# ReINTEL Challenge 2020: Vietnamese Fake News Detection using Ensemble Model with PhoBERT embeddings

**Cao Nguyen Minh, Hieu**
VNG Corporation
cnmhieu.hcmus.edu.vn@gmail.com

**Nguyen Hieu, Thuan**
Athena Studio
thuan.hieu301@gmail.com

**To Van, Hung**
Shopee
hungvanto123456@gmail.com

**Vo Quoc, Bang**
Tiki Corporation
bavo.imp@gmail.com

## Abstract

Along with the increasing traffic of social networks in Vietnam in recent years, the number of unreliable news has also grown rapidly. As we make decisions based on the information we come across daily, fake news, depending on the severity of the matter, can lead to disastrous consequences. This paper presents our approach for the Fake News Detection on Social Network Sites (SNSs), using an ensemble method with linguistic features extracted using PhoBERT (Nguyen and Nguyen, 2020). Our method achieves AUC score of 0.9521 and got 1st place on the private test at the 7th International Workshop on Vietnamese Language and Speech Processing (VLSP). For reproducing the result, the code can be found at https://gitlab.com/thuan.hieu301/vlsp2020-reintel-kurtosis

## 1 Introduction

Social network sites have become a very influential part of Vietnamese people's daily life. We use them to connect with each other, and get access to the latest information. However, such advances in large scale communication also bring their problems, one of which is fake news. It can be seen as information which is altered, manipulated, misguiding users to achieve personal gains, such as increase advertisement interaction, political power gain, or even terrorism. Without proper censoring, they can spread fear in the public community, causing panic and invoking violence.

Due to such dire consequences, a lot of researches have been done to prevent this type of harmful information. However, there has been little effort put in for the Vietnamese language. This is a challenging task, due to a lack of quality human-verified data, and the difficult nature of the fake contents. Fake news may have:

- Similar contents to the real ones, however some key information is twisted (figures, celebrities, locations, ...) in order to capture the attention of readers.
- Contents encapsulated inside images, which requires human verification
- Special slangs, acronyms, misspellings which makes it difficult for machine to automate the process
- Unseen information that can take times before it is verified, which then might be too late

In this paper, we present our approach to the problem of fake news detection presented at the VLSP 2020, shared-task Reliable Intelligence Identification on Vietnamese SNSs (ReINTEL) (Le et al., 2020). We experimented with 3 types of features: the time the news is posted, the community interaction to its (through number of share, like, comment) and, most importantly, the content of the news. After much preprocessing and exploration had been done, we combined the strength of basic handcrafted linguistic cues in the training data with term frequency encoding (TF-IDF) and PhoBERT as context embedding. These features are combined and used as input for an ensemble model using StackNet [1]. Our model achieved the AUC score of 0.9521, ranked first place on the private leader board of ReINTEL.

We discuss related work and previous approaches in section 2. We then describe our method workflow in section 3, starting with data cleaning and preprocessing, how we extracted the features we used, and the ensemble of models for our final result. Experiment's results and detailed description of parameters are shown in section 4. We

[1] A framework using stacked generalization to combine results of different models https://github.com/kaz-Anova/StackNet.

conclude our report and discuss what could be improved in section 5.

## 2 Related works

For the linguistic-based features, some approaches focus on extract special discriminative features such as acronymns, pronoun, special characters (Shu et al., 2017; Gupta et al., 2014). However, these features are not well understood, as well as require extensive labour for validation and can be domain specific. Ruchansky et al. extend the method by using doc2vec embeddings, which learn semantic representation of the posts. Recent advancement in Natural Language Processing, and most importantly BERT (Devlin et al., 2018), has helped to advance the research on this topic. Bhatt et al. combine the context generated by using LSTM and CNN, in combination with statistically handcrafted features to perform the final prediction.The work by Yang et al. use a combination of multiple Recurrent Neural Network (RNN) architectures as a natural language inference (NLI) mechanism, combining with BERT to make the final prediction. Research done by Huang and Chen focuses more on ensembling multiple deep learning architectures to achieve State Of The Art result for Fake News Detection. Ahmad et al. also shows that ensembling methods help achieve better performance on the current task.

## 3 Methodology

In this section, we will describe our approach to solve the problem. Linguistic features extracted with PhoBERT and tf-idf, in conjunction with metadata provided, are used as input to an ensemble of models to achieve the best result in the private dataset. Using models that don't require much computation power not only helps us to tune each model quickly, but also enable us to analyze the impact of each feature on the fake news detection problem as a whole.

### 3.1 Preprocessing

To extract valuable features, we started with some preprocessing steps, which is described as follow:

1. Convert numeric-like features to numeric type if possible, null value otherwise;
2. Remove rows having null or empty content;
3. Deduplicated rows having the same content and interactions.

The first step were applied on both training and test set, while the remain ones were done only on training set.

