification. *CoRR* abs/1607.01759 (2016). arXiv:1607.01759 http://arxiv.org/abs/1607.01759

[8] E. Kafeza, A. Kanavos, C. Makris, and P. Vikatos. 2014. T-PICE: Twitter Personality Based Influential Communities Extraction System. In *2014 IEEE International Congress on Big Data*. 212–219. https://doi.org/10.1109/BigData.Congress.2014.38

[9] Eunice Kim, Yongjun Sung, and Hamsu Kang. 2014. Brand followers' retweeting behavior on Twitter: How brand relationships influence brand electronic word-of-mouth. *Computers in Human Behavior* 37 (2014), 18 – 25. https://doi.org/10.1016/j.chb.2014.04.020

[10] Cristian Lumezanu, Nick Feamster, and Hans Klein. 2012. # bias: Measuring the Tweeting Behavior of Propagandists.

[11] Aasish Pappu, Roi Blanco, Yashar Mehdad, Amanda Stent, and Kapil Thadani. 2017. Lightweight Multilingual Entity Extraction and Linking. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining* (Cambridge, United Kingdom) *(WSDM '17)*. Association for Computing Machinery, New York, NY, USA, 365–374. https://doi.org/10.1145/3018661.3018724

[12] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. 1532–1543. http://www.aclweb.org/anthology/D14-1162

[13] Alexandre Salle, Marco Idiart, and Aline Villavicencio. 2016. Enhancing the LexVec Distributed Word Representation Model Using Positional Contexts and External Memory. *CoRR* abs/1606.01283 (2016). arXiv:1606.01283 http://arxiv.org/abs/1606.01283

[14] Alexandre Salle, Aline Villavicencio, and Marco Idiart. 2016. Matrix Factorization using Window Sampling and Negative Sampling for Improved Word Representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Berlin, Germany, 419–424. https://doi.org/10.18653/v1/P16-2068

[15] S. Stieglitz and L. Dang-Xuan. 2012. Political Communication and Influence through Microblogging–An Empirical Analysis of Sentiment in Twitter Messages and Retweet Behavior. In *2012 45th Hawaii International Conference on System Sciences*. 3500–3509. https://doi.org/10.1109/HICSS.2012.476

[16] Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas. 2010. Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology* 61, 12 (2010), 2544–2558.

[17] Soroush Vosoughi, Deb Roy, and Sinan Aral. 2018. The spread of true and false news online. *Science* 359, 6380 (2018), 1146–1151. https://doi.org/10.1126/science.aap9559 arXiv:https://science.sciencemag.org/content/359/6380/1146.full.pdf

[18] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. NPA: Neural News Recommendation with Personalized Attention. *CoRR* abs/1907.05559 (2019). arXiv:1907.05559 http://arxiv.org/abs/1907.05559