VLSP 2020

Proceedings of the 7th International Workshop on Vietnamese Language and Speech Processing

18 December, 2020

Hanoi University of Science and Technology

Hanoi, Vietnam

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

Introduction

In 2020, the Association for Vietnamese Language and Speech Processing (VLSP) is officially founded, as a chapter of Vietnam Vietnam Association for Information Processing. VLSP consortium is an initiative which came from Institute of Information Technology - Vietnam Academy of Science and Technology, to establish a community working on speech and text processing for Vietnamese language. The first national project KC01.03/06-10 from 2007-2009 which received strong support from Ministry of Science and Technology, gathered eight active research groups from universities and institutes in Vietnam and overseas. A main goal of the first project is to set up long term strategy on Vietnamese language processing in order to provide and to involve community to enrich shared language resources and tools for R&D purpose. Since 2012, the VLSP Consortium has organized a series of workshops, in conjunction with large international conferences organized in Vietnam. Until 2020, six events have taken place with different forms of activities such as technical reports, activity reports, discussion panel, shared tasks on VLSP. From VLSP 2016, organizing shared tasks on Vietnamese processing became the main activities of the workshop series in order to promote the development of essential tools and resources for VLSP.

The seventh VLSP workshop constitutes the first workshop organized by the Association for VLSP. held in Hanoi in December 2020. To mark this event, seven shared tasks have been proposed and in the end six challenges have been organized:

1. Vietnamese Universal Dependency Parsing (UDP): the task of determining syntactic dependencies between words in a sentence.
2. Vietnamese Relation Extraction (RE): the task of identifying and determining the semantic relations between pairs of named entity mentions within a single sentence.
3. English-Vietnamese Machine Translation (MT): text translation from English to Vietnamese in the news domain.
4. Reliable Intelligence Identification on Vietnamese Social Network Sites (ReINTEL): the task of identifying a piece of information shared on social network sites (SNSs) as reliable or unreliable.
5. Automatic Speech Recognition for Vietnamese (ASR): the task includes two evaluation sub-tasks. For the first sub-task, all participants are required to use only the provided data to develop ASR models including acoustic and language models. For the second one, participants can use all available data sources to develop their ASR models without any limitation.
6. Vietnamese Text-To-Speech on Common Datasets (TTS): the task of building a TTS system with a training voice from the same speech database released by the organizers.

The participants to the evaluation campaign were asked to present their system in a dedicated paper.

We very much hope that you have had an enjoyable and inspiring time!

Huyen T M. Nguyen, Xuan-Son Vu, Chi Mai Luong

Hanoi & Umeå

Feburary 2021

Organisers:

General Chair: Nguyễn Thị Minh Huyền
Organizing Co-chairs: Nguyễn Văn Huy, Hà Thành Lê, Vũ Xuân Sơn, Nguyễn Thị Thu Trang, Trần Thế Trung, Trần Mai Vũ
Program Committee Co-chairs: Nguyễn Lê Minh, Lương Chi Mai
Publication Co-chairs: Vũ Xuân Sơn, Nguyễn Thanh Sơn, Hà Thành Lê, Vũ Tiến Thành

Program Committee:

Ngô Xuân Bách (PTIT, Vietnam), Nguyễn Việt Cường (INT^2), Lê Anh Cường (TDTU, Vietnam), Phạm Hiển (IOL, VASS), Phan Xuân Hiếu (UET, VNU), Lê Quang Hùng (QNU, Vietnam), Lê Thanh Hương (HUST, Vietnam), Nguyễn Văn Huy (VINBDI), Nguyễn Thị Minh Huyền (HUS, VNU) Hà Thành Lê (KIT, Germany & VINBDI, Vietnam), Lương Chi Mai (IOIT, VAST), Nguyễn Lê Minh (JAIST, Japan), Phạm Quang Nhật Minh (Aimesoft Co., Vietnam), Nguyễn Lưu Thuỳ Ngân (UIT, VNUHCM), Lê Hồng Phương (HUS, VNU), Vũ Xuân Sơn (Umeå University, Sweden), Nguyễn Phương Thái (UET, VNU), Lê Công Thành (InfoRe), Nguyễn Thị Thu Trang (HUST, Vietnam), Trần Thế Trung (FPT, Vietnam), Nguyễn Văn Vinh (UET, VNU), Trần Mai Vũ (UET, VNU), Lê Đức Trọng (UET, VNU), Harry Nguyen (University of Glasgow, Singapore), Alex To, (ReML.AI) Nguyễn Thuỳ Trinh (ReML.AI), Nguyễn Quang (UET, VNU), Linh Le (ReML.AI), Hoàng Minh Đức (ReML.AI), Nguyễn Anh Tuấn (ReML.AI), Lê Nghĩa (ReML.AI)

Table of Contents

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.

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.

	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.

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
	max_depth=11
Neural Network	hidden_layer=(40,)
	learning_rate=0.001
	max_iter=100
LightGBM	n_estimators=1000
	learning_rate=0.012
	num_leaves=7
CatBoost	iterations=530
	learning_rate=0.015
	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*.

ReINTEL Challenge 2020: A Comparative Study of Hybrid Deep Neural Network for Reliable Intelligence Identification on Vietnamese SNSs

Hoang Viet Trinh, Tung Tien Bui, Tam Minh Nguyen
Huy Quang Dao, Quang Huu Pham, Ngoc N. Tran
trinh.viet.hoang@sun-asterisk.com
AI Research Team, R&D Lab, Sun* Inc.

Ta Minh Thanh
Le Quy Don Technical University, Ha Noi, Vietnam

Abstract

The overwhelming abundance of data has created a misinformation crisis. Unverified sensationalism that is designed to grab the readers' short attention span, when crafted with malice, has caused irreparable damage to our society's structure. As a result, determining the reliability of an article has become a crucial task. After various ablation studies, we propose a multi-input model that can effectively leverage both tabular metadata and post content for the task. Applying state-of-the-art fine-tuning techniques for the pretrained component and training strategies for our complete model, we have achieved a 0.9462 ROC-score on the VLSP private test set.

1 Introduction

1.1 Overview

The fast growth of social media and misinformed contents have posed an incremental challenge of exposing untrustworthy news to billions of their global users, including 65 million Vietnamese users (Social, 2020). Consequently, the spread of mistrust information on social cites has placed real damages on government, policymakers, organizations, and citizens of many countries (Cheng and Chen, 2020; Pham et al., 2020), resulting in an urge for fast and large-scale fact-checking online contents. With the enormous amount of news and information on the internet daily, this is impossible to be efficiently done only by human efforts, putting a quest to create a trustworthy system to perform the task automatically.

Reliable Intelligence Identification on Vietnamese SNSs (ReINTEL) is the task of reliable or unreliable social-network-sites (SNSs) identification. The main difficulties of these tasks, including:

- The given data (contents of social sites) is unstructured, containing mostly texts combined with metadata (including: images, dates, numbers, username, id, *etc*). The meta-information is partially missing and incorrect, making the usage of those data more challenging.
- The problem is multi-modal learning, which 'involves relating information from multiple sources' (Sachowski, 2016), resulting in the search for a proper combination of features from those sources to learn a unified model with high performance.

1.2 Our contributions

In this paper, we propose our methods to resolve these above-mentioned problems. With thorough experiments, we determined to answers two main questions: Should we incorporate multi-source data? Furthermore, how to combine them in terms of training strategies? Our contributions are as followed:

- We provide a reliable method of data cleansing, making metadata ready for prediction.
- More importantly, we are the first who construct a comprehensive comparative study to discover the effectiveness of models when incorporating multi-source data with different training strategies. Our experiment's results reveal that:
  - Models using text or meta-features alone has a crucial gap in performance, indicating that texture information is significantly more predictive than metadata.
  - Models utilize multi-source data with different training strategies results in a wide range of performance. This finding implies that combining data in training has a significant impact on the overall performance.
  - Combining data from multi-sources with particular training plans leads to our best

models. Additionally, the model trained with metadata alone performs significantly better than a random guess, shedding light on the meta data's informativeness.

- We apply state-of-the-art transfer learning methods for textual feature extractions and neural network (in comparison with other traditional machine learning methods) for tabular-data feature representation, achieving the competitive performance of 0.9418 ROC-score on the public test set (ranked 2nd) and 0.9462 ROC-score (ranked 3th) on the private test set.

1.3 Roadmap

In the following sections, we briefly review some related works involve with our methods. Next, in section 3, we illustrate our method in detail. Our experiments are described in Section 4, including dataset description, data preprocessing methods, and our model configurations, whereas Section 5 indicates all of our experimental results. Finally, section 6 is the conclusion for our proposed framework.

2 Related work

2.1 Contextual Representation For Text

Recent works on learning universal representation for text, namely Elmo (Peters et al., 2018), GPT (Radford, 2018), BERT (Devlin et al., 2018) have brought remarkable improvements for wide, diverse NLP downstream tasks: Text Classification, Question Answering and Named Entity Recognition. In contrast to traditional methods such as Word2vec (Mikolov et al., 2013) or Glove (Pennington et al., 2014) which learns context-independent word embeddings, universal language models were trained on a massively large amount of unlabeled data with different pretext tasks, including causal language modeling and masked language modeling, to learn a deep contextual representation of words given its context.

2.2 Fake News Detection on SNSs

Studies of fake news identification on social network sites have gained significant attention recently. Most of them utilize data from multiple sources. For example, CSI (Ruchansky et al., 2017), a framework with several modules based on Long Short-Term Memory (Hochreiter and Schmidhuber, 1997) and a fully connected layer that utilizes the article's contents, the users' responses and behaviors of source users who promote it. Another instance is dEFEND (Shu et al., 2019), which exploits both news contents and user comments with a deep hierarchical co-attention network to learn a rich representation for fake news detection. From a slightly different point of view, TriFN (Shu et al., 2017) models a tri-relationship between users, publishers, and new contents by several embedding methods and experiments promising results.

Table 1: Statistics of the datasets.

	Dataset
Total News	5172
Users	3706
Unique News	5087
News have images	1287
Reliable News	4238
Unreliable News	934

Although utilizing multi-source data, existing research appears to lack a comprehensive study on the effectiveness of input-combination strategies.

2.3 Vietnamese Natural Language Processing

Inspired by BERT's textual learning methods, PhoBERT (Nguyen and Nguyen, 2020) was proposed to extend the successes of deep pre-trained language models to Vietnamese. Its pretraining approach is based on RoBERTa (Liu et al., 2019) training strategies to optimize BERT training procedure. Additionally, PhoBERT also consists of two different settings, PhoBERT Base, which uses 12 Transformer Encoder layers and 24 layers with PhoBERT Large. It improves many Vietnamese NLP downstream tasks. For instance, Pham (Pham et al., 2020) introduced novel techniques to adapt general-purpose PhoBERT to a specific text classification task and archives state of the art on Vietnamese Hate Speech Detection (HSD) campaign.

3 Methodology

3.1 Dataset

In this paper, we use the dataset provided by VLSP organizers for ReINTEL task (Le et al., 2020), composed of contents from Vietnamese social network sites (SNSs), e.g., Facebook, Zalo, or Lotus (Social, 2020). There are approximately

5,000 labeled training examples, while the test set consists of 2,000 unlabeled examples. Each example is provided with information about the news's textual content, timestamp, number of likes, shares, comments, and attached pictures. Table 1 indicates the detailed statistic of the dataset, the data distribution of reliable and unreliable news was heavily imbalanced and skewed toward trustworthy contents.

3.2 Data preprocessing

Fake news can be studied with respect to four perspectives: (i) knowledge-based (focusing on the false knowledge in fake news); (ii) style-based (concerned with how fake news is written); (iii) propagation-based (focused on how fake news spreads); and (iii) credibility-based (investigating the credibility of its creators and spreaders) (Zhou and Zafarani, 2018). In this task, with the ReINTEL dataset, we focused on knowledge-based and credibility-based. Specifically, we performed the following preprocessing to extract the necessary information.

- **Deleted incorrect data rows**: While mining data, there are few incorrect rows due to the process of collecting and storing data. We decided to delete these rows from the data set.
- **Filled missing value**: To deal with missing values, we fill them with different strategies: numbers with 0, timestamps with the min timestamp and post messages with empty string
- **Extracted date time features from timestamp values**: For each timestamp value, we decoded these to date time values to enrich feature: minutes, hours, days, months, years, weekdays, etc.
- **Created `user_score` feature**: For user id, we created a user reputation score metric based on previous posts in dataset. This score is used to evaluate the user's future posts
- **Created `image_count` feature**: With images of each post, we compiled several information, including: number of images and image's aspect ratio
- **Preprocessed `post_message` feature**: We perform post messages preprocessing more carefully than the rest. The processing stages are listed below:
  - Filled missing value with empty string
  - Standardized Vietnamese punctuation
  - Removed HTML tags
  - Replaced email, links, phone, numbers, emoji, date time with new corresponding token

3.3 Model for Tabular Data

Metadata for the ReINTEL dataset is composed of all input features except post message (text data). We tried numerous machine learning algorithms to learn a classifier using only metadata, ranging from traditional methods: Logistic Regression, Linear Discriminant Analysis, K Nearest Neighbor, Decision Tree, Gaussian Naive Bayes, Support Vector Machine, Adaptive Boosting, Gradient Boosting, Random Forest (Hastie et al., 2001), and Extra Trees (Geurts et al., 2006) to a deep learning method: Multi-Layer Perceptron (Hastie et al., 2001)

We then proceeded to select a handful of model with high performances and complexities to serve as a base model for stacking (Wolpert, 1992). Meanwhile, for the meta-model used in stacking, we chose Logistic Regression. We also did the same for blending ensemble (Sill et al., 2009).

3.4 Deep learning-based Content Classification

BERT's layers capture a rich hierarchy of linguistic information, with surface features at the bottom, general syntactic knowledge in the middle, and specific semantic information at the top layer (Jawahar et al., 2019). Therefore, in order to better benefit for our downstream task, we incorporate as much as possible different kinds of information from our model backbone PhoBERT by concatenating [CLS] hidden states from each of 12 blocks, followed by a straightforward custom head, which is a multilayer perceptron with Dropout (Srivastava et al., 2014). The architecture of the model is shown in the Figure 1.

3.5 Deep Multi-input Model

Our experiments (details are in the below section) indicates that meta data is informative predictors for reliable and unreliable news classification. Therefore, we decided to combine both text and meta data to resolve the task. The structure of our

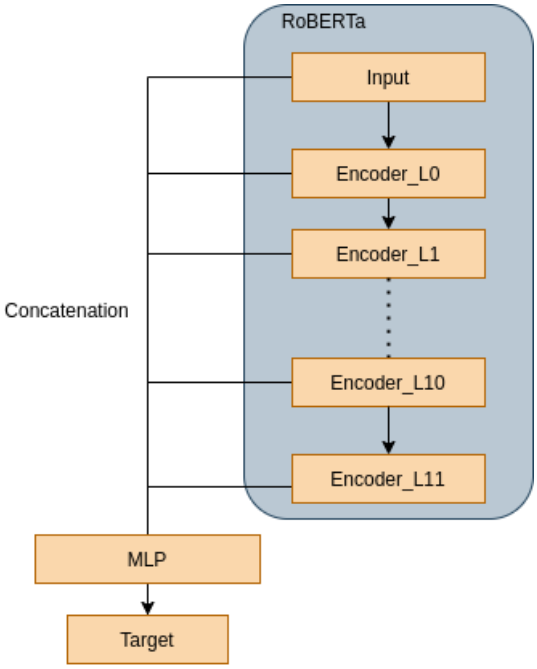


Figure 1: The architecture model for content classification using RoBERTa pre-trained model.

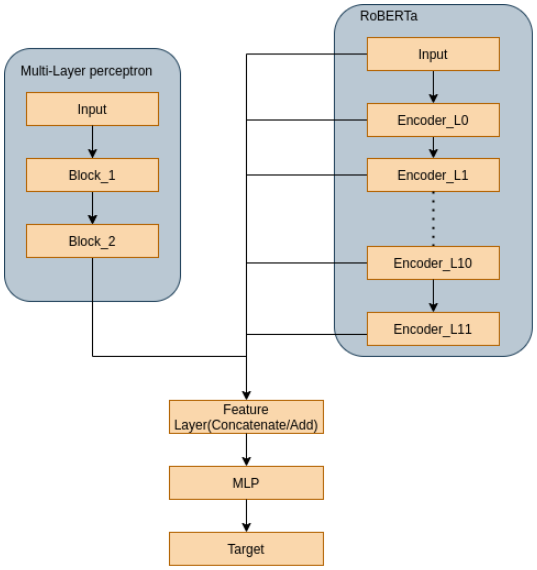


Figure 2: An illustration of our proposed deep multi-input architecture.

multi-input model is described (in Figure 2) as followed: output features of Multi-Layer Perceptron and RoBERTa models, after being concatenated or added together, were simply passed through a custom head classifier.

4 Experiments

4.1 Model Settings

We divide the dataset into a training set and a validation set with 10-fold cross validation method. Each fold, we use AdamW (Kingma and Ba, 2014) for optimization with a learning rate of 10^{-5} and a batch size of 32. Warm-up learning was applied, with the chosen maximum learning rate was 2×10^{-5}. Except for all bias parameters and coefficients of LayerNorm layers (Ba et al., 2016), the rest of the model's parameters were regularized with weight decay to reduce overfitting. We used a regularization coefficient of 0.01. The number of training epochs was 20.

Instead of using cross-entropy loss, we implemented a label smoothing cross-entropy loss function, a combination of cross-entropy loss and label smoothing (Müller et al., 2019). The smoothing rate is set to 0.15.

4.2 Fine-tuning technique

We applied state-of-the-art fine-tuning techniques including: gradual unfreezing, discriminate learning rate, warm-up learning rate schedule (Pham et al., 2020) to perform effective task adaptation (Gururangan et al., 2020).

4.3 Training Strategies

We apply four training strategies to study the effects of combining text and mate data on our above-mentioned multi-data model's performance. Notice here that we used the pre-trained weights of RoBERTa as the initialization for the textual-feature-extraction-model's backbone in all strategies. We refer to the textual and meta feature extraction parts of the multi-source model are referred as text and meta submodel for short. Our training policies are described as followed:

- Strategy 1 (S1): The parameters of both the text submodel's head and the meta submodel are initialized randomly
- Strategy 2 (S2): The meta submodel will be trained for the task first. Its feature extraction part (all layers except the output one used

for classification) is used to combine with the text submodel. The parameters of the text submodel's head are initialized randomly.

- Strategy 3 (S3): Meta submodel is un-trained when incorporates with the text submodel, which is already fine-tuned with the task.
- Strategy 4 (S4): Both the two submodels are trained/fine-tuned with the classification task before being combined for further training.

4.4 System configuration

Our experiments are conducted on a computer with Intel Core i7 9700K Turbo 4.9GHz, 32GB of RAM, GPU GeForce GTX 2080Ti, and 1TB SSD hard disk.

5 Experimental Results

5.1 Evaluation metrics

For this work, we used the Area Under the Receiver Operating Characteristic Curve (ROC-AUC), a common evaluation metrics for classification tasks. The Receiver Operating Characteristic (ROC) curve shows how well a model classify samples by plotting the true positive rate against the false positive rate at various thresholds. To turn the graph into a numerical metrics, the Area Under Curve (AUC) is then evaluated. A maximum value of 1.0 indicates that the model predicts correctly for all thresholds, and a minimum of 0.0 implies the model gets everything wrong all the time. The formula for ROC-AUC is

$$\text{ROC-AUC} = \int_{0}^{+\infty} \int_{-\infty}^{+\infty} f_1(u) f_0(u-v) du dv \tag{1}$$

where f_1 and f_0 are the density functions.

5.2 Our results

Our results are shown in Table 2 3 4 5

Table 2 compares the effectiveness of traditional machine learning algorithm on metadata. The performance ranges from a ROC-AUC score of 0.5450 with a simple Logistic Regression, to 0.7338 through employing Gradient Boosting across various models. Despite achieving results not as competitive as which of Gradient Boosting, the Multi-Layer Perceptron model was chosen due to its differentiability, which enabled joint training with the textual model (details in Section 3.5).

Table 2: Performance of models using only meta data.

Method	ROC-AUC
Logistic Regression	0.545037
Linear Discriminant Analysis	0.545037
K Nearest Neighbors	0.633251
Decision Tree	0.657217
Gaussian Naive Bayes	0.588978
Support Vector Machine	0.599256
Adaptive Boosting	0.673511
Gradient Boosting	**0.733850**
Random Forest	0.727192
Extra Tree	0.651323
Multi-Layer Perceptron	0.604653

Table 3: ROC-AUC score on public test of combining feature from blocks. Input model is the text content of the news.

Blocks	ROC-AUC
Block 1-6	0.913251
Block 6-12	0.937330
Block 9-12	0.921147
Block 1-12	0.939915
Block 1-12 (Ensemble)	**0.941811**

Most of the aforementioned model's performances are significantly better random guessing, indicating that metadata is an informative predictor for the news classification task.

Table 3 shows the ROC-AUC scores as we tried incorporating different embeddings from different RoBERTa blocks. Specifically, as illustrated in Figure 1, we selected a subset of all embeddings RoBERTa generated, which are then concatenated together and passed through a classifier. Amongst our trials, an ensemble of various combinations across all embeddings achieved the highest AUC-ROC score of 0.9418.

Table 4 highlights one of the major discoveries of our work. It presents our best results for models using only meta- or text data to classify SNS. The

Table 4: Performance of models using only either text or meta data.

Blocks	ROC-AUC
Only meta data	0.7338
Only text data	**0.9628**

Table 5: Performances of multi-data model with different training strategies.

Blocks	ROC-AUC
Strategy 1 (S1)	0.9058
Strategy 2 (S2)	0.9399
Strategy 3 (S3)	0.9552
Strategy 4 (S4)	**0.9628**

performance gap between the two models is significant (more than 0.20 in ROC-AUC score), pointing out that textual features are more predictive than metadata. Besides, using only meta-features is considerably more accurate than random guess (0.7338 ROC-AUC score), indicating that its information can be employed to train a better model.

Table 5 sheds lights on how to effectively combined multi-source data. S1, S2, S3, and S4 in the table refer to the previously-mentioned strategy 1, strategy 2, strategy 3, and strategy 4. S1 and S2 result in the least performance among the four, less than almost 0.05 and 0.02 ROC-AUC score than our second best strategies, S4. Additionally, compared to training with only textual features even better than S1 and inconsiderably worse than S2. This result indicates that fine-tuning text submodel with the task before combining with meta submodel is crucial to achieving high performance.

The worsen results of S1 compared to S2 and S3 compared to S4 points out that pretraining meta submodel before the combination of 2 submodels enhances the overall training.

6 Conclusion

This paper has constructed a comprehensive comparative study to discover the effectiveness of models with multiple inputs and mixed data. We have explored and proposed different training strategies to train the hybrid deep neural architecture for reliable intelligence identification task. By conducting experiments using PhoBERT, we have demonstrated that combining mixed data with particular training plans leads to our best results. With our proposed methods, we have achieved a competitive performance of 94.18% ROC-score on the public test and 94.62% ROC-score on the private test set in VLSP's ReINTEL 2020 campaign.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization.

Yang Cheng and Zifei Fay Chen. 2020. The influence of presumed fake news influence: Examining public support for corporate corrective response, media literacy interventions, and governmental regulation. *Mass Communication and Society*, 23(5):705–729.

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

Pierre Geurts, Damien Ernst, and Louis Wehenkel. 2006. Extremely randomized trees. *Mach. Learn.*, 63(1):3–42.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2001. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

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.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

Rafael Müller, Simon Kornblith, and Geoffrey Hinton. 2019. When does label smoothing help?

Dat Quoc Nguyen and Anh Tuan Nguyen. 2020. Phobert: Pre-trained language models for vietnamese.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *CoRR*, abs/1802.05365.

Quang Pham, Nguyen Viet Anh, Linh Doan, Ngoc Tran, and Ta Thanh. 2020. From universal language model to downstream task: Improving roberta-based vietnamese hate speech detection.

A. Radford. 2018. Improving language understanding by generative pre-training.

Natali Ruchansky, Sungyong Seo, and Yan Liu. 2017. CSI: A hybrid deep model for fake news. *CoRR*, abs/1703.06959.

Jason Sachowski. 2016. *Identify Potential Data Sources*, pages 63–72.

Kai Shu, Limeng Cui, Suhang Wang, Dongwon Lee, and Huan Liu. 2019. Defend: Explainable fake news detection. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*, KDD '19, page 395–405, New York, NY, USA. Association for Computing Machinery.

Kai Shu, Suhang Wang, and Huan Liu. 2017. Exploiting tri-relationship for fake news detection. *CoRR*, abs/1712.07709.

J. Sill, G. Takács, L. Mackey, and D. Lin. 2009. Feature-weighted linear stacking. *ArXiv*, abs/0911.0460.

We Are Social. 2020. Digital 2020 - global digital overview.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

David H. Wolpert. 1992. Stacked generalization. *Neural Networks*, 5(2):241 – 259.

Xinyi Zhou and Reza Zafarani. 2018. Fake news: A survey of research, detection methods, and opportunities. *CoRR*, abs/1812.00315.

An Empirical Study of Using Pre-trained BERT Models for Vietnamese Relation Extraction Task at VLSP 2020

Pham Quang Nhat Minh
Aimesoft JSC
Hanoi, Vietnam
minhpham@aimesoft.com

Abstract

In this paper, we present an empirical study of using pre-trained BERT models for the relation extraction task at the VLSP 2020 Evaluation Campaign. We applied two state-of-the-art BERT-based models: R-BERT and BERT model with entity starts. For each model, we compared two pre-trained BERT models: FPTAI/vibert and NlpHUST/vibert4news. We found that NlpHUST/vibert4news model significantly outperforms FPTAI/vibert for the Vietnamese relation extraction task. Finally, we proposed an ensemble model that combines R-BERT and BERT with entity starts. Our proposed ensemble model slightly improved against two single models on the development data and the test data provided by the task organizers.

1 Introduction

The relation extraction task is to extract entity mention pairs from a sentence and determine relation types between them. Relation extraction systems can be applied in question answering (Xu et al., 2016), detecting contradiction (Pham et al., 2013), and extracting gene-disease relationships (Chun et al., 2006), protein-protein interaction (Huang et al., 2004) from biomedical texts.

In VLSP 2020, the relation extraction task is organized to assess and advance relation extraction work for the Vietnamese language. In this paper, we present an empirical study of BERT-based models for the relation extraction task in VLSP 2020. We applied two state-of-the-art BERT-based models for relation extraction: R-BERT (Wu and He, 2019) and BERT with entity starts (Soares et al., 2019). Two models use entity markers to capture location information of entity mentions. For each model, we investigated the effect of choosing pre-train BERT models in the task, by comparing two Vietnamese pre-trained BERT models: NlpHUST/vibert4news and FPTAI/vibert (Bui et al., 2020). In our understanding, our paper is the first work that provides the comparison of pre-trained BERT models for Vietnamese relation extraction.

The remainder of this paper is structured as follows. In Section 2, we present two existing BERT-based models for relation classification, which we investigated in our work. In Section 3, we describe how we prepared datasets for the two BERT-based models and our proposed ensemble model. In Section 4, we give detailed settings and experimental results. Section 5 gives discussions and findings. Finally, in Section 6, we present conclusions and future work.

2 BERT-based Models for Relation Classification

In the following sections, we briefly describe BERT model (Devlin et al., 2019), problem formalization, and two existing BERT-based models for relation classification, which we investigated in this paper.

2.1 Pre-trained BERT Models

The pre-trained BERT model (Devlin et al., 2019) is a masked language model that is built from multiple layers of bidirectional Transformer encoders (Vaswani et al., 2017). We can fine-tune pre-trained BERT models to obtain the state-of-the-art results on many NLP tasks such as text classification, named-entity recognition, question answering, natural language inference.

Currently, pre-trained BERT models are available for many languages. For Vietnamese, in our understanding, there are three available pre-trained BERT models: PhoBERT (Nguyen and Nguyen, 2020), FPTAI/vibert (Bui et al., 2020), and NlpHUST/vibert4news[1]. Those models are differ-

[1] vibert4news is available on https://huggingface.co/NlpHUST/vibert4news-base-cased

ent in pre-training data, selected tokenization, and training settings. In this paper, we investigated two pre-trained BERT models including FPTAI/vibert and NlpHUST/vibert4news for the relation extraction task. Investigation of PhoBERT for the task is left for future work.

2.2 Problem Formalization

In this paper, we focus on the relation classification task in the supervised setting. Training data is a sequence of examples. Each sample is a tuple $\mathbf{r} = (\mathbf{x}, \mathbf{s}_1, \mathbf{s}_2, y)$. We define $\mathbf{x} = [x_0 ... x_n]$ as a sequence of tokens, where $x_0 =$ [CLS] is a special start marker. Let $\mathbf{s}_1 = (i, j)$ and $\mathbf{s}_2 = (k, l)$ are pairs of integers such that $0 < i \leq j \leq n, 0 < k \leq l \leq n$. Indexes of $\mathbf{s}_1$ and $\mathbf{s}_2$ are start and end indexes of two entity mentions in $\mathbf{x}$, respectively. y denotes the relation label of the two entity mentions in the sequence $\mathbf{x}$. We use a special label OTHER for entity mentions which have no relation between them. Our task is to train a classification model from the training data.

2.3 R-BERT

In R-BERT (Wu and He, 2019), for a sequence $\mathbf{x}$ and two target entities e_1 and e_2 which specified by indexes of $\mathbf{s}_1$ and $\mathbf{s}_2$, to make the BERT module capture the location information of the two entities, a special token '$' is added at both the beginning and end of the first entity, and a special token '#' is added at both the beginning and end of the second entity. [CLS] token is also added to the beginning of the sequence.

For example, after inserting special tokens, a sequence with two target entities "Phi Sơn" and "SLNA" becomes to:

"[CLS] Cầu thủ $ Phi Sơn $ đã ghi bàn cho # SLNA # vào phút thứ 80 của trận đấu ."

The sequence $\mathbf{x}$ with entity markers, is put to a BERT model to get hidden states of tokens in the sequence. Then, we calculate averages of hidden states of tokens within the two target entities and put them through a tanh activation function and a fully connected layer to make vector representations of the two entities. Let H'_0, H'_1, H'_2 be hidden states at [CLS] and vector representations of e_1 and $e2$. We concatenate three hidden states and add a softmax layer for relation classification. R-BERT obtained 89.25% of MACRO F1 on the SemEval-2010 Task 8 dataset (Hendrickx et al., 2010).

2.4 BERT with Entity Start

We applied the BERT model with entity starts (hereinafter, referred to as BERT-ES) presented in (Soares et al., 2019) for Vietnamese relation classification. In the model, similar to R-BERT, special tokens are added at the beginning and end of two target entities. In experiments of BERT-ES for Vietnamese relation classification, different from (Soares et al., 2019), we used entity markers '$' and '#' instead of markers '[E1]', '[/E1]', '[E1]', and '[/E2]'. We did not add [SEP] at the end of a sequence. In BERT-ES, hidden states at the start positions of two target entities are concatenated and put through a softmax layer for final classification. On SemEval-2010 Task 8 dataset, BERT-ES obtained 89.2% of MACRO F1.

3 Proposed Methods

In this work, we applied R-BERT and BERT-ES as we presented in Section 2 for Vietnamese relation extraction, and proposed an ensemble model of R-BERT and BERT-ES. In the following sections, we present how we prepared data for training BERT-based models and how we combined two single models: R-BERT and BERT-ES.

3.1 Data Preprocessing

Relation extraction data provided by VLSP 2020 organizers in WebAnno TSV 3.2 format (Eckart de Castilho et al., 2016). In the data, sentences are not segmented and tokens are tokenized by white spaces. Punctuations are still attached in tokens.

According to the task guideline, we consider only intra-sentential relations, so sentence segmentation is required in data preprocessing. We used VnCoreNLP toolkit (Vu et al., 2018) for both sentence segmentation and tokenization. For the sake of simplicity, we just used syllables as tokens of sentences. VnCoreNLP sometimes made mistakes in sentence segmentation, and as the result, we missed some relations for those cases.

3.2 Relation Sample Generation

From each sentence, for training and evaluation, we made relation samples which are tupes $\mathbf{r} = (\mathbf{x}, \mathbf{s}_1, \mathbf{s}_2, y)$ as described in Section 2. Since in the data, named entities with their labels are provided, a simple way of making relation samples is generating all possible entity mention pairs from entity mentions of a sentence. We used the label OTHER for entity mention pairs that lack relation

No.	Relation	Arguments	Directionality
1	LOCATED	PER - LOC, ORG – LOC	Directed
2	PART–WHOLE	LOC – LOC, ORG – ORG, ORG-LOC	Directed
3	PERSONAL–SOCIAL	PER – PER	Undirected
4	AFFILIATION	PER – ORG, PER-LOC, ORG – ORG, LOC-ORG	Directed

Table 1: Relation types permitted arguments and directionality.

between them. All entity mentions pairs that are not included in gold-standard data are used as OTHER samples.

In the annotation guideline provided by VLSP 2020 organizers, there are constraints about types of two target entities of relation types as shown in Table 1. Thus, we consider only entity mention pairs whose types satisfy those constraints. In training data, sometimes types of two target entities do not follow the annotation guideline. We accepted those entity pairs in making relation samples from provided train and development datasets. However, in processing test data for making submitted results, we consider only entity pairs whose types follow the annotation guideline.

Since the relation PERSONAL-SOCIAL is undirected, for this type, if we consider both pairs (e_1, e_2) and (e_2, e_1) in which e_1 and e_2 are PERSON entities, it may introduce redundancy. Thus, we added an extra constraint for PER-PER pairs that e_1 must come before e_2 in a sentence.

In the training data, we found a very long sentence with more than 200 relations. We omitted that sentence from the training data because that sentence may lead to too many OTHER relation samples.

3.3 Proposed Ensemble Model

In our work, we tried to combine R-BERT and BERT-ES to make an ensemble model. We did that by calculating weighted averages of probabilities returned by R-BERT and BERT-ES. Since in our experiments, BERT-ES performed slightly better than R-BERT on the development set, we used weights 0.4 and 0.6 for R-BERT and BERT-ES, respectively.

4 Experiments and Results

We conducted experiments to compare three BERT-based models on Vietnamese relation extraction data: R-BERT, BERT-ES, and the proposed ensemble model. We also investigated the effects of two Vietnamese pre-trained BERT models on the performance of models.

Relation	Train	Dev
LOCATED	507	304
PART-WHOLE	1,016	402
PERSONAL-SOCIAL	101	95
AFFILIATION	756	489
OTHER	23,904	13,239
Total	26,284	14,529

Table 2: Label distribution of relation samples generated from train and dev data.

Hyper-Parameters	Value
Max sequence length	384
Training epochs	10
Train batch size	16
Learning rate	2e-5

Table 3: Hyper-parameters used in training models.

4.1 Data

The provided training dataset contains 506 documents, and the development dataset contains 250 documents. After data preprocessing and relation sample generation, we obtained relations with label distributions shown in Table 2.

4.2 Experimental Settings

In development, we trained models on the training data and evaluated models on the development data. However, to generate results on the provided test dataset, we trained BERT-based models on the dataset obtained by combining the provided training dataset and the development dataset.

Table 3 shows hyper-parameters we used for training models. We trained all models on a single 2080 Ti GPU.

We used MICRO F1 and MACRO F1 of four relation labels which do not include the label OTHER as evaluation measures.

Model	Pre-trained BERT Model	MACRO F1	MICRO F1
R-BERT	NlpHUST/vibert4news	0.6392	0.7092
R-BERT	FPTAI/vibert	0.596	0.6736
BERT-ES	NlpHUST/vibert4news	**0.6439**	0.7101
BERT-ES	FPTAI/vibert	0.5976	0.6822
Ensemble Model	NlpHUST/vibert4news	0.6412	**0.7108**
Ensemble Model	FPTAI/vibert	0.6029	0.6851

Table 4: Evaluation results on dev dataset.

Model	MACRO F1	MICRO F1
R-BERT	0.6294	0.6645
BERT-ES	0.6276	0.6696
Ensemble Model	**0.6342**	**0.6756**

Table 5: Evaluation results on test dataset.

4.3 Results

Table 4 shows the evaluation results obtained on the development dataset. We can see that using NlpHUST/vibert4news significantly outperformed FPTAI/vibert in both MICRO F1 and MACRO F1 scores. BERT-ES performed slightly better than R-BERT. The proposed ensemble model is slightly improved against R-BERT and BERT-ES in terms of MICRO F1 score.

Table 5 shows the evaluation results obtained on the test dataset. We used NlpHUST/vibert4news for generating test results. Table 5 confirmed the effectiveness of our proposed ensemble model. The ensemble model obtained the best MACRO F1 and the best MICRO F1 score on the test data among the three models.

4.4 Result Analysis

We looked at details of precision, recall, and F1 scores for each relation type on the development data. Table 6 shows results of the ensemble model with vibert4news pre-trained model. PERSONAL-SOCIAL turned out to be a difficult label. The proposed ensemble obtained a low Recall, and F1 score for that label. The reason might be that the relations of PERSONAL-SOCIAL are few in the training data while the patterns of PERSONAL-SOCIAL relations are wider than other relation types.

5 Discussion

In experiments, we compared the effects of two pre-trained BERT models: NlpHUST/vibert4news and FPTAI/vibert on relation extraction. The two pre-trained models have the same BERT architecture (BERT base model) but are different in chosen tokenizers, vocabulary size, pre-training data, and training procedure. Table 7 shows a comparison of the two models.

FPTAI/vibert was trained on 10GB of texts collected from online newspapers while NlpHUST/vibert4news was trained on 20GB of texts in the news domain. FPTAI/vibert used subword tokenization, and vocabulay of FPTAI/vibert was modified from mBERT while tokenization of vibert4news is based on syllables.

We come up with some reasons why using NlpHUST/vibert4news significantly outperformed FPTAI/vibert for Vietnamese relation extraction.

- Pre-training data used to trained vibert4news is much larger than FPTAI/vibert.

- Tokenization used in NlpHUST/vibert4news is based on syllables while FPTAI/vibert used subwords and modified the original vocabulary of mBERT. We hypothesize that syllables which are basic units in Vietnamese are more appropriate than subwords for Vietnamese NLP tasks.

Due to the time limit, we did not investigate PhoBERT (Nguyen and Nguyen, 2020) which used word-level corpus to train the model. As future work, we plan to compare vibert4news that uses syllable-based tokenization with PhoBERT that uses word-level/subword tokenization for Vietnamese relation extraction.

	Precision	Recall	F1
AFFILIATION	0.7615	0.744	0.7528
LOCATED	0.7053	0.7007	0.7030
PART – WHOLE	0.65	0.8085	0.7206
PERSONAL - SOCIAL	0.6136	0.2842	0.3885

Table 6: Precision, Recall, F1 for each relation type on the dev dataset.

	FPTAI/vibert	vibert4news
Data size	10GB	20GB
Data domain	News	News
Tokenization	Subword	Syllable
Vocab size	38168	62000

Table 7: Comparison of NlpHUST/vibert4news and FPTAI/vibert.

6 Conclusion

We have presented an empirical study of BERT-based models for relation extraction task at VLSP 2020 Evaluation Campaign. Experimental results show that the BERT-ES model which uses entity markers and entity starts obtained better results than the R-BERT model, and choosing an appropriate pre-trained BERT model is important for the task. We showed that pre-trained model NlpHUST/vibert4news outperformed FPTAI/vibert for Vietnamese relation extraction task. In future work, we plan to investigate PhoBERT (Nguyen and Nguyen, 2020) for Vietnamese relation extraction to understand the effect of using word segmentation to the task.

References

The Viet Bui, Thi Oanh Tran, and Phuong Le-Hong. 2020. Improving sequence tagging for vietnamese text using transformer-based neural models. *arXiv preprint arXiv:2006.15994*.

Richard Eckart de Castilho, Éva Mújdricza-Maydt, Seid Muhie Yimam, Silvana Hartmann, Iryna Gurevych, Anette Frank, and Chris Biemann. 2016. A web-based tool for the integrated annotation of semantic and syntactic structures. In *Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH)*, pages 76–84, Osaka, Japan. The COLING 2016 Organizing Committee.

Hong-Woo Chun, Yoshimasa Tsuruoka, Jin-Dong Kim, Rie Shiba, Naoki Nagata, Teruyoshi Hishiki, and Jun'ichi Tsujii. 2006. Extraction of gene-disease relations from medline using domain dictionaries and machine learning. In *Biocomputing 2006*, pages 4–15. World Scientific.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38, Uppsala, Sweden. Association for Computational Linguistics.

Minlie Huang, Xiaoyan Zhu, Yu Hao, Donald G Payan, Kunbin Qu, and Ming Li. 2004. Discovering patterns to extract protein–protein interactions from full texts. *Bioinformatics*, 20(18):3604–3612.

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.