### 3.2 Feature Engineering

#### 3.2.1 Metadata

We considered all features except the content of the posts are metadata features.

***Number of likes, comments, and shares:*** We first transformed these 3 features to log scale for normalization. Then for each of them, a *is_null* feature were generated, equaling to 0 if the corresponding value is presented, and 1 otherwise.

***Timestamp of posts:*** We extracted the hour and the day of week from the timestamp of posts.

***Combinations:*** We tried to generate some combinations of the above numeric features. Particularly, we computed the divisions of the number of likes, comments, and shartes to each other and obtained 3 new numeric features.

Finally, any not-a-number value was filled by -1.

#### 3.2.2 Post content

***Term Frequency - Inverse Document Frequency (TF-IDF):*** TF-IDF is a simple but strong feature extraction technique for text data. We fitted a TF-IDF vectorizer from 1-gram to 3-gram on post contents of our training data, followed by a Single Value Decomposition (SVD) model to reduce the dimension of transformed TF-IDF features. A 300-dimensional vector of latent features was obtained for each post at the end of this step.

***PhoBERT Embedding:*** BERT (Devlin et al., 2018) is a robust language model recently boosting many NLP tasks to a new level of achievement. PhoBERT (Nguyen and Nguyen, 2020), in our knowledge, is the best pre-trained BERT model for Vietnamese. In our solution, we leveraged PhoBERT to extract document embeddings from the posts. Notably, to receive more meaningful contextual embedding, some cleaning operations were applied to the contents before feeding into PhoBERT, consisting of word tokenization, special characters removal, redundant content removal. Moreover, another SVD model was fitted on top of those embedding to map 768-d output vectors of the BERT model to 100-dimensional space.

***Characters Counting:*** After extensive exploratory analysis, it turned out that the occurrence of some special characters and patterns have impact on the performance of our model, such as question mark, exclamation mark, triple dot, link, and so on. Thus, we created a list of those characters and created corresponding features which present the number of each of them in the posts.

### 3.3 Modelling

Tree-based models are the first choice when dealing with tabular data, thanks to their strength in both predictability and explainability. Furthermore, ensemble learning, especially stacking, is a good way to prevent overfitting and improve the performance of the overall system. Pursuing these observations, we designed our modeling phase as an ensemble system including 25 different base models and 5 stacked models on top of them. Precisely, the base models are from 5 different kinds: 5 Random Forests, 5 LightGBM Gradient Boosting Trees (GDBTs), 5 CatBoost GDBTs, 5 shallow Neural Networks, and 5 Naive Bayes classifiers; and the stacked models are 5 CatBoost GDBTs.

*Training phase:* we formulate our training data in a 5-folds cross-validation manner. In each fold, 5 different-kind models were trained. After these training finished, 5 probability vectors were predicted and treated as 5 features, combined with the original features to form a new training set to train the corresponding stacked model of that fold.

*Inference phase:* probabilities from 5 trained stacked models are averaged to get final scores.

## 4 Experiments

### 4.1 Datasets

We evaluated our methods on the datasets provided by the 2020 VLSP competition, which contain totally about 6000 training and 2000 testing examples, divided into multiple sets described in table 1. The manually annotated labels equal to 1 if the news as potentially unreliable, and 0 otherwise. Our training set is composed of the public training and the warmup training set. Table 2 is a statistic summarization of our training set. After the feature engineering steps, our final training set consisted of 420 features and 4956 examples, 831 (16.8%) of which are label 1.

|  | **no. of examples** |
|---|---|
| warmup training set | 800 |
| warmup test set | 200 |
| public training set | 4372 |
| public test set | 1642 |
| private test set | 1646 |
| Total | 8600 |

Table 1: Datasets.

|  |  |
|---|---|
| # rows | 5172 |
| # label 1 | 934 |
| # *user_name* | 3706 |
| # unique *post_message* | 4868 |
| latest *timestamp_post* | Jan 2, 2014 |
| nearest *timestamp_post* | Sep 28, 2020 |

Table 2: Statistic summarization of our training set.

It should be noted that, although only the 2 training sets contain labels, we still leveraged the content of posts from all datasets except the private one to extract features described in section 3.2.2. This way of making full use of unlabeled data help the model generalize well and result in better performance.

### 4.2 Model hyper-parameters

|  |  |
|---|---|
| Tf-Idf vectorizer | n-gram range=(1, 3) |
| SVD on Tf-Idf | n_components=300 |
| SVD on embedding | n_components=100 |
| Naive Bayes | class_prior=[.75, .25] |
| Random Forest | n_estimators=800<br>max_depth=11 |
| Neural Network | hidden_layer=(40,)<br>learning_rate=0.001<br>max_iter=100 |
| LightGBM | n_estimators=1000<br>learning_rate=0.012<br>num_leaves=7 |
| CatBoost | iterations=530<br>learning_rate=0.015<br>depth=6 |

Table 3: Model hyper-parameters.

