SSN_NLP at CheckThat! 2020: Tweet Check Worthiness Using Transformers, Convolutional Neural Networks and Support Vector Machines

Sachin Krishan Thyaharajan, Kayalvizhi Sampath, Thenmozhi Durairaj, Rishivardhan Krishnamoorthy

SSN College Of Engineering, Chennai
sachinkrishnan18128@cse.ssn.edu.in, {kayalvizhis,theni_d}@ssn.edu.in, rishivardhan18126@cse.ssn.edu.in

Abstract. Social media has become a significant source of information for a large fraction of the population. One such popular social media is twitter. An excessive amount of misinformation spread has become ubiquitous. But it is immensely computationally expensive to verify every claim made in every tweet. In this paper, the authors have explored Machine learning solutions to score a tweet on its worthiness to be fact-checked. In this paper, we present approaches using CNN, Transformer models and SVM for CLEF-2020 CheckThat! Check-Worthiness task.

Keywords: Check Worthiness, Convolutional Neural Network, Transformers, Support Vector Machine

1 Introduction

The rampant spread of fake news on social media has become an all too familiar plight. Fake news has become indistinguishable from real news. The hazards caused by fake news create confusion and misunderstanding about important social and political issues. According to a study on Twitter, false news travels faster than true news. Research project finds humans, not bots, are primarily responsible for the spread of misleading information [17]. With important political figures and business tycoons active on Twitter, it has become a rather important stage for global information. It is unrealistic to check every tweet to verify the information it holds due to the exorbitant computational requirement with 500 million tweets posted every day.
This puts us in need of an algorithm that can filter or rank tweets based on their check worthiness which is the goal of the CLEF-2020 Check That!'s [3][1] Check Worthiness task 1. We use CNN, Transformer models and SVM to score each tweet based on their tweet worthiness.

2 Related Work

Prevention of fake news collides with freedom of speech, hence detection can be done based on objective facts to curb fake news [7]. Prevalent state-of-the-art fact-checking methods employ the use of feature engineering techniques to extract features from each sentence and thereby its context. ClaimBuster[10] system arose as the first work to check worthiness. The system extracts sentiment features, TF-IDF word representations, POS tags and named entities. These features were derived from sentence level, thereby no contextual information between sentences was captured. Extending ClaimBuster's work, [9] incorporated contextual awareness into representation by the inclusion of sentence positioning in a speaker segment, speaker mentioning the opponent, audience reactions and sentence similarity to segments as features. [14] proposes a deep learning framework for detecting Rumors from Microblogs with Recurrent Neural Networks for learning hidden representations that capture the variation of contextual information of relevant posts over time.
In last year's CheckThat! [2] Team Copenhagen [8] achieved the best performance. They made use of LSTM RNN that learned dual token embeddings, domain-specific embeddings and syntactic dependencies. Team TheEarthIsFlat [6] made use of a feed-forward neural network with two hidden layers which takes as input Standard Universal Sentence Encoder (SUSE) embeddings [4] for the current sentence as well as for the two previous sentences as a context.

Our system's approach can be considered to pivot on sentence-level features. We do not include context-aware features into our data due to the low amount of training data and less compute power.

3 Dataset description

The data for the task was given in two formats: *.tsv* and *.json*. The training set has 672 data points, the development set has 150 data points. We have used the *.tsv* format which is a TAB separated text file. The text encoding is UTF-8. A row of the *.tsv* file has the following format:

---

topic id $< Tab >$ tweet id $< Tab >$tweet url $< Tab >$ tweet text $< Tab >$ claim $< Tab >$ check worthiness

---

The column descriptions are as follows:

topic_id	unique ID for the topic of the tweet
tweet_id	unique ID for each tweet given by Twitter
tweet_url	URL of the given tweet
tweet_text	text content in the tweet
claim	It is a binary value of 1 if the tweet contains a claim else 0
check_worthiness	It is a binary value of 1 if the tweet is worthy to be fact checked else 0

4 Methodology

4.1 Data preparation

The tweets given were preprocessed initially to remove stop words and punctuation. Then, all the tweets were normalized. The whole data was lemmatized and tokenized using the *nltk* library[13].

4.2 Training the models

The data prepared is given to various models that include Convolutional Neural Network (CNN), Transformers and Support Vector Machine.

CNN
Convolutional neural networks (CNN) are trained on top of pre-trained word vectors for classification tasks [11]. The training data was vectorized using a Word2Vec vectorizer using a pre-trained Google News word vector model with 3 million 300-dimension English word vectors and then padded with zeros up to the maximum sequence length in the given sentences. The padded sequence was fed to the convolutional neural network made of an input layer, an embedding layer, 5 convolutional layers each followed by max-pooling layers, a dropout layer and 2 dense layers. The convolutional layers each use 200 filters and the layers have kernel sizes of 2,3,4,5 and 6 sequentially with RelU activation. The following dense layer has 128 nodes with RelU activation and the following dense layer has 2 nodes with sigmoid activation. The model was trained and saved for classifying test data.