Minh Quang Nhat Pham, Minh Le Nguyen, and Akira Shimazu. 2013. Using shallow semantic parsing and relation extraction for finding contradiction in text. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 1017–1021.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30:5998–6008.

Thanh Vu, Dat Quoc Nguyen, Dai Quoc Nguyen, Mark Dras, and Mark Johnson. 2018. VnCoreNLP: A Vietnamese natural language processing toolkit. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 56–60, New Orleans, Louisiana. Association for Computational Linguistics.

Shanchan Wu and Yifan He. 2019. Enriching pre-trained language model with entity information for relation classification. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2361–2364. ACM.

Kun Xu, Siva Reddy, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2016. Question answering on freebase via relation extraction and textual evidence. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2326–2336.

Improving prosodic phrasing of Vietnamese text-to-speech systems

Pham Ngoc Phuong
Thai Nguyen University
phuongpn@tnu.edu.vn

Chung Tran Quang
Hanoi University of Science and Technology
chungtran@vais.vn

Quang Minh Nguyen
Vietnam Artificial Intelligence Solution
minhnq@vais.vn

Quoc Truong Do
Vietnam Artificial Intelligence Solution
truongdo@vais.vn

Abstract

End-to-end TTS architecture which is based on Tacotron2 is the state-of-art system. It breaks the traditional system framework to directly converts text input to speech output. Although it is shown that Tacotron2 is superior to traditional piping systems in terms of speech naturalness, it still has many defects in building Vietnamese TTS: 1) Not good at prosodic phrasing for long sentences, 2) Not good at expression for foreign words. In this paper, we used 2 methods to solve these defects: 1) Pause detection system for predicting and inserting punctuation into long sentences to improve speech naturalness. 2) Translation system for transcribing foreign words to Vietnamese words. In the VLSP 2020 evaluation campaign, our model achieved a mean opinion score (MOS) of 3.31/5 compared to 4.22/5 of humans.

Index Terms— Text-to-speech, TTS, Vietnamese TTS, end-to-end speech synthesis

1 Introduction

Text-to-Speech (TTS) study is widely applied in real-life but it is still a challenge in the field of speech processing. Many techniques have been proposed such as concatenative synthesis (Hunt and Black, 1996), statistical parametric speech synthesis (SPSS). Although concatenative synthesis can reach highly natural synthesized speech, the approach is inherently limited by properties of the speech corpus used for the unit selection process. Meanwhile, SPSS allows product direct speech smoothly and intelligibly by a vocoder. A full SPSS system consists of text analysis, feature generation, and waveform generation modules a, some SPSS techniques are used for Vietnamese TTS: Hidden Markov model (HMM) (Tokuda et al., 2000), Deep neural networks(DNN) (Ze et al., 2013), generative adversarial networks (GAN)(Saito et al., 2017) and End-to-end architectures(Wang et al., 2017). Currently, DNN approaches have gradually replaced HMM models for the duration model and acoustics model. However, the generated voice is often muffled and becomes unnatural. Wavenet (Oord et al., 2016), Wave RNN (Kalchbrenner et al., 2018), GAN (Saito et al., 2017) produces audio with significantly improved naturalness but requirements deep experience and voices that are not as realistic as they are in reality. An end-to-end architecture (Tacontron 2 and WaveGlow vocoder) include five components: linguistic analysis, acoustic model, duration model, parameter generation, and post-filtering are replaced by encoder-attention-decoder networks (Wang et al., 2017; Shen et al., 2018), to be able to effectively optimize the mapping from input text to acoustic features. Finally, a neural vocoder such as Waveglow generated a waveform from the generated mel-spectrogram.

However, in a long sentences or long phrases, speech synthesis results will not be natural. This comes from the fact that human speakers usually break phrases by inserting word transitions instead of punctuation for the sake of expressivity, better comprehension or only taking a breath. The term phrasing is used to describe the phenomenon of grouping words into phrases and separating these phrases with pauses or punctuation inserts. In addition, there are many foreign words in the sentences that are not in the Vietnamese phonetic dictionary. If only replacing foreign words with International Phonetic Alphabet (IPA), the synthesized sentence will not be pronounced in Vietnamese standard. In this paper, 2 methods are applied to synthesize sen-

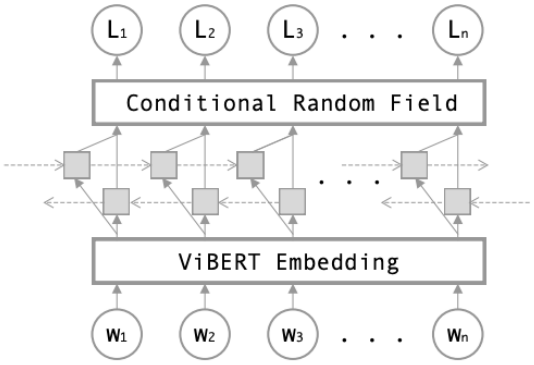


Figure 1: The CaPu model insert the punctuation into the sentences.

tences more naturally: 1) Pause detection module will insert punctuation into sentences to improve prosody of the TTS system, 2) Translation module will transforms foreign words into the Vietnamese standard pronunciation word.

2 Prosodic and pronunciation modeling

2.1 Prosodic modeling

When reading long sentences, the reader always stops at the punctuation or at the position of two or more words of equal syntactic importance (such as noun, verb, etc). So, pause prosodic detection is extremely important affecting the prosody of the TTS system. However, the provided data from the VLSP organization (Trang et al., 2020) was the result of the ASR system, so it had the text only. The synthetic sound quality of the deep neural network depends on the input data. Thus, adding the punctuation at a suitable position can enhance the prosody of our system. To solve this challenging problem, we integrate the Capitalization and Punctuation (CaPu) model (Nguyen et al., 2020) to recover the punctuation of the sentences. The CaPu model not only inserts the punctuation automatically to correct the text format but also places the punctuation at the location relating to breathing.

The CaPu model includes three components that is the embedding layer, the recurrent layer, and the classification layer. More specifically, the embedding layers is ViBERT model that embedded the input sentences to the fixed vectors. The fixed vectors passed through the bidirectional GRU layers. followed by the conditional random field layer to classify the punctuation-tag of each input word. ViBERT is a variation of RoBERTa$_{base}$ model with fewer layers than the original model, it contains 4 encoder layers, the number of heads is 4 and the hidden dimension size is 512. The model has 4 bidirectional GRU layers, the hidden size of GRU cell is 512. The figure 1 depicts CaPu architecture.

To train CaPu model, we collected a huge of text from many domains on the internet including wikipedia, law, politics, etc. This document has the punctuation in accordance with Vietnamese standard style. To mimic the pause of the reader, we use word time-stamp of the ASR system. If the silent time is more than 0.3 second, we put the commas at this silent position. Finally, we trained the CaPu model with the processed data. As a result, CaPu model can insert the punctuation at the proper location by 2 strategy, Vietnamese standard and reader style. Besides, we also added a dot at the end of transcript text to present the end of audio. The result of the CaPu model:

Raw transcript:

cảm giác đó đến một cách đột ngột nhưng mụ xua đuổi nó đi không cho nó chạm tới mụ cũng như không để cho nó chạm tới nền cộng hòa

After add commas to transcript:

cảm giác đó đến một cách đột ngột **,** nhưng mụ xua đuổi nó đi **,** không cho nó chạm tới mụ **,** cũng như không để cho nó chạm tới nền cộng hòa **.**

2.2 Pronunciation modeling

One of the biggest challenges for the VLSP Text-To-Speech (Trang et al., 2020) is that the transcript text has many foreign words. Because foreign words are out of the Vietnamese vocabulary and can not convert to the phoneme directly. This leads to trouble for the participants when joining and building the Vietnamese TTS system. To handle and tackle this problem, we used Vietnamese sound to pronounce these English words. For example, “kuttner” will be pronounced by “cắt nơ”, seeing more examples in Table 1. In order to transform from foreign words to Vietnamese words, we used the popular translation model-Transformer$_{base}$ (Vaswani et al., 2017) model.

The Transformer architecture has two modules, the encoder, and the decoder, and 2 component is connected through an attention mechanism. The Transformer model that we used for this challenge is composed of a stack of N=6 identical layers for both the encoder and decoder.

To train this translation model, we must create a large number of pair of English-Vietnamese words. The total dataset that we produced is more than 1

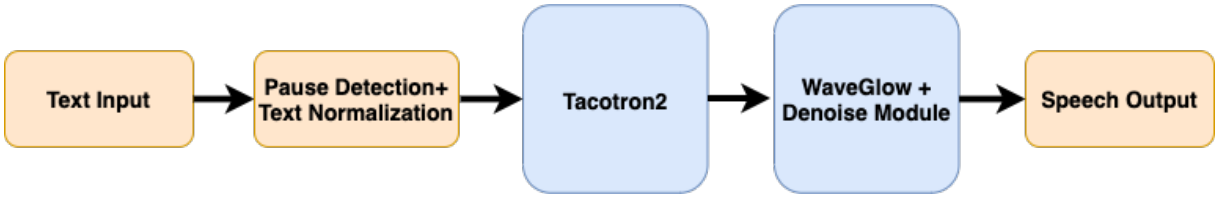


Figure 2: Our TTS pipeline, the input text passes to the pause detection and text normalization module. Subsequently, the processed data passes to Tacotron2 and WaveGlow to generate speech synthesis

English word	Vietnamese word
kuttner	cắt nơ
Anderson	an đơ sơn
vera	vê ra
reme	rê mi

Table 1: Convert English words to Vietnamese words

hundred million pairs. The result of the translation model was displayed in Table 1.

3 Text-to-Speech System

Nowadays, for the TTS task, the end-to-end speech synthesis pipeline consists of two phases, 1. converting text to Mel-spectrogram and 2. converting Mel-spectrogram to waveform synthesis. The model Tacotron2 combining with WaveGlow vocoder is still state-of-the-art for the TTS task. Tacotron2 is a deep neural network receiving a text to predict Mel-spectrogram signal. Then Mel-spectrogram will be converted to waveform thanks to WaveGlow. However, we realized that synthetic speech was noisy. Therefore, we used a denoiser model, attaching at the end of the WaveGlow model.

- *Tacotron2*: The network has two components an encoder and a decoder. We had a small change comparing with the original model. To adapt to the characteristic of the Vietnamese language, the input model was phoneme level instead of character level. Phoneme character passed to the embedding layer, which represented by 512-dimensional. Afterward, these vectors passed through a stack of 3 convolutional layers, followed by single bi-directional LSTM layers to generate the encoded features. The encoder output was consumed by an attention network which yielded a fixed-dimensional vector. Finally, the decoder had the mission of converting this vector to a Mel-spectrogram. To train the Tacotron2 model, we minimized the output of the model with ground trust using mean squared error(MSE).

- *WaveGlow*: The network that we used for the TTS challenge was similar to the original model. The model transformed the output of the Tacotron2 model to the waveform signals. WaveGlow is deployed using only a single network and single cost function, so it is fast, efficient and can produce high quality audio synthesis. The network has 12 coupling layers and 12 invertible 1 x 1 convolutions. In coupling module has 8 layers of dilated convolutions with 512 channels used as residual connections and 256 channels in the skip connection. For the challenge, we used the pre-trained model provided by the author to synthesize the audio.

- *Denoise Module*: This module will reduce the noise of synthetic audio generated from WaveGlow. Firstly, we produced bias audio by using WaveGlow infer a zero Mel-spectrogram with shape 1x80x88. Then both synthetic audio and bias audio will be transformed to Mel-spectrogram by the short-time Fourier transform method. Next, we used the synthetic Mel-spectrogram minus the bias Mel-spectrogram. As a result, we received the final Mel-spectrogram and we used the inverse Fourier transform function to convert it back to audio.

4 Experimental Setup

4.1 Dataset

The duration of the training dataset is about 5-6 hours of a single female speaker and has 7770 audio files. The duration of each file is from 2s to 11s. The sample rate is 44100Hz, 2 channels. To train the model, we resampled to a sample rate of 20500Hz and also convert it to mono channel (1 channel). Besides, we decreased the volume of each file audio by 50%. To reduce noise for the training data, audio in training dataset will be trimmed the silence at start and end position. All transcript text in the dataset is spelled out, for example, “30” is written as “ba mươi”.

Data Processing	Evaluation
No	Speech synthesis can not read the foreign words, the pause in the sentences is unnatural
Pause detection	Speech synthesis can pause at the punctuation correctly, prosody seem naturally
Pause detection + Text Normalization	Speech synthesis can pronounce foreign words.

Table 2: Data processing and evaluate the system

4.2 Experimental Setup

Both CaPu and translation model were implemented by Fairseq (Ott et al., 2019) framework. We used Adam optimizer with beta factor (0.9, 0.98), the learning rate of 0.0005. Conditional Random Field (CRF) loss was applied to train the model and the learning rate scheduler was the inverse square root. The warm-up initial learning rate is 1e-7, and the batch size is 64.

To train the Tacotron2 model, we use GeForce RTX 2080 Ti, 11GB, the learning rate is 1e-3, the weight decay is 1e-6 , the batch size is 64. Adam optimizer with β_1=0.9 and β_2=0.999, $epsilon$=1e-6.

5 Result

We used Tacotron2+Waveglow to evaluate the TTS system. We conducted many experiments relating to data processing, see Table 2 for more detail. Finally, when we combined 2 methods processing pause detection and text normalization, the TTS system yielded speech synthesis naturally. Not only prosody seem natural, but also our system can pronounce foreign words similar to Vietnamese people.

MOS was applied to evaluate the system. The speech synthesis was evaluated by three groups of listeners: speech experts, volunteers, and undergraduates. The listeners will have 5 options to give a score from 1-5: excellent(5), good(4), fair(3), poor(2), 1(bad).

In the VLSP 2020's challenge, as shown in Table 3, our architecture achieved a MOS of 3.31 for the naturalness. For intelligibility, the rate of hearing correct words is 83.10% and the rate of listening to correct syllabi's is 82.90%

	MOS
Our system	3.31
Human	4.22

Table 3: MOS Result for the VLSP Dataset

6 Conclusion and future works

In this paper, we describe our architecture for the Vietnamese Text-to-speech system. For the data from an organization, our approach yielded a MOS of 3.31. By conducting many experiments, we realized that data processing is very important in this challenge. By converting English words to Vietnamese words, also add commas to transcript text, these techniques assist model producing utterance synthesis very naturally.

In the future, we can experiment with more state-of-the-art architecture such as Hifi-Gan, Mel-Gan, Glow-TTS. Also, exploring many challenges of TTS such as how to training TTS with small data, TTS adaptation, etc.

References

Andrew J Hunt and Alan W Black. 1996. Unit selection in a concatenative speech synthesis system using a large speech database. In *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, volume 1, pages 373–376. IEEE.

Nal Kalchbrenner, Erich Elsen, Karen Simonyan, Seb Noury, Norman Casagrande, Edward Lockhart, Florian Stimberg, Aaron van den Oord, Sander Dieleman, and Koray Kavukcuoglu. 2018. Efficient neural audio synthesis. *arXiv preprint arXiv:1802.08435*.

Thai Binh Nguyen, Quang Minh Nguyen, Thi Thu Hien Nguyen, Quoc Truong Do, and Chi Mai Luong. 2020. Improving vietnamese named entity recognition from speech using word capitalization and punctuation recovery models. *Proc. Interspeech 2020*, pages 4263–4267.

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Yuki Saito, Shinnosuke Takamichi, and Hiroshi Saruwatari. 2017. Statistical parametric speech synthesis incorporating generative adversarial networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(1):84–96.

Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. 2018. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4779–4783. IEEE.

Keiichi Tokuda, Takayoshi Yoshimura, Takashi Masuko, Takao Kobayashi, and Tadashi Kitamura. 2000. Speech parameter generation algorithms for hmm-based speech synthesis. In *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 00CH37100)*, volume 3, pages 1315–1318. IEEE.

Nguyen Thi Thu Trang, Nguyen Hoang Ky, Pham Quang Minh, and Vu Duy Manh. 2020. Remaining problems with state-of-the-art techniques in proceedings of the seventh international workshop on vietnamese language and speech processing.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc.

Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al. 2017. Tacotron: Afully end-to-end text-to-speech synthesis model. *arXiv preprint arXiv:1703.10135*.

Heiga Ze, Andrew Senior, and Mike Schuster. 2013. Statistical parametric speech synthesis using deep neural networks. In *2013 ieee international conference on acoustics, speech and signal processing*, pages 7962–7966. IEEE.

Development of Smartcall Vietnamese Text-to-Speech for VLSP 2020

Manh Cuong Nguyen
Smartcall JSC
dodo.proptit.99@gmail.com

Khuong Duy Trieu
Smartcall JSC
duytrkh@gmail.com

Ba Quyen Dam
Smartcall JSC
dambaquyen.ptit@gmail.com

Thu Phuong Nguyen
ICTU, Thai Nguyen University
ntphuong@itcu.edu.vn

Quoc Bao Nguyen
Smartcall JSC and ICTU, Thai Nguyen University

Abstract

An end-to-end text-to-speech (TTS) system (e.g. consisting of Tacotron-2 and WaveGlow vocoder) can achieve the state-of-the art quality in the presence of a large, professionally-recorded training database. However, the drawbacks of using neural vocoders such as WaveGlow include 1) a time-consuming training process, 2) a slow inference speed, and 3) resource hunger when synthesizing waveform from spectral features. Moreover, the synthesized waveform from the neural vocoder can inherit the noise from an imperfect training data. This paper deals with the task of building Vietnamese TTS systems from moderate quality training data with noise. Our system utilizes an end-to-end TTS system that takes advantage of the Tacotron-2 acoustic model, and a custom vocoder combining a High Fidelity Generative Adversarial Networks (HiFiGAN)-based vocoder and a WaveGlow denoiser. Specifically, we used the HiFiGAN vocoder to achieve a better performance in terms of inference efficiency, and speech quality. Unlike previous works, we used WaveGlow as an effective denoiser to address the noisy synthesized speech. Moreover, the provided training data was thoroughly preprocessed using voice activity detection, automatic speech recognition and prosodic punctuation insertion. Our experiment showed that the proposed TTS system (as a combination of Tacotron-2, HiFiGAN-based vocoder, and WaveGlow denoiser) trained on the preprocessed data achieved a mean opinion score (MOS) of 3.77 compared to 4.22 for natural speech, which is the best result among participating systems of VLSP 2020's TTS evaluation.

Index Terms— End-to-end TTS, Tacotron-2, HiFi-GAN, WaveGlow, vocoder

1 Introduction

Text-to-speech synthesis plays a crucial role in speech-based interaction systems. In the last two decades, there have been many attempts to build high quality Vietnamese TTS systems. A data processing scheme proved its efficacy in optimizing naturalness of end-to-end TTS systems trained on Vietnamese found data (Phung et al., 2020). Text normalization methods were explored; utilizing regular expressions and language model (Tuan et al., 2012). New prosodic features (e.g. phrase breaks) were investigated, which showed their efficacy in improving naturalness of Vietnamese hidden Markov models (HMM)-based TTS systems (Dinh et al., 2013; Trang et al., 2013; Phan et al., 2013). Different types of acoustic models were investigated such as HMM (Dinh et al., 2013), deep neural networks (DNN) (Nguyen et al., 2019), and sequence-to-sequence models (Phung et al., 2020). For postfiltering, it was shown that a global variance scaling method may destroy the tonal information; therefore, exemplar-based voice conversion methods were utilized in postfiltering to preserve the tonal information (Tuan et al., 2016). To our knowledge, there is little to none research on vocoders for Vietnamese TTS systems, especially when the training data is moderately noisy.

In the International Workshop on Vietnamese Language and Speech Processing (VLSP) 2020, a TTS challenge (Trang et al., 2020) required participants to build Vietnamese TTS systems from a provided moderately noisy corpus. The corpus included raw text and corresponding audio files. However, the corpus has incorrect pronunciation of a foreign language, the slight buzzer sounds in audio data, and many incorrectly labeled words, which pose significant challenges to participants. For example, a general neural vocoder will learn the buzzer sounds from the corpus, and introduce

it to the synthesized speech.

In previous VLSP 2019's TTS evaluation, Tacotron-2 and WaveGlow neural vocoder were combined to achieve the best speech quality in Vietnamese speech synthesis (Lam et al.). However, HiFiGAN vocoder significantly outperformed WaveGlow vocoder in term of vocoding quality and efficiency (Kong et al., 2020). In the paper, we present the complete steps of building our end-to-end TTS system combining data preprocessing (Phung et al., 2020) and end-to-end modeling which showed that the system addressed the data problems and achieved high performance and high efficiency.

In particular, we introduced a solution that combines HiFiGAN and WaveGlow denoiser as a custom vocoder to enhance the quality of the final synthesized sound. Specifically, in Section II, we present the TTS system architecture consisting of a Tacotron-2 network followed by the HiFiGAN model as a vocoder and the WaveGlow model as a denoiser. The use of HiFiGAN has both improved aggregation speed and reduced resource size, and utilizing WaveGlow denoiser significantly reduces unexpected noise of synthesized speech. The challenges of naturalness, background noise and buzzer noises in the artificial sound were also overcome by combining Tacotron-2, a HiFiGAN-based vocoder and a WaveGlow denoiser.

2 SYSTEM ARCHITECTURE

2.1 Data Preprocessing

We inherited the data processing method (as shown in Figure 1) proposed in (Phung et al., 2020). We remove non-speech segments from the audio files using Voice Activity Detection (VAD) model (Kim and Hahn, 2018). As for textual data, we normalized the original text to lower case without punctuation, then use the results from an Automatic Speech Recognition (ASR) (Peddinti et al., 2015) model to define unvoiced intervals to automatic punctuation to improve the naturalness and prosody of synthesized voices (Phung et al., 2020). Moreover, there is an enormous number of English words in the provided databases, so our solution is to borrow Vietnamese sounds to read the English words. Even, the English words can consist of Vietnamese syllables and English fricative sounds (for example, x sound) if necessary (for instance, "study" becomes 'x-ta-đi'), which can make it easier for the model to learn the fricative sounds. Also, by selecting the pronunciation of English words, we introduced uncommon Vietnamese syllables, which enriched the vocabulary of the training data set. The overall text normalization was carried out using regular expressions and a dictionary. Finally, we manually reviewed and corrected the transcription. The data processing scheme is shown in Figure 1

2.1.1 Voice Activity Detection

We used the Voice Activity Detection (VAD) module to split long audio files of many sentences into short speech segments corresponding to many new sentences. Additionally, large silences at the beginning and the end of each audio were removed. We utilized the a VAD model (Kim and Hahn, 2018) including a Long Short Term Memory Recurrent Neural Network (LSTM-RNN)-based classification.

2.1.2 Automatic Speech Recognition and Speech Punctuation

We utilized a Automatic Speech Recognition (ASR) system to obtain the time stamps of each word or each sound in each sentence. Moreover, the within-sentence pauses were identified and considered as potential punctuation. We marked a pause as a punctuation when its duration is greater than a threshold of 0.12 seconds. Then, the punctuation was added to input text. Without the added punctuation, the Tacotron-2 may align short pauses to any word or phoneme; which significantly reduce the quality of the synthesized voice.

The ASR acoustic model is the state-of-the art Time Delay Neural Network (Peddinti et al., 2015). To achieve the best performance on provided VLSP data, the language model is trained to over-fit the provided data.

2.2 Proposed text-to-speech systems

We proposed a text-to-speech system which is robust to noisy training data. Our system (as shown in Figure 2) was composed of a recurrent sequence-to-sequence feature prediction network called Tacotron-2, which mapped text embedding to acoustic features, followed by a Generative Adversarial Networks for Efficient and High Fidelity Speech Synthesis (HiFiGAN)-based vocoder. When using the HiFiGAN-based vocoder alone, we realized that the synthesized speech was noisy. As a result, we utilized the WaveGlow model to denoise the synthesized sound. Therefore, our proposed speech synthesis system includes a Tacotron-2 as a

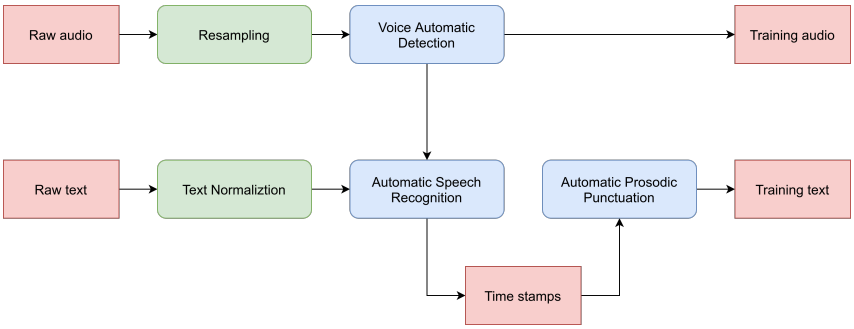


Figure 1: Data Processing Scheme

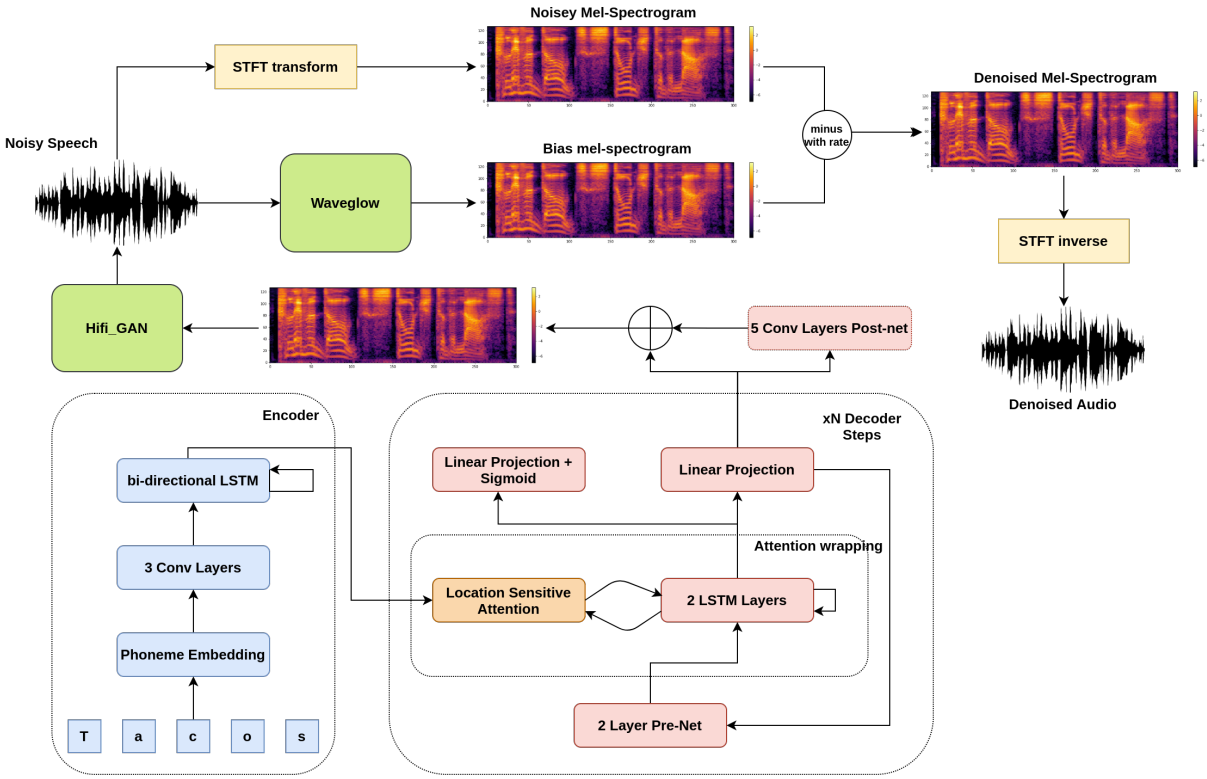


Figure 2: End-to-end system architecture

acoustic model, a HiFiGAN-based vocoder, and a WaveGlow denoiser.

- *Tacotron-2*: In previous VLSP 2019's TTS evaluation, Tacotron-2 was utilized in Vietnamese speech synthesis to achieve the best speech quality (Lam et al.). Therefore, we utilized Tacotron-2 as our TTS acoustic model. Our network architecture was almost similar to (Shen et al., 2017), with some modifications. Firstly, character embedding was used instead of phoneme embedding, which can take advantage of a more flexible and diverse pronunciation dictionary for the Vietnamese dataset. Lastly, we changed some parameters to better fit the data set which has a sampling rate of 22050 Hz, a minimum frequency of 75 Hz, and a amaximum frequency of 7600 Hz.

- *HiFiGAN*: To achieve better vocoding quality and higher efficiency, we utilized a HiFiGAN-based vocoder instead of WaveGlow vocoder. Our network architecture was similar to config V1 (Kong et al., 2020). A mel-spectrogram was used as input of generator and upsamples it through transposed convolutions until the length of the output sequence matches the temporal resolution of a raw waveform.

- *WaveGlow*: Our network architecture was similar to (Prenger et al., 2019). However, we only use WaveGlow for audio's noise reduction. First, we generate bias audio with mel-spectrogram from Tacotron-2 (sigma=0.0). And then we transform bias audio to bias mel-spectrogram. Next, for audio's noise reduction, we took the converted mel-spectrogram from the HiFi-GAN output minus the mel-spectrogram bias by a "denoiser strength" of 0.15. Finally, we obtained the last mel-spectrogram and converted it back to sound.

3 Experiments

The goal of the subjective experiments is to show the efficacy of our proposed method when the training data is noisy. We used the Tacotron-2 acoustic model in combination with different vocoders including 1) WaveGlow vocoder (denoted as WaveGlow), 2) HiFiGAN vocoder (denoted as HiFiGAN), and 3) our proposed method combining HiFiGAN-based vocoder and WaveGlow denoiser (denoted as HiFiGAN+Denoiser). The target natural speech is denoted as NAT.

3.1 Network Training

The original corpus contained 9 hours and 23 minutes of speaking from a female speaker. And after removing the unvoiced parts, the corpus had 8 hours and 21 minutes of speech. All data has been entered to train from scratch for the Tacotron-2 model. We also trained our HiFiGAN and WaveGlow model on the ground truth-aligned predictions.

3.2 Experimental Results

We submitted our proposed system (described in Section 2) to the VLSP 2020's TTS evaluation. The system was evaluated using the VLSP organizer's subjective MOS test. There were 24 participants listening to the stimuli of synthesized and natural speech. The participants gave each utterance a score on a 5-point scale including "very bad", "bad", "fair", "good", and "very good". Details of the results of the second MOS test are given in Table 1.

Our system	NAT
3.77	4.22

Table 1: Average MOS of our proposed system (described in Section 2) from VLSP's TTS evaluation

We conducted the second Mean Opinion Score (MOS) test to evaluate the performance of four vocoders (WaveGlow, HiFiGAN, and HiFiGAN+WaveGlow) in speech synthesis. Each listener listened to 20 test sentences and rate the quality of each sentence in a 5-point scale including "very bad", "bad", "fair", "good", and "very good". In total, there are 20 (sentences) $\times$ 4 (systems) = 80 (trials) [1] in a Latin-square design. We need 80 $\div$ 20 = 4 listeners to cover all the trials.There were 12 participants in the the test.

We summarize the perceptual characteristics of each speech synthesis systems in Table 2. The Figure 3 showed that our proposed system (denoted as HiFiGAN+Denoiser) has a highest MOS. The proposed system is better than natural speech (NAT) due to the fact that the target natural speech is noisy. The results showed that HiFiGAN vocoder outperformed WaveGlow vocoder when the training data is noisy.

We also ran the benchmarks for three models on the same Nvidia GTX 1080 Ti GPU hardware,

[1] Samples are available at: https://proptitclub.github.io/paper/index.html

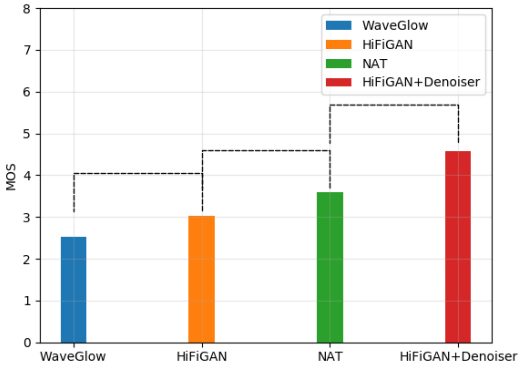


Figure 3: Average MOS of four systems. Dashed lines show statistically significant differences with p-value < 10^{-8}

Systems	Evaluate
WaveGlow	Each pronouncing word has a buzzer, however, the background noise is noticeable
HiFiGAN	The sound quality of each word has been improved, the background noise is moderate
HiFiGAN+Denoiser	The sound is clean

Table 2: Experimental reviews

with the same set of samples to show the inference efficicenty of using HiFiGAN-based vocoder. Statistics of real-time factor (RTF) values, which tells how many seconds of speech are generated in 1 second of wall time, are shown in Table 3. The results show that the speech synthesis rate of the model with HiFiGAN vocoder compared to the model with WaveGlow vocoder is 1.8 times, which hugely improves the speed performance of the system. For the system with both HiFiGAN and WaveGlow, the speed performance is approximate to the model using only HiFiGAN, because the denoising process of WaveGlow is not computationally exhausting. The results indicate that the HiFiGAN-based vocoder has better inference efficiency than the WaveGlow vocoder.

On the other hand, the resource consumption of our proposed model increases due to the use of both HiFiGAN and WaveGlow denoiser. While the number of HiFiGAN's parameters is 13.92 million, the WaveGlow has six times more parameters than HiFiGAN (as shown in Table 4). And the total

Systems	RTF
WaveGlow	4.00
HiFiGAN	7.37
HiFiGAN+Denoiser	7.25

Table 3: RTF results

number of parameters using both models is 101.65 million.

Models	Param (M)
WaveGlow	87.73
HiFiGAN	13.92
HiFiGAN and WaveGlow	101.65

Table 4: Number of parameters

4 CONCLUSION AND FUTURE WORKS

In this report, we have presented our Vietnam TTS system for VLSP 2020. As for the challenge, our approach yields MOS result pretty close to this of natural speech. By testing various solutions to these challenges, we found that combining the methods to develop a custom vocoder played a significant role in the quality of synthesized speech. And the system efficiency was also significantly improved. As a result, the challenges of naturalness, background noise and buzzer noises in the artificial sound have been overcome. We plan to investigate other types of neural vocoders for improving the quality of speech synthesis.

References

Anh Tuan Dinh, Thanh Son Phan, Tat Thang Vu, and Chi Mai Luong. 2013. Vietnamese hmm-based speech synthesis with prosody information. In *Eighth ISCA Workshop on Speech Synthesis, Barcelona, Spain*.

J. Kim and M. Hahn. 2018. Voice activity detection using an adaptive context attention model. *IEEE Signal Processing Letters*, 25(8):1181–1185.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. 2020. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *ArXiv*, abs/2010.05646.

Phung Viet Lam, Phan Huy Kinh, Dinh Anh Tuan, Trieu Khuong Duy, and Nguyen Quoc

Bao. Development of zalo vietnamese text-to-speech for vlsp 2019. http://vlsp.org.vn/sites/default/files/2019-10/VLSP2019-TTS-PhungVietLam.pdf. Accessed: Oct, 2019.

Thinh Van Nguyen, Bao Quoc Nguyen, Kinh Huy Phan, and Hai Van Do. 2019. Development of vietnamese speech synthesis system using deep neural networks. In *Journal of Computer Science and Cybernetics*, volume 34, pages 349–363.

V. Peddinti, D. Povey, and S. Khudanpur. 2015. A time delay neural network architecture for efficient modeling of long temporal contexts. In *Sixteenth Annual Conference of the International Speech Communication Association*.

T. Phan, T. Duong, A. Dinh, T. Vu, and C. Luong. 2013. Improvement of naturalness for an hmm-based vietnamese speech synthesis using the prosodic information. In *The 2013 RIVF International Conference on Computing Communication Technologies - Research, Innovation, and Vision for Future (RIVF)*, pages 276–281.

Viet Lam Phung, Phan Huy Kinh, Anh Tuan Dinh, and Quoc Bao Nguyen. 2020. Data processing for optimizing naturalness of vietnamese text-to-speech system. *arXiv:2004.09607*.

R. Prenger, R. Valle, and B. Catanzaro. 2019. Waveglow: A flow-based generative network for speech synthesis. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3617–3621.

Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, R. J. Skerry-Ryan, Rif A. Saurous, Yannis Agiomyrgiannakis, and Yonghui Wu. 2017. Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions. *CoRR*, abs/1712.05884.

Nguyen Thi Thu Trang, Nguyen Hoang Ky, Pham Quang Minh, and Vu Duy Manh. 2020. Vietnamese text-to-speech shared task vlsp 2020: Remaining problems with state-of-the-art techniques in proceedings of the seventh international workshop on vietnamese language and speech processing (vlsp 2020). In *International workshop on Vietnamese Language and Speech Processing (VLSP 2020)*.

Nguyen Thi Thu Trang, Albert Rilliard, Tran Do Dat, and Christophe d'Alessandro. 2013. Prosodic phrasing modeling for vietnamese tts using syntactic information. In *Proceedings of Interspeech, Lyon, France*.

Dinh Anh Tuan, Phi Tung Lam, and Phan Dang Hung. 2012. A study of text normalization in vietnamese for text-to-speech system. In *Proceedings of Oriental COCOSDA Conference, Macau, China*.

Dinh Anh Tuan, Phan Thanh Son, and Masato Akagi. 2016. Quality improvement of vietnamese hmm-based speech synthesis system based on decomposition of naturalness and intelligibility using non-negative matrix factorization. In *Advances in Information and Communication Technology. ICTA 2016. Advances in Intelligent Systems and Computing, vol 538. Springer, Cham.*

Vietnamese Relation Extraction with BERT-based Models at VLSP 2020

Thuat Nguyen and Hieu Man Duc Trong
Hanoi University of Science and Technology, Hanoi, Vietnam
{thuat.nh163964,hieu.mdt161530}@sis.hust.edu.vn

Abstract

In recent years, BERT-based models have achieved the state-of-the-art performance over many Natural Language Language tasks. Because of that, BERT-based model becomes a trend and is widely used for so many NLP task. And in this paper, we present our approach on how we apply BERT-based model to Relation Extraction shared-task of VLSP 2020 campaign. In detail, we present: (1) our general idea to solve this task; (2) how we preprocess data to fit with the idea and to yield better result; (3) how we use BERT-based models for Relation Extraction task; and (4) our experiment and result on public development data and private test data of VLSP 2020.

1 Introduction

Nowadays, Natural Language Processing (NLP) is a very interesting and necessary field of research. The results of the works in the field of natural language processing can bring many benefits to human. As an interesting task in the field of NLP research, the result of Information Extraction (IE) works in general and Relation Extraction (RE) works in particular can help people a lot on automating text processing tasks. However, compared to other popular languages (e.g., English, Chinese), evaluations and research results for Relation Extraction in Vietnamese language are still limited. In this year's international workshop on Vietnamese Language and Speech Processing (VLSP 2020)[1], for the first time, there is a shared task about Relation Extraction in Vietnamese. This is really great as it means that Relation Extraction in Vietnamese is gaining more attention from the research and industry communities. In the Relation Extraction shared task in VLSP Campaign 2020, organizers will release training, development and test data. Training and development data contain Vietnamese electronic newspapers, labeled entity types of all entity mentions in the articles (there are only three types of entity entities) and labeled relations between entity mentions that belong to the same sentence. In the meantime, the test data also contains the similar information contained in the training and development data (newspapers and entity mentions), but will not be provided with the labels of relation between entities. And participating groups are asked to build learning systems based on training and development data, capable of predicting the relationship labels between entities belonging to the same sentence in the test data. And in the next section of this paper, we describe in detail about VLSP 2020 RE task's dataset, how we preprocess the data and about our BERT-based model's architecture that we use for this year's VLSP RE task.

2 Data and Methodology

2.1 Data

All three sets of data (training, development and test) contain files in WebAnno TSV 3.2 File format[2]. Each file only contains one raw document (electronic newspapers) that has not been split into sentences. There are three types of Named Entities (NE): Locations (LOC), Organizations (ORG), and Persons (PER). And four types of relation between annotated entities; three of four relation types are directed and the last one is undirected. These relation types are described in Table 1.

The detailed information is given in the VLSP 2020 RE task's page[3] and the annotation guideline of this task.

[1] https://vlsp.org.vn/vlsp2020/eval

[2] https://webanno.github.io/webanno/releases/3.6.6/docs/user-guide.html#sect_webannotsv

[3] https://vlsp.org.vn/vlsp2020/eval/re

No.	Relation	Arguments	Directionality
1	LOCATED	PER–LOC, ORG–LOC	Directed
2	PART-WHOLE	LOC–LOC, ORG–ORG, ORG-LOC	Directed
3	PERSONAL–SOCIAL	PER–PER	Undirected
4	ORGANIZATION–AFFILIATION	PER–ORG, PER-LOC, ORG–ORG, LOC-ORG	Directed

Table 1: Relation types in the VLSP 2020 dataset.