Table 3 shows the tuned hyper-parameters we used for each model described in Section 3.3. All classifiers except Naive Bayes used our predefined class weights of 0.15 for class 0 and 0.75 for class 1.

| | Time (seconds) |
|---|---|
| Fitting TF-IDF and SVD | 282.71 |
| Getting embedding | 375.18 |
| All steps before training | 779.42 |
| Training model | 474.96 |
| Whole training stage | 1254.38 |
| Whole inference stage | 14.76 |

Table 4: Approx. run time of proposed method.

### 4.3 Evaluation

All steps were executed on the same machine with the following specs: 4 Intel Xeon CPUs 2.20GHz, 1 16GB RAM, and 1 Tesla T4 16GB GPU. The step that occupied the most amount of RAM (~10GB) is fitting SVD on vectorized TF-IDF features. Only the training step of ensemble model used all of CPU cores, the others only used one core at a time. GPU was only used for extracting document embeddings from PhoBERT model. Table 4 summarizes approximate time of some time-consuming steps of the proposed method on our training set.

We use Area Under the Curve (AUC) score as our evaluation metric and a 5-folds cross-validation scheme to evaluate our models. Though lots of experiments were made, we only shows the main versions that improve the performance significantly. All versions before ensemble were trained with a tuned CatBoost classifier. Comparison to top teams in the competition are shown in table 5. Our experiments were conducted as follow:

- Version 1: no embedding, no combination features (described in section 3.2.1).
- Version 2: add PhoBERT embedding.
- Version 3: add ensemble learning manner.
- Version 4: add combination features
- Final version: leverage unlabeled data.

## 5 Conclusion

### 5.1 Summary

We list out some remarkable insights that we discovered in this task:

- Combining high-importance features is a good way of feature generation
- TF-IDF should be applied on raw contents to capture their original form, while document embedding should be applied on cleaned ones to obtain contextual features.

| | CV | PublicLB | PrivateLB |
|---|---|---|---|
| Ours (V1) | 0.8633 | 0.8482 | - |
| Ours (V2) | 0.9104 | 0.8895 | - |
| Ours (V3) | 0.9454 | 0.9326 | - |
| Ours (V4) | 0.9508 | 0.9399 | 0.9406 |
| Ours (Final) | 0.9647 | - | **0.9521** |
| Other teams | | | |
| NLP_BK | - | 0.9360 | 0.9513 |
| Toyo-Aime | - | **0.9427** | 0.9449 |

Table 5: AUC scores of proposed method and other teams on different datasets.

- The more the content the model learnt, the better the performance.
- Stacking with complementary bagging is very powerful.

### 5.2 Future work

Due to the time limit, a lot of methods we tried still need more validation and tuning, therefore were left out of the final submission. Other information, such as post images, can also give a boost in performance, due to the content is embedded in the images, or special information such as watermarks. Other Natural Language Processing features like sentiment of the comments, Part Of Speech tagging, bias, although tried, but haven't tuned carefully to produce good result, could be helpful. We also believe the URL, if provided, could also help improve the performance.

## References

Iftikhar Ahmad, Muhammad Yousaf, Suhail Yousaf, and Muhammad Ovais Ahmad. 2020. Fake news detection using machine learning ensemble methods. *Complexity*, 2020.

Gaurav Bhatt, Aman Sharma, Shivam Sharma, Ankush Nagpal, Balasubramanian Raman, and Ankush Mittal. 2017. On the benefit of combining neural, statistical and external features for fake news identification. *arXiv preprint arXiv:1712.03935*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Aditi Gupta, Ponnurangam Kumaraguru, Carlos Castillo, and Patrick Meier. 2014. Tweetcred: Real-time credibility assessment of content on twitter. In *International Conference on Social Informatics*, pages 228–243. Springer.

Yin-Fu Huang and Po-Hong Chen. 2020. Fake news detection using an ensemble learning model based on self-adaptive harmony search algorithms. *Expert Systems with Applications*, page 113584.

Duc-Trong Le, Xuan-Son Vu, Nhu-Dung To, Huu-Quang Nguyen, Thuy-Trinh Nguyen, Linh Le, Anh-Tuan Nguyen, Minh-Duc Hoang, Nghia Le, Huyen Nguyen, and Hoang D. Nguyen. 2020. Reintel: A multimodal data challenge for responsible information identification on social network sites.

Dat Quoc Nguyen and Anh Tuan Nguyen. 2020. PhoBERT: Pre-trained language models for Vietnamese. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1037–1042.

Natali Ruchansky, Sungyong Seo, and Yan Liu. 2017. Csi: A hybrid deep model for fake news detection. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 797–806.

Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake news detection on social media: A data mining perspective. *ACM SIGKDD explorations newsletter*, 19(1):22–36.

Kai-Chou Yang, Timothy Niven, and Hung-Yu Kao. 2019. Fake news detection as natural language inference. *arXiv preprint arXiv:1907.07347*.