Transformers
The models used are derivatives of Google's BERT [5]. BERT stands for Bidirectional Encoder Representations from Transformers. The models used in this paper are RoBERTa [12] and XLNet [18] which are derivatives of BERT that give better performance. XLNet is developed to work seamlessly with the Auto Regression objective, including integrating Transformer-XL and the careful design of the two-stream attention mechanism. It manages to overcome the deficiencies of BERT whilst requiring more compute power and memory (GPU/TPU memory). The XLNet model was fine-tuned over the pre-trained XLNet Base Cased language model that comprises 12 Transformer blocks, 12 self-attention

heads and 768 hidden dimensions. RoBERTa makes use of a robustly optimized method that improves on BERT by modifying key hyperparameters in BERT. It was fine-tuned over the RoBERTa base language model that comprises 12 Transformer blocks, 12 self-attention heads and 768 hidden dimensions with a total parameter of 215M. For our case of experimentation, both transformer models were trained with hyperparameters as 50 epochs with training batch size set to 128 and the learning rate set to 4e-5.

Support Vector Machine
SVM or Support Vector Machine is a traditional machine learning algorithm. The principle behind the algorithm is it creates a hyperplane which separates the data into classes. The text is vectorized using a count vectorizer. The resulting count matrix is transformed to a normalized TF-IDF representation then classified using SVM [16] classifiers of scikit-learn [15].

4.3 Choosing models

Amongst these four models experimented, RoBERTa was submitted. It was selected based on the f1 score and because of the proven robustness of transformer models for natural language processing tasks. The performance of our models in the development set is shown in Table 1.

Model	**F1**
RoBERTa	0.730
XLNet	0.642
CNN	0.654
SVM	0.590

Table 1. Performance of our model in development set

5 Results

Table 3 shows the performance of the selected Roberta model on the evaluation data. Table 2 shows the test results of all participants. The main metric used for evaluation is the Mean Average Precision (MAP). The performance of the model is ranked 5th in comparison to the other teams' models. We can see that the Precision @ k value decreases as k increases suggesting the model is reliable with top tweets. The model got a perfect score on the Reciprocal Rank again suggesting the model performs well for the top tweets. We can infer from this that the model gives valid top rankings.

Team	MAP	RR	R-P	P@1	P@3	P@5	P@10	P@20	P@30
Accenture	**0.8064**	**1.0000**	**0.7167**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**0.9500**	**0.7400**
Team_Alex	0.8034	**1.0000**	0.6500	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**0.9500**	**0.7400**
contr.-1	0.7988	1.0000	0.6500	1.0000	1.0000	1.0000	1.0000	0.9500	0.7400
contr.-2	0.7809	1.0000	0.6667	1.0000	1.0000	1.0000	1.0000	0.8500	0.6800
check_square	0.7217	1.0000	0.6667	**1.0000**	0.6667	0.8000	0.8000	0.8000	0.7000
contr.-1	0.6249	0.5000	0.6000	0.0000	0.6667	0.8000	0.8000	0.6500	0.5800
contr.-2	0.7139	0.5000	0.6833	0.0000	0.6667	0.6000	0.8000	0.8500	0.7000
QMUL-SDS	0.7141	**1.0000**	0.6333	**1.0000**	**1.0000**	**1.0000**	0.9000	0.8000	0.6400
contr.-1	0.7820	1.0000	0.7000	1.0000	1.0000	1.0000	1.0000	0.8500	0.7000
contr.-2	0.7288	1.0000	0.6333	1.0000	1.0000	1.0000	0.9000	0.8500	0.6800
Tobb Etu	0.7062	**1.0000**	0.6000	**1.0000**	**1.0000**	**1.0000**	0.9000	0.8000	0.6600
contr.-1	0.5635	0.2000	0.6000	0.0000	0.0000	0.2000	0.3000	0.6000	0.6600
contr.-2	0.7102	1.0000	0.6333	1.0000	1.0000	1.0000	1.0000	0.7500	0.6800
SSN_NLP	0.6739	**1.0000**	0.6000	**1.0000**	**1.0000**	0.8000	0.8000	0.8000	0.6200
Factify	0.6561	0.5000	0.6833	0.0000	0.3333	0.6000	0.7000	0.7500	0.7000
contr.-1	0.6963	1.0000	0.6833	1.0000	0.3333	0.6000	0.8000	0.8000	0.7400
BustingMisinformation	0.6172	**1.0000**	0.5833	**1.0000**	**1.0000**	0.8000	0.7000	0.6000	0.6000
nlpir01	0.6069	1.0000	0.5667	1.0000	**1.0000**	**1.0000**	0.7000	0.6000	0.5800
contr.-1	0.5546	0.2500	0.5500	0.0000	0.0000	0.4000	0.7000	0.7500	0.5200
contr.-2	0.5193	0.5000	0.4500	0.0000	0.6667	0.4000	0.5000	0.6000	0.4800
ZHAW	0.5052	0.3333	0.5333	0.0000	0.3333	0.4000	0.6000	0.5000	0.5200
contr.-1	0.6648	1.0000	0.6333	1.0000	1.0000	0.8000	0.9000	0.7000	0.6600
UAICS	0.4950	1.0000	0.4667	**1.0000**	0.3333	0.4000	0.6000	0.6000	0.4600
TheUniversityofSheffield	0.4746	0.2500	0.5333	0.0000	0.0000	0.4000	0.2000	0.3500	0.4800
contr.-1	0.6459	1.0000	0.5833	1.0000	1.0000	1.0000	0.8000	0.6000	0.5800