2.2 General Idea

In this section, we describe our general idea about how we process data:

- We need to split original raw documents by sentences since the dataset contains only pre-labeled relationships between entities belonging to the same sentence.

- Assuming that there are total n entities in a sentence, we create $\frac{n(n-1)}{2}$ sentences corresponding to $\frac{n(n-1)}{2}$ pairs of entities. Each of these sentences is a data point that is passed to our BERT-based model later. The label for each data point is the relation label between the pair of entities in this sentence.

- These are four types of relation. Three of them are directed, so we create new two undirected relations for each directed relations, depending on whether the directed relation label is on the preceding or following entity in the sentence. See below EXAMPLE I and EXAMPLE II for more clarity.

  EXAMPLE 1: In the sentence: "Hà Nội là thủ đô của Việt Nam", the relation between two entities ("Hà Nội" and "Việt Nam") is PART-WHOLE. This relation label is on the "Việt Nam" entity, which is the entity that comes after in the sentence. We set this data point's label to PART-WHOLE.

  EXAMPLE 2: In the sentence: "Việt Nam có thủ đô là Hà Nội", the relation between two entities ("Hà Nội" and "Việt Nam") is PART-WHOLE. This relation label is on "Hà Nội" entity, which is the entity that comes first in the sentence. We set this data point's label to WHOLE-PART.

- There are many entities in the same sentence but there are no relations between them, so we create a new type of relation called "OTHERS" for them.

- Finally, we pass these data points into our BERT-based model.

In the end, we have a total of seven types of relations.

2.3 Preprocessing data

This section presents details on how we preprocess data. Because the dataset contains only pre-labeled relationships between entities belonging to the same sentence. So we need to split original raw documents by sentences. To do that, we try to use two of the best libraries out there for Vietnamese language processing: VnCoreNLP[4] (VNC) and Underthesea[5] (UTS). In our own experiment, Underthesea seem better to us when compared to VnCoreNLP:

- VNC has problems with Unicode normalized: "Thanh Thủy" will be "Thanh Thuỷ". While UTS seem to have better Unicode normalized.

- VNC has problems with splitting a correct sentence into two sentences. While UTS seems or very rarely has this problem. It is quite hard for us to fix this problem.

- VNC can split sentences perfectly by some characters like single dot, three dots ... while UTS sometimes does not split sentences by these characters. However, we can find, and fix these sentences easily.

Besides, there are some other small problems when we use these two libraries. But results from Underthesea seem to be better than results from VnCoreNLP. So we decide to use Underthesea for preprocessing data.

We follow the following steps to preprocess data:

- Normalize data with "NFC" form.

[4] https://github.com/vncorenlp/VnCoreNLP

[5] https://github.com/undertheseanlp/underthesea

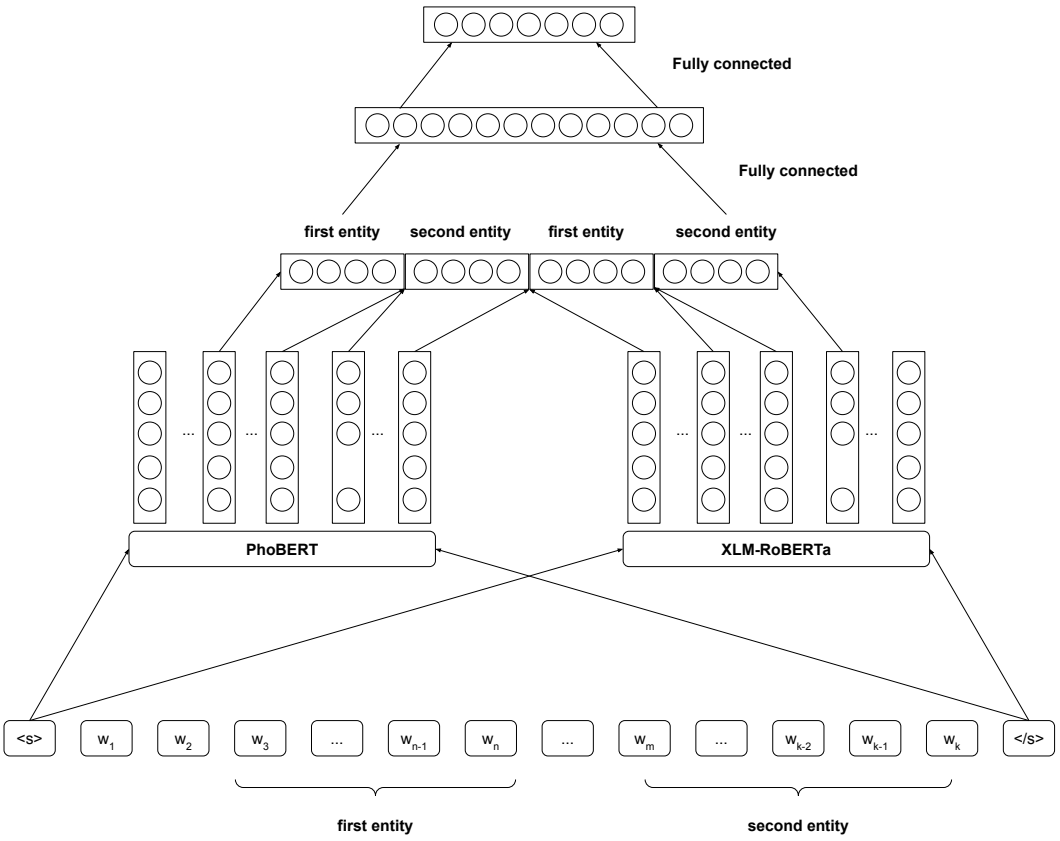


Figure 1: Our BERT-based model for Relation Extraction.

- Using Underthesea to split raw documents to sentences.
- Find and review sentences contain characters like: dot, three dots. However, these characters are not the ending characters of these sentences. Then if there are mistakes in these sentences (two different sentences are combined into a single sentence), we will split these sentences using rules.
- Split sentences by colon punctuation using rules.
- Remove characters that are not alphanumeric (either alphabets or numbers) at the beginning or at the end of an entity.
- Fix the problem with faulty Word Segmentation of Underthesea.

Besides, we also do some other preprocess steps like: Check and fix if there is a relation between entities belonging to different sentences to make sure that data extracted from raw data is correct.

2.4 BERT-based model

In this section, we present our BERT-base model's architecture. We use two BERT-based models that support Vietnamese language: PhoBERT (PB) (Nguyen and Tuan Nguyen, 2020) and XLM-RoBERTa (XLMR) (Conneau et al., 2019). We use these two BERT-based models to generate embedding vectors for each pair of entities of each sentence. Then we combine (using pooling methods) these embeddings into one single embedding vector, and pass it into a multi layer neural network with seven (the number relation types) units and Softmax activation function in the last layer. The architecture of our model is shown in Figure 1.

About details, we follow the following steps to process sentences:

- We pass sentences into the BERT-based models to generate embedding vectors for each pair of entities of each sentence. We try to use both of two BERT-base models PB and XLMR; we also try to use only PB or only XLMR.

- In particular, each entity may have multiple word pieces. So with each entity's word pieces, we try to use and combine embeddings of it from different BERT layers to only one single embedding vector for that word piece. We tried several combinations like: concating embeddings from the last four layers, element wise max pooling embeddings from the last two layers.

- Then, with each entity, we do the same process like each entity's word pieces to generate only one single embedding vector for an entity from its word pieces embedding vectors.

- Each sentence has two entities, so we have two embedding vectors. Let the first entity's embedding vector be h_1; the second entity's embedding vector be h_2. From these two vectors, we generate one single embedding vector for the current sentence: $[h_1, h_2]$.

- Each PB and XLMR model have its own final sentence embedding. In the combination model of PB and XLMR, we concatenate two sentence embedding of these two models to obtain one single sentence embedding vector.

- Finally, we pass the final sentence embedding vector to a multi layer neural network with seven (relation types amount) units and Softmax activation function in the last layer.

3 Experiments and results

In our experiments, we try to use only one of the two BERT-based models (PB or XLMR) and compare with using both models, but using both models always gives much better results. We use Google Colab[6] GPU for training. Since the maximum GPU memory of Colab is 16GB, our biggest model is a combination of fine-tuned PB base model with non fine-tuned XLMR Large (Model 1). We found that if we fine tune PB with high epoch numbers (about 8) and with small learning rate of E-05 can give results that are close to the best we have ever had. And the model results seem more stable when using average pooling instead of using max pooling.

[6] https://colab.research.google.com

Model	Development data	Test data
Model 1	**93.23**	71.19
Model 2	93.10	69.30
Model 3	93.09	**72.06**

Table 2: The performance of the models (Micro-averaged F-score) on the public development data and the private test data.

Each participating team can submit three final results on the test set. The official evaluation measures are micro-averaged F-score. So we choose three models that have the highest micro-averaged F-score on the public development data. Details of the results (on both public development data and private test data) are presented in Table 2.

All of our three best models using PB base and XLMR base model, with PB base is fine-tuned with a learning rate of E-05. Our worst model on the development data (Model 3) give the best result on the private data. We think that two other models may too overfit on the training data tuning on public development data.

With results in Table 2, we achieved the best result with Model 3, ranking the 1st of the scoreboard on the private test set of Relation Extraction shared-task at VLSP 2020 campaign.

4 Conclusion and Future Work

In this paper, we have presented our approach to solve the Relation Extraction task proposed at the VLSP Shared Task 2020. We find out that the BERT-base model is actually really good, since our models are quite simple but achieve acceptable results. In the future, we want to use better GPU to train bigger models like fine tuned PB large with fine tuned XLMR large, since bigger models seem to have better results.

References

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Dat Quoc Nguyen and Anh Tuan Nguyen. 2020. PhoBERT: Pre-trained language models for Viet-

namese. In *Findings of the Association for Computational Linguistics: EMNLP 2020*.

Vietnamese Text-To-Speech Shared Task VLSP 2020: Remaining problems with state-of-the-art techniques

NGUYEN Thi Thu Trang[1,2], NGUYEN Hoang Ky[2], PHAM Quang Minh[2] and VU Duy Manh[2]

[1]*School of Information and Communication Technology*
Hanoi University of Science and Technology
Hanoi, Vietnam
trangntt@soict.hust.edu.vn, trangntt@vbee.vn

[2]*R&D Lab*
Vbee Services and Data Processing Solution Jsc.
Hanoi, Vietnam
kynh@vbee.vn, minhpq@vbee.vn, manhvd@vbee.vn

***Abstract*— The VLSP 2020 is the seventh annual international workshop whose campaign was organized at the Hanoi University of Science and Technology (HUST). This was the third time we organized the Text-To-Speech shared task. In order to better understand different speech synthesis techniques on a common Vietnamese dataset, we conducted a challenge that helps us better compare research techniques in building corpus-based speech synthesizers. Participants were provided with a single training dataset including utterances and their corresponding texts. There are 7,770 utterances of a female Southwest professional speaker (about 9.5 hours). There is a total of 59 teams registered to participate in this shared task, and finally, 7 participants were evaluated online with perceptual tests. The best synthetic voice with Tacotron 2 and Hifigan vocoder with Waveglow denoiser achieved 89.3% compared to the human voice in terms of naturalness, i.e. 3.77 over 4.22 points on a 5-point MOS scale). Some reasons for a quite-big gap between the best synthetic voice with state-of-the-art synthetic techniques and the human voice were: (i) improper prosodic phrasing for long sentences and (ii) wrong/bad pronunciation for loan words.**

Keywords—VLSP Campaign 2020, TTS shared task, speech synthesis, text-to-speech, evaluation, perception test, Vietnamese

I. Introduction

VLSP stands for Vietnamese Language and Speech Processing Consortium. It is an initiative to establish a community working on speech and text processing for the Vietnamese language [2]. The VLSP 2020 was the sixth annual international workshop. The Text-To-Speech (TTS) shared task was a challenge in the VLSP Campaign 2020, which was organized at Hanoi University of Science and Technology. This was the third time we organized the challenge in speech synthesis.

To the best of our knowledge, Vietnamese TTS systems can be divided into three main types:(i) Hidden Markov Model (HMM) based systems, (ii) Deep Neural Network (DNN) based systems, and (iii) state-of-the-art end-to-end systems. HMM-based TTS systems [6][10] and DNN-based TTS systems [4][9] need to provide pause position and loanword pronunciation in the text pre-processing step. Some end-to-end TTS systems, such as Tacotron [3][11], could use a massive amount of text and audio data pairs to learn prosody and loanword modeling directly from the TTS training process. Nevertheless, corpora do not always design to support that purpose.

This shared task has been designed for understanding and figuring out remaining problems in Vietnamese TTS with state-of-the-art speech synthesis techniques on the same dataset. Based on some subjective feedback from listeners of the last year's TTS shared task, three main problems have been raising for this year: prosodic phrasing (mainly focusing on pause detection) [5], text normalization (mainly focusing on loanwords) [6] [8], and removing noise for Internet datasets.

Participants took the released speech dataset, build a synthetic voice from the data and submit the TTS system. We then synthesized a prescribed set of test sentences using each submitted TTS system. The synthesized utterances were then imported to an online evaluation system. Some perception tests were carried out to rank the synthesizers focusing on evaluating the intelligibility and the naturalness of participants' synthetic utterances.

The rest of this paper is organized as follows. Section II presents the common dataset and its preparation. Section III introduces participants and a complete process of the TTS shared task in VLSP Campaign 2020. We then show the evaluation design and experimental results in Section IV. We finally conclude the task and give some possible ideas for the next challenge in Section V.

II. Common Dataset

The topic of this shared task is to address remaining problems of TTS systems using state-of-the-art synthesis techniques. Based on some analyses on the previous task results, aforementioned, we raised the following issues for this shared task: (i) prosodic phrasing (focusing on pause detection for long input sentences), (ii) text normalization (focusing on expanding loanwords), and (iii) removing background noises (of Internet audios).

Due to the topic of this year's task, we decided to collect audiobooks from the Internet. Vbee Jsc supported to build the dataset for this task. The corpus was taken from a novel called "Bell to Whom the Soul" by Hemingway, a famous American novelist. Audio stories were downloaded manually, divided into 28 long audio files, each had 30 to 60 minutes in length. These files were then automatically split into smaller audio files that are less than 10 seconds in length (using Praat scripting tool). After this process, the number of sound files was up to nearly 20,000 sound files with different lengths.

However, approximately 10,000 sound files that were too short in length (i.e. less than 750 ms) were discarded. Next, we used the ASR API of Vais Jsc to convert the remaining 10,000 audio files into text. These data were checked by the teams participating in the contest. Each team only had to check xxx files for participation. Finally, 7,770 best quality utterances and their corresponding texts were selected as the final dataset. Even though the speaker's voice was professional and pretty, the voice still contained some background noise due to the recording device's low quality.

III. PARTICIPANTS

For TTS shared task this year, participants had to follow a complete process (Fig. 1), which was managed in the website of the TTS shared task of VLSP Campain 2020 (https://tts.vlsp.org.vn).

First, each team registered to participate in the challenge. They were then provided with accounts to log into. On this site, all teams were asked to check the audio files to see if they match the corresponding text and edit if necessary. If they found that the text was exactly the content of the audio, they voted for that transcription. Each audio file needs to be checked by at least 3 teams. Audio files that had no vote after the validation process, we had to check them manually. The participants who completed the required task were asked to send their user license agreement with valid signatures. They were then able to download the training dataset. The dataset includes utterances and their corresponding texts in a text file.

Participants were asked to build only one synthetic voice from the released database. All teams had 20 days for training and optimizing their voices. Each team then submitted the result with a TTS API following the announced specification requirement. We also supported teams that could not deploy their TTS systems to a public server by accepting their docker images that contain the TTS API.

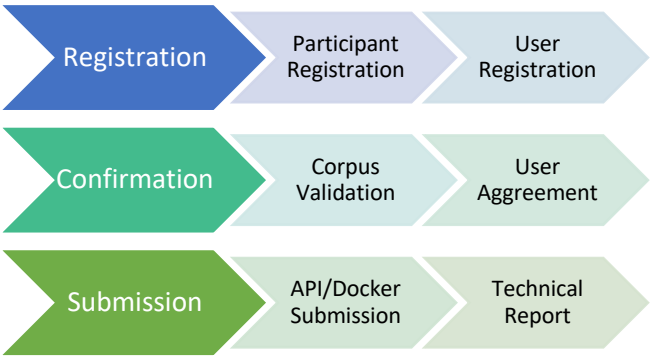


Fig. 1. A complete process for participating TTS shared task VLSP 2020.

We then synthesized audio files from the text files in the test dataset using teams' TTS API. Synthesized files will be evaluated. After receiving evaluation results, the teams proceed to write and submit technical reports.

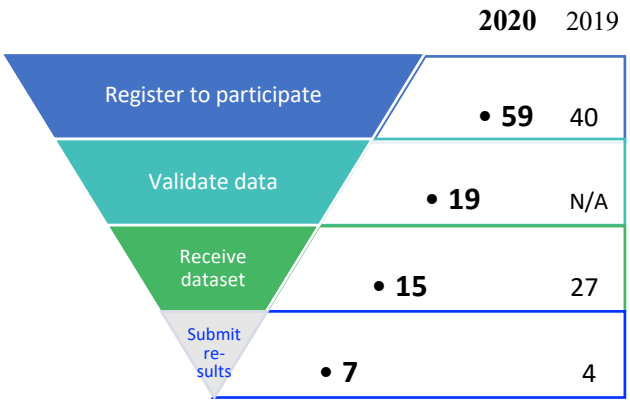


Fig. 2. Participants in VLSP TTS 2020 and 2019.

Fig. 2 compares the number of participants of last year to this year. Fifty-nine teams registered for this year's challenge. Unlike last year, participants were asked to validate the provided dataset, and 19 joined the data validation process, and 15 teams obtained the data after sending the signed user agreement. Finally, nine teams, compared to four in 2019, submitted their TTS system. We synthesized testing audio through the TTS API of each team. Unfortunately, we could not use the TTS API of the two teams due to problems with their TTS system or their server. Table I gives the list of participants that had final submissions to the VLSP TTS shared task 2020.

TABLE I. LIST OF TEAMS PARTICIPATING IN VSLP TTS 2020

No	Team ID	Affiliation	Submission
1	Team1	Unaffiliated	API (error)
2	Team2	Smartcall	API
3	Team3	Unaffiliated	Docker image (error)
4	Team4	Viettel Telecom	API
5	Team5	IC	IC
6	Team6	VAIS	API
7	Team7	Falcon	API
8	Team8	Sun Asterisk Inc.	Docker Image
9	Team9	UET	Docker Image

IV. EVALUATION

Perceptual testing was chosen for evaluating synthetic voices. First, an intelligibility test was conducted to measure the understandability, then the MOS test, which allowed us to score and compare the global quality of TTS systems with respect to natural speech references. All subjects conducted the online evaluation via a web application. This online evaluation system was built by the School of Information and Communication Technology, Hanoi University of Science and Technology, and Vbee Jsc. This system was integrated into https://tts.vlsp.or.vn.

They first registered on the website with necessary information including their hometowns, ages, genders, occupations. They were trained on how to use the website and how to conduct a good test. They were strictly asked to do the test in a controlled listening condition (i.e. headphones and in a quiet distraction-free environment). To ensure that the subjects focused on the test, we designed several sub-tests for each test due to a big number of testing voices (i.e. 8 voices including natural speech). As a result, each sub-test lasts from 25 to 30 minutes.

On completion of any sub-test, or after logging in again, a progress page showed listeners how much they had completed. Detailed instructions for each sub-test were only shown on the page with the first part of each sub-test; subsequent parts had briefer instructions in order to achieve a simple layout and a focussed presentation of the task.

In order to address the issue of duplicate contents of stimuli, we adopted the Latin square (nxn) [1] for all sub-tests, where n is a number of voices in the sub-test. To be more specific, each subject listened to one n^{th} of the utterances per voice, without any duplicate content. With the Latin square design, the number of subjects should be at least twice more than the ones with the normal design.

Stimuli were randomly and separately presented only once to subjects. Each stimulus was an output speech of a TTS system or a natural speech for a sentence. Details of the two tests are described in the following subsections.

A. *Intelligibility Test*

In the intelligibility test, subjects were asked to write down the text of the audio they heard (Fig. 3). The subjects might listen again a second time if they do not hear clearly or have long sentences. They only listened to the utterances the third time when the subjects were distracting, or the sentence were very long.

TABLE II. DESIGN FOR INTELLIGIBILITY SUB-TESTS

Sub-test 1	**Sub-test 2**	**Sub-test 3**
IntelligibilityTest-1	IntelligibilityTest-2	IntelligibilityTest-3
Team 7 Team 8 Team 9	NATURAL Team 5 Team 6	NATURAL Team 2 Team 4

There are three sub-tests in the intelligibility test, following the Latin Square design aforementioned. In each sub-test, there were 3 voices of 3 different teams with or without the natural speech reference (NATURAL). Details for each sub-test is presented in Table II. Each sub-test included voices of two (sub-test 2 and sub-test 3) or three teams (sub-test 1). The natural speech was put in both sub-test 2 and sub-test 3 for more reference. As a result, each sub-test had a total of 3 voices.

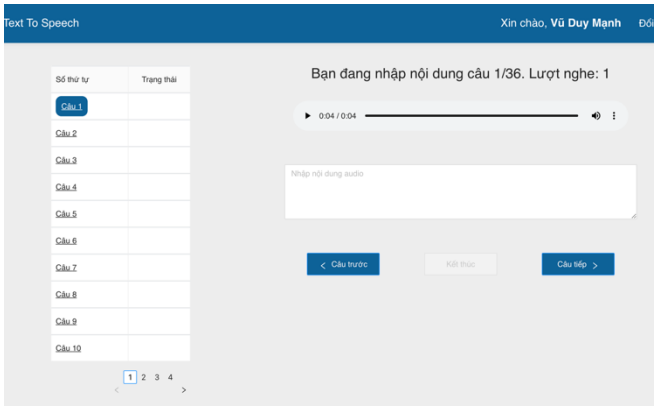


Fig. 3. Online Tool for Intelligibility Test.

Twenty-seven subjects participated in this test. There were two main types of subjects who participated in the test: (i) 19 students (19-22 years-old, 10 females) from Hanoi University of Science and Technology, VNU University of Science; (ii) 8 speech experts (23-38 years-old, 4 female).

The testing dataset included 36 sentences. Each subject needs to participate in at least two of the three sub-tests.

B. *MOS Test*

Subjects (i.e. listeners) were asked to assess by giving scores to the speech they had heard (Fig. 4). When taking this test, subjects listen to the voice once, unless they do not hear it clearly, then listen for a second time.

Subjects randomly listened to utterances and then gave their scores for the naturalness of the utterances. The question presented to subjects was "How do you rate the naturalness of the sound you have just heard?". Subjects could choose one of the following five options (5-scale):

- 5: Excellent, very natural (human speech)
- 4: Good, natural
- 3: Fair, rather natural
- 2: Poor, rather unnatural (rather robotic)
- 1: Bad, very unnatural (robotic).

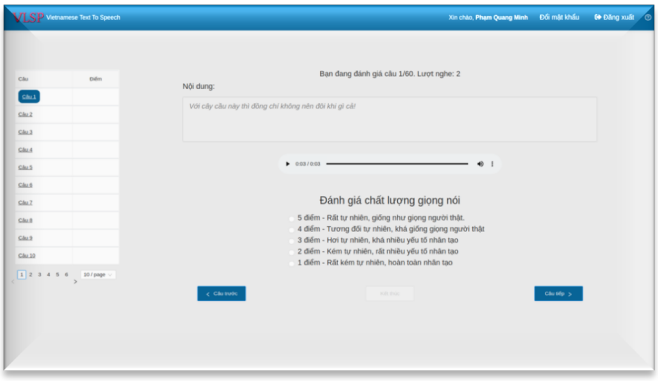


Fig. 4. Online Tool for MOS Test.

Testing text set includes 60 sentences. There are two sub-tests, including 60 random utterances each (taken from 480 utterances). Table III illustrates the design for the two MOS sub-tests. We put the natural speech (NATURAL) as a reference in both sub-tests. Due to an odd number of final participated teams, sub-test 1 included 3 teams (Team 2,4,5) while sub-test 2 had voices from the remaining 4 teams (Team 6,7,8,9).

TABLE III. DESIGN FOR MOS TEST SUB-TESTS

Sub-test 1	**Sub-test 2**
MOS Test 1	MOS Test 2
NATURAL Team 2 Team 4 Team 5	NATURAL Team 6 Team 7 Team 8 Team 9

Subjects participated in two sub-tests for voices built from the common dataset. Due to a rather big number of voices in each sub-test (i.e. 5 including the natural reference), we let the subjects to heard randomly half of the utterances for each voice. The number of subjects who listened to each sub-test was 48 (20 females). Each subject needs to participate in all two sub-tests, estimated at 25 to 30 minutes.

V. EVALUATION RESULTS

A. *Intelligibility Score*

Due to a large number of loanwords in the test set, the intelligibility results were not good, at about 68-89% at both word and syllable levels, even with natural speech. The subjects might do not know how to write these loanwords or present different orthography from the original text. We should have a special design and more analyses for this type of test in the future.

B. *MOS Score*

The perceptual evaluation of the general naturalness was carried out on different voices of participants and a natural speech reference (NATURAL) of the same speaker as the

training corpus. Fig. 5 and Table IV show the final MOS test results. Only three teams submitted technical reports, i.e. Team2, Team6, and Team7.

We can see that Team2 was the best team (i.e. 3.769) – about 89.3% compared to the natural speech (i.e. 4.220/5). This team adopted Tacotron-2 as the acoustic model, and HiFi-GAN as a real-time vocoder, and Waveglow as a denoiser. Team7 was the second place with a 3.698 score (only less than the first place 0.07 point). This team used FastSpeech and PostNet, which could be considered as a faster acoustic model, compared to Tacotron-2 or only FastSpeech. Team6 was the fifth place with a 3.313 score. Their acoustic model was Tacotron2, and their vocoder was Waveglow.

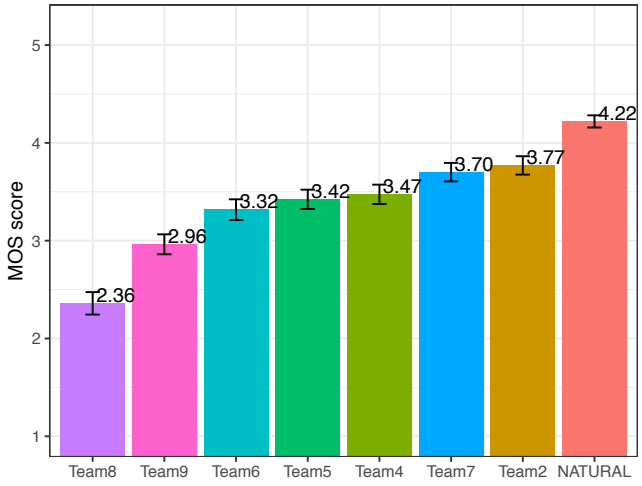


Fig. 5. MOS Test Final Results.

TABLE IV. MOS TEST RESULTS WITH SYNTHESIS TECHNIQUES

Testing voice	MOS Score (5-scale)	Synthesis Techniques
NATURAL	4.220	
Team2	**3.769**	• Acoustic model: Tacotron 2; Vocoder: HiFi-GAN; • Denoiser: Waveglow
Team7	3.698	• Acoustic model: FastSpeech + PostNet; • Vocoder: Waveglow
Team6	3.313	• Acoustic model: Tacotron 2; • Vocoder: Waveglow

Although using state-of-the-art synthesis techniques that lead to a high-quality synthetic voice, there were still some remaining problems in the results of participants. Some reasons were found for a quite-big gap between the best synthetic voice with state-of-the-art synthetic techniques and the human voice: (i) improper prosodic phrasing for long sentences and (ii) wrong/bad pronunciation for loan words.

C. Analysis and Discussion

Several two-factorial ANOVAs were run on the MOS results, illustrated in Table V. The two factors were the TTS system (8 levels) and the Sentence (60 levels) or the Subject (48 levels). All factors and their interactions in both ANOVAs had significant effect ($p<0.0001$).

The TTS system factor alone explained an important part of the variance over levels of both Sentence (29%) and Subject factors (30%). The Sentence factor explained only about 8% of the variance (partial $\eta^2 = 0.08$) while the Subject did 19% (partial $\eta^2 = 0.19$). The interaction between the System and Sentence or Subject explained a quite important part of the variance, i.e. 21% and 14% respectively.

TABLE V. ANOVA RESULTS OF MOS TEST

Factor	df	df error	F	p	η^2
System	7	5,688	335.38	0.0000	0.29
Sentence	59	5,688	8.71	0.0000	0.08
System:Sentence	412	5,688	3.57	0.0000	0.21
System	7	5,798	353.37	0.0000	0.30
Subject	47	5,798	29.29	0.0000	0.19
System:Subject	314	5,798	2.89	0.0000	0.14

We did observe the sentences with bad scores and found that they were long sentences or had a number of loanwords. Synthetic utterances having consecutive loanwords are extremely bad intelligible. These problems led to bad scores for both Intelligibility and MOS Test.

VI. CONCLUSIONS

We did some valuable experiments on TTS systems from different participants using a common dataset in the TTS shared task in the VLSP Campaign 2020. Participants had to validate a piece of training data before receiving the common dataset. There are 7,770 utterances of a female Southwest professional speaker (about 9.5 hours) in the released training dataset. Although using state-of-the-art synthesis techniques that lead to a high-quality synthetic voice, there were still some remaining problems in the results of participants. The best synthetic voice with Tacotron 2 and Hifigan vocoder with Waveglow denoiser achieved 89.3% compared to the human voice, i.e. 3.77 over 4.22 point on a 5-point MOS scale). Some reasons were found for a quite-big gap between the best synthetic voice with state-of-the-art synthetic techniques and the human voice: (i) improper prosodic phrasing for long sentences and (ii) wrong/bad pronunciation for loan words. For the next speech synthesis task of the VLSP Campaign in 2021, we may have more advanced topics for Vietnamese speech synthesis, such as speaker adaptation or expressive speech synthesis.

ACKNOWLEDGMENT

The VLSP 2020 TTS shared task was mainly supported by the R&D Lab, Vbee Services and Data Processing Solution Jsc, and School of Information and Communication Technology. They supported this shared task in developing, deploying, and conducting the online evaluation, based on perception tests as well as building the dataset for the challenge. This task was funded by the Vingroup Innovation Foundation (VINIF) under the project code DA116_14062019 / year 2019. We would like to thank Vais Jsc. for their ASR in building the dataset, and last but not least, the subjects who gave time and effort for the experiments.

REFERENCES

[1] Cochran William G. and Cox Gertrude M. "*Experimental Designs, 2nd Edition*". Wiley, 2 edition, April 1992. ISBN 0471545678.

[2] Luong Chi Mai. "*Special Issue in VLSP 2018*". Journal of Computer Science and Cybernetics, V.34, N.4 (2018).

[3] Shen, J., Pang, R., Weiss, R.J., et al. 2017. "*Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions*". 2018 IEEE

International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018.

[4] H. Ze, A. Senior, and M. Schuster, *"Statistical Parametric Speech Synthesis using Deep Neural Networks"* on 2013 IEEE International Conference on Acoustics, Speech and Signal Processing 2013, pp. 7962–7966. IEEE.

[5] Nguyen Thi Thu Trang, Albert Rilliard, Tran Do Dat, and Christophe d'Alessandro, *"Prosodic Phrasing Modeling for Vietnamese TTS using Syntactic Information"* in 15th Annual Conference of the International Speech Communication Association. Singapore. 2014.

[6] Nguyen Thi Thu Trang, Alessandro Christophe, Rilliard Albert, and Tran Do Dat. *"HMM-based TTS for Hanoi Vietnamese: Issues in Design and Evaluation*" in 14th Annual Conference of the International Speech Communication Association (Interspeech 2013), pages 2311-2315. Lyon, France, August 2013b. ISCA.

[7] Nguyen Thi Thu Trang, Pham Thi Thanh, and Tran Do Dat. *"A method for Vietnamese Text Normalization to Improve the Quality of Speech Synthesis"* in Proceedings of the 2010 Symposium on Information and Communication (SoICT 2010), Hanoi, Vietnam. 2010.

[8] Nguyen Thi Thu Trang, Dang Xuan Bach, and Nguyen Xuan Tung. *"A Hybrid Method for Vietnamese Text Normalization"* in Proceedings of the 2019 3rd International Conference on Natural Language Processing and Information Retrieval (NLPIR 2019). Japan. 2019.

[9] Nguyen Van Thinh, Nguyen Quoc Bao, Phan Huy Kinh, Do Van Hai, *Development of Vietnamese Speech Synthesis System using Deep Neural Networks*, Journal of Computer Science and Cybernetics, V.34, N.4 (2018), 349-363.

[10] Vu Thang Tat, Luong Mai Chi, and Nakamura S. *"An HMM-based Vietnamese Speech Synthesis System"* in Proceedings of the Oriental COCOSDA International Conference on Speech Database and Assessments, pages 116–121, Beijing, China, 2009.

[11] Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, YonghuiWu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, YingXiao, Zhifeng Chen, Samy Bengio, et al. *"Tacotron:Towards End-to-end Speech Synthesis"* in 18th Annual Conference of the International Speech Communication Association. Sweden. 2017.

ReINTEL Challenge 2020: A Multimodal Ensemble Model for Detecting Unreliable Information on Vietnamese SNS

Nguyen Manh Duc Tuan
Toyo Unversity, Japan
ductuan024@gmail.com

Pham Quang Nhat Minh
Aimesoft JSC, Vietnam
minhpham@aimesoft.com

Abstract

In this paper, we present our methods for unrealiable information identification task at ReINTEL Challenge 2020. The task is to classify a piece of information into reliable or unreliable category. We propose a novel multimodal ensemble model which combines two multimodal models to solve the task. In each multimodal model, we combined feature representations acquired from three different data types: texts, images, and metadata. Multimodal features are derived from three neural networks and fused for classification. Experimental results showed that our proposed ensemble model improved against single models in term of AUC score. We obtained 0.9445 AUC score on the private test of the challenge.

1 Introduction

Recently, fake news detection have received much attention in both NLP and data mining research community. This year, for the first time, VLSP 2020 Evaluation Campaign held ReINEL Challenge (Le et al., 2020) to encourage the development of algorithms and systems for detecting unreliable information on Vietnamese SNS. In ReINTEL Challenge 2020, we need to determine whether a piece of information containing texts, images, and metadata is reliable or unreliable. The task is formalized as a binary classification problem and training data with annotated labels was provided by VLSP 2020 organizers.

In this paper, we present a novel multimodal ensemble model for identifying unreiable information on Vietnamese SNS. We use neural networks to obtain feature representations from different data types. Multimodal features are fused and put into a sigmoid layer for classification. Specifically, we use BERT model to obtain feature representations from texts, a multi-layer perceptron to encode metadata and text-based features, and a fine-tuned VGG-19 network to obtain feature representations from images. We combined two single models in order to improve the accuracy of fake news detection. Our proposed model obtained 0.9445 ROC AUC score on the private test of the challenge.

2 Related Work

Approaches to fake news detection can be roughly categorized into categorises: content-based methods, user-based methods and propagation-based methods.

In content-based methods, content-based features are extracted from textual aspects, such as from the contents of the posts or comments, and from visual aspects. Textual features can be automatically extracted by a deep neural network such as CNN (Kaliyar et al., 2020; Tian et al., 2020). We can manually design textual features from word clues, patterns, or other linguistic features of texts such as their writing styles (Ghosh and Shah, 2018; Wang et al., 2018; Yang et al., 2018). We can also analyze unreliable news based on the sentiment analysis (Wang et al., 2018). Furthermore, both textual and visual information can be used together to determine fake news by creating a multimodal model (Zhou et al., 2020; Khattar et al., 2019; Yang et al., 2018).

We can detect fake news by analysing social network information including user-based features and network-based features. User-based features are extracted from user profiles (Shu et al., 2019; Krishnan and Chen, 2018; Duan et al., 2020). For example, number of followers, number of friends, and registration ages are useful features to determine the credibility of a user post (Castillo et al., 2011). Network-based features can be extracted from the propagation of posts or tweets on graphs (Zhou and Zafarani, 2019; Ma et al., 2018).

3 Methodology

In this section, we describe methods which we have tried to generate results on the private test dataset of the challenge. We have tried three models in total and finally selected two best models for ensemble learning.

3.1 Preprocessing

In the pre-processing steps, we perform following steps before putting data into models.

- We found that there are some emojis written in text format such as ":)", ";)", "=]]", ":(", "=[", etc. We converted those emojis into sentiment words "happy" and "sad" in Vietnamese respectively.
- We converted words and tokens that have been lengthened into short form. For example, "Coooool" into "Cool" or "*****" into "**".
- Since many posts are related to COVID-19 information, we changed different terms about COVID-19 into one term, such as "covid", "ncov" and "convid" into "covid", for consistency.

Since meta-data of news contains a lot of missing values, we performed imputation on four original metadata features. We used the mean values to fill missing values for three features including the number of likes, the number of shares, and the number of comments. For the timestamp features, we applied the MICE imputation method (Azur et al., 2011).

We found that there are some words written in incorrect forms, such as 's.áthại' instead of 'sát hại'. One may try to convert those words into standard forms, but as we will discuss in Section 4, keeping the incorrect form words actually improved the accuracy of models.

We converted the timestamp feature into 5 new features: day, month, year, hour and weekday. In addition to metadata features provided in the data, we extracted some statistic information from texts: number of hashtags, number of urls, number of characters, number of words, number of question-marks and number of exclaim-marks. For each user, we counted the number of unreliable news and the number of reliable news that the user have made and the ratio between two numbers, to indicate the sharing behavior (Shu et al., 2019). We also created a Boolean variable to indicate that a post contains images or not. In total, we got 17 features including metadata features. All the metadata-based features will be standardized by subtracting the mean and scaling to unit variance, except for the Boolean feature.

3.2 Model Architecture

Figure 1 shows the general model architecture of three models we have tried. In all models, we applied the same strategy for image-based features and meta-data based features. For metadata-based features, we passed it into a fully-connected layer layer with batch normalization. We found that there are posts having one or more images and there are posts having have no image. For posts containing images, we randomly chose one image as the input. For other posts, we created a black image (all pixels have zero values) as the input. We then fine-tuned VGG-19 model on the images of the training data. After that, we used the output prior the fully-connected layer as image-based features. Instead of taking averages of all vectors of pixels, we applied the attention mechanism as shown in Figure 1b to obtain the final representation of images.

In the following sections, we describe three variants that we made from the general architecture.

Model 1

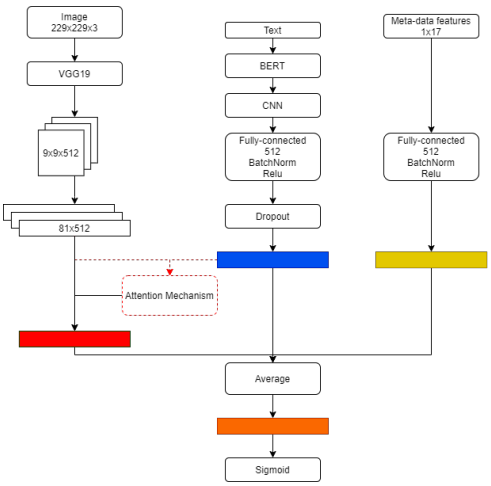
In the first model (Figure 2a), we obtained the embedding vector of a text using BERT model (Devlin et al., 2019). After that, we used 1D-CNN (Kim, 2014) with filter sizes 2, 3, 4, and 5. By doing that, we can use more information from different sets of word vectors for prediction. We flattened and concatenated all the output from 1D-CNN and passed into a fully-connected layer with with a batch normalization layer. Finally, we took averages of features of texts, images and metadata and passed them into a sigmoid layer for classification.

Model 2

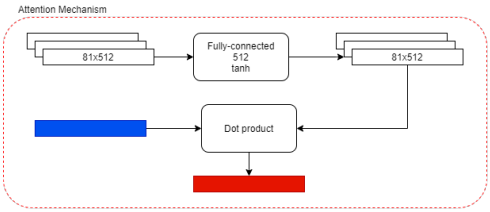
In the second model (Figure 2b), there are some changes in comparison with the first model. After passing the embedding vectors through various layers of 1D-CNN, we stacked those outputs vertically and passed into three additional 1D-CNN layers.

Model 3

In the third model (Figure 2b), we just slightly changed the second model by adding a shortcut connections between input and the output of each 1D-CNN layer.

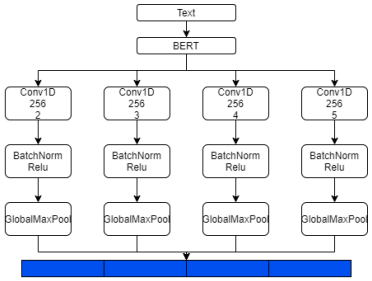


(a) General structure for each model.

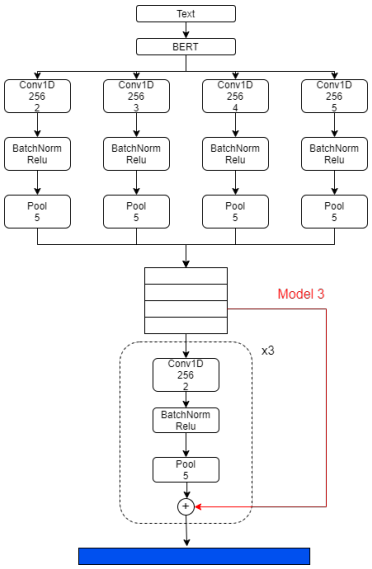


(b) Attention Mechanism.

Figure 1: General Model Architecture



(a) Model 1.



(b) Model 2 and 3.

Figure 2: Text-based features extractor for each model.

Ensemble Model

For the final model, we selected two best models among three above models and took averages of probabilities returned by the two models to obtain the final result.

4 Experiments and Results

In experiments, we used the same parameters as showed in Table 1 for all proposed models[1]. We reported ROC-AUC scores on the private test data.

In the first experiment, we compared two ways of preprocessing texts: 1) converting words in incorrect forms into corrected forms; and 2) keeping the incorrect forms of words. The text is put through PhoBERT (Nguyen and Nguyen, 2020) to get the embedded vectors. In this experiment, we did not apply the attention mechanism. Table 2 shows that keeping the original words obtained better ROC-AUC score.

Next, we compared the effects of two different pre-trained BERT models for Vietnamese: PhoBERT and Bert4news[2]. Table 3 shows that Bert4news model is significantly better than PhoBERT model. Furthermore, when we added the proposed attention mechanism to get feature representations for images, we obtained 0.940217

[1]Our code: https://github.com/dt024/vlsp2020_toyoaimesoft

[2]Bert4News is available on: https://github.com/bino282/bert4news

Hyper-parameter	Value
FC layers	512
Dropout	0.2
Pooling size	5
1D-Conv filters	256
Learning parameter	2e-5
Batch size	16

Table 1: Parameters Setting

Exp	ROC-AUC
Convert words to correct forms	0.918298
Keep words in incorrect forms	0.920608

Table 2: Two ways of preprocessing texts.

Exp	ROC-AUC
PhoBERT	0.920608
Bert4news	0.927694
Bert4news + attention	0.940217

Table 3: Comparison of different pre-trained models and using attention mechanism

Exp	ROC-AUC
Model 1	0.939215
Model 2	0.919242
Model 3	0.940217
Ensemble	0.944949

Table 4: Final results

AUC score.

Table 4 shows results for three models which we have described in section 3. We got 0.939215 with model 1, 0.919242 with model 2, and 0.940217 with model 3. The final model is derived from model 1 and model 3 by calculating the average of results returned by model 1 and model 3. We obtained 0.944949 of ROC-AUC using that simple ensemble model.

5 Discussion

Since there may be more than one images in a post, we have tried to use one image as input or multiple images (4 images at most) as input. In preliminary experiments, we found that using only one image for each post obtained higher result in development set, so we decided to use one images in further experiments.

We have showed that keeping words in incorrect forms in the text better than fixing it to the correct forms. A possible explanation might be that those texts may contain violent contents or extreme words and users use that forms in order to bypass the social media sites' filtering function. Since those words can partly reflect the sentiment of the text, the classifier may gain benefit from it. The reason is that unreliable contents tend to use more subjective or extreme words to convey a particular perspective (Wang et al., 2018).

We also showed that by using the proposed attention mechanism, the result improved significantly. This result indicates that images and texts are co-related. In our observation, images and texts of reliable news are often related while in many unreliable news, posters use images that do not relate to the content of the news for click-bait purpose.

We found that convolution layers are useful and textual features can be well extracted by CNN layers. Conneau et al., 2017 has showed that a deep stack of local operations can help the model to learn the high-level hierarchical representation of a sentence and increasing the depth leads to the improvement in performance. Also, deeper CNN with residual connections can help to avoid overfitting and solves the vanishing gradient problem (Kaliyar et al., 2020).

6 Conclusion

6.1 Summary

We have presented a multimodal ensemble model for unreliable information identification on Vietnamese SNS. We combined two neural network models which fuse multimodal features from three data types including texts, images, and metadata. Experimental results confirmed the effectiveness of our methods in the task.

6.2 Future work

As future work, we plan to use auxiliary data to verify if a piece of information is unreliable or not. We believe that the natural way to make a judgement in fake news detection task is to compare a piece of information with different information sources to find out relevant evidences of fake news. We also want to see whether or not choosing one image randomly can affects the results and find solution to use more than one image.

References

Melissa Azur, Elizabeth Stuart, Constantine Frangakis, and Philip Leaf. 2011. Multiple imputation by chained equations: What is it and how does it work? *International journal of methods in psychiatric research*, 20:40–9.

Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. 2011. Information credibility on twitter. In *Proceedings of the 20th International Conference on World Wide Web*, WWW '11, page 675–684, New York, NY, USA. Association for Computing Machinery.

Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. 2017. Very deep convolutional networks for text classification.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Xinhuan Duan, Elham Naghizade, Damiano Spina, and Xiuzhen Zhang. 2020. RMIT at PAN-CLEF 2020: Profiling Fake News Spreaders on Twitter. In *CLEF 2020 Labs and Workshops, Notebook Papers*. CEUR Workshop Proceedings.

Souvick Ghosh and Chirag Shah. 2018. Towards automatic fake news classification. *Proceedings of the Association for Information Science and Technology*, 55(1):805–807.

Rohit Kumar Kaliyar, Anurag Goswami, Pratik Narang, and Soumendu Sinha. 2020. Fndnet – a deep convolutional neural network for fake news detection. *Cogn. Syst. Res.*, 61(C):32–44.

Dhruv Khattar, Jaipal Singh Goud, Manish Gupta, and Vasudeva Varma. 2019. Mvae: Multimodal variational autoencoder for fake news detection. In *The World Wide Web Conference*, WWW '19, page 2915–2921, New York, NY, USA. Association for Computing Machinery.

Yoon Kim. 2014. Convolutional neural networks for sentence classification.

S. Krishnan and M. Chen. 2018. Identifying tweets with fake news. In *2018 IEEE International Conference on Information Reuse and Integration (IRI)*, pages 460–464.

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.

Jing Ma, Wei Gao, and Kam-Fai Wong. 2018. Rumor detection on twitter with tree-structured recursive neural networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1980–1989, Melbourne, Australia. Association for Computational Linguistics.

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.

Kai Shu, Xinyi Zhou, Suhang Wang, Reza Zafarani, and Huan Liu. 2019. The role of user profile for fake news detection.

Lin Tian, Xiuzhen Zhang, Yan Wang, and Huan Liu. 2020. Early detection of rumours on twitter via stance transfer learning. In *Advances in Information Retrieval*, pages 575–588, Cham. Springer International Publishing.

L. Wang, Y. Wang, G. de Melo, and G. Weikum. 2018. Five shades of untruth: Finer-grained classification of fake news. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 593–594.

Yaqing Wang, Fenglong Ma, Z. Jin, Ye Yuan, G. Xun, Kishlay Jha, Lu Su, and Jing Gao. 2018. Eann: Event adversarial neural networks for multi-modal fake news detection. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

Yang Yang, Lei Zheng, Jiawei Zhang, Qingcai Cui, Zhoujun Li, and Philip S. Yu. 2018. Ti-cnn: Convolutional neural networks for fake news detection.

Xinyi Zhou, Jindi Wu, and Reza Zafarani. 2020. Safe: Similarity-aware multi-modal fake news detection.

Xinyi Zhou and Reza Zafarani. 2019. Network-based fake news detection: A pattern-driven approach.

ReINTEL Challenge 2020: Exploiting Transfer Learning Models for Reliable Intelligence Identification on Vietnamese Social Network Sites

Kim Thi-Thanh Nguyen[1,2], **Kiet Van Nguyen**[1,2]
[1]University of Information Technology, Ho Chi Minh City, Vietnam
[2]Vietnam National University, Ho Chi Minh City, Vietnam
18520963@gm.uit.edu.vn, kietnv@uit.edu.vn

Abstract

This paper presents the system that we propose for the Reliable Intelligence Identification on Vietnamese Social Network Sites (ReINTEL) task of the Vietnamese Language and Speech Processing 2020 (VLSP 2020) Shared Task. In this task, the VLSP 2020 provides a dataset with approximately 6,000 training news/posts annotated with reliable or unreliable labels, and a test set consists of 2,000 examples without labels. In this paper, we conduct experiments on different transfer learning models, which are bert4news and PhoBERT fine-tuned to predict whether the news is reliable or not. In our experiments, we achieve the AUC score of 94.52% on the private test set from ReINTEL's organizers.

Index Terms: Reliable, news, transfer learning, bert4news, PhoBERT.

1 Introduction

With the explosion of The Fourth Industrial Revolution in Vietnam, SNSs such as Facebook, Zalo, Lotus have attracted a huge number of users. SNSs have become an essential means for users to not only connect with friends but also freely share information and news. In the context of the COVID-19 pandemic, as well as prominent political and economic events that are of great interest to many people, some people tend to distribute unreliable information for personal purposes. The discovery of unreliable news has received considerable attention in recent times. Therefore, VLSP opens ReINTEL (Le et al., 2020) shared-task with the purpose of identifying being shared unreliable information on Vietnamese SNSs.

Censoring news to see if it is trustworthy is tedious and frustrating. It is sometimes difficult to determine whether the news is credible or not. Fake news discovery has been studied more and more by academic researchers as well as social networking companies such as Facebook and Twitter. Many shared-task to detect rumors were held, such as SemEval-2017 Task 8: Determining rumour veracity and support for rumours (Derczynski et al., 2017) and SemEval-2019 Task 7: RumourEval, Determining Rumour Veracity and Support for Rumours (Gorrell et al., 2019).

In this task, we focus on finding a solution to categorize unreliable news collected in Vietnamese, which is a low-resource language for natural language preprocessing. Specifically, we implement deep learning and transfer learning methods to classify SNSs news/posts. The problem is stated as:

- **Input**: Given a Vietnamese news/post on SNSs with the text of news/post (always available), some relative information, and image (may be missing).
- **Output**: One of two labels (unreliable or reliable) that are predicted by our system.

Figure 1 shows an example of this task.

The rest of the paper is organized as follows. In Section 2, we present the related work. In Section 3, we explain some proposed approaches and its result. In Section 4, we present the experimental analysis. Finally, Section 5 draws conclusions and future work.

2 Related work

Ruchansky et al. (2017) used a hybrid model called CSI to categorize real and fake news. The CSI model includes three components: capture, source, and integrate. The first module is used to detect a user's pattern of activity on news feeds. The second module learns the source characteristics of user behavior. The last module combines both previous modules to categorize news is real or fake. The CSI model does not make assumptions about user behavior or posts, although it uses both user-profiles and article data for classification.

Id: 0.
User id: 2167074723833130000.
Post message: Cần các bậc phụ huynh xã Ngũ Thái lên tiếng, không ngờ xã mình cũng nhận thịt nhiễm sán... Cho các cháu Mầm non ăn uống thế này thật vô nhân tính! VTV đăng tin rồi nhé các anh chị.
English translation: *Needing the parents of Ngu Thai commune to speak up, astonishing my commune accept contaminated meat ... Feeding preschool children like this is so inhumane! VTV posted the news, guys.*
Timestamp post: 1584426000.
Number of post's like: 45.
Number of post's comment: 15.
Number of post's share: 8.
Label: 1 (unreliable).
Image: NAN.

Figure 1: An example extracted from the dataset.

Slovikovskaya (2019) focused on improving the results of the Fake News Challenge Stage 1 (FNC-1) stance detection task using transfer learning. Specifically, this work improved the FNC-1 best performing model adding BERT (Devlin et al., 2018) sentence embedding of input sequences as a model feature and fine-tuned XLNet (Yang et al., 2019) and RoBERTa (Liu et al., 2019b) transformers on FNC-1 extended dataset.

3 Approaches

In this study, we concentrate on SOTA models, including deep neural network models and transfer learning models.

3.1 Experimental approaches

3.1.1 Deep neural network models

In studying the fundamental theories and methods of detecting fake news, Zhou and Zafarani (2020) have come up with some fundamental theories of detecting fake news. The authors wrote, *"theories have implied that fake news potentially differs from the truth in terms of, e.g., writing style and quality (by Undeutsch hypothesis)"*. Therefore, we choose text-feature as the primary input of our experimental models. Firstly, we run deep learning models like Text CNN (Kim, 2014), BiLSTM (Zhou et al., 2016) combine with some pre-trained word embedding models such as FastText[1] (Bojanowski et al., 2016) and PhoW2V[2] (Tuan Nguyen et al., 2020) to predict the credibility of news. The results of this approach get an AUC score of 0.84 to 0.86, as shown in Table 1. We also plan to experiment with incorporating other features that ReINTEL's organizers provide, such as user id, the number of likes, shares, comments, and image, but the lack of information (shown in Table 2) leads to enormous dynamic causes us to ignore this approach.

Model	AUC
Text CNN + FastText	0.865996
Text CNN + PhoW2V	0.846567
BiLSTM + FastText	0.863183
BiLSTM + PhoW2V	0.854487

Table 1: Experimental results of deep neural models on public test set.

Feature name	Train set	Public test set	Private test set
Id	0	0	0
User name	0	0	0
Post message	1	0	0
Timestamp post	96	28	34
Number of like	115	41	616
Number of comment	10	7	677
Number of share	725	280	742
Label	0	0	0
Image	3,085	1,148	1,138

Table 2: Statistics of missing values in the dataset.

3.1.2 BERT and RoBERTa for Vietnamese

One of the problems of deep learning is its massive data requirements as well as the need for computing resources. This has spurred the development of large models and transfer learning methods. Nguyen et al. (2020) presents two BERT fine-tuning methods for the sentiment analysis task on datasets of Vietnamese reviews and gets slightly outperforms other models using GloVe and FastText. Liu et al. (2019a) fine-tuned BERT under the multi-task learning framework and obtains new state-of-the-art results on ten NLU tasks, including SNLI, SciTail, and eight out of nine GLUE tasks, pushing the GLUE benchmark to 82.7% (an improvement of 2.2%)[3]. Therefore, we attempt to fine-tune PhoBERT[4] (Nguyen and Tuan Nguyen, 2020) and bert4news[5], pre-trained models for Vietnamese which is based on BERT architecture. And transfer learning shows strength in these experi-

[1] https://fasttext.cc/docs/en/crawl-vectors.html
[2] https://github.com/datquocnguyen/PhoW2V
[3] As of February 25, 2019 on the latest GLUE test set
[4] https://github.com/VinAIResearch/PhoBERT
[5] https://github.com/bino282/bert4news

Model	AUC
PhoBERT	0.932424
bert4news	0.935163
PhoBERT+bert4news	0.945169

Table 3: Results of PhoBERT, bert4news, and results that combine these two models on the private test set.

Model	Epochs	Random Seed	Learning Rate	AUC
PhoBERT	5	42	1.00e-5	0.901628
PhoBERT	6	42	3.00e-5	0.920835
PhoBERT	7	38	2.00e-5	0.924961
PhoBERT	7	42	2.00e-5	0.932424
bert4news	5	42	3.00e-5	0.930596
bert4news	6	24	2.00e-5	0.922787

Table 4: Parameter changes lead to a change of results on the public test set.

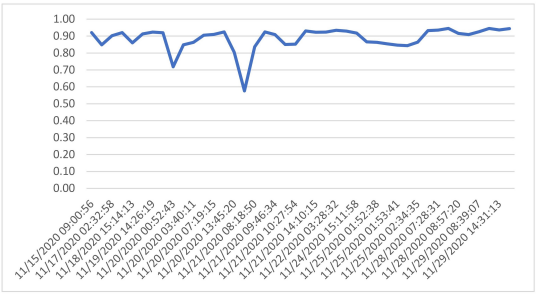


Figure 2: Performances of the team during the challenging task.

ments, we get an AUC score of between 0.92 to almost 0.95, as shown in Table 3.

3.2 Fine-tuning BERT and RoBERTa for Vietnamese

After many experiments, we find that the deep learning models do not achieve higher performance than fine-tuned bert4news and PhoBERT. Therefore, we decided to focus on only improving the results on transfer learning methods. Besides, we also try to combine the results of these two models.

The fine-tuning idea is taken from the study (Sun et al., 2019). The BERT base model creates an architecture of 12 sub-layers in the encoder, 12 heads in multi-head attention on each sub-layer. BERT input is a sequence of not more than 512 tokens; the output is a set of self-attention vectors equal to the input length. Each vector is 768 in size. The BERT input string represents both single text and text pairs explicitly, where a special token [CLS] is used for string sorting tasks, and a special token [SEP] marks the end position of the single text or the position that separates the text pair. For fine-tuning the BERT architecture for text classification, we concatenated the last four hidden representations of the [CLS] token, which will be passed into a small MLP network containing the full connection layers to transform into the distribution of discrete label values.

Our fine-tuning process consists of two main steps: tokenize the text content and retrain the model on the dataset. For PhoBERT, we use VNcoreNLP (Vu et al., 2018) library to tokenize content, while for bert4news, we use BertTokenizer.

4 Experiments

4.1 Experimental settings

In this paper, we conduct various experiments on Google Colab (CPU: Intel(R) Xeon(R) CPU @ 2.20GHz; RAM: 12.75 GB; GPU Tesla P100 or T4 16GB with CUDA 10.1). We fine-tune PhoBERT and bert4news with different parameters as batch size, learning rate, epoch, random seed. To save time and cost, we set batch size 32 for all models. With the same hyperparameter values, distinct random seeds can lead to substantially different results (Dodge et al., 2020). With the above configuration, we spend about 2.40 minutes per epoch for both bert4news and PhoBERT. Table 4 shows the parameter setting and the performance, respectively.

4.2 Performances over time

Figure 2 shows the results of our testing process. It is easy to see that our results are not stable in the first phase due to trying many methods. Our results are more stable in the later stage of the competition, but there are not many mutations.

5 Conclusion and future work

In summary, we have proposed the following methods for classifying untrustworthy news: combining deep learning model with pre-trained word embedding, fine-tune bert4news, and PhoBERT, combining text, numeric, and visual features. Accordingly, the best result belongs to the transfer learning models when achieving an AUC score of 94.52% for the combined model of bert4news and PhoBERT.

In the future, we plan to combine other features offered by ReINTEL's organizers with transfer learning models due to classifying based on news content alone is not enough (Shu et al., 2019). While we are doing well in transfer learning, we also aim to build a system for the fast and accu-

rate detection of fake news at the early stages of propagation, which is much more complicated than detecting long-circulated news. Besides, we hope to develop a system to score users based on the news they post and share to reduce unreliable news on Vietnam SNSs.

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *CoRR*, abs/1607.04606.

Leon Derczynski, Kalina Bontcheva, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Arkaitz Zubiaga. 2017. SemEval-2017 task 8: RumourEval: Determining rumour veracity and support for rumours. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 69–76, Vancouver, Canada. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping.

Genevieve Gorrell, Elena Kochkina, Maria Liakata, Ahmet Aker, Arkaitz Zubiaga, Kalina Bontcheva, and Leon Derczynski. 2019. SemEval-2019 task 7: RumourEval, determining rumour veracity and support for rumours. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 845–854, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *CoRR*, abs/1408.5882.

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.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019a. Multi-task deep neural networks for natural language understanding. *CoRR*, abs/1901.11504.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

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, Online. Association for Computational Linguistics.

Quoc Thai Nguyen, Thoai Linh Nguyen, Ngoc Hoang Luong, and Quoc Hung Ngo. 2020. Fine-tuning bert for sentiment analysis of vietnamese reviews.

Natali Ruchansky, Sungyong Seo, and Yan Liu. 2017. CSI: A hybrid deep model for fake news. *CoRR*, abs/1703.06959.

Kai Shu, Suhang Wang, and Huan Liu. 2019. Beyond news contents: The role of social context for fake news detection. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, WSDM '19, page 312–320, New York, NY, USA. Association for Computing Machinery.

Valeriya Slovikovskaya. 2019. Transfer learning from transformers to fake news challenge stance detection (FNC-1) task. *CoRR*, abs/1910.14353.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune BERT for text classification? *CoRR*, abs/1905.05583.

Anh Tuan Nguyen, Mai Hoang Dao, and Dat Quoc Nguyen. 2020. A pilot study of text-to-SQL semantic parsing for Vietnamese. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4079–4085, Online. Association for Computational Linguistics.

Thanh Vu, Dat Quoc Nguyen, Dai Quoc Nguyen, Mark Dras, and Mark Johnson. 2018. VnCoreNLP: A Vietnamese natural language processing toolkit. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 56–60, New Orleans, Louisiana. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, abs/1906.08237.

Peng Zhou, Zhenyu Qi, Suncong Zheng, Jiaming Xu, Hongyun Bao, and Bo Xu. 2016. Text classification improved by integrating bidirectional LSTM with two-dimensional max pooling. *CoRR*, abs/1611.06639.

Xinyi Zhou and Reza Zafarani. 2020. A survey of fake news: Fundamental theories, detection methods, and opportunities. *ACM Comput. Surv.*, 53(5).

A Joint Deep Contextualized Word Representation for Deep Biaffine Dependency Parsing

Xuan-Dung Doan
Viettel Cyberspace Center, Viettel Group
Hanoi, Vietnam
dungdx4@viettel.com.vn

Abstract

We propose a joint deep contextualized word representation for dependency parsing. Our joint representation consists of five components: word representations from ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019) language models for Vietnamese (Che et al., 2018; Nguyen and Nguyen, 2020), Word2Vec (Mikolov et al., 2013) embeddings trained on baomoi dataset (Xuan-Son Vu, 2019), character embeddings (Kim, 2014), and part-of-speech tag embeddings. When using the joint representation with a deep biaffine dependency parser (Dozat and Manning, 2016), our model ranks 2nd in Vietnamese Universal Dependency Parsing Shared-Task at VLSP 2020 (Linh et al., 2020).

1 Introduction

Dependency parsing is the task of automatically identifying binary grammatical relations between tokens in a sentence. There are two common approaches to dependency parsing: transition-based (Nivre, 2003; McDonald and Pereira, 2006), and graph-based (Eisner, 1996; McDonald et al., 2005a).

Recently, there has been a surge in the use of deep learning approaches to dependency parsing (Chen and Manning, 2014; Dyer et al., 2015; Kiperwasser and Goldberg, 2016; Dozat and Manning, 2016; Ma et al., 2018; Fernández-González and Gómez-Rodríguez, 2019; Zhang et al., 2020), which help alleviate the need for hand-crafted features, take advantage of the vast amount of raw data through word embeddings, and achieve state-of-the-art results.

Contextualized word representations, such as ELMo and BERT, have shown to be extremely helpful in a variety of NLP tasks. The contextualized model is used as a feature extractor, which is able to encode semantic and syntactic information of the input into a vector.

In this work, we further improve dependency parsing performance by making good use of external contextualized word representations.

2 Related works

Che et al. (2018) incorporated ELMo into both dependency parser and ensemble parser training with different initialization. Their system achieved the best result in CoNLL 2018 shared task.

Li et al. (2019) captured contextual information by combining the power of both BiLSTM and self-attention via model ensembles. The results led to a new state-of-the-art parsing performance.

Nguyen and Nguyen (2020) replaced the pre-trained word embedding of each word in an input sentence by corresponding contextualized embedding computed for the first subword token of the word. They achieve the state-of-the-art performance on VnDT dependency treebank v1.1 (Nguyen et al., 2014).

3 Methodology

In our model, an input sentence of n words $w = w_1, w_2, ..., w_n$ is fed to each of the component networks to learn separate token embeddings. We describe the learning process below.

3.1 Graph-based Dependency Parsing

Graph-based Dependency Parsing follows the common structured prediction paradigm (McDonald et al., 2005a; Taskar et al., 2005):

$$predict(w) = \underset{y \in \mathcal{Y}(w)}{\operatorname{argmax}}\, score_{global}(w, y) \quad (1)$$

$$score_{global}(w, y) = \sum_{part \in y} score_{local}(w, part) \quad (2)$$

Given an input sentence w (and the corresponding sequence of the vectors $w_{1:n}$), we look the highest-score parse tree y in the space $\mathcal{Y}$(w) of valid dependency trees over w. In order to make the search tractable, the scoring function is decomposed to the sum of local scores for each part independently.

3.2 Word Embedding

The input layer maps each input word w_i into a dense vector representation x_i. We use word2vec (Mikolov et al., 2013) embeddings trained on baomoi dataset (Xuan-Son Vu, 2019) $emb^{word}_{w_i}$, a CNN-encoder character representation (Kim, 2014) $emb^{char}_{\hat{w}_i}$, and POS-tag embedding is created randomize to enrich each word's representation $emb^{tag}_{t_i}$ further.

$$x_i = emb^{word}_{w_i} \oplus emb^{char}_{\hat{w}_i} \oplus emb^{tag}_{t_i} \quad (3)$$

3.3 Deep Contextualized Word Representations

3.3.1 ELMO

ELMO uses an LSTM (Hochreiter and Schmidhuber, 1997) network to encode words in a sentence and training the LSTM network with language modeling objective on large-scale raw text. $ELMo_i$ calculates the hidden representation $h^{(LM)}_i$ as

$$h^{(LM)}_i = BiLSTM^{(LM)}(h^{(LM)}_0, (\hat{w}_1, ..., \hat{w}_n))_i \quad (4)$$

where $\hat{w}_i$ is the output of a CNN over characters. ELMo representational power is computed by a linear combination of BiLSTM layers:

$$ELMo_i = \gamma \sum_{j=0}^{L} s_j h^{(LM)}_{i,j} \quad (5)$$

where s_j is a softmax-normalized task-specific parameter and γ is a task-specific scalar. We use the Vietnamese ELMo model released by Che et al. (2018).

3.3.2 BERT

BERT introduced an alternative language modeling objective to be used during training of the model. Instead of predicting the next token, the model is expected to guess a masked token. BERT is based on the Transformer architecture (Vaswani et al., 2017), which carries the benefit of learning potential dependencies between words directly. For use in downstream tasks, BERT extract the Transformer's encoding of each token at the last layer, which effectively produces $BERT_i$.

PhoBERT (Nguyen and Nguyen, 2020) was introduced for the Vietnamese NLP community as a Roberta-based model (Liu et al., 2019). PhoBERT achieves the state-of-the-art in Vietnamese POS-tag and Named Entity Recognition. Therefore, we use PhoBERT to produce $BERT_i$.

After getting $ELMo_i$ and $BERT_i$, we use them as an additional word embedding. The calculation of x_i becomes:

$$x_i = emb^{word}_{w_i} \oplus emb^{char}_{\hat{w}_i} \oplus emb^{tag}_{t_i} \oplus ELMo_i \oplus BERT_i \quad (6)$$

The BiLSTM is used to capture the context information of each word. Finally, the encoder outputs a sequence of hidden states s_i.

3.4 Biaffine Attention Mechanism

We use the Biaffine attention mechanism described in (Dozat and Manning, 2016) for our dependency parser. The task is posed as a classification problem, where given a dependent word, the goal is to predict the head word (or the incoming arc). Formally, let s_i and h_t be the BiLSTM output states for the dependent word and a candidate head word respectively, the score for the arc between s_i and h_t is calculated as:

$$e^t_i = h^T_t W s_i + U^T h_t + V^T s_i + b \quad (7)$$

Where W, U, V, b are parameters, denoting the weight matrix of the bi-linear term, the two weight vectors of the linear terms, and the bias vector.

Similarly, the dependency label classifier also uses a biaffine function to score each label, given the head word vector h_t and child vector s_i as inputs.

3.5 Training Loss

The parser defines a local cross-entropy loss for each position i. Assuming w_j is the gold-standard head of w_i, the corresponding loss is

$$loss(s, i) = -log \frac{e^{score(i \leftarrow j)}}{\sum_{0 \le k \le n, k \ne i} e^{score(i \leftarrow k)}} \quad (8)$$

3.6 Dependency Parsing Decoding

The decoding problem of this parsing model is solved by using the Maximum Spanning Tree (MST) algorithm (McDonald et al., 2005b).

4 Experiments

4.1 Dataset

The VLSP organizers released the datasets in two phases. We split the first dataset into training, development, and test data, according to the 7:1:2 ratio. We then merge the second dataset into the first training data. The final statistics are summarized in Table 1.

Table 1: Statistics of the public dataset

	Number of sentences
Train set	6626
Develop set	507
Test set	1010

4.2 Setup

Table 2 summarizes the hyper-parameters that we use in our experiments. We implement an additional model that trains on lowercased input data, since the dataset also includes text from social media, which contains many word-form errors. We compare our results with the graph-based Deep Biaffine (BiAF) (Dozat and Manning, 2016) parser. Since the private test set of the VLSP Shared Task contains raw text only, we use VncoreNLP (Vu et al., 2018) to segment and POS-tag the raw data. Parsing performance is measured using UAS metric (Unlabeled Attachment Score) and LAS metric (Labeled Attachment Score) by comparing the gold relations of the test set and relations returned by the system. We use the evaluation script published at CoNLL 2018 [1].

Table 2: Hyper-parameters in our experiments

	Layer	Hyper-Parameter	Value
Input	Word	dimension	300
	POS	dimension	50
	Char	dimension	50
LSTM	Encoder	encoder layer	6
		encoder size	500
	MLP	arc MLP size	512
		label MLP size	128
	Training	Dropout	0.33
		optimizer	Adam
		learning rate	0.001
		batch size	80
ELMo		dimension	1024
BERT		dimension	768

4.3 Main Results

The results on the test set are shown in Table 3.

Table 3: The results (UAS%/LAS%) on the test set

	UAS/LAS
BiAF	80.83/69.40
Our model	82.86/**71.16**
Our lowercase model	**83.02**/71.05

The raw private test set after segmentation and POS tagging by VncoreNLP is the input to our model. The results on the raw private test set are shown in Table 4.

Table 4: The results (UAS%/LAS%) on each file of the raw private test set

	Our model	Our lowercase model
VTB	**76.33/67.46**	75.68/66.59
vn1	**74.79/65.38**	72.17/62.61
vn3	74.22/66.73	**74.95/67.28**
vn7	**68.33/61.67**	66.11/61.11
vn8	**74.81**/65.71	74.29/**65.97**
vn10	**80.64/72.46**	78.45/69.98
vn14	72.61/62.45	**73.36/63.69**
Total	**76.12/67.32**	75.48/66.53

Beside providing the private raw data set, VLSP organizers also provide the data in CoNLL-U (Ginter et al., 2017) format. The results on the private CoNLL-U format test set are shown in Table 5.

Table 5: The results (UAS%/LAS%) on each file of the private CoNLL-U format test set

	Our model	Our lowercase model
VTB	**84.81/76.44**	84.58/76.29
vn1	**78.98/70.94**	77.43/70.17
vn3	**85.89**/76.97	85.46/**77.58**
vn7	**82.22/75.56**	80.00/73.89
vn8	**82.49/73.93**	81.32/73.8
vn10	**85.46/77.53**	81.20/72.69
vn14	**84.04**/75.31	83.54/**76.81**
Total	**84.65/76.27**	84.23/76.05

The final result is calculated by averaging UAS and LAS scores on the raw private data and the private CoNLL-U format data. The official rank

[1] https://universaldependencies.org/conll18/conll18_ud_eval.py

is based on average the final UAS and LAS score. The final result of all teams is shown in Table 6.

Table 6: The final results (UAS%/LAS%/Average%) of all teams

	UAS	LAS	Aver.	Rank
Our model	80.39	71.80	76.09	2
DP2	80.89	71.36	76.12	1
DP3	78.58	70.04	74.31	4
DP4	79.28	70.47	74.87	3
DP5	77.28	68.77	73.03	5

Our model ranks 1st in LAS and 2nd in UAS. Finally, we rank 2nd on average UAS and LAS, officially.

5 Conclusion

We present joint ELMO and BERT as features for dependency parsing. In the future, we plan to analyze the effectiveness of our model when ELMO and/or BERT are excluded. We also plan to improve our model by using the self-attention mechanism as a replacement for the BiLSTM-based encoder in our current model.

References

Wanxiang Che, Yijia Liu, Yuxuan Wang, Bo Zheng, and Ting Liu. 2018. Towards better UD parsing: Deep contextualized word embeddings, ensemble, and treebank concatenation. *CoRR*, abs/1807.03121.

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750, Doha, Qatar. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Timothy Dozat and Christopher D. Manning. 2016. Deep biaffine attention for neural dependency parsing. *CoRR*, abs/1611.01734.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 334–343, Beijing, China. Association for Computational Linguistics.

Jason M. Eisner. 1996. Three new probabilistic models for dependency parsing: An exploration. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.

Daniel Fernández-González and Carlos Gómez-Rodríguez. 2019. Left-to-right dependency parsing with pointer networks. *CoRR*, abs/1903.08445.

Filip Ginter, Jan Hajič, Juhani Luotolahti, Milan Straka, and Daniel Zeman. 2017. CoNLL 2017 shared task - automatically annotated raw texts and word embeddings. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *CoRR*, abs/1408.5882.

Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional LSTM feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327.

Ying Li, Zhenghua Li, Min Zhang, Rui Wang, Sheng Li, and Luo Si. 2019. Self-attentive biaffine dependency parsing. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5067–5073. International Joint Conferences on Artificial Intelligence Organization.

HA My Linh, NGUYEN Thi Minh Huyen, VU Xuan Luong, NGUYEN Thi Luong, PHAN Thi Hue, and LE Van Cuong. 2020. Vlsp 2020 shared task: Universal dependency parsing for vietnamese. In *Proceedings of The seventh international workshop on Vietnamese Language and Speech Processing (VLSP 2020)*, Hanoi, Vietnam.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Xuezhe Ma, Zecong Hu, Jingzhou Liu, Nanyun Peng, Graham Neubig, and Eduard H. Hovy. 2018. Stack-pointer networks for dependency parsing. *CoRR*, abs/1805.01087.

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005a. Online large-margin training of dependency parsers. In *Proceedings of the 43rd*

Annual Meeting of the Association for Computational Linguistics (ACL'05), pages 91–98, Ann Arbor, Michigan. Association for Computational Linguistics.

Ryan McDonald and Fernando Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy. Association for Computational Linguistics.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005b. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 523–530, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26 (NIPS 2013)*.

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.

Dat Quoc Nguyen, Dai Quoc Nguyen, Son Bao Pham, Phuong-Thai Nguyen, and Minh Le Nguyen. 2014. From treebank conversion to automatic dependency parsing for vietnamese. In *Natural Language Processing and Information Systems*, pages 196–207, Cham. Springer International Publishing.

Joakim Nivre. 2003. An efficient algorithm for projective dependency parsing. In *Proceedings of the Eighth International Conference on Parsing Technologies*, pages 149–160, Nancy, France.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *CoRR*, abs/1802.05365.

Ben Taskar, Vassil Chatalbashev, Daphne Koller, and Carlos Guestrin. 2005. Learning structured prediction models: A large margin approach. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, page 896–903, New York, NY, USA. Association for Computing Machinery.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Thanh Vu, Dat Quoc Nguyen, Dai Quoc Nguyen, Mark Dras, and Mark Johnson. 2018. VnCoreNLP: A Vietnamese natural language processing toolkit. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 56–60, New Orleans, Louisiana. Association for Computational Linguistics.

Son N. Tran Lili Jiang Xuan-Son Vu, Thanh Vu. 2019. Etnlp: A visual-aided systematic approach to select pre-trained embeddings for a downstream task. *In: Proceedings of the International Conference Recent Advances in Natural Language Processing (RANLP)*.

Yu Zhang, Zhenghua Li, and Min Zhang. 2020. Efficient second-order TreeCRF for neural dependency parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3295–3305, Online. Association for Computational Linguistics.

Applying Graph Neural Networks for Vietnamese Dependency Parsing

NGUYEN Duc Thien **NGUYEN Thi Thu Trang*** **TRUONG Dang Quang**
Hanoi University of Science and Technology
Hanoi, Vietnam
Óbuda University
Budapest, Hungary
ndthien98@gmail.com *trangntt@soict.hust.edu.vn* *dangquangtruong98@gmail.com*

Abstract

This paper presents a state-of-the-art model to solve the Vietnamese dependency parsing task (HA My Linh, 2020) in VLSP 2020[1] Evaluation Campaign. In this model, the Bidirectional Long Short-Term Memory (BiLSTM) network is used to extract the contextual information, while the graph neural network captures high-order information. Some preprocessing for Vietnamese raw texts are included for the training, such as word segmentation, part-of-speech (POS) tagging for the model.

We modified the network with suitable word embedding mechanisms, i.e., fastText, to represent the semantic information of words more accurately. Therefore, Vietnamese words that are marked as unknown tokens now can have the right embedding; thus, they will be well modeled in dependency parsing.

Experiments on the raw text dataset show that the model achieved an average of 72.85% of unlabeled attachment score (UAS) and 64.35% of labeled attachment score (LAS). With the Segmentation and POS tagging dataset, we achieved a higher average of 81.71% (UAS) and 73.19% (LAS).

1 Introduction

In recent years, dependency parsing is a fascinating research topic and has a large number of applications in natural language processing. This task is to automatically identify the relationship between words in a sentence and label the relationship between the head and the dependency word, and thus, establish the grammatical structure of the sentence. Traditional graph-based dependency parsing only extracts the parent-child relationship and ignores deeper relationships. Hence, we decided to experiment with the idea of extracting deeper relationships of the neighbor nodes, which is extensively covered in the paper Ji et al. (2019).

This state-of-the-art model achieved good performance due to its ability to represent incorrect Out-Of-Vocabulary (OOV) words in the input layer for Vietnamese. Normally, words that are not found in the vocabulary will be marked as unknown tokens before feeding to the embedding layer. This caused the model to embed OOV words incorrectly; therefore, it created the loss of information in calculating attention distribution. In this paper, we modified the pre-trained layer of word embedding for the graph neural networks with a more suitable embedding mechanism for Vietnamese, which solved the issue well.

The rest of this paper is organized as follows: Section 2 presents the architecture and its components of graph neural networks. The experiments are shown in Section 3. Finally, Section 4 concludes the paper and gives some perspectives for the work.

2 Methodology

Normally, Graph-based dependency parser search through the space of possible trees for a given sentence encoded as directed graphs and use methods from graph theory (Maximum Spanning Tree or greedy algorithm) for the optimal solutions. However, in the Graph Neural Network (GNN) model, the dependency parser utilizes the neural network to assign a weight to each edge, then construct a MST from the edge weight (Dozat et al., 2017). For maximum accuracy, we need to analyze the surface form and the deep structure of the graph. There are three main components in the model: Encoder extracts the surface form and the contextual information and turns them into the nodes (words) representations for the next components; The graph attention network (a subset of GNN, using the structure from Veličković et al. (2017)) layers then extract the deep structure and high-order information

*Corresponding author
[1]Vietnamese Language and Speech Processing

to illustrate the head-dependent relationships of the nodes; the final component is the decoder, used to create the dependency tree from the output of the GNN. We will discuss the details in the following sections.

2.1 Pre-processing

First, we used the VNCoreNLP - suggested by Vu et al. (2018) - to segment and perform the POS tagging on the raw text. VNCoreNLP used a transformation rule-based learning model for the segmentation of the Vietnamese document, thus, obtained faster and better accuracy than all previous segmentation tools, as the model accounted for the fact that Vietnamese words are created from syllables including the space character (Nguyen et al., 2017). The VNCoreNLP performed the task of labeling words with POS tag Vu et al. (2018) via MarMot (a CRF framework), state of the art POS and morphological tagger (Müller et al., 2013)

Word embedding is the most popular representation method for words in a document because it captures the context of words, semantic and syntactic similarity, relation with other words, etc. Using word embedding makes it easier to represent words with less memory than using a one-hot vector while also showing the relationship between words.

With a huge training corpus (e.g., a total of 100 billion words with a 3-million-word vocab in Google News), the pre-trained model can cover much more context for word embedding than the auto-updating mechanism of the word embedding in the end-to-end abstractive summarization model with its training corpus (e.g., a total of 240 million words with a 50k-word vocab in Daily Mail/CNN) (Anh and Trang, 2019).

In this paper, we adopted a suitable pre-trained model for Vietnamese with 300-dimensional word embeddings, i.e., fastText from Facebook (Joulin et al., 2016), for the word embedding layer. The fastText trained on the Wikipedia dataset with character n-grams of length 5 by CBOW[2] method. fastText is more suitable in our case as when the GNN model meets unknown vocab, the fastText generates an embedding of the vocab with value 0, resulting in error reductions; meanwhile, the Word2Vec and the GloVe does not do that. This method enables fastText to handle OOV[3] words by constructing the vector for OOV words from its characters.