Table 2. Test set results

Run	MAP	RR	R-P	P@1	P@3	P@5	P@10	P@20	P@30
RoBERTa	0.6739	1.0000	0.6000	1.0000	1.0000	0.8000	0.8000	0.8000	0.6200

Table 3. Performance of test data

6 Conclusion

We have presented our submission for Task 1 of CheckThat! @ CLEF-2020 to predict the check-worthiness of tweets. The task was approached with various methods that include transformers, CNN and SVM. Among these models, the Roberta transformer model performed better than other methodologies on the development set. Hence, the output from RoBERTa was submitted for evaluation. From table 2, the results show that our team has a perfect recall score and a good MAP score. The performance can further be improved with some other transformer models such as XLNet, Electra, BERT, etc and inclusion of more data samples for training.

7 Acknowledgement

We express our sincere thanks to DST-SERB and HPC laboratory for providing space and materials needed for our work.

References

1. Arampatzis, A., Kanoulas, E., Tsikrika, T., Vrochidis, S., Joho, H., Lioma, C., Eickhoff, C., Névéol, A., Cappellato, L., Ferro, N. (eds.): Experimental IR Meets Multilinguality, Multimodality, and Interaction Proceedings of the Eleventh International Conference of the CLEF Association (CLEF 2020). LNCS (12260), Springer (2020)
2. Atanasova, P., Nakov, P., Karadzhov, G., Mohtarami, M., Da San Martino, G.: Overview of the clef-2019 checkthat! lab: Automatic identification and verification of claims. task 1: Check-worthiness. In: CLEF (Working Notes) (2019)
3. Barrón-Cedeño, A., Elsayed, T., Nakov, P., Da San Martino, G., Hasanain, M., Suwaileh, R., Haouari, F., Babulkov, N., Hamdan, B., Nikolov, A., Shaar, S., Sheikh Ali, Z.: Overview of CheckThat! 2020: Automatic identification and verification of claims in social media. In: Arampatzis et al. [1]
4. Cer, D., Yang, Y., yi Kong, S., Hua, N., Limtiaco, N., John, R.S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., Sung, Y.H., Strope, B., Kurzweil, R.: Universal sentence encoder (2018)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding (2018)
6. Favano, L., C.M.L.P.: TheEarthIsFlat's submission to CLEF'19 CheckThat! challenge. In: CLEF 2019 Working Notes. Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum. CEUR Workshop Proceedings, CEUR-WS.org, Lugano, Switzerland (2019) (2019)
7. Figueira, , Oliveira, L.: The current state of fake news: challenges and opportunities. Procedia Computer Science **121**, 817–825 (12 2017). https://doi.org/10.1016/j.procs.2017.11.106
8. Hansen, C., H.C.S.J.L.C.: Neural weakly supervised fact check-worthiness detection with contrastive sampling-based ranking loss. In: CLEF 2019 Working Notes. Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum. CEUR Workshop Proceedings, CEUR-WS.org, Lugano, Switzerland (2019) (2019)
9. Hassan, N., Arslan, F., Li, C., Tremayne, M.: Toward automated fact-checking: Detecting check-worthy factual claims by claimbuster. pp. 1803–1812 (08 2017). https://doi.org/10.1145/3097983.3098131
10. Hassan, N., Li, C., Tremayne, M.: Detecting check-worthy factual claims in presidential debates. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. p. 1835–1838. CIKM '15, Association for Computing Machinery, New York, NY, USA (2015). https://doi.org/10.1145/2806416.2806652, `https://doi.org/10.1145/2806416.2806652`
11. Kim, Y.: Convolutional neural networks for sentence classification (2014)
12. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach (2019)
13. Loper, E., Bird, S.: Nltk: The natural language toolkit (2002)

14. Ma, J., Gao, W., Mitra, P., Kwon, S., Jansen, J., Wong, K.F., Cha, M.: Detecting rumors from microblogs with recurrent neural networks (07 2016)
15. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
16. Prasetijo, A.B., Isnanto, R.R., Eridani, D., Soetrisno, Y.A.A., Arfan, M., Sofwan, A.: Hoax detection system on indonesian news sites based on text classification using svm and sgd. In: 2017 4th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE). pp. 45–49 (2017)
17. Vosoughi, S., Roy, D., Aral, S.: The spread of true and false news online. Science **359**(6380), 1146–1151 (2018). https://doi.org/10.1126/science.aap9559, `https://science.sciencemag.org/content/359/6380/1146`
18. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., Le, Q.V.: Xlnet: Generalized autoregressive pretraining for language understanding (2019)