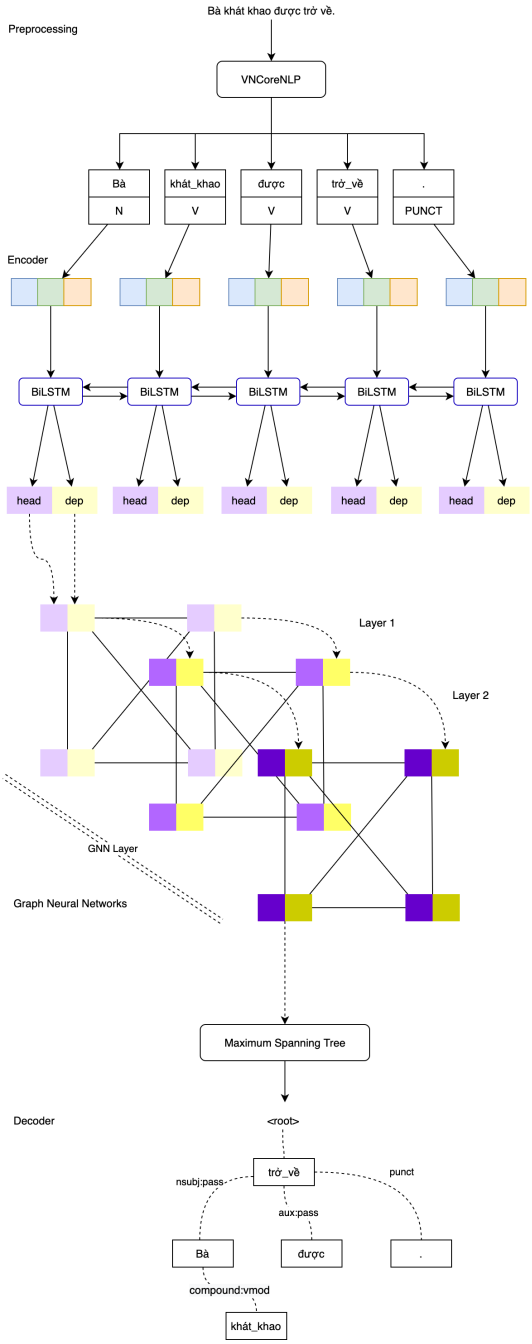


Figure 1: The architecture of Graph Neural Networks

[2]Continuous Bag of Words
[3]Out-of-vocabulary

Both GloVe and Word2Vec are unable to do so.

2.2 Encoder

According to Kiperwasser and Goldberg (2016), we can apply BiLSTM model to create the dependency tree as illustrated in Figure [1] Firstly, each word is embedded using a vector combined from three different vectors: randomly initialized word embedding, pre-trained word embedding, and part-of-speech embedding.

$$x_i = e(w_i) \oplus e'(w_i) \oplus e(pos_i) \tag{1}$$

As a result, the x_i illustrated the sentence of the word i in [2]. Given the position i of the word, the BiLSTM model can compute state vectors $\overrightarrow{c_i}$ and $\overleftarrow{c_i}$ where the $\overrightarrow{c_i}$ is draw from the start of the sentence to the position i and $\overleftarrow{c_i}$ is from the end of the sentence to i.

$$\overrightarrow{c_i} = \overrightarrow{LSTM(x_i)} \oplus \overleftarrow{LSTM(x_i)} \tag{2}$$

The two vectors $\overrightarrow{c_i}$ and $\overleftarrow{c_i}$ then concatenate to become the context-dependent representation of the word i. Thus we can use multilayers perceptron (MLP) to define two-node representations of the word i the probability of being the head role vector and probability of being the dependent role vector (Dozat et al., 2017):

$$\mathbf{h}_i = MLP_h(c_i), \mathbf{d}_i = MLP_d(c_i) \tag{3}$$

The score function is a SoftMax function, where the representations of the word i and j is the input, therefore complementing the analysis of the surface form of the segmented sentence. As a result, the output of the BiLSTM component is a complete weight graph model. (Dozat et al., 2017)

$$\sigma(i,j) = \text{Softmax}_i(h_j^T A d_j + b_1^T h_j + b_2^T h_j) \tag{4}$$

2.3 GNN Layers

In the implementation, the GNN component can utilize at most three layers, each layer consists of 4 graph neural network units as illustrated in Figure [1] - where the representation of the vectors is calculated from the same representation in the previous layer using this formula where g is the *LeakyReLU* function, t is the layer, v_i is the vector representation of i, and a_{ij} is the edge weight of

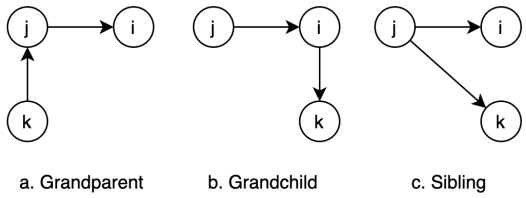


Figure 2: Relations between nodes

v_i and v_j (i and j are forming the neighborhood) (Wang and Chang, 2016):

$$v_i^t = g\left(W \sum_{j \in \mathcal{N}(i)} \alpha_{ij}^t v_j^{t-1} + B v_i^{t-1}\right) \tag{5}$$

We can apply the formula [5] to analyze the high order information of the nodes which is represented in three ways: grandparents, grandchildren, and siblings (Figure [2]) (Eisner, 1997).

Specifically, the head representation of node i should attend to the neighbors' representation as they are the parents of the i. Therefore the model can calculate h_i from the h_j of the previous layer $t-1$ using the formula [5]:

$$\begin{cases} h_i^t = g\left(W_1 \sum_{j \in \mathcal{N}(i)} \alpha_{ji}^t h_j^{t-1} + B_1 h_i^{t-1}\right) \\ d_i^t = g\left(W_2 \sum_{j \in \mathcal{N}(i)} \alpha_{ij}^t d_j^{t-1} + B_2 d_i^{t-1}\right) \end{cases} \tag{6}$$

The dependent node d_i's computation operation is the same as the head node's one i. Thus the equation [6] can assist to analyse the order of the relationship of grandparents and grandchild.

To examine the sibling relationships, the head representation of the node i check the neighborhood where they are dependent on node i. Thus the formula will update the h_i in the following way:

$$\begin{cases} h_i^t = g\left(W_1 \sum_{j \in \mathcal{N}(i)} \alpha_{ji}^t d_j^{t-1} + B_1 h_i^{t-1}\right) \\ d_i^t = g\left(W_2 \sum_{j \in \mathcal{N}(i)} \alpha_{ij}^t h_j^{t-1} + B_2 d_i^{t-1}\right) \end{cases} \tag{7}$$

Finally we combine the two equations [6] and [7] above to update the grandparents, grandchild

Param	Ji et al. (2019)	Our paper
Word Embedding	300 dim	500 dim
POS Embedding	100 dim	100 dim
arc MLP size	500 dim	500 dim
rel MLP size	100 dim	100 dim
Dropout	0.33	0.33
Optimizer	Adam	Adam
Learning rate	0.002	0.002
Graph layers	2	2

Table 1: Hyper-parameter.

and siblings.

$$\begin{cases} h_i^t = & g(W_1 \sum\limits_{j \in \mathcal{N}(i)} (\alpha_{ji}^t h_j^{t-1} + \alpha_{ji}^t d_j^{t-1}) \\ & + B_1 h_i^{t-1}) \\ d_i^t = & g(W_2 \sum\limits_{j \in \mathcal{N}(i)} (\alpha_{ij}^t h_i^{t-1} + \alpha_{ji}^t d_j^{t-1}) \\ & + B_2 d_i^{t-1}) \end{cases} \tag{8}$$

As the equations [8] illustrated, the edge weight a_{ij} is the decisive element responsible for the update of the relationship information. The edge weight is figured with the following formula:

$$\alpha_{ij}^t = \begin{cases} \text{Softmax}_i & (h_i^T A d_j + b_1^T h_i + b_2^T d_j) \\ & i \in \mathcal{N}_k^t(j) \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

2.4 Decoder

After the high-order information is extracted from the GNN and enhanced the nodes representations, the node representation will be used to build the dependency tree via Biaffine parser (the setting is identical to Dozat et al. (2017))

3 Experiments

3.1 Dataset

The VLSP provided the datasets and separated them into training datasets and raw text datasets. The data for training was further divided into two packages: the first package consists of 5070 sentences, with a large domain from the social media comments on restaurants and hotels (100 sentences), to the story of the Little Prince (1570 sentences) and the VietTreeBank - VTB (3400 sentences); the second package includes 3000 sentences with diverse origins.

The raw text data for prediction includes the two packages above, and 20 raw text files crawled from VnExpress news articles. The VTB files and the files with index 1,3,4,7,8,10,14 were accurately tokenized and labeled.

The graph-based dependency parsing neural network model has one important characteristic: the raw text dataset's sentences have to be tokenized for the training to be carried out successfully. Therefore the VNCoreNLP - an NLP pipeline used for POS tagging, named entity recognition and dependency parsing is useful here in this case [4]. This tool is capable of providing highly accurate annotation for the input sentences, therefore improving the score of the training model.

3.2 Training

The training operation consists of two methods: First, we have to decode the output of the final layer of the GNN component (denoted by)

$$\alpha_{ij}^t = \sigma^t(i,j) = P^t(i|j) \tag{10}$$

which are the tree structures (computed by $P(i|j)$) and the dependency edge labels (measured by $P(r|i,j)$, which indicated the probability a tree (i,j) holds a dependency relation r, using another MLP from biaffine parser (Dozat et al., 2017), the loss function of the classifier is computed with the equation:

$$\mathcal{L}_0 = -\frac{1}{n} \sum_{(i,j,r) \in T} (\log P^\tau(i|j) + \log P(r|i,j)) \tag{11}$$

Second, the model can supervise on $P^t(i|j)$ from each layer of the GNN component, therefore the layer-wise loss will be computed with the equation:

$$\mathcal{L}' = \sum_{t=1}^{\tau} \mathcal{L}_t = \sum_{t=1}^{\tau} -\frac{1}{n} \sum_{(i,j,r) \in T} \log P(r|i,j) \tag{12}$$

The main objective is to minimize the loss of combination of them:

$$L = \lambda_1 \mathcal{L}_0 + \lambda_2 \mathcal{L}' \tag{13}$$

3.3 Results

We have implemented and operated the model on the AWS Server (AWS Deep Learning AMI

Dataset	UAS	LAS
Test from VTB	81.89	73.34
VNExpress 1	75.12	66.15
VNExpress 3	84.36	75.38
VNExpress 7	76.67	67.22
VNExpress 8	79.25	71.98
VNExpress 10	80.47	72.54
VNExpress 14	80.55	73.57
Total	**81.71**	**73.19**

Table 2: Test on labeled datasets.

Dataset	UAS	LAS
Test from VTB	73.18	64.66
VNExpress 1	68.77	58.75
VNExpress 3	74.10	65.81
VNExpress 7	61.67	55.56
VNExpress 8	68.96	61.43
VNExpress 10	73.19	64.13
VNExpress 14	68.4	60.72
Total	**72.85**	**64.35**

Table 3: Test on raw-text datasets.

(Ubuntu 18.04) Version 34.0 installed in the EC2 Instance p3.2xlarge - GPU NVIDIA Tesla v100 16 GB, Memory 61 GB, SSD 100 GB, CPU 8 Virtual Cores) successfully. The hyperparameters configuration in Table [1] has slight modifications. For the word embedding, we used fastText (Bojanowski et al., 2016) with Vietnamese data as the primary pre-trained model, which has 300 dimensions instead of 100 dimensions of GloVe that Ji et al. (2019) used. Then, we concatenate the pre-trained word embedding with 200-dimension randomly initialize word embedding and 100-dimension part-of-speech embedding. Randomly embedding vectors obtained from binomial distribution. The training operation took approximately one hour.

The main evaluators for the dependency parsing problem are LAS and UAS. The results are coming from the script evaluator 2018. For the labeled data, the highest UAS is 81.89% from the VTB package, meanwhile the package Test VNExpress 14 achieved the highest LAS 73.57%.

Table [2] shows results from VLSP 2020 private tests for dependency parsing on labeled datasets, meanwhile raw-text datasets' results are shown on Table [3].

4 Conclusion

To conclude, our experiment on using the graph neural network for graph-based dependency parsing suggests that understanding the deep structure of the representations of words via nodes' message passing improved a slightly better accuracy and efficiency than other traditional graph-based dependency parsers. In future works, we are planning to improve the performance of the model by applying Conditional Random Fields in the labeling process for the nodes before extracting the high-order information via graph neural network.

Acknowledgments

The authors wish to thank VLSP organizers for their reviews and encouragement.

References

Dang Trung Anh and Nguyen Thi Thu Trang. 2019. Abstractive text summarization using pointer-generator networks with pre-trained word embedding. In *Proceedings of the Tenth International Symposium on Information and Communication Technology*, pages 473–478.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Timothy Dozat, Peng Qi, and Christopher D. Manning. 2017. Stanford's graph-based neural dependency parser at the CoNLL 2017 shared task. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 20–30, Vancouver, Canada. Association for Computational Linguistics.

Jason Eisner. 1997. Three new probabilistic models for dependency parsing: An exploration. *arXiv preprint cmp-lg/9706003*.

VU Xuan Luong NGUYEN Thi Luong PHAN Thi Hue LE Van Cuong HA My Linh, NGUYEN Thi Minh Huyen. 2020. Vlsp 2020 shared task: Universal dependency parsing for vietnamese. In *Proceedings of The seventh international workshop on Vietnamese Language and Speech Processing (VLSP 2020)*, Hanoi, Vietnam.

Tao Ji, Yuanbin Wu, and Man Lan. 2019. Graph-based dependency parsing with graph neural networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2475–2485.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional lstm feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327.

Thomas Müller, Helmut Schmid, and Hinrich Schütze. 2013. Efficient higher-order crfs for morphological tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 322–332.

Dat Quoc Nguyen, Dai Quoc Nguyen, Thanh Vu, Mark Dras, and Mark Johnson. 2017. A fast and accurate vietnamese word segmenter. *CoRR*, abs/1709.06307.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Thanh Vu, Dat Quoc Nguyen, Dai Quoc Nguyen, Mark Dras, and Mark Johnson. 2018. VnCoreNLP: A Vietnamese natural language processing toolkit. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 56–60, New Orleans, Louisiana. Association for Computational Linguistics.

Wenhui Wang and Baobao Chang. 2016. Graph-based dependency parsing with bidirectional LSTM. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2306–2315, Berlin, Germany. Association for Computational Linguistics.

Implementing Bi-LSTM-based deep biaffine neural dependency parser for Vietnamese Universal Dependency Parsing

Nguyen Thi Thuy Lien
FPT University
Hanoi, Vietnam
liennttmse0096@fpt.edu.vn

Abstract

This paper presents our approach to resolve the Vietnamese Universal Dependency Parsing task in VLSP 2020 Evaluation Campaign. On the basis of Deep Biaffine Attention for Neural Dependency Parsing(Dozat and Manning, 2017), we adapted the dependency parser for Vietnamese. Our best model obtained a pretty good performance on the test datasets, achieving 84.08% UAS score and 75.64% LAS on average for the ConLL-U dataset. On the raw text data-set, the results we reached still quite limited, on average 74.47% of UAS and 65.3% of LAS.

1 Introduction

Dependency grammars is a family of grammar formalisms that are quite important in contemporary speech and language processing systems (Daniel Jurafsky, 2019). The dependency parsing task is to identify pairs of a dependent token and a head token that have dependency relation and their dependency relation labels in a given sentence. For decades, researchers have applied dependency parsing in many tasks of natural language processing such as information extraction, coreference resolution, question-answering, semantic parsing, etc.

Universal dependency parsing shared-task was proposed in VLSP 2020 evaluation campaign to promote the development of dependency parsers for Vietnamese(HA My Linh, 2020). The shared-task published a training corpus of approximately 10,000 dependency-annotated sentences. There are two parts of testing, the first one requires the participant to parse from the input as raw texts where no linguistic information is available. And the second, participant systems will have to parse dependencies information from linguistics annotated sentences. On the CoNLL-U formated test dataset, with the best model, we reached 84.08% UAS score and 75.64% LAS score (averaged on seven test sets). With the raw text dataset, we obtained 74.47% UAS score and 65.30% LAS score.

2 Related Works

Dependency parsing consists of transition-based, graph-based, and grammar-based parser (Nivre and Kübler, 2009). A graph-based algorithm finds the highest scoring parse tree from all possible outputs of an input sentence, scoring each complete tree, while a transition-based algorithm builds a parse by a sequence of actions and scoring each action individually (Zhang and Clark, 2008).

In 2016, Kiperwasser & Goldberg presented a scheme for dependency parsing which is based on bidirectional-LSTMs. The BiLSTM is trained jointly with the parser objective(Kiperwasser and Goldberg, 2016). The effectiveness was demonstrated in two ways by integrating it into a greedy transition-based parser and a globally optimized first-order graph-based parser. In both cases, this approach yields extremely competitive parsing accuracies.

In 2017, Dozat & Manning build off recent work from Kiperwasser & Goldberg, they use a larger but more thoroughly regularized parser than other recent BiLSTM-based approaches, with biaffine classifiers to predict arcs and labels (Dozat and Manning, 2017). Their parser gained state of the art or near state of the art performance on standard treebanks for six different languages.

3 Methodology

3.1 Data preprocessing

Training data includes 6 files, including 8150 sentences. In which, the number of different UPOS labels assigned is 30 and the number of XPOS labels is 56, these labels are unevenly distributed across the dataset.

Realizing that the appearance of some labels with a low sample count may negatively interfere with the results, we converted the group POS tag to accordingly non-group label, such as 'ADV:G' to 'ADV'. Simultaneously, we merge the labels with the same meanings but the different writing styles, such as Adv and ADV. The histogram of UPOS tag labels and XPOS tag labels after handling are shown in Figure 1 and Figure 2.

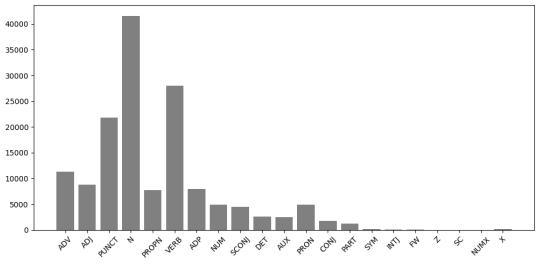


Figure 1: Histogram of processed Upos

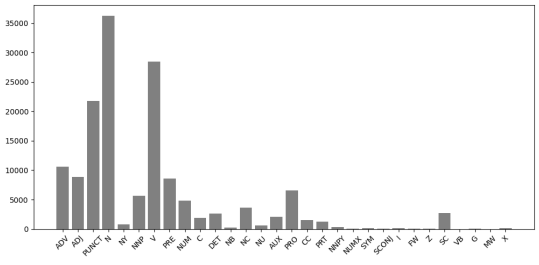


Figure 2: Histogram of processed Xpos

3.2 Proposed System

Tokenization and Sentence Splitting The first step of processing is tokenizing the raw text sentences. We used the VNCoreNLP toolkit to deal with this stage. In Vietnamese, lemmas are the same as the word forms.

POS Tagging To predict POS, we build a BERT-based (Devlin et al., 2019) classifier using bert-base-multilingual-cased pretrained-model available in HuggingFace (Wolf et al., 2020). The bert-base-multilingual-cased includes 12-layer, 768-hidden, 12-heads, 179M parameters, trained on cased text in the top 104 languages with the largest Wikipedia. This model was fine-tuned on the training data on total of 8 epochs using the hyper-parameters shown in Table 1.

Dependency Parsing We implemented a BiLSTM-based deep biaffine neural dependency parser (Dozat and Manning, 2017).

Hyper-parameters	Value
`lr`	2e-5
`eps`	1e-8
Optimizer	AdamW

Table 1: Hyper-parameters of the BERT classifier for pos.

Hyper-parameters	Value
Embedding size	100
Word embedding	fastText
`lr`	3e-3
Optimizer	Adam
LSTM size	400
Deep biaffine size	400
LSTM dropout	0.5
LSTM depth	3

Table 2: Hyper-parameters of the dependency parse.

We used Adam optimizer (Kingma and Ba, 2014) to optimize the network with the learning rate of 0.003 and fastText for word representations (Joulin et al., 2016). With fastText pre-trained word vectors, each word vector has 300 dimensions.

We set the max-steps to 50,000. However, after 3,000 steps without improvement in the validation accuracy, the training process is terminated instead of running through the whole 50,000 steps. After every 100 steps, a model checkpoint will be saved if there is an increase in validation accuracy. Table 2 summarises the hyper-parameters of the dependency parser we used in the parser.

In our experiments, we built two different models for dependency parsing, the first model uses both UPOS and XPOS information as training and predict data and the second model only uses UPOS information during the entire process.

4 Experiments & Results

The VLSP 2020 workshop provides two dependency parsing test datasets. The first one includes data files in raw text format and the other contains data files in which the sentences have been tokenized and stored in the CoNLL-U format.

Table 3 shows the evaluation results on the raw text dataset. Our system achieves 65.30% of LAS and 74.47% of UAS on average. The best result is obtained on the vn3 set which was crawled from VnExpress, with 69.29% of LAS and 77.09% of UAS. In contrast, the result recorded on the vn8 set is the lowest, just 59.35% of LAS and 69.61% of

Model		VTB	vn1	vn3	vn7	vn8	vn10	vn14	Avg. Score
First	UAS(%)	74.55	71.7	**77.09**	70.56	*69.61*	76.41	71.62	**74.47**
Model	LAS(%)	65.34	58.91	**69.29**	65.56	*59.35*	68.22	63.69	**65.30**

Table 3: Results on the raw text dataset.

Model		VTB	vn1	vn3	vn7	vn8	vn10	vn14	Avg. Score
First	UAS(%)	84.41	*74.34*	84.42	**85.56**	82.88	83.55	82.79	**84.08**
Model	LAS(%)	75.94	*63.37*	76.48	**78.89**	74.84	75.48	76.31	**75.64**
Second	UAS(%)	**83.20**	*68.32*	76.60	71.11	70.56	73.13	75.56	**81.58**
Model	LAS(%)	**75.14**	*55.95*	68.48	61.11	61.09	63.29	68.08	**73.32**

Table 4: Results on the CoNLL-U formated dataset.

UAS. One of the reasons that can be mentioned is that the subject of vn8 is somewhat different from the other data sets.

Table 4 presents the evaluation results on the CoNLL-U datasets. The model using both UPOS and XPOS information for training gives better results, 84.08% of UAS and 75.64% of LAS on average of seven datasets. This model works best on the vn7 dataset, reaching 85.56% of UAS and 78.89% of LAS. However, it performs worse on the vn1 set, obtains only 74.34% of UAS and 63.37% of LAS. The second model which uses only UPOS and tokens as input on the training process achieves a bit lower performance, with 81.58% averaged UAS score and 73.32% averaged LAS score. The result obtained when adding xpos feature are higher than using only upos feature. It proves that xpos feature has a relatively vital meaning in universal dependency parsing.

Experimental results indicate that the results obtained on raw text dataset is substantially worse than those obtained on data in CoNLL-U format. UAS decreased 9,61% and LAS reduced even more, up to 10.34% on average. A plausible explanation is that the raw data processing is not done effectively enough. On the other hand, the results that we achieved are relatively low compared to the evaluation on English data (Wilie et al., 2020). However, it implies that there will probably still be room for improvement.

5 Conclusion

In this paper, we present our experiments for the Vietnamese universal dependency parsing task at VLSP 2020 Evaluation Campaign. For raw text processing, we combine several toolkits and models. At the first step, we choose the VNCoreNLP toolkit as a tokenizer. Then a BERT classifier is used to detect the universal part-of-speech tags and Vietnamese part-of-speech tags. At the end, a Bi-LSTM-based deep biaffine neural dependency parser is implemented to produce dependency parsing results. We have obtained promising results on the test dataset, although the results are still lower than results on English datasets. It indicates that our approach probably still has space for growth. Our experiment includes separate modules, which are not inextricably linked. In the future works, we plan to continue doing experiments and improving the dependency parsing model. Next, we plan to build a comprehensive and unified pipeline system which processes raw text and generates dependencies information. In addition, we will also analyze more carefully the pre-processing and processing stages to give a convincing explanation for the difference between the results on the CoNLL-U formatted dataset and raw-text dataset, as well as the difference between files in these datasets.

References

James H. Martin Daniel Jurafsky. 2019. Chapter 15 Dependency Parsing. In *Speech and Language Processing*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing.

VU Xuan Luong NGUYEN Thi Luong PHAN Thi Hue LE Van Cuong HA My Linh, NGUYEN Thi Minh Huyen. 2020. Vlsp 2020 shared task: Universal dependency parsing for vietnamese. In *Proceedings of The seventh international workshop on*

Vietnamese Language and Speech Processing (VLSP 2020), Hanoi, Vietnam.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext.zip: Compressing text classification models.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional LSTM feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327.

Joakim Nivre and Sandra Kübler. 2009. Dependency parsing. *Synthesis Lectures on Human Language Technologies*, 2.

Bryan Wilie, Karissa Vincentio, Genta Indra Winata, Samuel Cahyawijaya, Xiaohong Li, Zhi Yuan Lim, Sidik Soleman, Rahmad Mahendra, Pascale Fung, Syafri Bahar, and Ayu Purwarianti. 2020. Indonlu: Benchmark and resources for evaluating indonesian natural language understanding.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yue Zhang and Stephen Clark. 2008. A tale of two parsers: Investigating and combining graph-based and transition-based dependency parsing. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 562–571, Honolulu, Hawaii. Association for Computational Linguistics.

Vietnamese-English Translation with Transformer and Back Translation in VLSP 2020 Machine Translation Shared Task

LE Duc Cuong **NGUYEN Thi Thu Trang***
School of Information and Communication Technology
Hanoi University of Science and Technology
cuongad1999@gmail.com *trangntt@soict.hust.edu.vn*

Abstract

Transformers have been proven to be more effective for machine translation and many NLP tasks. However, those networks may not work well to low-resource translation tasks, such as the one for the English-Vietnamese language pair. Therefore, this paper aims to enhance the quality of the machine translation model by using the transformer model with a back-translation technique. An intermediate translation system was built using the bilingual dataset as a training corpus. This system was then used with a large monolingual dataset to generate the back-translation data, which can be considered as augmented training data for the translation model. The experimental result on the IWSLT'15 English-Vietnamese test set showed that the system with back-translation outperforms about 2.4 BLEU points than the system with the only transformer. With the test set of the Machine Translation shared task in VLSP 2020, the proposed system with back-translation was ranked as the first place with the highest score of human evaluation (1.55 points, compared to 1.33 points for second place). With the automatic evaluation, the system achieved a 32.1 BLEU score and a 0.50 TER score on VLSP 2020 Machine translation task test data.

1 Introduction

The demand for translation from one language to another is increasing due to the explosion of the Internet and the exchange of information between various regions using different regional languages. Machine translation has long been a major problem in the field of Natural Language Processing (NLP). Neural Machine Translation (NMT) has recently been put into research and has made huge improvements to machine translation systems. Most NMT systems are based on an encoder-decoder architecture consists of two neural networks (Bahdanau et al., 2016; Luong et al., 2015). The encoder compresses the source strings into a vector, used by the decoder to generate the target sequence. Sequence-to-sequence networks consist of two Recurrent Neural Networks (RNNs) and an attention mechanism has significant improvements compared to the traditional statistical machine translation approach.

To the best of our knowledge, transformer architecture networks have achieved the best results for many languages (Vaswani et al., 2017; Wang et al., 2019; Edunov et al., 2018). Transformer is a network architecture based on a self-attention mechanism. Transformers are good at machine translation and many NLP tasks because they totally avoid recursion, by processing sentences as a whole and by learning relationships between words thanks to multi-head attention mechanisms and positional embeddings. Recent networks include a number of parameters and they mostly focus on high-resource language pairs data.

However, those networks may not work well to low-resource translation tasks such as English-Vietnamese. Preparing a good quality bilingual data set is quite difficult, while the amount of monolingual data is quite abundant and available online. That raises a basic idea of using this single language data source to enhance the quality of the machine translation model. Some approaches to solving this problem include creating a language model to improve the quality of the machine translation model (Sennrich et al., 2016) or using back-translation.

In this paper, we propose a machine translation system participating in the Machine Translation Shared Task in VLSP 2020 (Thanh-Le et al., 2020). The main translation model in this system is the transformer with back-translation. This technique can be considered semi-supervised learning, whose

*Corresponding author

main purpose is data augmentation. Despite being simple, the back translation technique has achieved great improvements in both SMT (Bojar and Tamchyna) and NMT (Edunov et al., 2018).

The rest of this paper is organized as follows. Section 2 presents related works using encoder-decoder and back-translation architecture. Our methodology is presented in Section 3. The experiments are shown in Section 4 and Section 5. Finally, Section 6 concludes the paper and gives some perspectives for the work.

2 Related work

We build upon recent work on neural machine translation which is typically a neural network with an encoder/decoder architecture. The encoder represents information of the source sentence, while the decoder is a neural language model based on the output of the encoder. The parameters of both models are learned together to maximize the occurrence of target sentences with corresponding source sentences from a parallel corpus (Sutskever et al., 2014). At inference, a target sentence is generated by left-to-right decoding. Different neural architectures have been proposed with the goal of improving the efficiency of the translation system. This includes recurrent networks (Sutskever et al., 2014; Bahdanau et al., 2016; Luong et al., 2015), convolutional networks (Kalchbrenner et al., 2014; Gehring et al., 2017) and transformer networks (Vaswani et al., 2017). Recent work is based on the attention mechanism in which the encoder generates a sequence of vectors for each target token, the decoder pays attention to the most relevant part of the source through the weights of the vectors encoder (Bahdanau et al., 2016; Luong et al., 2015). Attention has been refined with self-attention and multi-head attention (Vaswani et al., 2017). The baseline model of our system is the transformer architecture (Vaswani et al., 2017).

The idea of back-translation has been suggested since statistical machine translation, where it was used for semi-supervised learning (Bojar and Tamchyna) or self-training (Vandeghinste, 2011). In the modern NMT study, (Sennrich et al., 2016) reported significant increases in terms of WMT and IWSLT shared tasks (Edunov et al., 2018), while (Currey et al., 2017) reported similar findings on low resource conditions, suggesting that even poor translations can make progress.

3 Methodology

3.1 Our proposed system architecture

Aforementioned, for the low-resource bilingual dataset like English-Vietnamese, we proposed to use the back-translation technique as an augmentation technique to build more data for the training corpus. Back-Translation can be considered as a semi-supervised learning technique. Firstly, an intermediate machine translation system is trained using existing parallel data. This system is used to translate the target to the source language. The result is a new parallel corpus in which the source side is a translation synthesizer while the target is the text is written by humans (monolingual dataset). Then, the synthesized parallel corpus is combined with the real text (bilingual dataset) to train the final system. Back-Translation does not need to change model architecture unlike using a language model. The basic idea to use the language model is scoring the candidate words proposed by the translation model at each time step or concatenating the hidden states of the language model and the decoder.

Figure 1 illustrates our proposed system architecture. In this paper, we adopted Transformer as the main translation model. Both monolingual and bilingual datasets must be cleaned and preprocessed before feeding to the Transformer model, which will be presented in subsection 3.2.

To build the final translation model, three main phases have to be performed:

- **Phase 1:** Training a Vietnamese-English translation model with transformer using the bilingual dataset.
- **Phase 2:** Generating an extra bilingual dataset from the monolingual dataset using the Vietnamese-English translation model in the previous phase. During this phase, we used greedy decoding to speed up the data generation process because the monolingual data set was quite large.
- **Phase 3:** Combining generated extra bilingual dataset with origin bilingual one and train the final Vietnamese-English translation model.

We use the same transformer architecture for the English-Vietnamese or Vietnamese-English translation model. Detail description of this architecture is presented in Subsection 3.3.

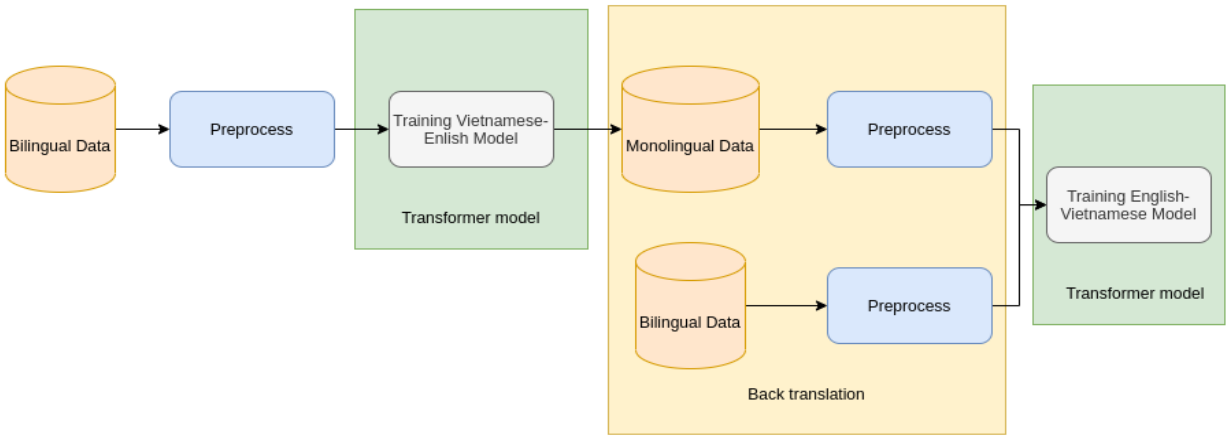


Figure 1: The proposed system architecture.

3.2 Text Pre-processing

3.2.1 VSLP 2020 Datasets

We received and only used two datasets from VLSP 2020 translation task (Thanh-Le et al., 2020) to develop our model. The monolingual dataset included about 20 million sentences crawled from a number of different e-newspapers. The bilingual database had about 4.14 million sentences from many different domains, presented in Table 1.

The bilingual dataset was used to train both English-Vietnamese and Vietnamese-English model while the monolingual dataset was used to create the back-translation dataset.

Table 1: The bilingual dataset on multi-domains

Dataset	Domain	Size (sentences)
News	News (in-domain)	20.0K
Basic	Basic conversations	8.8K
EVBCorpus	Mixed domains	45.0K
TED-like	EduTech talks	546.0K
Wiki-ALT	Wikipedia articles	20.0K
OpenSubtitle	Movie Subtitles	3.5M

3.2.2 Data cleaning and Pre-processing

The bilingual dataset was manually labeled by VLSP organizers so the problems with low translation quality are few. Therefore, we only need to remove too long sentence pairs in this dataset. All sentences having more than 250 words were eliminated.

Meanwhile, the monolingual dataset was crawled on the Internet. Therefore, this dataset had some problems in the raw text, e.g. too long sentences (due to the fault of the sentence tokenizer), non-Vietnamese language, HTML characters. We need a number of steps for data cleaning and preprocessing for this dataset. Some main steps were taken as follows.

- Removing non-Vietnamese sentences: Filtering out sentences that are not in Vietnamese using a language detection model;
- Removing sentences that are too long or too short;
- Cleaning HTML characters and some special characters.

After the data cleaning and pre-processing, the monolingual dataset had nearly 20 million remaining sentences, while the bilingual one had a total of 4.1 million sentence pairs. The data were cleaned, normalized, then lower-cased and tokenized using the Moses[1] tool. The data were learned a BPE set of 35,000 items using the Subword Neural Machine Translation toolkit[2].

3.3 The Transformer Model

The core idea behind the Transformer model is self-attention, the ability to attend to different positions of the input sequence to compute a representation of that sequence. The transformer creates stacks of self-attention layers to build both encoder and decoder instead of RNNs or CNNs. This general architecture helps transformer model calculated in parallel, instead of a series like RNNs, and learn long-range dependencies. The transformer architecture is presented in Figure 2.

Without the recurrence or the convolution, the transformer encodes the positional information of each input token by a position encoding function.

[1]Moses Open Source Toolkit for Machine Translation
[2]https://github.com/rsennrich/subword-nmt

Thus the input of the bottom layer for each network can be expressed as $Input = Embedding + PositionalEncoding$. The positional encoding is added on top of the actual embeddings of each word in a sentence.

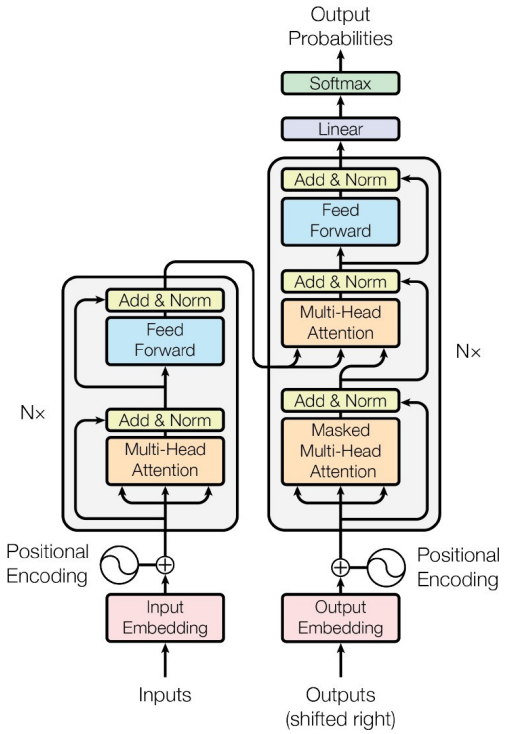


Figure 2: Transformer architecture.

The encoder has several layers stacked together. Each layer consists of a multi-head self-attention mechanism and a position-wise fully connected feed-forward network. Multi-head self-attention mechanism help model can pay "attention" to many certain pieces of content of the input.

The decoder is also a stack of identical layers, each layer comprising three sub-layers. At the bottom is a masked multi-head self-attention, which ensures that the predictions for position i depend only on the known outputs at the positions less than i. In the middle is another multi-head attention which performs the attention over the encoder output. The top of the stack is a position-wise fully connected feed-forward sub-layer. The decoder output finally goes through a linear transform with softmax activation to produce the output probabilities.

4 Experiment

4.1 Experimental setup

Transformer setup. We use the Transformer model in PyTorch from the fairseq toolkit[3]. All experiments were based on the Big Transformer architecture with 6 blocks in the encoder and decoder. We used the same hyper-parameters for all experiments, word representations of size 1024, feed-forward layers with inner dimension 4096. We used 16 attention heads, and we average the checkpoints of the last ten epochs. Models were optimized with Adam optimization using $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\varepsilon = 1e^{-}8$.

Back-translation set up. We run experiments on 2 GPU Tesla V100 and spent about 36 hours training the final model.

4.2 Automatic Evaluation and Human Evaluation

VLSP organizers provided two evaluation results for each model: (i) Automatic evaluation, and (ii) Human evaluation.

4.2.1 Automatic evaluation

In VLSP 2020, the automatic evaluation was used for reference, but not for the final decision for system ranking. The two metrics were BLEU and TER scores.

BLEU is a quality metric score for MT systems that attempts to measure the correspondence between a machine translation output and a human translation, as illustrated in Equation 1. The central idea behind BLEU is that the closer a machine translation is to a target human translation, the better it is.

$$\frac{\sum_{C \in Candidates} \sum_{ngram \in C} Count_{clip}(ngram)}{\sum_{C' \in Candidates} \sum_{ngram' \in C'} Count_{clip}(ngram')} \quad (1)$$

Translation Edit Rate (TER) is a method to determine the amount of Post-Editing required for machine translation jobs. The automatic metric measures the number of actions required to edit a translated segment inline with one of the reference translations, as illustrated in Equation 2.

$$TER = \frac{number\ of\ edits}{length\ of\ reference\ sentence} \quad (2)$$

[3]https://github.com/pytorch/fairseq

4.2.2 Human Evaluation

Human evaluation is the main metrics for ranking participating systems. There were 5 experts who were professional Vietnamese-English translators or interpreters. Each subject was asked to rate all systems from 1 to 6 based on Adequacy and Fluency. The overall rank was calculated by using the TrueSkill algorithm. TrueSkill is a rating system among game players. It was developed by Microsoft Research and has been used on Xbox LIVE for ranking and matchmaking services. This system quantifies players' TRUE skill points by the Bayesian inference algorithm.

5 Experimental Results

5.1 Experiment for Back-Translation

To find out the role of back-translation, we did some experiments on the IWSLT'15 English-Vietnamese test set. This test set is used from Stanford NLP group and has 1268 pairs Vietnamese-English sentence. Table 2 presents the results of the systems that used and did not use the back-translation (the baseline model with the transformer only). The experimental result showed that the model with back translation outperforms to the baseline one about 2.4% in BLUE score.

Table 2: Experimental results for the back translation on the IWSLT'15 English-Vietnamese test set

Model	BLEU score
Transformer (baseline)	36.3
Transformer + Back-Translation	38.7

5.2 VLSP 2020 Experimental Results

VLSP organizers released 2 test sets: a public test set and a private test set. The public test has 1220 pairs in the news domain while the private test is collected from online newspapers about Covid-19 news articles, about 789 pairs.

The result running on the private test is shown in Table 3. The final model that we submitted was our proposed system, which used Transformer and Back-Translation. Our system achieved a 32.10 BLEU score and a 0.5 TER score. According to the results of VLSP organizers, our BLEU score was at third and TER score is at second. However, the human evaluation of our system got the best result, which was 1.554. This led our system to be the first rank in the Machine Translation shared task in VLSP 2020.

As in Table 3, the automatic evaluation (BLEU) was on pair with human evaluation except in the case of our system. A possible reason was found that our system did not do casing recovery. The automatic evaluation metrics do consider casing, but the experts do not.

Table 3: Score of systems by VLSP organizer

Team	BLEU	TER	Human score
Our System*	***32.10***	***0.50***	***1.554***
EngineMT	38.39	0.45	1.327
RD-VAIS	33.89	0.53	0.864

5.3 Observations

After having some observations on the outputs of the baseline and the system with back translation, we find that the model using back translation gave more natural results than the baseline one.

For instance, as shown in the Table 4 by removing the duplicated pronounce "họ" (them) in the output, model using back translation avoids repeating words and makes the sentence more natural.

Table 4: Removing duplicated pronounces with back translation

Input: they will go back home to celebrate tet together with their families.
Baseline model: họ sẽ về nhà để ăn mừng tết với gia đình **họ**.
Baseline model + Back translation: họ sẽ trở về nhà để ăn mừng tết với gia đình.

With back translation, more suitable terms were selected in a specific context. As illustrated in Table 5, with the back translation mechanism, the "characteristics" word was translated into "đặc điểm" (properties), which suited best in the context. Whereas, the baseline model without back translation translated to "tính cách" (traits), typically one belonging to a person.

In addition, in some cases, back translation also helps the model generate some additional words, which can help to increase the fluency of the translation sentences (Table 6). This enhances the nat-

Table 5: More suitable terms with back translation

Input: typhoid's characteristics are continuous fever , high fever up to 40°C , excessive sweating , gastroenteritis and uncolored diarrhea.
Baseline model: **tính cách** của bệnh thương hàn là bệnh sốt liên tiếp, sốt cao lên đến 40 độ c, đổ mồ hôi quá nhiều, viêm dạ dày ruột và tiêu chảy không có màu.
Baseline model + Back translation: **đặc điểm** của bệnh thương hàn là sốt liên tiếp, sốt cao lên tới 40 độ c, đổ mồ hôi quá nhiều, viêm dạ dày và tiêu chảy không có màu.

Table 6: More natural expression with back translation

Input: thuan suggest to the delegation, in the short term to hurry up to prevent the epidemy, treat the disease, moreover, in the long term to make the whole team understand about malaria prevention method and therefore they will prevent disease for themselves which is also prevent disease for the whole team.
Baseline model: thuận gợi ý với phái đoàn, trong thời gian ngắn để nhanh chóng ngăn chặn sự phát bệnh, điều trị bệnh, hơn nữa, trong lâu dài để làm cho toàn bộ đội hiểu về phương pháp phòng ngừa bệnh sốt rét và do đó họ sẽ ngăn chặn bệnh này cho chính họ cũng sẽ ngăn chặn bệnh này cho cả đội.
Baseline model +Back translation: **ông** thuận gợi ý cho phái đoàn, trong thời gian ngắn để nhanh chóng ngăn chặn biểu mô, điều trị bệnh, hơn nữa, về lâu dài để cả nhóm hiểu về phương pháp phòng ngừa bệnh sốt rét và do đó họ sẽ ngăn ngừa bệnh tật cho bản thân, điều này cũng sẽ ngăn ngừa bệnh cho toàn đội.

uralness of the generated expression for the target language.

6 Conclusion

Participating in the machine translation shared task on VLSP 2020, we proposed some data cleaning and pre-processing for both monolingual and bilingual datasets. We did eliminate some very long or very short sentences as well as invalid characters (e.g. HTML, special ones). Some non-Vietnamese sentences in the monolingual dataset were also automatically removed. We proposed to use the transformer as the main translation model with back-translation as a data augmentation technique. An intermediate translation system was built using the bilingual dataset as a training corpus. The back-translation data were generated from the monolingual dataset by using the intermediate translation system. This back-translation data were then combined with the bilingual dataset to form the final training dataset for the final translation system.

The experiment results on the IWSLT'15 English-Vietnamese test set suggested that the back-translation is an effective data augmentation technique for deep learning machine translation models, which made an enhancement from 36.3 to 38.7 of the BLEU score. With the test set of Machine Translation shared task of VLSP 2020, this technique seemed can adapt quite well on the news domain. Our system with the back-translation technique was ranked as the first place with the highest score of human evaluation (i.e. 1.55 points, compared to 1.33 of the second place). With the automatic evaluation, the system achieved a 32.1 BLEU score and a 0.50 TER score on VLSP 2020 Machine translation task test data.

We will do some experiments on a number of sampling data methods during the preparation of back-translation datasets. We also consider analyzing and investigating the correspondences between human evaluation and automatic ones.

Ackknowledgement

This work was supported by the Vingroup Innovation Foundation (VINIF) under the project code DA116_14062019 / year 2019.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2016. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv:1409.0473 [cs, stat]*. ArXiv: 1409.0473.

Ondr˘ej Bojar and Ales˘ Tamchyna. Improving Translation Model by Monolingual Data. page 7.

Anna Currey, Antonio Valerio Miceli Barone, and Kenneth Heafield. 2017. Copied Monolingual Data Improves Low-Resource Neural Machine Translation. In *Proceedings of the Second Conference on*

Machine Translation, pages 148–156, Copenhagen, Denmark. Association for Computational Linguistics.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding Back-Translation at Scale. *arXiv:1808.09381 [cs]*. ArXiv: 1808.09381.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional Sequence to Sequence Learning. *arXiv:1705.03122 [cs]*. ArXiv: 1705.03122.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A Convolutional Neural Network for Modelling Sentences. *arXiv:1404.2188 [cs]*. ArXiv: 1404.2188.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. *arXiv:1508.04025 [cs]*. ArXiv: 1508.04025.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving Neural Machine Translation Models with Monolingual Data. *arXiv:1511.06709 [cs]*. ArXiv: 1511.06709.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. *arXiv:1409.3215 [cs]*. ArXiv: 1409.3215.

Ha Thanh-Le, Tran Van-Khanh, and Nguyen Kim-Anh. 2020. Goals, challenges and findings of the vlsp 2020 english-vietnamese news translation shared task. *Proceedings of the Seventh International Workshop on Vietnamese Language and Speech Processing (VLSP 2020)*.

V. Vandeghinste. 2011. Learning Machine Translation. * Cyril Goutte, Nicola Cancedda, Marc Dymetman, and George Foster. *Literary and Linguistic Computing*, 26(4):484–486.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. *arXiv:1706.03762 [cs]*. ArXiv: 1706.03762.

Dilin Wang, Chengyue Gong, and Qiang Liu. 2019. Improving Neural Language Modeling via Adversarial Training. *arXiv:1906.03805 [cs, stat]*. ArXiv: 1906.03805.

The UET-ICTU Submissions to the VLSP 2020 News Translation Task

Thi-Vinh Ngo[1], Minh-Thuan Nguyen[2], Minh Cong Nguyen Hoang[2]
Hoang-Quan Nguyen[2], Phuong-Thai Nguyen[2], Van-Vinh Nguyen[2]

[1]*University of Information and Communication Technology, TNU, Viet Nam*
[2]*University of Engineering and Technology, VNU, Viet Nam*
ntvinh@ictu.edu.vn, npthai@vnu.edu.vn

Abstract

Our UET-ICTU team includes members from the University of Engineering and Technology (UET) and Thai Nguyen University of Information and Communication Technology (ICTU). We participate in the VLSP 2020 Shared Task for Machine Translation which focuses on the news domain translation in one direction English → Vietnamese. Our neural machine translation (NMT) system uses Back Translation (BT) of monolingual data in the target language to augment synthetic training data. Besides, we leverage the Term Frequency and Inverse Document Frequency (TF-IDF) method to data selection close to the in-domain from other monolingual and parallel resources. To enhance the effectiveness of the system translation, we also employ other techniques such as fine-tuning and assembly translation. Our experiments showed that the system can achieve a significant improvement in BLEU score up to + 16.57 overcoming the in-domain baseline system.

1 Introduction

The University of Engineering and Technology (UET) and Thai Nguyen University of Information and Communication Technology (ICTU) participate in the VLSP 2020 Shared Task for Machine Translation on news domain translation from English to Vietnamese (Ha et al., 2020). From datasets in different domains of the Shared Task, we use various strategies to improve the quality of translation in the news domain.

Data selection Data selection techniques help MT systems better translate on a specific domain by eliminating irrelevant data from resources outside the in-domains. This reduces training time but still preserve performance when using smaller datasets instead of training on the large ones. Many works show several methods to select sentences close to background corpus such as: (Axelrod et al., 2011; van der Wees et al., 2017) compute scores for sentences out of domain corpus based on cross-entropy difference (CED) (Moore and Lewis, 2010) from language models; (Wang et al., 2017; Zhang and Xiong, 2018) use sentence embeddings to rank source sentences. This method is only suitable for recurrent networks in NMT. (Wang et al., 2018; Zhang and Xiong, 2018) investigate the translation probability $P(y|x, \theta)$ to be a dynamic criterion to extract sentence pairs during the training process. (Peris et al., 2016) train a neural network classifier to classify sentences into negative or positive fields. These works require training either language models or neural networks and they are less effective in the data sparse situations. (Silva et al., 2018) show empirical results in three various strategies as CED (Moore and Lewis, 2010), TF-IDF (Salton and Yang, 1973) and Feature Decay Algorithms (FDA) (Poncelas et al., 2017). They show that the TF-IDF method has achieved the best improvements in both BLEU and TER (Translation Error Rate) measures. This technique is simple, fast, and does not require training language models or neural networks. Therefore, in this paper, we will leverage it to rank sentences in the scenario that in-domain corpus is small. The detail of this method will be presented in section 3.

Using monolingual resource Monolingual data is used widely in machine translation (MT) (Sennrich et al., 2015; Ha et al., 2017; Lample et al., 2018; Siddhant et al., 2020) due to its widely available. In this paper, we create additional synthetic parallel training data using BT method in (Sennrich et al., 2015) and investigate its effectiveness in our MT systems by combining with genuine parallel data.

Fine-tuning (Luong and Manning, 2015; Zoph et al., 2016) have proposed the fine-tuning process to transfer some of the learned parameters from the parent model to the child model and have

shown significant improvements in many translation tasks. Our systems also fine-tuning on subcorpus (a smaller corpus is extracted from a large corpus) to achieve the best translation effectiveness.

Ensemble translation Ensemble translation (Luong et al., 2015) enable to incorporate the outputs of trained models to enhance translation systems. We attempt to investigate this strategy in our MT system.

Our paper demonstrates a substantial improvement in translating the news domain from the VLSP 2020 Shared Task when combining the aforementioned techniques.

In Section 2, we present an overview of Neural Machine Translation and focus on the transformer architecture. The details of the methods in our paper are presented in Section 3. The settings of the translation system and experimental results are discussed the Section 4. Related works are showed in Section 5. Finally, conclusions and future works are described in Section 6.

2 Neural Machine Translation

Neural Machine Translation (Cho et al., 2014; Sutskever et al., 2014) uses memory units such as Gated Recurrent Units (GRU) or Long Short-Term Memory (LSTM) to overcome the exploding or vanishing gradient problem in recurrent networks. They suggest a new architectural type for MT systems in the form of end-to-end. It includes an encoder to present the sentence in the source language including n tokens $X = (x_1, x_2, ..., x_n)$ into the continue space and a decoder to generate the predicted sentence $Y = (y_1, y_2, ..., y_m)$ in the target language containing m tokens.

The attention mechanism (Luong et al., 2015a; Bahdanau et al., 2015) is considered as the soft-alignment between a source sentence and the corresponding target sentence to enhance the effectiveness of the systems.

Due to the fact that recurrent neural networks (RNN) have limited parallelization in the training process, (Vaswani et al., 2017) propose the transformer architecture that may be highly parallelizable as well as better in translating long sentences. In the transformer, instead of using GRU or LSTM units, a word attends to the other words in a sentence using the self-attention mechanism as the following:

$$Attention(Q, K, V) = Softmax(\frac{QK^T}{\sqrt{d}})V \quad (1)$$

where K (key), Q (query), V (value) present the hidden states of tokens in the input sentence from encoder or decoder and d is the size of the input.

The attention mechanism in the transformer is the variant of the original attention (Luong et al., 2015a; Bahdanau et al., 2015) when we replace queries by the decoder's hidden states while keys and values come from the encoder's hidden states in the equation 1.

The NMT system is trained to optimize its parameters θ through minimizing the maximum likelihood of all sentence pairs.

$$\mathcal{L}(\theta) = \frac{1}{T}\sum_{k=1}^{k=T} logP(Y^k|X^k;\theta) \quad (2)$$

where T is the number of sentence pairs in the bilingual corpus.

3 The strategies improve our MT system

3.1 Data selection

As mentioned in section 1, in this paper, we utilize the TF-IDF method (Salton and Yang, 1973) to extract a subset of data from large datasets. In the method, TF is the term frequency which presents the ratio between the number of times a term (a word or a sub-word) appears in a sentence and the total number of terms in the sentence. IDF is the inverse document frequency which specifies the ratio between the total number of documents and the number of documents containing the term. Thus, an in-domain corpus D contains T sentence pairs, the TF-IDF score of the token w in the sentence s in the general domain G is evaluated as:

$$score_w = TF - IDF_w = \frac{F_w^G}{W_s^G}.\frac{T^D}{K_w^D} \quad (3)$$

where F_w^G is the frequency of w in s, W_s^G is the length of s, and K_w is the number of sentences in D contain w.

The score of the sentence $s \in G$ is calculated as :

$$score_s = \sum_{i=1}^{i=W_s^G} score_{w_i} \quad (4)$$

These scores are then used to rank sentences in corpus G. The sentence which has the highest

score is nearest to the background corpus, and vice versa.

Our work employs this technique to extract both bilingual and monolingual data.

3.2 Back Translation

In order to improve the translation system from the source language X to the target language Y, (Sennrich et al., 2015) trained the backward translation system from Y X, and it is then used to infer monolingual data from the language Y to predict hypotheses in the language X. We will gain the synthetic bilingual data and it is then mixed with the original bilingual data to augment the training corpus. This technique is called Back Translation (BT).

Our paper applied BT to generate pseudo parallel data English-Vietnamese in the limited bilingual data scenario. In reality, the monolingual data is available but the inference in NMT takes a long time, so we leverage the data selection mentioned in section 3.1 to filter monolingual data.

3.3 Fine-tuning

NMT systems are trained on a large corpus, and then continuously fine-tuned on the in-domain corpus to achieve better performance. We train the NMT system on the mixed datasets from various domains, and then fine-tuning on a smaller corpus extracted from original generic corpus using the strategy in section 3.1.

3.4 Ensemble Translation

The outputs of NMT models can be saturated together to predict better hypotheses. We call this ensemble translation (Luong et al., 2015). The combination vector is simply selected from maximum, or minimum or, average (can be then normalized) probabilities of the output vectors. In this work, we attempt to exhaustive the mean of probabilities from three models and find that a trivial improvement comparing to an individual one.

4 Experiments

4.1 Datasets

Our work only employs the datasets from the VLSP 2020 Shared Task for Machine Translation. It includes six bilingual corpora in divergent domains and one Vietnamese monolingual corpus. This Shared Task focuses on translating the News domain. The bilingual datasets are described in Table 1.

No.	Domains	Training	dev	test
1	News (in-domain)	20K	1007	1220
2	Basic	8.8K	-	-
3	EVBcorpus	45K	-	-
4	TED-like	546K	-	-
5	Wiki-ALT	20K	-	-
6	Open subtitle	3.5M	-	-

Table 1: The English-Vietnamese parallel datasets are used in our work

We use 5 datasets from (1) to (5) for training experiments, the Open subtitle corpus is only used for learning sub-word units in English. The Vietnamese monolingual corpus which includes 20M sentences is exploited for the back translation.

4.2 Preprocessing

We firstly tokenized and true-cased English texts using Moses's scripts. Next, we concated all 6 bilingual corpora to learn 40.000 operators Byte Pair Encoding (BPE) codes like (Sennrich et al., 2016). Lastly, the tokenized and true-cased texts were applied to BPE codes.

Vietnamese texts were tokenized and true-cased using Moses's scripts.

4.3 Systems and Training

We conduct our experiments using the source code from `NMTGMinor`[1]. Our NMT system included four layers for both encoder and decoder and the embedding and hidden sizes are 512. The systems are trained with each mini-batch size of 64 sentence pairs (except the baseline system uses 32 sentence pairs). The vocabulary sizes are 50K tokens for both source and target sides. We use dropout with a probability of 0.2 for embedding and attention layers. The Adam optimizer is applied for updating parameters with an initial learning rate of 1.0. A beam size of 10 is employed for the decoding process.

We train our NMT systems after 50 epochs, and then they are fine-tuned on extracted and in-domain corpus to enhance the accuracy.

4.4 Results

We present empirical results in two measures: BLEU (Papineni et al., 2002) and Translation Er-

[1] https://github.com/quanpn90/NMTGMinor

ror Rate (TER) (Snover et al., 2006). They are implemented in sacreBLEU[2].The higher scores in BLEU specify the better translations while the lower scores in TER indicate better ones. Table 2 shows our experimental results.

In-domain system (baseline) We train the baseline system on News corpus. We learn 10K operators BPE codes and then English texts are applied them.

News + 4 corpus We find that the Open subtitle corpus contains sentences that are not news domain. Therefore, we only combine the background corpus with the 4 remaining corpora. We have shown the improvements of +14.13 BLEU points and -0.259 TER scores.

+ Back Translation We rank sentences from Vietnamese monolingual corpus using the data selection method mentioned in section 3.1, and then extract the top 200K sentences from the ranked text. We employ the backward translation system from Vietnamese → English to generate synthetic bilingual data. The synthetic data are then concatenated to the corpus in the system (2) to train again. We obtain +15.31 BLEU and -0.295 TER points.

+ Fine-tuning on ranked corpus We rank 4 parallel corpora from (2) to (5) in Table 1 using the TF-IDF method in section 3.1 again, and then we also extract the top 200K sentence pairs. The extracted data is combined with the background corpus to continuously fine-tuning the system (3) with an initial learning rate at 0.5. The improvements can be found as +16.21 BLEU and -0.297 TER scores.

+ Fine-tuning on News domain We continue to fine-tune the system (4) with an initial learning rate at 0.25 in the in-domain corpus to gain the best performance, + 16.57 BLEU and -0.3 TER scores.

+ Ensemble translation We combine the output of three best models from the system (5) using the method mentioned in 3.4. We see that our system does not improve.

5 Related Work

NMT systems are restricted in domain translation, therefore, previous works have proposed a variety of data selection techniques to retrieve sentences that are the most related to a specific domain. (Axelrod et al., 2011; van der Wees et al., 2017) leverage language model to estimates the cross-entropy difference (CED) (Moore and Lewis, 2010) for sentences from generic domain. (Wang et al., 2017; Zhang and Xiong, 2018) employed the embedding vectors in the source space from NMT systems to rank sentences. (Wang et al., 2018; Zhang and Xiong, 2018) suggested a dynamic selection based on translation probability to classify sentences during the training process. (Peris et al., 2016) train a neural network to separate sentences into individual domains. These methods are quite complex because they require training neural networks or language models. (Silva et al., 2018) conducted experiments on CED, TF-IDF, FDA, and observe that the TF-IDF strategy is very fast and effective for data selection. In this works, we investigate this method again in the English-Vietnamese translation task.

Due to the lack of bilingual data, some prior studies exploited monolingual data in different ways. (Sennrich et al., 2015) proposed BT method by using used monolingual from the target language. (Ha et al., 2017) shown the mix-source technique to create synthetic data by making a copy of the target language. (Lample et al., 2018) used monolingual data for unsupervised NMT. (Siddhant et al., 2020; Ngo et al., 2020) investigated monolingual data in multilingual NMT. Our work also attempts to using BT method to enhance our NMT system in the data sparse issue.

To gain the best performance in the background domain, (Luong and Manning, 2015; Zoph et al., 2016) demonstrate the effectiveness when transferring the knowledge from the parent model to then child model by the fine-tuning technique. We also apply this approach to our NMT system to achieve better improvements. Besides, we attempt to estimates the quality of the system when using ensemble translation in (Luong and Manning, 2015)

6 Conclusion and Future Work

Our NMT systems have achieved significant improvements when integrating simple techniques such as data section, BT, fine-tuning. In the future, we will leverage more data from other resources as well as using pre-trained models to improve the translation system.

7 Acknowledgments

We would like to thank the organizers and sponsors of the VLSP 2020. We also thank reviewers

[2] https://github.com/mjpost/sacrebleu

No.	Systems	dev		test		offical test	
		BLEU	TER	BLEU	TER	BLEU	TER
1	News corpus (In-domain, baseline)	33.42	0.550	31.66	0.568	21.82	0.753
2	News + 4 corpus (basic + evb + Ted-like + wiki-alt)	46.40	0.427	45.13	0.436	36.12	0.494
3	+ Back Translation	46.35	0.418	45.47	0.436	37.13	0.458
4	+ fine-tuning on ranked bilingual data	48.23	0.399	47.32	0.415	38.03	0.456
5	+ fine-tuning on News corpus	48.94	0.399	48.03	0.405	**38.39**	**0.453**
6	+ Ensembly translation	**49.02**	**0.393**	**48.08**	**0.404**	38.32	**0.453**

Table 2: The results of our English → Vietnamese MT systems are measured in BLEU and TER scores.

who review our paper carefully and give us helpful comments.

References

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078.

Thanh-Le Ha, Jan Niehues, and Alexander Waibel. 2017. Effective Strategies in Zero-Shot Neural Machine Translation.

Thanh-Le Ha, Van-Khanh Tran, and Kim-Anh Nguyen. 2020. Goals, challenges and findings of the vlsp 2020 english-vietnamese news translation shared task. In *VLSP 2020*.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised machine translation using monolingual corpora only.

Minh-Thang Luong and Christopher D. Manning. 2015. Stanford neural machine translation systems for spoken language domain. In *International Workshop on Spoken Language Translation*, Da Nang, Vietnam.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015a. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

Thang Luong, Ilya Sutskever, Quoc Le, Oriol Vinyals, and Wojciech Zaremba. 2015. Addressing the rare word problem in neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 11–19, Beijing, China. Association for Computational Linguistics.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224, Uppsala, Sweden. Association for Computational Linguistics.

Thi-Vinh Ngo, Phuong-Thai Nguyen, Thanh-Le Ha, Khac-Quy Dinh, and Le-Minh Nguyen. 2020. Improving multilingual neural machine translation for low-resource languages: French, english - vietnamese. In *Proceedings of the 3rd Workshop on Technologies for MT of Low Resource Languages*, pages 55–61.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Álvaro Peris, Mara Chinea-Rios, and Francisco Casacuberta. 2016. Neural networks classifier for data selection in statistical machine translation. *CoRR*, abs/1612.05555.

Alberto Poncelas, Andy Way, and Antonio Toral. 2017. Extending feature decay algorithms using alignment entropy. pages 170–182.

G. Salton and C. S. Yang. 1973. On the specification of term values in automatic indexing. *Journal of Documentation.*, 29(4):351–372.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *CoRR*, abs/1511.06709.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Association for Computational Linguistics (ACL 2016)*.

Aditya Siddhant, Ankur Bapna, Yuan Cao, Orhan Firat, Mia Chen, Sneha Kudugunta, Naveen Arivazhagan, and Yonghui Wu. 2020. Leveraging monolingual data with self-supervision for multilingual neural machine translation.

Catarina Cruz Silva, Chao-Hong Liu, Alberto Poncelas, and Andy Way. 2018. Extracting in-domain training corpora for neural machine translation using data selection methods. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 224–231, Brussels, Belgium. Association for Computational Linguistics.

Matthew Snover, Bonnie J. Dorr, R. Schwartz, and L. Micciulla. 2006. A study of translation edit rate with targeted human annotation.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Rui Wang, Andrew Finch, Masao Utiyama, and Eiichiro Sumita. 2017. Sentence embedding for neural machine translation domain adaptation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 560–566, Vancouver, Canada. Association for Computational Linguistics.

Rui Wang, Masao Utiyama, and Eiichiro Sumita. 2018. Dynamic sentence sampling for efficient training of neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 298–304, Melbourne, Australia. Association for Computational Linguistics.

Marlies van der Wees, Arianna Bisazza, and Christof Monz. 2017. Dynamic data selection for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1410, Copenhagen, Denmark. Association for Computational Linguistics.

Shiqi Zhang and Deyi Xiong. 2018. Sentence weighting for neural machine translation domain adaptation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3181–3190, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, Texas. Association for Computational Linguistics.

VLSP 2020 Shared Task: Universal Dependency Parsing for Vietnamese

HA My Linh[1], NGUYEN Thi Minh Huyen[1], VU Xuan Luong[2], NGUYEN Thi Luong[3], PHAN Thi Hue[4], LE Van Cuong[5]

[1]VNU University of Science, Hanoi, Vietnam
{huyenntm, hamylinh}@hus.edu.vn
[2]Vietnam Lexicography Center, Hanoi, Vietnam
vuluong@vietlex.com
[3]Dalat University, Lamdong, Vietnam
luongnt@dlu.edu.vn
[4]Coc Coc Company, Hanoi, Vietnam
phanhuek51nn@gmail.com
[5]University of Social Sciences and Humanities, Hanoi, Vietnam
cuongle.ussh@gmail.com

Abstract

This paper describes the shared task on Vietnamese universal dependency parsing at the seventh workshop on Vietnamese Language and Speech Processing (VLSP 2020[1]). This challenge, following the first edition in 2019, aims to provide the VLSP community with gold universal dependency annotated datasets for Vietnamese and to evaluate dependency parsing systems based on the same training and test sets. Consequently, the best systems made available to the community would be promoted for using in further applications. Each participant was provided with the same training data with more than 8000 annotated sentences and returned the result on a test set of more than 1000 sentences. Contrary to the first edition, where the test set was pre-processed with word segmentation and part-of-speech (POS) tagging in CoNLL-U format, participants of this year compete on two tracks: one track with raw texts and the other with pre-processed texts as test input. In this report, we define the shared task and describe data preparation, as well as make an overview of methods and results performed by VLSP 2020 participants.

1 Introduction

Dependency parsing is the task of determining syntactic dependencies between words in a sentence. The dependencies include, for example, the information about the relationship between a predicate and its arguments, or between a word and its modifiers. Dependency parsing can be applied in many tasks of natural language processing such as information extraction, co-reference resolution, question-answering, semantic parsing, etc.

Many shared-tasks on dependency parsing have been organized since 2006 by CoNLL (The SIGNLL Conference on Computational Natural Language Learning), not only for English but also for many other languages in a multilingual framework. The CoNLL 2017 Shared Task was done on 81 test sets from 49 languages and in CoNLL 2018 Shared Task (Zeman et al., 2018), there were 82 test sets from 57 languages. From 2017, a Vietnamese dependency treebank containing 3,000 sentences is included for the CoNLL shared-task "Multilingual Parsing from Raw Text to Universal Dependencies". However, this Vietnamese dependency treebank is still small and contains several errors because of automatic conversion from the version 1 to version 2 of Universal Dependencies[2] (UD v2).

In the framework of the VLSP 2019 and 2020 workshops, one of the shared-tasks is on Vietnamese dependency parsing, in order to promote the development of dependency parsers for Vietnamese. Based on newly revised guidelines for Vietnamese dependency treebank following the UD v2 annotation scheme, training and test sets have been annotated. The label set and guidelines on word segmentation and POS tagging were equally revised, in agreement with the universal principles. In 2020, participants are provided with more than 8,000 sentences for the training dataset. The test set includes more than 1000 sentences provided in two formats as two tracks of the challenge: one is raw text and the other is text segmented in words and POS tagged. The tool provided for evaluating

[1]https://vlsp.org.vn/vlsp2020

[2]https://universaldependencies.org/v2/index.html

dependency parsing models by the CoNLL 2018 shared task is used in the framework of VLSP 2019 and 2020 dependency parsing shared task.

Five participant systems have been evaluated in VLSP 2020. After the description of the datasets and evaluation methods, we give an overview of models developed by participant systems and discuss the results obtained by these systems on the two tracks of the shared tasks.

2 Data preparation

Training and test datasets have been automatically generated by a draft parsing system and manually revised by annotators. We introduce the set of dependency labels first, then the annotation process and finally the datasets built for the shared task of dependency parsing.

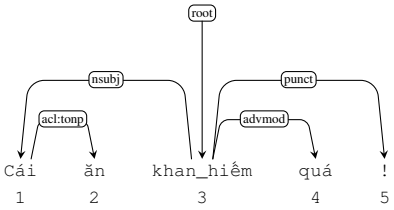
2.1 Dependency labels

In 2017, the NLP group of the VNU University of Science (Nguyen et al., 2018) has developed a Vietnamese dependency dataset of 3,000 sentences which were then integrated into Stanford University's dependency project. The label set is composed of 48 dependency labels, defined based on Universal Dependency label set (version 1). The 3,000 sentences of this dataset are extracted from VietTreebank - a constituency treebank, then automatically transformed into a dependency treebank. The process is terminated by a manual revision, although there exist inevitably some errors from inexperienced annotators. The UD v2 version of this dataset in Universal Dependency repositories was automatically generated from the version 1, it contains consequently much more errors.

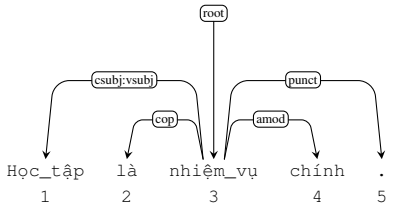
For the dependency shared task organized in the framework of VLSP 2019 and VLSP 2020 workshops, we have reviewed entirely the set of dependency labels and defined a set of 38 types and 47 language-specific subtypes of dependency relations in accordance with the guidelines for Universal dependency relations[3] version 2.

Here are some new dependency labels specific for Vietnamese language.

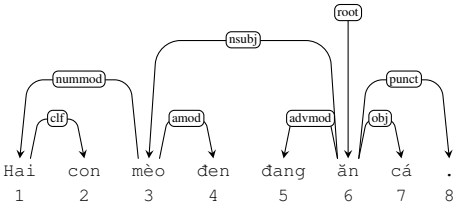
- *acl:tonp*: Usually a verb in Vietnamese can be nominalized by adding a classifier noun such as *cái*, *việc*, *sự*, ... before the verb. Example: *Cái*[classfier] *ăn*[to eat] *khan hiếm*[scarce] *quá*[too]!/The food is too scarce!



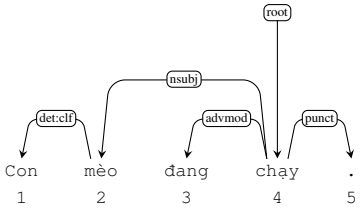
- *csubj:vsubj*: This relationship is used in the case that a verb is the subject of a sentence. In Vietnamese, the subject is usually nominal, but in some cases, adjective or verb without any derivation can be subject. Example: *Học tập*[Study] *là*[to be] *nhiệm vụ*[task] *chính*[main]./Studying is the main task.



- *clf*: Syntactically, classifier noun is rather bound to the numeral in the nominal group than the head noun. Therefore, in agreement with the guidelines of UD version 2, we treat classifiers as functional dependents of numerals, using the *clf* relation. Example: *Hai*[Two] *con*[classifer] *mèo*[cat] *đen*[black] *đang*[tense/aspect marker] *ăn*[to eat] *cá*[fish]./Two black cats are eating fish.



- *det:clf*: When a classifier noun does not appear with a specified quantity or pronoun words, the classifier noun have the same properties as a determiner. The relation of a classifier noun with the head noun is thus *det:clf*. Example: *Con*[Classifier] *mèo*[cat] *đang*[tense/aspect marker] *chạy*[to run]./The cat is running.



[3] https://universaldependencies.org/u/dep/index.html

Other relations, including *nsubj:nn*, *obl:tmod*, could be consulted in the dependency annotation guidelines released on the web of VLSP.

Regarding multiword expressions (MWEs), we have defined 16 subtypes for capturing different cases of MWEs in Vietnamese.

Dependency labels are described in detail in the guidelines published along with training data. Each relation is accompanied by a definition, examples and notes on ambiguous cases.

Besides the dependency labels, we also map Vietnamese POS tagset to Universal POS tagset. This work is important for the integration of Vietnamese dependency corpus to Universal dependency project. Guidelines for Vietnamese word segmentation and POS tagging used for VietTreebank published in 2009 have equally been revised, and the corpus published for the dependency shared task is annotated in accordance with these guidelines.

2.2 Annotation process

For an easy annotation of dependency relations, we have designed a tool exceptionally for this task.

The data annotation is performed by two linguists, one computer scientist and approved by one linguistic annotator expert. Finally, annotators cross-checked labeling results and discussed among them for obtaining the most accurate annotation.

Table 1 shows the inter-annotator agreement between each couple of annotators.

Table 1: Agreement between three annotators

Agreement	UAS	LAS
Ano1-Ano2	96.28	92.74
Ano1-Ano3	94.44	89.98
Ano2-Ano3	95.55	92.53
Average	95.42	91.75

2.3 Datasets

In 2019, the datasets are collected from three sources: 4000 sentences in VietTreebank corpus (articles crawled from the "Tuổi trẻ" news website), "Little Prince" corpus (a famous French novella, translated in hundreds of languages around the world), and a set of hotel and restaurant reviews (social network data).

All the training and test datasets from VLSP 2019 are provided as training data for VLSP 2020 shared task, in addition to about 4000 sentences from VietTreebank newly annotated in 2020. In total, the training set contains 8,152 sentences. The test data is composed of two sets: 906 sentences from VietTreebank and 217 sentences randomly collected from *VnExpress*[4].

For VLSP 2019, participants worked only with pre-processed datasets: all the sentences in the training and test set are segmented and POS tagged. In VLSP 2020, participants competed on two tracks: one with raw data and the other with data already segmented in words and POS tagged. At the first step, all the teams received the raw data and had one-day deadline to submit their result. At the second step, participants have been sent the same test set with word segmentation and POS tagging.

Table 2 gives some statistics on the datasets: the number of sentences and the average number of words per sentence.

Table 2: Number of sentences and average number of words per sentence

Data	Number of Sentences	Length <30	Length 30-50	Length >50	Length Average
Training Package1	5069	4882	159	28	14.40
Training Package2	3083	1942	1005	136	24.96
Test Data	1123	852	229	42	23.29

It can be seen that the sentences in the training dataset Package2 and testing data are much longer than sentences in Package1 from the previous year. This is not a small challenge for participants, because the longer the sentence is, the greater the complexity is. To tackle this problem, one needs to have smoother and more efficient pre-processing steps.

3 Parsing Methods

VLSP dependency parsing shared task counted 15 registered teams, but finally only 5 teams could submit results. All these teams (DP1, DP2, DP3, DP4 and DP5) actually deployed parsing models based on graph neural networks (Dozat et al., 2017), combining with different models of word embeddings.

3.1 Team DP1

The team DP1 proposed a joint deep contextualized word representation for dependency parsing. Their joint representation consists of five components: word representations from ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019) language

[4] https://vnexpress.net/

models for Vietnamese (Nguyen and Tuan Nguyen, 2020), Word2Vec embeddings trained on Baomoi dataset (Xuan-Son Vu, 2019), character embeddings (Kim, 2014), and POS tag embeddings. This joint representation is finally deployed in a deep biaffine dependency parser (Dozat et al., 2017).

For raw data input, they used VnCoreNLP (Vu et al., 2018) for segmentation and POS tagging. A POS tag mapping was defined to convert from VnCoreNLP POS tagset into the universal tagset used in VLSP dependency data.

3.2 Team DP2

The team DP2 proposes a combining architecture of two state-of-the-art moels: PhoBERT - the Vietnamese language model (Nguyen and Tuan Nguyen, 2020), and the Biaffine Attention mechanism for universal dependency parsing.

For the encoder, they extract word vectors from two last layers of PhoBERT-base and concatenate them to form 1536-D word representations. The outputs of PhoBERT are passed through a word alignment layer to obtain aggregated word-based representations.

For decoding, they develop equally models for jointly learning POS taging and dependency parsing as proposed in (Nguyen and Verspoor, 2018). However, their experiments show that the best performance on their validation set obtained with the use of PhoBERT-large and biaffine attention mechanism without POS learning. The package is available on github[5].

VnCoreNLP (Vu et al., 2018) was used for preprocessing raw texts.

3.3 Team DP3

The team DP3 chose equally the model of Stanford's graph-based neural dependency parser to build their dependency parsing models. The team focused on testing four different configurations of embeddings: word embeddings or pre-trained word embeddings (Xuan-Son Vu, 2019) combined with character embeddings or with POS tag embeddings.

In case of raw data input, the team DP3 used underthesea[6] for word segmentation. For POS tagging, they have trained a POS tagger using bidirectional LSTM-CRF models for sequence tagging (Huang et al., 2015) and the same pre-trained word embeddings as above.

The results show that for raw text input, the use of character-level embeddings proves a better performance than POS tag embeddings. For CoNLL data input, using pre-trained word embeddings in combination with POS tag embeddings gives the best performance.

3.4 Team DP4

The solutions adopted by Team DP4 for building their parsing systems are as follows.

For dependency parsing, they implemented a BiLSTM-based deep biaffine neural dependency parser. They used Adam optimizer to optimize the network and fastText for word representations (Joulin et al., 2016). Two different models for dependency parsing have been built: the first uses both UPOS and XPOS information for training and predicting data, while the second uses only UPOS information during the entire process. Experiments show that the model using both UPOS and XPOS information generally gives better results.

For the preprocessing of raw data, VnCoreNLP (Vu et al., 2018) was used for sentence splitting and word segmentation. The POS tagging was performed by a BERT-based (Devlin et al., 2019) classifier using bertbase-multilingual-cased pretrained-model available in HuggingFace (Wolf et al., 2019).

3.5 Team DP5

The team DP5 uses Bidirectional Long Short-Term Memory (BiLSTM) (Kiperwasser and Goldberg, 2016) network to extract the contextual information, while the graph neural network captures high-order information. The pre-processing of raw texts, such as word segmentation and POS tagging, is performed by using VnCoreNLP (Vu et al., 2018).

For the word embedding layer, they adopted a pre-trained model for Vietnamese with 300-dimensional word embeddings, i.e. fastText (Joulin et al., 2016). Each word is embedded using three different vectors: randomly initialized word embedding, pre-trained word embedding, and POS embedding.

4 Evaluation

4.1 Data format

The dependency annotated texts are encoded in CoNLL-U format[7].

[5] https://github.com/quangph-1686a/VUDP
[6] https://pypi.org/project/underthesea/
[7] https://universaldependencies.org/format.html

Each sentence consists of one or more word lines, and each word line contains 10 fields as follows.

1. ID: Word index, integer starting at 1 for each new sentence.
2. FORM: Word form or punctuation symbol.
3. LEMMA: Lemma of the word, which is the same as the word form for Vietnamese.
4. UPOS: Universal POS tag. X if not available.
5. XPOS: Vietnamese POS tag; _ if not available.
6. FEATS: Morphological features; _ if not available.
7. HEAD: Head of the current word, which is either a value of ID or zero (0).
8. DEPREL: Universal dependency relation to the HEAD (root if HEAD = 0) or a defined language-specific sub-type of one.
9. DEPS: Enhanced dependency graph in the form of a list of head-deprel pairs.
10. MISC: Any other annotation.

An example is given in Table 3.

Table 3: A sentence in training set

1	Tôi	tôi	PROPN	Pro	_	3	nsubj	_	_
2	đã	đã	ADV	Adv	_	3	advmod	_	_
3	sống	sống	VERB	V	_	0	root	_	_
4	nhiều	nhiều	ADJ	Adj	_	3	advmod:adj	_	_
5	với	với	SCONJ	C	_	7	case	_	_
6	những	những	DET	Det	_	7	det	_	_
7	người lớn	người lớn	N	N	_	3	obl:with	_	_
8	.	.	PUNCT	PUNCT	_	3	punct	_	_

The 9th and 10th columns remain empty (_) in current datasets. For test data, the 7th and 8th columns are empty (_).

4.2 Evaluation metrics

VLSP 2020 participant systems are evaluated and ranked using the standard evaluation metric in dependency parsing which is Labeled Attachment Score (LAS), defined in comparing the gold relations of the test set and relations returned by the system:

$$P = \frac{correctRelations}{systemNodes}$$

$$R = \frac{correctRelations}{goldNodes}$$

$$LAS = \frac{2 * P * R}{(P + R)}$$

As in CoNLL 2018 dependency shared task (Zeman et al., 2018), for scoring purposes, only universal dependency labels will be taken into account, which means that language-specific subtypes such as acl:relcl (relative clause), a subtype of the universal relation acl (clausal modifier of noun), will be truncated to acl both in the gold standard and in the parser output in the evaluation.

In addition, UAS (Unlabeled Attachment Score) metric is also provided, showing the percentage of words that are assigned the correct syntactic head. We use the evaluation script published at CoNLL 2018[8].

4.3 Results

Table 4 shows the results obtained on raw text input of each system. The teams DP1 and DP3 submitted results of multiple models. Results from both the UAS and LAS measurements show the uniformity of the teams' models across all different data sets. Last two columns show the results of each team on the whole test set with the best systems highlighted.

Table 5 shows the results for the segmented and POS tagged text input in CoNLL-U format. More models have been submitted for this track. It can be seen that the results with this format are significantly higher than the raw data input, which is quite understandable, especially as the pre-processing tools are in agreement with older guidelines of word segmentation and POS tagging. An interesting observation is that the team DP2 achieves the first rank for raw text input but only the third rank for this pre-processed input: the model submitted by this team is the only model that doesn't use POS information. A possible interpretation is that erroneous POS labels had a strong negative impact on the results.

A statistic shows that all teams share a high intersection of 55.49% lines with the gold test dataset. An analysis in detail in the future would help us to understand better the characteristics of these common results.

Table 6 gives a closer look of the results regarding the sentence length. For all models, the accuracy decreases as the sentence length increases. This confirms the bigger challenge of the VLSP 2020 dependency parsing shared task in comparison with the task in VLSP 2019. In addition, given

[8] https://universaldependencies.org/conll18/conll18_ud_eval.py)

Table 4: Input: Raw text

Team	Model	VTB		vnexpress1		vnexpress3		vnexpress7		vnexpress8		vnexpress10		vnexpress14		Total	
		UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS
DP1	1	76.33	67.46	74.79	65.38	74.22	66.73	68.33	61.67	74.81	65.71	80.64	72.46	72.61	62.45	**76.12**	**67.32**
	2	75.68	66.59	72.17	62.61	74.95	67.28	66.11	61.11	74.29	65.97	78.45	69.98	73.36	63.69	**75.48**	**66.53**
DP2	1	78.49	68.94	79.72	70.62	78.37	70.08	68.89	65.56	78.31	70.00	81.08	74.80	74.85	68.15	**78.45**	**69.21**
DP3	1	76.44	67.68	73.05	63.78	75.91	68.05	66.67	64.44	62.52	55.86	73.36	67.16	68.16	61.19	**75.63**	**67.12**
	2	74.97	65.50	69.65	59.00	75.00	67.74	61.11	58.33	60.60	51.63	70.85	62.73	70.90	63.93	**74.15**	**64.93**
DP4	1	74.55	65.34	71.70	58.91	77.09	69.29	70.56	65.56	69.61	59.35	76.41	68.22	71.62	63.69	**74.47**	**65.3**
DP5	1	73.18	64.66	68.77	58.75	74.1	65.81	61.67	55.56	68.96	61.43	73.19	64.13	68.4	60.72	**72.85**	**64.35**

Table 5: Input: CoNLLU

	Model	VTB		vnexpress1		vnexpress3		vnexpress7		vnexpress8		vnexpress10		vnexpress14		Total	
		UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS
DP1	1	84.81	76.44	78.98	70.94	85.89	76.97	82.22	75.56	82.49	73.93	85.46	77.53	84.04	75.31	**84.65**	**76.27**
	2	84.58	76.29	77.43	70.17	85.46	77.58	80.00	73.89	81.32	73.80	81.20	72.69	83.54	76.81	**84.23**	**76.05**
DP2	1	83.36	73.29	82.84	73.42	83.81	74.59	81.11	75.00	80.67	72.11	85.76	78.85	81.80	73.32	**83.32**	**73.5**
DP3	1	80.12	70.71	75.73	66.15	80.15	71.72	74.44	70.56	76.01	66.28	82.09	74.74	76.81	69.58	**79.86**	**70.62**
	2	81.89	73.71	67.70	57.96	78.68	70.98	69.44	61.67	74.97	66.54	76.36	69.02	75.81	68.58	**80.81**	**72.66**
	3	80.81	71.71	76.20	67.23	79.47	71.29	74.44	70.00	76.26	68.09	82.09	74.16	79.05	71.07	**80.44**	**71.5**
	4	82.11	73.47	73.88	65.84	80.82	72.02	70.00	64.44	76.39	69.26	81.64	71.95	80.55	73.32	**81.53**	**72.96**
DP4	1	84.41	75.94	74.34	63.37	84.42	76.48	85.56	78.89	82.88	74.84	83.55	75.48	82.79	76.31	**84.08**	**75.64**
	2	83.20	75.14	68.32	55.95	76.60	68.48	71.11	61.11	70.56	61.09	73.13	63.29	75.56	68.08	**81.58**	**73.32**
DP5	1	81.89	73.34	75.12	66.15	84.36	75.38	76.67	67.22	79.25	71.98	80.47	72.54	80.55	73.57	**81.71**	**73.19**

Table 6: Statistics by the length of sentence

CoNLLU	< 30		30-50		> 50	
	UAS	LAS	UAS	LAS	UAS	LAS
DP1	86.11	77.37	82.75	75.12	80.89	72.61
DP2	84.28	74.52	81.90	72.49	81.34	69.87
DP3	82.61	74.01	80.10	71.62	78.88	70.23
DP4	84.83	76.15	83.05	75.08	82.30	74.05
DP5	83.06	74.36	79.81	71.89	78.60	69.47

Table 7: The final rank

No.	UAS	LAS	Aver.	Rank
DP1	80.39	71.80	76.09	2
DP2	80.89	71.36	76.12	1
DP3	78.58	70.04	74.31	4
DP4	79.28	70.47	74.87	3
DP5	77.28	68.77	73.03	5

the best model in 2019 obtained a performance of 73.53% for UAS and 61.28% for LAS, we can hope for improvement of all systems by enlarging the training dataset.

The teams are finally ranked based on the average of the best models for 2 testing data formats, as shown in Table 7.

5 Conclusion

We have presented the VLSP 2020 shared task on Dependency Parsing for Vietnamese. Although the number of registered participants for receiving the training datasets is 15, only 5 teams could submit the results. The other teams may not have enough time for achieving a satisfactory result, as many teams registered for several shared tasks at VLSP 2020. This shared task provides useful resources for building Vietnamese dependency parser and other applications that use dependency parsing results. We will continue to improve the quantity and quality of annotated sentences in order to get better performance in dependency parsing systems.

Acknowledgement

This shared task was supported by VINIF and VNG Zalo, as well as the NLP group at VNU University of Science.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of

deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Timothy Dozat, Peng Qi, and Christopher D. Manning. 2017. Stanford's graph-based neural dependency parser at the CoNLL 2017 shared task. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 20–30, Vancouver, Canada. Association for Computational Linguistics.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional LSTM feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327.

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, Online. Association for Computational Linguistics.

Dat Quoc Nguyen and Karin Verspoor. 2018. An improved neural network model for joint POS tagging and dependency parsing. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 81–91, Brussels, Belgium. Association for Computational Linguistics.

Thi Luong Nguyen, My Linh Ha, Nguyên Thi Minh Huyên, and Phuong Le-Hong. 2018. Using BiLSTM in Dependency Parsing for Vietnamese. *Computación y Sistemas*, 22(3).

M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Thanh Vu, Dat Quoc Nguyen, Dai Quoc Nguyen, Mark Dras, and Mark Johnson. 2018. VnCoreNLP: A Vietnamese natural language processing toolkit. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 56–60, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Son N. Tran Lili Jiang Xuan-Son Vu, Thanh Vu. 2019. Etnlp: A visual-aided systematic approach to select pre-trained embeddings for a downstream task. *In: Proceedings of the International Conference Recent Advances in Natural Language Processing (RANLP)*.

Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018. CoNLL 2018 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–21, Brussels, Belgium. Association for Computational Linguistics.

ReINTEL: A Multimodal Data Challenge for Responsible Information Identification on Social Network Sites

Duc-Trong Le[1], **Xuan-Son Vu**[2], **Nhu-Dung To**[3], **Huu-Quang Nguyen**[4], **Thuy-Trinh Nguyen**[4], **Linh Le**[4], **Anh-Tuan Nguyen**[4], **Minh-Duc Hoang**[4], **Nghia Le**[4] **Huyen Nguyen**[5], **Hoang D. Nguyen**[6]

[1]University of Engineering and Technology, Vietnam National University, Vietnam.
trongld@vnu.edu.vn
[2]Dept. of Computing Science, Umeå University, Sweden.
sonvx@cs.umu.se
[3]School of Computer Science, University of Sydney, Australia.
duto3894@uni.sydney.edu.au
[4]ReML.AI - Reliable Machine Learning Lab, International.
{quang, trinh, linh, tuan, duc, nghia}@reml.ai
[5]Hanoi University of Science, Vietnam National University, Vietnam.
huyenntm@hus.edu.vn
[6]School of Computing Science, University of Glasgow, Singapore.
harry.nguyen@glasgow.ac.uk

Abstract

This paper reports on the ReINTEL Shared Task for Responsible Information Identification on social network sites, which is hosted at the seventh annual workshop on Vietnamese Language and Speech Processing (VLSP 2020). Given a piece of news with respective textual, visual content and metadata, participants are required to classify whether the news is *'reliable'* or *'unreliable'*. In order to generate a fair benchmark, we introduce a novel human-annotated dataset of over 10,000 news collected from a social network in Vietnam. All models will be evaluated in terms of AUC-ROC score, a typical evaluation metric for classification. The competition was run on the Codalab platform. Within two months, the challenge has attracted over 60 participants and recorded nearly 1,000 submission entries.

1 Introduction

This challenge aims at identifying the reliability of information shared on social network sites (SNSs). With the blazing-fast spurt of SNSs (e.g. Facebook, Zalo and Lotus), there are approximately 65 million Vietnamese users on board with the annual growth of 2.7 million in the recent year, as reported by the Digital 2020 [1]. SNSs have become widely accessible for users to not only connect friends but also freely create and share diverse content (Shu et al., 2017; Zhou et al., 2019). A number of users, however, has exploited these social platforms to distribute fake news and unreliable information to fulfill their personal or political purposes (e.g. US election 2016 (Allcott and Gentzkow, 2017)). It is not easy for other ordinary users to realize the unreliability, hence, they keep spreading the fake content to their friends. The problem becomes more seriously once the unreliable post becomes popular and gains belief among the community. Therefore, it raises an urgent need for detecting whether a piece of news on SNSs is reliable or not. This task has gained significant attention recently (Ruchansky et al., 2017; Shu et al., 2019a,b; Yang et al., 2019).

The shared task focuses on the responsible (i.e. reliable) information identification on Vietnamese SNSs, referred to as ReINTEL. It is a part of the 7th annual workshop on Vietnamese Language and Speech Processing, VLSP 2020[2] for short. As a binary classification task, participants are required to propose models to determine the reliability of SNS posts based on their content, image and metadata information (e.g. number of likes, shares, and comments). The shared task consists of three phases namely *Warm up, Public Test, Private Test*, which is hosted on Codalab from October 21st, 2020 to November 30th, 2020. In summary, there are around 1000 submissions created by 8 teams and over 60 participants during the challenge period.

[1] https://wearesocial.com/digital-2020

[2] https://vlsp.org.vn/vlsp2020

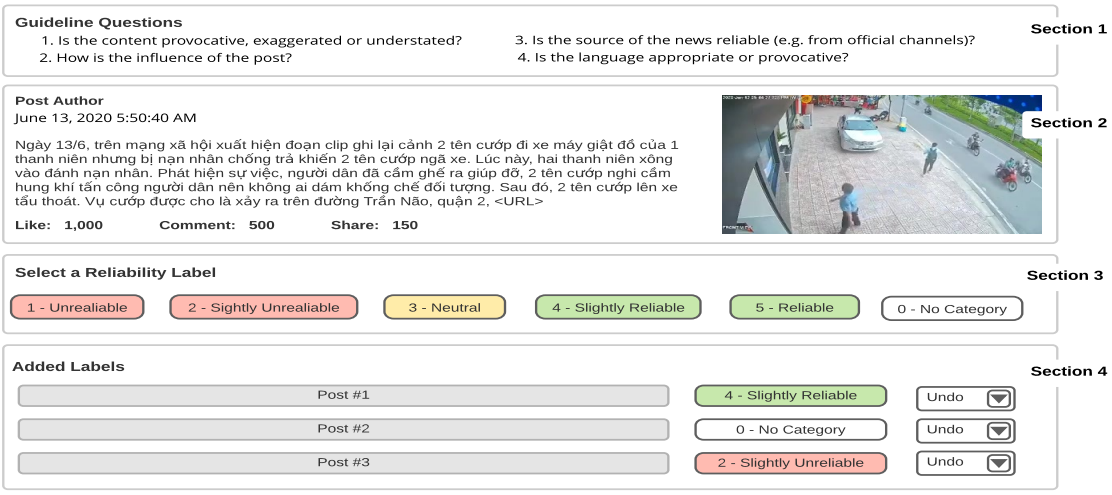


Figure 1: Data Annotation Tool

As our first contribution, this shared task provides an evaluation framework for the reliable information detection task, where participants could leverage and compare their innovative models on the same dataset. Their knowledge contribution may help improve safety on online social platforms. Another valuable contribution is the introduction of a novel dataset for the reliable information detection task. The dataset is built based on a fair human annotation of over 10,000 news from SNSs in Vietnam. We hope this dataset will be a useful benchmark for further research. In this shared task, AUC-ROC is utilized as the primary evaluation metric.

The remainder of the paper is organized as follows. The next section describes the data collection and annotation methodologies. Subsequently, the shared task description and evaluation are summarized in Section 3. In Section 4, we discusses the potentials of language and vision transfer learning for the detection task. Section 5 describes the competition, approaches and respective results. Finally, Section 6 concludes the paper by suggesting potential applications for future studies and challenges.

2 The ReINTEL 2020 Dataset

2.1 Data Collection

We collect the data for two months from August to October 2020. There are two main sources of the data: SNSs and Vietnamese newspapers. As for the former source, public social media posts are retrieved from news groups and key opinion leaders (KOLs). Many fake news, however, has been flagged and removed from the social networking sites since the enforcement of Vietnamese cybersecurity law in 2019 (Son, 2018). Therefore, to include the deleted fake news, we gather newspaper articles reporting these posts and recreate their content.

All the collected data were originally posted in the period of March - June 2020. During this time, Vietnam was facing a second wave of Covid-19 with a drastic increase from 20 to 355 cases (WHO, 2020). The spread of Covid-19 results in an 'infodemic' in which misleading information is disseminated rapidly especially on social media (Hou et al., 2020; Huynh et al., 2020). Hence, this period is a potential source of fake news. Besides Covid-19, the items in our dataset cover a wide range of domains including entertainment, sport, finance and healthcare. The result of the data collection stage is 10,007 items that are prepared for the annotation process.

2.2 Data Annotation

2.2.1 Annotator and Training

We recruit 23 human annotators to participate in the annotation process. The annotators receive one week training to identify fact-related posts and how to evaluate the reliability of the post based on primary features including the news source, its image and content.

2.2.2 Annotation Tool

Figure 1 demonstrates the annotation tool interface, which is designed to support quick and easy annotation. The first section contains guideline questions

to remind the annotators of the labeling criterion including the news source credibility, the language appropriateness and fact accuracy. The second section is the post content, image and influence (i.e. number of likes, comments and shares). In Section 3, the annotators select a Reliability score for the post. There is a 5-point reliability Likert scale for fact-based posts with the following labels: 1 - Unreliable, 2 - Slightly unreliable, 3 - Neutral, 4 - Slightly reliable, 5 - Reliable. On the other hand, if the post is opinion-based and does not contain facts, the annotators should select label '0 - No category' instead.

The last section is a list of labeled items for the annotators to review and update their decision, if necessary, using the 'Undo' button.

2.2.3 Annotation Process

The annotation process is conducted from 9th to 19th October 2020. The annotators are divided into three groups to annotate 10,007 items independently. Therefore, each item will be annotated three times by different annotators.

Once the annotators finish 30,021 annotations (i.e. 10,007 items annotated three times), we filter and summarise the result based on majority vote basis. Firstly, we combine labels of the same essence: Category 1 and 2 (Unreliable and Sightly unreliable) and Category 4 and 5 (Slightly reliable and Reliable). After merging the categories, we select the majority votes to be the final labels. If the majority vote is 1 or 2, the final label should be 1 - Unreliable. If the majority vote is 4 or 5, the final label should be 0 - Reliable. When the majority vote is 3 - Neutral, we finalise using ground truth labels. Lastly, if the majority agrees that the post is not fact-based (i.e. 0 - No Category), we remove it from the set.

For items with no majority votes (i.e. three annotators have different opinions), we follow an alternate procedure. If the ground truth label is 1 - unreliable, the final label should be 1. On the other hand, if the ground truth label is 0 - reliable, we double check to separate reliable news from opinion-based items. The process is illustrated in Figure 2.

2.2.4 Content Filtering

Once the annotation process is finished, data needs to go through the last step before being published for the competition – the content filtering. In this step, we manually check to ensure that data, including both text and image, published for the competition:

1. Does not violate any law, statue, ordinance, or regulation
2. Will not give rise to any claims of invasion of privacy or publicity
3. Does not contain, depict, include or involve any of the following:
    - Political or religious views or other such ideologies
    - Explicit or graphic sexual activity
    - Vulgar or offensive language and/or symbols or content
    - Personal information of individuals such as names, telephone numbers, and addresses
    - Other forms of ethical violations

3 The ReINTEL 2020 Challenge

3.1 Dataset Splitting

Data splitting for data challenge is a difficult process in order to avoid evidence ambiguity and concept drifting which are the main cause of unstable ranking issue in data challenges.

In this competition, we apply RDS (Nguyen et al., 2020) to split ReINTEL data into three sets including public train, validation, and private test sets. It is worth to mention that, RDS is a method to approximate optimum sampling for model diversification with ensemble rewarding to attain maximal machine learning potentials. It has a novel stochastic choice rewarding is developed as a viable mechanism for injecting model diversity in reinforcement learning.

3.1.1 Baselines

To apply RDS (Nguyen et al., 2020) for the data splitting process, it requires to have baseline learners to obtain rewards for the reinforced process. It is recommended to choose representative baseline learners, to let the reinforced learner better capture different learning behaviors. The use of these baseline learners is important since each learner will behave differently depending on the patterns contained in the target data. As a result, RDS helps to increase the diversity of the data samples in different sets. Here we employ three models to classify reliable news using textual features as follows:

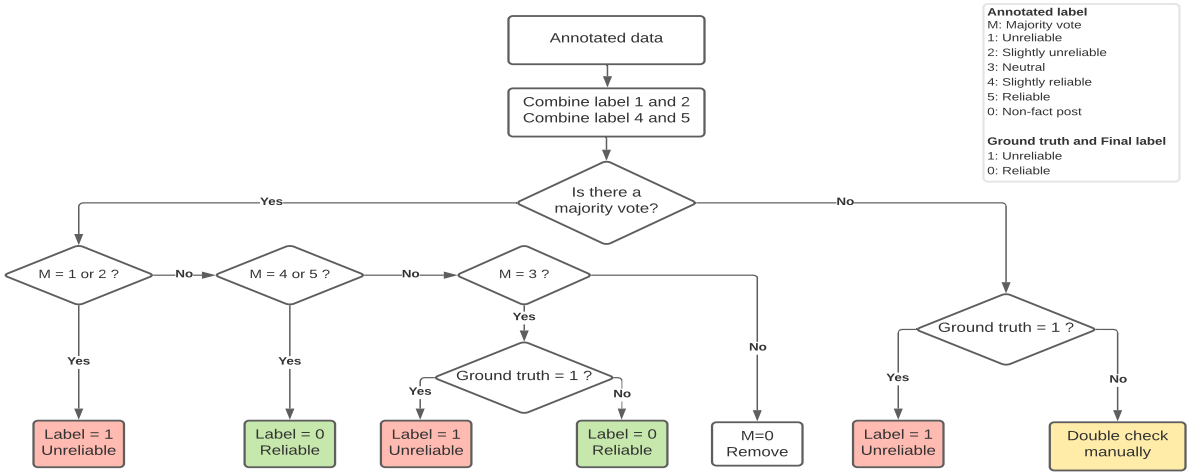


Figure 2: Data Annotation Process

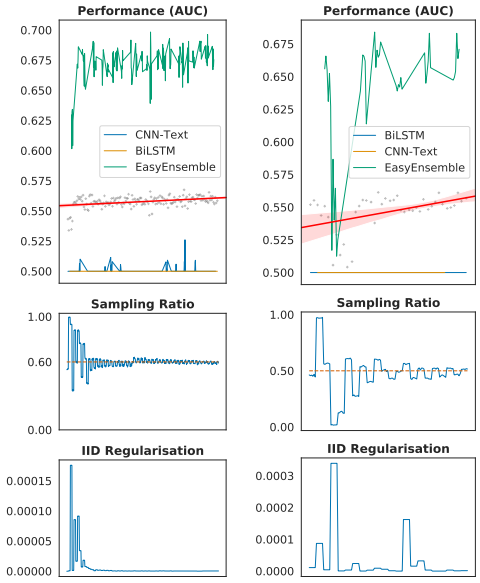
- **Bi-LSTM** (Schuster and Paliwal, 1997) is a bi-directional LSTM model. It has two LSTMs in which, one LSTM takes input sequence in a forward direction, and another LSTM takes input sequence in a backward direction. The use of Bi-LSTM architecture helps to increase the amount of information available to the network, to gain better performance in most of sequence related tasks. Bi-LSTM network is a standard baseline for most of text classification tasks.

- **CNN-Text** (Kim, 2014) is the use of CNN (LeCun et al., 1989) network on word embeddings to perform the classification tasks. The simple architecture outperformed all other models at the publication time.

- **EasyEnsemble** (Liu et al., 2009) is used to represent a tradition approach in dealing with im-balanced dataset. For the vectorization, we trained a Sent2Vec (Pagliardini et al., 2018) using the combined 1GB texts of Vietnamese Wikipedia data (Vu et al., 2019) and 19 GB texts of Vuong (2018).

3.1.2 Learning Dynamics

To disentangle dataset shift and evidence ambiguity of the data splitting strategy, we apply RDS stochastic choice reward mechanism (Nguyen et al., 2020) to create public training, public- and private testing sets. Figure 3 illustrates the learning dynamic towards the goal.



(a) Training Set (b) Val & Test Sets

Figure 3: Learning Dynamics for splitting data into 3 sets (public training, public testing, and private testing) using RDS Stochastic Choice Reward Mechanism (Nguyen et al., 2020).

4 Transfer Learning

Knowledge transfer has been found to be essential when it comes to downstream tasks with new datasets. If this transfer process is done correctly, it would greatly improve the performance of learning. Since ReINTEL challenge is a multimodal challenge, both visual based knowledge transfer and language based knowledge transfer are used by different teams.

To be fair between participants, we required all teams to register for the use of pre-trained models.

Model	Language	Vision	Description
Word2VecVN (Vu, 2016)	x		Trained on 7GB texts of Vietnamese news
FastText (Vietnamese version) (Joulin et al., 2016)	x		Trained on Vietnamese texts of the CommonCrawl corpus
ETNLP (Vu et al., 2019)	x		Trained on 1GB texts of Vietnamese Wikipedia
PhoBERT (Nguyen and Nguyen, 2020)	x		Trained on 20GB texts of both Vietnamese news and Vietnamese Wikipedia
Bert4News (Nha, 2020)	x		Trained on more than 20GB texts of Vietnamese news
vElectra and ViBERT (The et al., 2020)	x		vElectra was trained on 10GB texts, whereas ViBERT was trained on 60GB texts of Vietnamese news
VGG16 (Simonyan and Zisserman, 2015)		x	Trained on ImageNet (Deng et al., 2009)
YOLO (Redmon et al., 2015)		x	Trained on ImageNet (Deng et al., 2009)
EfficientNet B7 (Tan and Le, 2019)		x	Trained on ImageNet (Deng et al., 2009)

Table 1: List of pre-trained models registered by all participants of ReINTEL challenge in 2020.

Table 1 lists all pre-trained language and vision models registered by all participants.

4.1 Language Transfer Learning

For natural language processing tasks in Vietnamese, there have been many pre-trained language models are available. In 2016, Vu (2016) introduced the first monolingual pre-trained models for Vietnamese based on Word2Vec (Mikolov et al., 2013). The use of pre-trained Word2VecVN models was proved to be useful in various tasks, such as the name entity recognition task (Vu et al., 2018). In 2019, Vu et al. (2019) introduced the use of multiple pre-trained language models to achieve new state-of-the-art results in the name entity recognition task (Nguyen et al., 2019). Up to date, there have been many other new monolingual language models for Vietnamese are available such as PhoBERT (Nguyen and Nguyen, 2020), vElectra and ViBERT (The et al., 2020).

4.2 Vision Transfer Learning

Different from language models, visual models are normally universal and existing pre-trained models can be directly applied in most of image processing tasks. For the use of visual features, there is only one team using multimodal features among top 6 teams of the leader board. This team, in fact, achieved the 1^{st} rank on the public test (see Table 3); but they did not get the same rank on the private test. This hints that the reliability of news mainly depends on content of news and other meta information, such as number of likes on social networks. Moreover, it is yet to be explored to capture the reliability of news using both vision and language information.

4.3 Language and Vision Transfer Learning

The use of both language and vision transfer learning is important for multimodal tasks. This line of research has attracted much attention with various new language-vision models, such as VilBERT (Lu et al., 2019), 12-in-1 (Lu et al., 2020). No participants employ into this approach in the ReINTEL challenge due to the lack of language and vision pre-trained models in Vietnamese. Moreover, it is required to have extensive computer resources for applying this approach in a data challenge. In the future, we expect to see more research done in this direction because both images and texts are essential to SNS issues.

No	Attribute	Description
1	id	Unique ID of each post
2	user_name	Anomynized post owner's identity
3	post_message	Text content of the post
4	timestamp_post	The time when the post is uploaded
5	num_like_post	Number of likes that the post received
6	num_comment_post	Number of comments that the post received
7	num_share_post	Number of shares that the post received
8	image	The image uploaded with the post
9	label	Manually annotated label indicating the reliability of the post 1: Unreliable 0: Reliable

Table 2: Data attributes

5 Results

5.1 Data Format

Each instance includes 8 main attributes with/without a binary target label. Table 2 summarizes the key features of each attribute.

5.2 Training/Testing Data

The challenge provides approximately 8,000 training examples with the respective target labels. The testing set consists of 2,000 examples without labels.

5.3 Result Submission

Participants must submit the result in the same order as the testing set in the following format:

```
id1, label probability 1
Id2, label probability 2
...
```

5.4 Evaluation Metric

The challenge task is evaluated based on Area Under the Receiver Operating Characteristic Curve (AUC-ROC), which is a typical metric for classification tasks. Let us denote X as a *continuous random variable* that measures the 'classification' score of a given a news. As a binary classification task, this news could be classified as *"unreliable"* if X is greater than a threshold parameter T, and *"reliable"* otherwise. We denote $f_1(x), f_0(x)$ as probability density functions that the news belongs to *"unreliable"* and *"reliable"* respectively, hence the true positive rate $TPR(T)$ and the false positive rate $FPR(T)$ are computed as follows:

$$TPR(T) = \int_T^{\infty} f_1(x)dx \quad (1)$$

$$FPR(T) = \int_T^{\infty} f_0(x)dx \quad (2)$$

and the AUC-ROC score is computed as:

$$AUC\_ROC = \int_{-\infty}^{\infty} TPR(T)FPR'(T)dT \quad (3)$$

Here, submissions are evaluated with ground-truth labels using the *scikit-learn*'s implementation [3].

5.5 Participation

During the course two months of the competition, 61 participants sign up for the challenge. 30% of the participants compete in groups of 2 (6 teams) and 4 members (2 teams). 19 participants sign our corpus usages agreement.

From top 8 of the Private test leaderboard, 6 teams/participants submit their technical reports that demonstrate their strategies and findings from the challenge. The summary of the competition participation can be seen in Table 4.

5.6 Outcomes

In total, 657 successful entries were recorded. The highest results of the Public test and Private test phase were 0.9427 and 0.9521 respectively. Key descriptive statistics of the results in each phase is illustrated in Table 5.

[3] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_auc_score.html

Table 3: Top 6 teams on public-test and private-test with submitted papers and their final approaches. The rank is based on the ROC-AUC scores on the private-test.

#	Team	ROC-AUC		Final Approach	Ensemble?	Multimodal?
		Public-test	Private-test			
1	Kurtosis	0.9399	**0.9521**	TF-IDF + SVD; Emb + SVD; NB, LightGBM, CatBoost	Yes	No
2	NLP_BK	0.9360	0.9513	Bert4News + phoBERT + XLM + MetaFeatures	Yes	No
3	SunBear	0.9418	0.9462	RoBerta + MLP	Yes	No
4	uit_kt	-	0.9452	phoBERT + Bert4News	Yes	No
5	Toyo-Aime	**0.9427**	0.9449	CNN + Bert + Fully connected	Yes	Yes
6	ZaloTeam	-	0.9378	viBERT + viELECTRA + phoBERT	Yes	No

Metric	Value
Number of participants	61
Number of teams	8
Number of signed agreements	19
Number of submitted papers	6

Table 4: Participation summary

	Public Test	**Private Test**	**Overall**
Total Entries	571	86	657
Highest ROC	0.9427	0.9521	0.9474
Mean ROC	0.8463	0.8942	0.8703
Std. ROC	0.1215	0.1022	0.1119

Table 5: Results summary

6 Conclusion

The rise of misleading information on social media platforms has triggered the need for fact-checking and fake news detection. Therefore, the reliability of news has become a critical question in the modern age. In this paper, we introduce a novel dataset of nearly 10,000 SNSs entries with reliability labels. The dataset covers a great variety of topics ranging from healthcare to entertainment and economics. The annotation and validation process are presented in details with several filtering rounds. With both linguistic and visual features, we believe that the corpus is suitable for future research on fake news detection and news distributor behaviours using NLP and computer vision techniques. In Vietnam, where datasets on SNSs are scarce, our corpus will serve as a reliable material for other research.

Acknowledgment

The authors would like to thank the InfoRE company for the data contribution, the ReML-AI research group[4] for the data contribution and financial support, and the twenty three annotators for their hard work to support the shared task. Without their support, the task would not have been possible.

References

Hunt Allcott and Matthew Gentzkow. 2017. Social media and fake news in the 2016 election. *Journal of economic perspectives*, 31(2):211–36.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*.

Zhiyuan Hou, Fanxing Du, Hao Jiang, Xinyu Zhou, and Leesa Lin. 2020. Assessment of public attention, risk perception, emotional and behavioural responses to the covid-19 outbreak: social media surveillance in china. *Risk Perception, Emotional and Behavioural Responses to the COVID-19 Outbreak: Social Media Surveillance in China (3/6/2020)*.

Toan Luu Huynh et al. 2020. The covid-19 risk perception: A survey on socioeconomics and media attention. *Econ. Bull*, 40(1):758–764.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

[4] https://reml.ai

Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. 1989. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551.

X. Liu, J. Wu, and Z. Zhou. 2009. Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(2):539–550.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *CoRR*, abs/1908.02265.

Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 2020. 12-in-1: Multi-task vision and language representation learning. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

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.

Hoang D. Nguyen, Xuan-Son Vu, Quoc-Tuan Truong, and Duc-Trong Le. 2020. Reinforced data sampling for model diversification.

Huyen Nguyen, Quyen Ngo, Luong Vu, Vu Tran, and Hien Nguyen. 2019. Vlsp shared task: Named entity recognition. *Journal of Computer Science and Cybernetics*, 34(4):283–294.

Nguyen Van Nha. 2020. Pre-trained bert4news. https://github.com/bino282/bert4news.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2018. Unsupervised Learning of Sentence Embeddings using Compositional n-Gram Features. In *NAACL 2018 - Conference of the North American Chapter of the Association for Computational Linguistics*.

Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. 2015. You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640.

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*, CIKM '17, page 797–806, New York, NY, USA. Association for Computing Machinery.

M. Schuster and K. K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Kai Shu, Limeng Cui, Suhang Wang, Dongwon Lee, and Huan Liu. 2019a. defend: Explainable fake news detection. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 395–405.

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 Shu, Suhang Wang, and Huan Liu. 2019b. Beyond news contents: The role of social context for fake news detection. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 312–320.

Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition.

Tuan Son. 2018. Vietnam passes cyber security law.

Mingxing Tan and Quoc V. Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. *CoRR*, abs/1905.11946.

Viet Bui The, Oanh Tran Thi, and Phuong Le-Hong. 2020. Improving sequence tagging for vietnamese text using transformer-based neural models.

Thanh Vu, Dat Quoc Nguyen, Dai Quoc Nguyen, Mark Dras, and Mark Johnson. 2018. Vncorenlp: A vietnamese natural language processing toolkit. In *Proceedings of the 2018 NAACL: Demonstrations*, pages 56–60, New Orleans, Louisiana. Association for Computational Linguistics.

Xuan-Son Vu. 2016. Pre-trained word2vec models for vietnamese. https://github.com/sonvx/word2vecVN.

Xuan-Son Vu, Thanh Vu, Son N. Tran, and Lili Jiang. 2019. Etnlp: A visual-aided systematic approach to select pre-trained embeddings for a downstream task. *In: Proceedings of the International Conference Recent Advances in Natural Language Processing (RANLP)*.

Quoc Binh Vuong. 2018. Vietnamese news corpus. https://github.com/binhvq/news-corpus.

WHO. 2020. Who coronavirus disease (covid-19) dashboard.

Shuo Yang, Kai Shu, Suhang Wang, Renjie Gu, Fan Wu, and Huan Liu. 2019. Unsupervised fake news detection on social media: A generative approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5644–5651.

Xinyi Zhou, Reza Zafarani, Kai Shu, and Huan Liu. 2019. Fake news: Fundamental theories, detection strategies and challenges. In *Proceedings of the twelfth ACM international conference on web search and data mining*, pages 836–837.

Overview of VLSP RelEx shared task: A Data Challenge for Semantic Relation Extraction from Vietnamese News

Mai-Vu Tran[1], **Hoang-Quynh Le**[1], **Duy-Cat Can**[1]
Huyen Nguyen[2], **Linh Nguyen Tran Ngoc**[3] **and Tam Doan Thanh**[4]
[1]VNU University of Engineering and Technology, Hanoi, Vietnam.
{vutm, lhquynh, catcd}@vnu.edu.vn
[2]Hanoi University of Science, Vietnam National University, Vietnam.
huyenntm@hus.edu.vn
[3]Viettel Big Data Analytics Center, Viettel Telecommunication Company, Viettel Group.
linhntn3@viettel.com.vn
[4]doanthanhtam283@gmail.com

Abstract

This paper reports the overview of RelEx shared task for semantic relation extraction from Vietnamese News, which is hosted at the seventh annual workshop on Vietnamese Language and Speech Processing (VLSP 2020). This task focuses on classifying entity pairs in Vietnamese News text into four different, non-overlapping categories of semantic relations defined in advance. In order to generate a fair benchmark, we build a human-annotated dataset of 1,056 documents and 5,900 instances of semantic relations, collected from Vietnamese News in several domains. All models will be evaluated in terms of macro- and micro-averaged F1 scores, two typical evaluation metrics for semantic relation extraction problem.

1 Introduction

The rapid growth of volume and variety of news brings an unprecedented opportunity to explore electronic text but an enormous challenge when facing a massive amount of unstructured and semi-structured data. Recent research progress in text mining needs to be supported by Information Extraction (IE) and Natural Language Processing (NLP) techniques. One of the most fundamental sub-tasks of IE is Relation Extraction (RE). It is the task of identifying and determining the semantic relations between pairs of named entity mentions (or nominals) in the text (Aggarwal, 2015). Receiving the (set of) document(s) as an input, the relation extraction system aims to extract all predefined relationships mentioned in this document by identifying the corresponding entities and determining the type of relationship between each pair of entities (see examples in Figure 1).

Evidence: 23353704	Tại buổi họp báo, ông **Nguyễn Quang Huyền**, Phó Cục trưởng **[Cục Quản lý và Giám sát Bảo hiểm]** [**Bộ Tài chính**] cho biết,	
Relation type: AFFILIATION	*Entity 1:* PER Nguyễn Quang Huyền	*Entity 2:* ORG Cục Quản lý và Giám sát Bảo hiểm
Relation type: AFFILIATION	*Entity 1:* ORG Nguyễn Quang Huyền	*Entity 2:* ORG Bộ Tài chính
Relation type: PART-WHOLE	*Entity 1:* ORG Cục Quản lý và Giám sát Bảo hiểm	*Entity 2:* ORG Bộ tài chính

Figure 1: Relation examples.

RE is of significant importance to many fields and applications, ranging from ontology building (Thukral et al., 2018), improving the access to scientific literature (Gábor et al., 2018), question answering (Lukovnikov et al., 2017; Das et al., 2017) to major life events extraction (Li et al., 2014; Cavalin et al., 2016) and many other applications. However, manually curating relations is plagued by its high cost and the rapid growth of the electronic text.

For English, several challenge evaluations have been organized such as Semantic Evaluation (SemEval) (Gábor et al., 2018; Hendrickx et al., 2010), BioNLP shared task (Deléger et al., 2016), and Automatic Content Extraction (ACE) (Walker et al., 2006). These challenges evaluations attracted many scientists worldwide to attend and publish their latest research on semantic relation extraction. Many approaches are proposed for RE in English texts, ranging from knowledge-based methods to machine learning-based methods (Bach and Badaskar, 2007; Dongmei et al., 2020). Studies on this problem for Vietnamese text are still in the early stages with a few initial achievements. In recent years, there has been a growing interest to develop computational ap-

proaches for extracting semantic relations in Vietnamese text automatically with proposals of several methods. Despite these attempts, the lack of a comprehensive benchmarking dataset has limited the comparison of different techniques. RelEx challenge task in VLSP was set up to provide an opportunity for researchers to propose, assess and advance their researches.

The remainder of the paper is organized as follows. Section 2 gives the description about RelEx shared task. The next section describes the data collection and annotation methodologies. Subsequently, section 4 describes the competition, approaches and respective results. Finally, Section 5 concludes the paper.

2 RelEx 2020 Challenge

As the first shared task of relation extraction for Vietnamese text, we go from typical relations between three fundamental entities in News domain: $Location$, $Organization$ and $Person$. All semantic relations between nominals other than the aforementioned entities were excluded. Based on these three types of annotated entities, we selected four relation types with coverage sufficiently broad to be of general and practical interest. Our selection is referenced and modified based on the relation types and subtypes used in the ACE 2005 task (Walker et al., 2006). We aimed at avoiding semantic overlap as much as possible. Four relation types are described in Table 1 and as follow.

- The $LOCATED$ relation captures the physical or geographical location of an entity.
- The $PART - WHOLE$ relation type captures the relationship when the parts contribute to the structure of the wholes.
- The $PERSONAL-SOCIAL$ relations describe the relationship between people.
- The $ORGANIZATION - AFFILIATION$ relation type represents the organizational relationship of entities.
- We do not annotate non-relation entity pairs ($NONE$). These negatives instances need to be self-generated by participated teams, if necessary.

In the case of $PERSONAL - SOCIAL$, an undirected relation type, two entities are symmetric (i.e., not ordered). Other relation types are directed, i.e., their entities are asymmetry (i.e., order sensitive). We restrict the direction of these relation types always come from entity 1 to entity 2. The participated system needs to define which entity mention plays the role of entity 1 and which entity mention plays the role of entity 2.

This task only focused on intra-sentence relation extraction, i.e., we limit relations to only those that are expressed within a single sentence. The relations between entity mentions are annotated if and only if the relationship is explicitly referenced in the sentence that contains the two mentions. Even if there is a relationship between two entities in the real world (or elsewhere in the document), there must be evidence for that relationship in the local context where it is tagged. We do not accept the case of bridging relations (i.e., a relationship derived from two other consecutive relationships), uncertain relations, inferred relations, and relation in the future tenser (i.e., allusion/mean to happen in the future).

A relation is defined by two entities participating in this relationship. In other words, a sentence can contain several different relations if it has more than one pairs of entities. Any qualifying relations must be predicted, even if the text mentions them is overlap or nested with range text of other relations. We do not allow the multi-label cases, i.e., a pair of entities must have only one relationship or no relation. If there is an ambiguity between some relation types, the participated system needs to decide to choose the most suitable label.

Only binary relations are accepted. N-nary relations should be predicted if and only if they can be split into several binary relations without changing the semantic meaning of the relationships.

3 Task Data

3.1 Data Statistics

For the task, we prepared a total of $1,056$ News documents: 506 documents for the training, 250 documents for development and 300 documents in the test set. Of all $1,056$ news documents, 815 documents were selected in a single crawler process. The remaining 241 documents were selected in another crawler process to represent difference features and were incorporated into the test set. We

No	Relation	Agruments	Directionality
1	LOCATED	PER – LOC, ORG – LOC	Directed
2	PART – WHOLE	LOC – LOC, ORG – ORG, ORG – LOC	Directed
3	PERSONAL – SOCIAL	PER – PER	Undirected
4	ORGANIZATION –AFFILIATION	PER – ORG, PER – LOC, ORG – ORG, LOC – ORG	Directed

Table 1: Relation types permitted arguments and directionality.

	Training set	Development set	Test set
Number of documents	506	250	300
LOCATED	612	346	294
PART-WHOLE	1176	514	815
PERSONAL - SOCIAL	102	98	449
ORGANIZATION -AFFILIATION	771	518	205

Table 2: Statistics of the RelEx dataset.

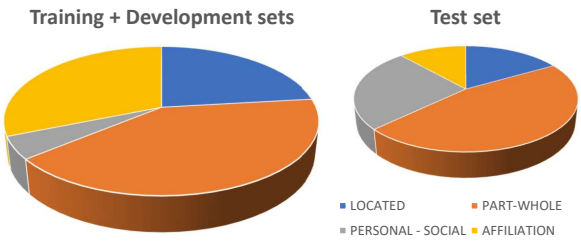


Figure 2: The distribution of relation types in Datasets.

then prepared the manual annotations, Table 2 describes statistics of the RelEx dataset in detailed. Figure 2 show the distribution of relation types in training/development set and the test set. Due to the effect of adding 'strange' data to the test set, the rate is partly inconsistent between training/development and test set.

3.2 Data Annotation

3.2.1 Annotators and Annotation Tool

There are 6 human annotators to participate in the annotation process. An annotation guideline with full definition and illustrative examples was provided. We used a week to train annotators about the markable and non-markable cases in documents. In the following week, annotators conducted trial annotations, then raised some issues that need clarification. An expert then preliminarily assessed the quality of the trial annotation process before started the full annotation process.

We used WebAnno[1] as the Annotation tool. It is a general purpose web-based annotation tool for a wide range of linguistic annotations including various layers of morphological, syntactical, and semantic annotations.

3.2.2 Annotation Process

The annotators were divided into two groups and used their account to conduct independent annotations, i.e., each document was annotated at least twice. The annotation process is described in Figure 3. First, the supervisor separated the whole dataset into several small parts. Each part was given to two independent annotators for annotating. For finding out the agreement between annotators, the committee then calculated the Inter-Annotator Agreement (IAA). Follow (Dalianis, 2018), IAA can be carried out by calculating the Precision, Recall, F-score, and Cohen's kappa, between two annotators. If the IAA is very low, for example, $F1$ is under 0.6, it may be due to the complexity and difficulty of the annotation task or the low quality of the annotation. For the RelEx task, the committee selected the IAA based on $F1$, and chose an acceptable threshold of 0.7. If the IAA between two annotators on a subset was smaller than 0.7, we went through the curation process with a third annotator to decide the final annotation.

4 Challenge Results

4.1 Data Format and Submission

The test set are formatted similarly with the training and development data, but without information for the relation label. The task is to predict, given a sentence and two tagged entities, which of the relation labels to apply. The participated teams must submit the result in the same format with the training and development data.

The participating systems had the following task: Given a documents and tagged entities, predict the semantic relations between those entities and the directions of the relations. Each teams can submit up to 3 runs for the evaluation.

4.2 Evaluation Metrics

The participated results were evaluated using standard metrics of Precision (P), Recall (R) and $F1$. In which, Precision indicates the percentage of

[1] http://webanno.github.io/webanno/

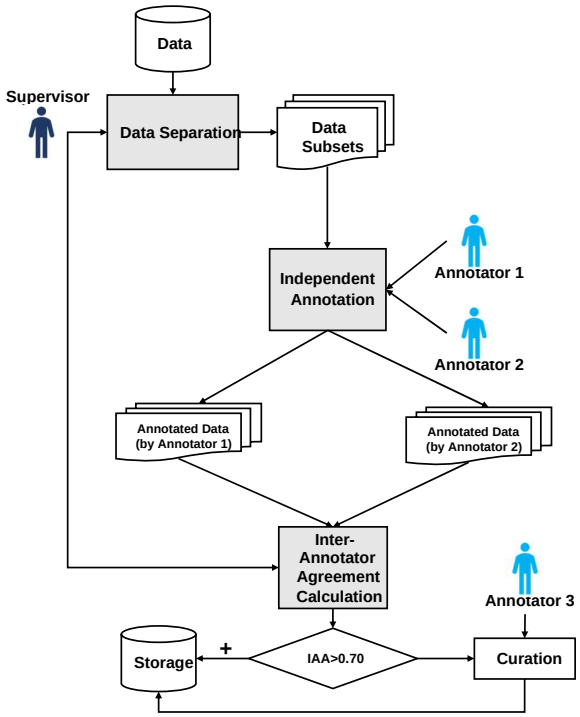


Figure 3: The annotation process.

system positives that are true instances, Recall indicates the percentage of true instances that the system has retrieved. $F1$ is the harmonic mean of Recall and Precision, calculated as follows:

$$F1 = \frac{2 \times P \times R}{P + R} \quad (1)$$

We released a detailed scorer which outputs:

- A confusion matrix,
- Results for the individual relations with P, R and $F1$,
- The micro-averaged P, R and $F1$,
- The macro-averaged P, R and $F1$.

Our official scoring metric is macro-averaged $F1$, taking the directionality into account (except $PERSONAL - SOCIAL$ relations).

4.3 Participants and Results

4.3.1 Participants

A total of 4 teams participated in the RelEx task. Since each team was allowed to submit up to 3 runs (i.e., 3 different version of their proposal method), a total of 12 runs were submitted. Table 3 lists the participants and provides a rough overview of the system features. VnCoreNLP[2] and underthesea[3] are used for preprocessing. All proposed model are based on the deep neural network architectures with different approaches, go from a simple method (i.e., multilayer perceptron) to Bidirectional Long Short-Term Memory and more complex architectures (e.g., BERT with entity start). With the application of deep learning models, participated teams use several pre-trained embedding model. In addition to word2vec (Mikolov et al., 2013; Vu, 2016), RelEx challenge acknowledgement several BERT-based word embedding for Vietnamese, including PhoBERT (Nguyen and Nguyen, 2020), NlpHUST/vibert4news[4], FPTAI/vibert (The et al., 2020) and XLMRoBERTa (Conneau et al., 2020).

4.3.2 Results

As shown in Table 4, the macro-averaged $F1$ score of participated teams (only considering the best run) ranges from 57.99% to 66.16% with an average of 62.42%. For reference information, the micro-averaged $F1$ score ranges from 61.84% to 72.06% with an average of 66.99%. The highest macro-averaged P and R is 80.38% and 66.75%, respectively. However, the team with the highest P has quite low R, and vice versa, the team with the highest R has the lowest P. The first and second-ranked teams have the right balance between P and R.

We ranked the teams by the performance of their best macro-averaged $F1$ score. Team of Thuat Nguyen and Hieu Man Duc Trong from Hanoi University of Science and Technology, Hanoi, Vietnam submitted the best system, with a performance of 66.16% of $F1$, i.e., 2.74% better than the runner-up system. The second prize was awarded to Pham Quang Nhat Minh with 63.42% of $F1$. The third prize was awarded to SunBear Team from AI Research Team, R&D Lab, Sun Inc, who proposed many improvements in their model. The detailed results of all teams are shown in Table 5.

4.4 Discussion

4.4.1 Relation-specific Analysis

We also analyze the performance for specific relations on the best results of each team for each relation. $PART - WHOLE$ seems to be

[2] https://github.com/vncorenlp
[3] https://github.com/undertheseanlp
[4] http://huggingface.co/NlpHUST/vibert4news-base-cased

No	Team	Main method	Pre-processing	Embeddings	Additional Techniques
1	HT-HUS	Multi layer neural network	+ VnCoreNLP + Underthesea + Pre-processing rules	+ PhoBert + XLMRoBERTa	
2	MinhPQN	+ R-BERT + BERT with entity start	No information	+FPTAI/vibert + NlpHUST/vibert4news	+ Ensemble model
3	SunBear	+ PhoBert + Linear classification + Multi-layer Perceptron	Underthesea	+ PhoBert	+ Join training Named Entity Recognition and Relation Extraction + Data sampling + Label embedding
4	VC-TUS	Bidirectional Long Short -Term Memory network	VnCoreNLP	+ Word2Vec + PhoBert	+ Position features + Ensemble

Table 3: Overview of the methods used by participating teams in RelEx task.

Team	Macro-averaged			Micro-averaged		
	P	R	F1	P	R	F1
HT-HUS	73.54	62.34	**66.16**	76.17	68.37	**72.06**
MinhPQN	73.32	57.09	63.42	76.83	60.28	67.56
SunBear	58.44	**66.75**	62.09	60.82	**73.29**	66.48
VC-Tus	**80.38**	46.43	57.99	**83.51**	49.09	61.84

Results are reported in %.

Highest result in each column is highlighted in bold.

Table 4: The final results of participated teams (best run results).

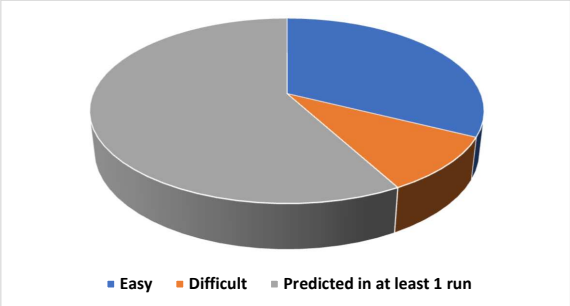


Figure 4: The annotation process.

the easiest relation. Comparing the best runs of teams, the lowest result for this relation is 79.57%, and the highest result was over 84.35%, i.e., the difference is comparatively small (4.78%). $ORGANIZATION - AFFILIATION$ is the relation that has the most difference between the best and worst system (16.73%). The most challenging relation is $PERSONAL - SOCIAL$. It is proved that being a problematic relation for all teams. This note can be clarified from the data statistics, although $PERSONAL - SOCIAL$ is a relation that has many different patterns in realistic, it accounts for only $\sim$ 5% of training and development data. It becomes even more difficult when it takes up $\sim$ 25% of test data. $LOCATED$ follows $PERSONAL - SOCIAL$ in terms of difficulty. Some of its patterns are confused with the $ORGANIZATION - AFFILIATION$ relation, i.e., whether a person is/do something in a particular location or is citizen/resident of a (geopolitical) location. An interesting observation shows that directional relations were not a difficult problem for participated teams. The submission with the most misdirected error failed only 7 examples out of the total number of results returned. Many submission does not have any errors in the directionality.

4.4.2 Difficult Instances

Figure 4 shows the ratio between easy cases (correctly predicted in all runs), difficult cases (did not found by any run), the rest are the number of examples that correctly predicted in at least one run (but not all runs). There were 140 examples ($\sim$ 10%) that are classified incorrectly by all systems. Except for a handful of errors

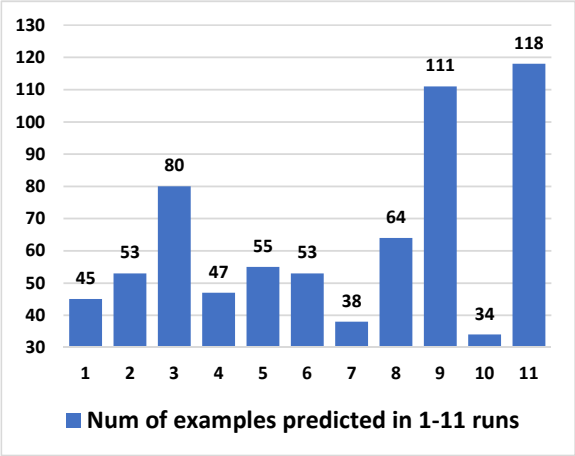


Figure 5: Number of examples predicted in 1-11 runs.

Team/Run	LOC	AFF	P-W	P-S	Macro-averaged			Micro-averaged		
	F1	F1	F1	F1	P	R	F1	P	R	F1
HT-HUS_1	**62.74**	72.33	84.05	40.43	78.76	57.90	64.89	80.49	63.82	71.19
HT-HUS_2	60.70	68.08	**84.35**	44.37	78.17	57.07	64.37	78.68	61.91	69.30
HT-HUS_3	62.50	**74.60**	82.87	44.67	73.54	62.34	**66.16**	76.17	68.37	**72.06**
MinhPQN_1	61.04	65.87	80.77	43.37	72.21	56.78	62.76	75.63	60.08	66.96
MinhPQN_2	62.41	66.38	81.00	43.87	73.32	57.09	63.42	76.83	60.28	67.56
MinhPQN_3	60.40	64.68	80.14	**46.56**	74.36	55.94	62.94	76.87	58.52	66.45
SunBear_1	59.74	67.54	79.57	41.50	58.44	66.75	62.09	60.82	73.29	66.48
SunBear_2	54.43	68.10	76.33	38.83	55.39	64.15	59.42	59.69	70.08	64.47
SunBear_3	49.29	62.10	71.52	31.24	53.11	55.27	53.54	55.91	59.16	57.49
VC-TUS_1	46.37	56.21	74.11	28.68	75.92	40.18	51.34	80.29	44.38	57.16
VC-TUS_2	55.23	57.87	79.70	39.16	80.38	46.43	57.99	83.51	49.09	61.84
VC-TUS_3	54.67	56.96	79.12	38.87	80.83	45.76	57.40	83.38	48.38	61.23

Results are reported in %. Highest result in each column is highlighted in bold.
LOC: LOCATED, AFF: ORGANIZATION-AFFILIATION,
P-W: PART-WHOLE, P-S: PERSONAL-SOCIAL.

Table 5: Detailed results of all submissions.

caused by annotation errors, most of them are made up of examples illustrating the limits of current approaches. We need a more in-depth survey on linguistic patterns and knowledge, as well as more complex reasoning techniques to resolve these cases. A case in point: *"Đừng quên trong tay của HLV Tom Thibodeau vẫn còn đó bộ 3 ngôi sao Karl-Anthony Towns – Andrew Wiggins – Jimmy Buter*." (ID 24527838). In this instance, *[Tom Thibodeau]* is participated in three $PERSONAL - SOCIAL$ relations with *[Karl-Anthony Towns]*, *[Andrew Wiggins]*, and *[Jimmy Buter]*. In which, two relations of *[Tom Thibodeau] - [Andrew Wiggins]* and *[Tom Thibodeau] - [Jimmy Buter]* were not predicted by any team, probably on account of their complex semantics presenting with a conjunction. Another example: *[Hassan được cho là người Iraq , được một cặp vợ chồng người Anh nhận làm con nuôi và cùng sinh sống tại Sunbury , vùng ngoại ô London]* (ID 23352918). Instance *[Hassan] - [Sunbury]* of $LOCATED$ relation is misclassified either as $ORGANIZATION - AFFILIATION$ or as no relation.

Figure 5 gives statistics on how many instances are correctly found in 1 to 11 out of 12 submissions. It shows that the proposed systems of participated teams produce multiple inconsistent results. It also notes the difficulty of the challenge and data.

5 Conclusions

The RelEx task was designed to compare different semantic relation classification approaches and provide a standard testbed for future research. The RelEx dataset constructed in this task is expected to make significant contributions to the other related researches. RelEx challenge is an endorsement of machine learning methods based on deep neural networks. The participated teams have achieved some exciting and potential results. However, the deeper analysis also shows some performance limitations, especially in the case of semantic relations presented in a complex linguistic structure. This observation raises some research problems for future works. Finally, we conclude that the RelEx shared task was run successfully and is expected to contribute significantly to Vietnamese text mining and natural language processing communities.

Acknowledgments

This work was supported by the Vingroup Innovation Foundation (VINIF) under the project code DA137_15062019/year 2019. The shared task committee would like to grateful DAGORAS data technology JSC. for their technical and financial support, and the six annotators for their hardworking to support the shared task.

References

Charu C Aggarwal. 2015. Mining text data. In *data mining*, pages 429–455. Springer.

Nguyen Bach and Sameer Badaskar. 2007. A review of relation extraction. *Literature review for Language and Statistics II*, 2:1–15.

Paulo R Cavalin, Fillipe Dornelas, and Sérgio MS da Cruz. 2016. Classification of life events on social

media. In *29th SIBGRAPI (Conference on Graphics, Patterns and Images)*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

Hercules Dalianis. 2018. Evaluation metrics and evaluation. In *Clinical Text Mining*, pages 45–53. Springer.

Rajarshi Das, Manzil Zaheer, Siva Reddy, and Andrew McCallum. 2017. Question answering on knowledge bases and text using universal schema and memory networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 358–365.

Louise Deléger, Robert Bossy, Estelle Chaix, Mouhamadou Ba, Arnaud Ferré, Philippe Bessieres, and Claire Nédellec. 2016. Overview of the bacteria biotope task at bionlp shared task 2016. In *Proceedings of the 4th BioNLP shared task workshop*, pages 12–22.

Li Dongmei, Zhang Yang, Li Dongyuan, and Lin Danqiong. 2020. Review of entity relation extraction methods. *Journal of Computer Research and Development*, 57(7):1424.

Kata Gábor, Davide Buscaldi, Anne-Kathrin Schumann, Behrang QasemiZadeh, Haifa Zargayouna, and Thierry Charnois. 2018. Semeval-2018 task 7: Semantic relation extraction and classification in scientific papers. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 679–688.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38.

Jiwei Li, Alan Ritter, Claire Cardie, and Eduard Hovy. 2014. Major life event extraction from twitter based on congratulations/condolences speech acts. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1997–2007.

Denis Lukovnikov, Asja Fischer, Jens Lehmann, and Sören Auer. 2017. Neural network-based question answering over knowledge graphs on word and character level. In *Proceedings of the 26th international conference on World Wide Web*, pages 1211–1220. International World Wide Web Conferences Steering Committee.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. *arXiv preprint arXiv:1310.4546*.

Dat Quoc Nguyen and Anh Tuan Nguyen. 2020. Phobert: Pre-trained language models for vietnamese. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 1037–1042.

Viet Bui The, Oanh Tran Thi, and Phuong Le-Hong. 2020. Improving sequence tagging for vietnamese text using transformer-based neural models. *arXiv preprint arXiv:2006.15994*.

Anjali Thukral, Ayush Jain, Mudit Aggarwal, and Mehul Sharma. 2018. Semi-automatic ontology builder based on relation extraction from textual data. In *Advanced Computational and Communication Paradigms*, pages 343–350. Springer.

Xuan-Son Vu. 2016. Pre-trained word2vec models for vietnamese. https://github.com/sonvx/word2vecVN.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus. *Linguistic Data Consortium, Philadelphia*, 57:45.

Goals, Challenges and Findings of the VLSP 2020 English-Vietnamese News Translation Shared Task

Thanh-Le Ha[1,2], **Van-Khanh Tran**[2], **Kim-Anh Nguyen**[2]

[1]Interactive Systems Lab, Karlsruhe Institute of Technology, Germany
thanh-le.ha@kit.edu
[2]Speech and Language Processing Department, Vingroup Big Data Institute, Vietnam
{v.leht6, v.khanhtv13, v.anhnk9}@vinbigdata.org

Abstract

This paper reports the VLSP 2020 English-Vietnamese News Translation shared task, which is one of the six shared tasks organized at the seventh annual workshop on Vietnamese Language and Speech Processing (VLSP 2020). In this task, we provided parallel and monolingual data for training machine translation systems translating English texts into Vietnamese, with the focus of *news* domain. There were 6 teams participating into the tasks, with 13 submissions in total. We performed both automatic and human evaluations on the submissions and presented the results and our findings at the conference. We hope this would boost the research of Vietnamese machine translation community and start maintaining annual machine translation tasks at VLSP conferences.

1 Introduction

VLSP stands for Vietnamese Language and Speech Processing Consortium. It is an initiative to establish a community working on speech and text processing for the Vietnamese language. The VLSP 2020 is the seventh annual international workshop and evaluation campaign.

Machine Translation (MT) is one of the six shared tasks for the VLSP evaluation campaign this year and it is the first time that MT is organized as a VLSP shared task after being a trial task in 2013. As an important research problem of Language and Speech Processing (LSP), MT often attracts interests from the research community. However, research in Vietnamese language-related MT often conducted by several R&D departments from big companies and research labs in large universities. In 2015, the prestigious MT campaign IWSLT (Cettolo et al., 2015), whose conference was organized in Da Nang, Vietnam, featured English-Vietnamese MT as one of the MT task of that year's campaign and it has been the first and only MT evaluation featuring Vietnamese language to date. We set the following goals when organizing this VLSP 2020 MT evaluation campaign:

- Reviving a traditional task in any LSP community and making it to be a recurrent event. Encouraging research for Vietnamese-related MT and engaging researcher into solving interesting problems of MT
- Motivating the contribution of free data and basic LSP tools supporting Vietnamese-related MT research.
- Extending practical applications of MT into smart tools and workflows, e.g. developing multilingual education channels, fighting again fake news in any languages and overcoming language barrier in business, tourism, entertainment and international communication.

Concretely, we have the following contributions while organizing VLSP 2020 English-Vietnamese News Translation task:

- Crawl, collect, compile and release free parallel and monolingual datasets for training and testing English-Vietnamese MT systems[1].
- Establishing a standard benchmark for research on English-Vietnamese Translation.
- Conduct automatic and human evaluations of the participating MT systems.

This paper is organized as follows. We describe the dataset for training and testing MT systems in Section 2. Section 3 lists the participating teams and summarizes the approaches they employed in

[1]The datasets are published at https://github.com/thanhleha-kit/EnViCorpora

Dataset Name	Domain	Size (# Sentence Pairs)
News	News (in-domain)	20K
Basic	Basic and short conversation	8.8K
EVBCorpus	Mixed domains	45K
TED-like	Educational & Tech Talks	546K
Wiki-ALT	Wikipedia articles	20K
OpenSubtitle	Movie Subtitles	3.5M
Corpus.2M.shuf	Monolingual Corpus of Vietnamese News	2M

Table 1: *Training Datasets for VLSP 2020 MT task*

their systems. Section 4 presents how we evaluated the translation outputs. We then show the evaluation results in Section 5. Finally, we conclude the the paper by giving our findings and drawing our future plans for the task.

2 Dataset

Although English-Vietnamese is the most popular language pair in the Vietnamese MT community, it is currently considered as "low-resource" language pair, where there are only a few public English-Vietnamese parallel corpora with adequate quality for training MT systems. They are Wikipedia articles extracted for the Asian Language Treebank project (Riza et al., 2016), mixed-domain EVB parallel corpus collected by Ngo et al. (2013), a multilingual corpus of short and basic sentences from Tatoeba project[2](Tiedemann, 2012) and a COVID-19 multilingual corpus created by ELRC[3] and compiled by Tiedemann (2012). In total, those corpora contain around 75,000 English-Vietnamese sentence pairs. Besides those high quality datasets, OPUS[4](Tiedemann, 2012), a website collecting translated texts from the web, compiles and publishes clean versions of movie subtitle datasets extracted from OpenSubtitles[5] (Lison and Tiedemann, 2016), as well as religious news and bible translations. Although they are large corpora with the number of sentence pairs varying from hundreds thousands to more than three millions, they are unusable without any filtering method since the domain are very narrow (religious) and the quality is not good (movie subtitles).

[2]https://tatoeba.org/eng/
[3]https://elrc-share.eu/
[4]https://opus.nlpl.eu/
[5]https://www.opensubtitles.org/en

2.1 Training Data

Parallel Data. We decide to create more parallel data for English Vietnamese. Our crawling sources are high-quality bilingual or multilingual websites of news and one-speaker educational talks of various topics, mostly technology, entertainment and design (hereby referred as TED-like talks). Because those websites are required to convey the original content in English to other languages (including Vietnamese) and often gone through several review stages before publishing, the quality is assured.

First, we extracted some basic conversations from English teaching websites and coupled them to the *Tatoeba* dataset. For the news domain, we crawled the data from and then applied some simple filtering methods to remove short sentences. Finally, we combined the crawled data with the *COVID-19 ELRC* data to produce a 20,000-sentence-pair parallel corpus.

For the TED-like domain, we downloaded *TED* talks monolingual data of English and Vietnamese from WIT3[6] (Cettolo et al., 2012), then aligned them based on the sentence ids. Furthermore, we extracted a parallel corpus from the subtitles of the TED-like videos uploaded on Amara[7] - a platform to assist its users to produce captions and subtitles of the videos they uploaded. As the result, more than five hundreds thousands sentence pairs were crawled.

Since the quality of the large *OpenSubtitle* dataset varies in movies, we decided to include it into the training data and let the participants choose how to use it. In the end, we released the following training data in which *news* is the in-domain data:

Monolingual Data. For this evaluation, we provided target monolingual data which is 2 million Vietnamese sentences, crawled from Vietnamese

[6]https://wit3.fbk.eu/
[7]https://amara.org/en/

Team	Affiliation	Submitted
Bluesky	Unknown	2
EngineMT (Ngo et al., 2020)	UET-ICTU	6
Lab-914 (Le and Nguyen, 2020)	HUST	2
NLP-HUST	HUST	1
THORLab	D-Soft	1
RD-VAIS (Pham et al., 2020)	TNU-HUST-VAIS	1

Table 2: *The teams participated to VLSP 2020 MT task*

newspapers from various topics. The text has adequate quality to train language models or to conduct back translation. Similar to the parallel data, we let the participants decide how to preprocess the data.

2.2 Validation and Test Data

Validation Data. While crawling the news data for training, we also reserved a small part to be validation data. We released a development dataset and a public test dataset at the same time with the training data. The development set contains 1007 English-Vietnamese sentence pairs and the public test set contains 1220 English-Vietnamese sentence pairs. The participants could use one of the validation sets to turn their models' hyperparameters and the other sets for choosing the primary system to be submitted.

Official Test Data. We informed the participants in advance that the in-domain data is `News`, but we did not reveal the theme is *Covid-19 News* until the report of the evaluation campaign. In order to avoid cheating and accidentally inclusion of the test data into training or validation data, we manually selected up-to-date English news about *Covid-19* from international online newspapers and then asked professional translators to translate them into Vietnamese. The translators need to conform some strict guidelines while translating the official test set, in order to keep it high quality. As the result, the official test set contains 789 sentence pairs. We mixed them with other crawled 2000 sentence pairs and distributed the English part to the participants, asking them to produce the Vietnamese translation using their models.

3 Participants and their Approaches

The organizers received submissions from 6 different teams with the total number of 13 submissions. Table 2 lists the teams. Among them, there are only 3 teams sending their paper describing their approaches and models.

3.1 Architecture

All of the three teams submitted neural machine translation systems. And all of them implemented their systems using the state-of-the-art *Transformer* architecture (Vaswani et al., 2017). The configurations are different, however. In *EngineMT* and *RD-VAIS* systems, the number of layers is 4 and the model size is 512 while in *Lab-914* the number of layers is 6 and the model size is 1024.

3.2 Preprocessing

In the preprocessing phase, the teams utilized common techniques on the parallel data. They all removed long sentences, tokenized the words simply by white-spaces and applied some casing treatments. In addition, *Lab-914* performed those techniques plus further filtering methods to remove noisy sentences from the Vietnamese monolingual corpus. For casing treatments, *Lab-914* simply lower-cased the data, *RD-VAIS* marked capitalized and upper-cased words by some special tokens before lowercasing and *EngineMT* applied smart casing using Moses toolkit (Koehn et al., 2007). All the teams performed subword tokenization using *Byte-Pair Encoding* algorithm (Sennrich et al., 2016b) implemented in `subword-nmt`[8] framework with the number of merging operations set at 35,000.

3.3 Back Translation

All the teams employed *Back Translation* (Sennrich et al., 2016a) as the sole technique to exploit monolingual data. However, each team had different strategies on how to use the monolingual data. *Lab-914* used all the monolingual data provided while *EngineMT* used much smaller monolingual corpus after filtering out most of them using their proposed data selection techniques. *RD-VAIS*, on the other hand, built two systems different on the

[8] https://github.com/rsennrich/subword-nmt

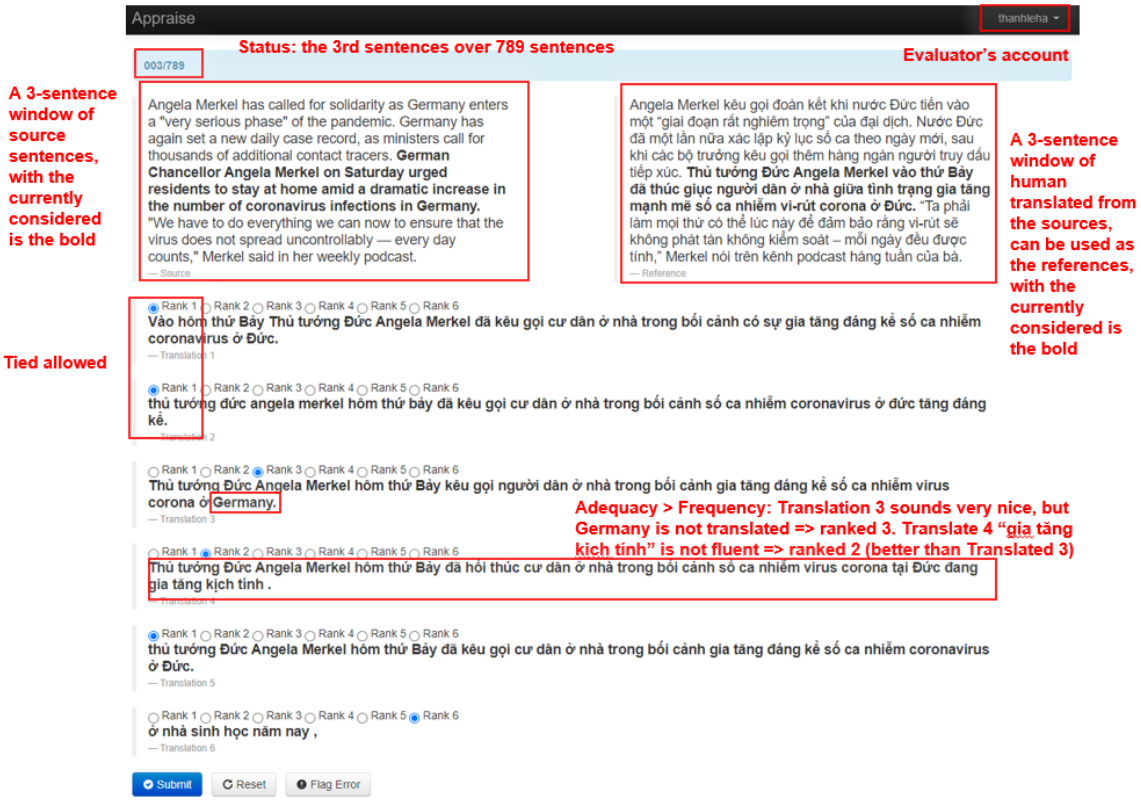


Figure 1: *Appraise's main interface to rank translation outputs*

size of the monolingual corpus. One contains 1 million sentences and the other contains all 2 million sentences. At the end, they chose the system trained with 1 million sentences back translation based on the performance on the public test set.

3.4 Domain Adaptation

We organized the task in a way that we would expect to see some domain adaptation techniques. *Lab-914* did not employ any specific domain adaptation when they used all the provided data in their systems and treated the in-domain news data the same as other data. *RD-VAIS*, besides the monolingual data which is news, they used only the parallel in-domain data. This might affect badly on their systems since the in-domain data is small and most of their training data come from back translated data. *EngineMT* is the team who employed several domain adaptation approaches. First they select subsets of data, both monolingual and parallel, which are relevant to the in-domain data with their TF-IDF-based data selection technique. Then they fine-tuned their models on the in-domain data and ensembled all the models they had.

4 Evaluation

VLSP 2020 is the first Machine Translation Evaluation Campaign for Vietnamese that has both automatic and human evaluation. Furthermore, the human evaluation result is used to rank the teams in the campaign.

4.1 Automatic Evaluation

For this campaign, we employed two metrics to evaluate the submissions: BLEU and TER. Since BLEU is the most popular automatic evaluation metric in Machine Translation, it is the main metrics to rank submissions in the automatic evaluation section.

4.2 Human Evaluation

Five experts which are professional translators and interpreters were invited to conduct the human evaluation for 6 primary systems from 6 teams. Each of them was asked to independently rank the translation outputs of 789 sentences. They were required to follow the evaluation guidelines in which the quality of the translations is rated based on two main criteria: *Adequacy* and *Fluency*. *Adequacy* is rated higher than *Fluency*, however.

Rank	Team	BLEU	TER
1	EngineMT	38.39	0.45
2	RD-VAIS	33.89	0.53
3	Bluesky	32.38	0.56
	Lab-914	32.10	0.50
5	NLP-HUST	23.72	0.62
6	THORLab	2.53	-

Table 3: *Automatic evaluation results of the MT task*

Rank	Prize	Team	Score
1	1st prize	Lab-914	1.554
2	2nd prize	EngineMT	1.327
3	3rd prize	RD-VAIS	0.864
4	-	Bluesky	0.536
5	-	NLP-HUST	-0.043
6	-	THORLab	-4.239

Table 4: *Human evaluation results of the MT task*

We used *Appraise*[9](Federmann, 2018) - an open-source web-based MT evaluation framework to assist the experts for the evaluation process. Figure 1 is the main interface that the evaluator can rate the outputs of all submitted systems. For each sentence, the evaluator is shown the English source sentence in a context of three sentences: the previous sentence, the currently considered sentence and the followed sentence. Also the golden translation of those three sentences are displayed as the references. The evaluator needs to rank each system's output from 1 (best) to 6 (worst), and tied ranking is allowed for two or more systems having the same translation quality.

The rankings from 5 experts were converted to pair-wise rankings (number of wins, loses and ties between a pair of two systems). Then they were combined into overall scores using a variant of *TrueSkill* (Sakaguchi et al., 2014), a sophisticated algorithms considering not only the average number of wins but also how difficult the task is and the variance of each system's translation quality.

5 Evaluation Results

5.1 Automatic Evaluation

We evaluated all the submissions, including contrastive systems and informed the participants BLEU and TER of their systems. But only the primary systems are ranked, and by their BLEU scores within statistically significant differences ($p \leq 0.05$). The ranking of the teams with corresponding BLEU and TER scores are described in Table 3.

Excepts the team *THORLab* seemed to have some errors in their submission, other teams produced decent outputs. Unsurprisingly, *EngineMT* led the board with a considerably large margin to the second team *RD-VAIS*, maybe because of their domain adaptation techniques. *Bluesky* and *Lab-914* were shared the third rank when the differences between their BLEU scores is not significantly obvious. Notably, based on TER, *Lab-914* were ranked second, only after *EngineMT*.

In some internal test, we realized that other data excepts the *OpenSubtitle* were high quality and would bring improvements to the systems that use them, even their domain are not news. *Lab-914* used a large transformer model and all the provided data, but their BLEU score are not on pair with *RD-VAIS* which used only the small, in-domain parallel data. We looked into their outputs and their system description as an attempt to explain the possible inconsistency and we discovered that they did not recover casing of their outputs. BLEU is based on the number of overlapping n-grams so that it is more sensitive to upper-cased and capitalized words than TER which is based on the accuracy of individual words. Later, the human evaluation verified our discovery.

5.2 Human Evaluation

As described in Section 4.2, we gathered the ranking of all the systems from 5 experts and produced a unique score for each systems by using the *TrueSkill* algorithm with the bootstrap resampling at p-level of $p \leq 0.05$. Table 4 lists the final ranking of the teams by human evaluation.

While in automatic evaluation, *EngineMT* is ranked first, here it goes runner-up, after *Lab-914*. This verifies our assumption about casing recovery. The automatic evaluation metrics do consider casing in their calculation, but the evaluators do not, following their evaluation guidelines.

6 Findings and Future Plans

VLSP 2020 English-Vietnamese News Translation task is the first official MT task hosted by VLSP organizers and it is also the first Vietnamese-related MT evaluation campaign featuring both automatic

[9] https://github.com/cfedermann/Appraise

and human evaluation. We hope that it would bring scientific and practical values to the VLSP community as well as our society in dealing with the Covid-19 pandemic and in developing useful AI tools.

These are our findings from this VLSP 2020 English-Vietnamese News Translation task:

- English-Vietnamse MT is still a low-resource task with the lack of large-size, high-quality datasets. *Back Translation* on news data helps improving the overall translation quality. More data, even with mediocre quality and out-of-domain (e.g. *OpenSubtitle*), when being used to train large models, also brings significantly gains, especially in the human evaluation.

- News might not be a good domain in case we would like to encourage domain adaptation techniques. Monolingual corpora are often crawled from online newspapers and Back Translation might outperform your finest domain adaptation techniques.

- The approaches and techniques are very common and well-known. There is no interesting research finding from the participants.

- There was no submission considering the linguistic characteristics of Vietnamese language or the differences between two languages: English and Vietnamese.

- There were a few participating teams.

We would like to continue hosting MT evaluation tasks in the near future with these plans in mind:

- More language directions in both well-resource and low-resource conditions

- More data in the popular MT tasks

- Consider some useful and interesting domains such as medical, law or technical domains.

- Spread the words to attract more participants working on interesting MT tasks.

7 Acknowledgment

This task is mainly conducted by the Speech and Language Processing Department, VinGroup Big Data Institute with the help of professional translators and interpreters. We would like to thank Vingroup Innovation Foundation (VINIF) for the financial support and the Association for VLSP for organizing support during the campaign.

References

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. WIT3: Web Inventory of Transcribed and Translated Talks. In *Conference of European Association for Machine Translation*, pages 261–268.

Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, Roldano Cattoni, and Marcello Federico. 2015. The IWSLT 2017 Evaluation Campaign. In *International Workshop on Spoken Language Translation (IWSLT'15)*, Danang, Vietnam.

Christian Federmann. 2018. Appraise - Evaluation Framework for Machine Translation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 86–88.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*, pages 177–180.

Duc Cuong Le and Thi Thu Trang Nguyen. 2020. Vietnamese-English Translation with Transformer and Back Translation in VLSP 2020 Machine Translation Shared Task. In *Proceedings of the Sixth Conference of the Association for Vietnamese Language and Speech Processing (VLSP 2020)*.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting Large Parallel Corpora from Movie and TV Subtitles. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC'2016)*.

Quoc Hung Ngo, Werner Winiwarter, and Bartholomäus Wloka. 2013. EVBCorpus - A Multi-layer English-Vietnamese Bilingual Corpus for Studying Tasks in Comparative Linguistics. In *Proceedings of the 11th Workshop on Asian Language Resources*, pages 1–9.

Thi-Vinh Ngo, Minh-Thuan Nguyen, Hoang-Minh-Cong Nguyen, Hoang-Quan Nguyen, Phuong-Thai Nguyen, and Van-Vinh Nguyen. 2020. The UET-ICTU Submissions to the VLSP 2020 News Translation Task. In *Proceedings of the Sixth Conference of the Association for Vietnamese Language and Speech Processing (VLSP 2020)*.

Ngoc Phuong Pham, Quang Chung Tran, Quang Minh Nguyen, and Hong Quang Nguyen. 2020. A Report on the Neural Machine Translation in VLSP Campaign 2020. In *Proceedings of the Sixth Conference of the Association for Vietnamese Language and Speech Processing (VLSP 2020)*.

Hammam Riza, Michael Purwoadi, Teduh Uliniansyah, Aw Ai Ti, Sharifah Mahani Aljunied, Luong Chi Mai, Vu Tat Thang, Nguyen Phuong Thai, Vichet Chea, Sethserey Sam, et al. 2016. Introduction of the Asian Language Treebank. In *2016 Conference of The Oriental Chapter of International Committee for Coordination and Standardization of Speech Databases and Assessment Techniques (O-COCOSDA)*, pages 1–6. IEEE.

Keisuke Sakaguchi, Matt Post, and Benjamin Van Durme. 2014. Efficient elicitation of annotations for human evaluation of machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 1–11.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving Neural Machine Translation Models with Monolingual Data. In *Association for Computational Linguistics (ACL 2016)*, Berlin, Germany.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural Machine Translation of Rare Words with Subword Units. In *Association for Computational Linguistics (ACL 2016)*, Berlin, Germany.

Jörg Tiedemann. 2012. Parallel Data, Tools and Interfaces in OPUS. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC'2012)*, pages 2214–2218.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.