VNU Journal of Science:
Computer Science and Communication Engineering

Journal homepage: http://www.jcsce.vnu.edu.vn/index.php/jcsce

Original Article

VNNLI - VLSP 2021: Leveraging Contextual Word Embedding for NLI Task on Bilingual Dataset

Duong Quoc Loc*, Nguyen Duc Vu

University of Information Technology, Vietnam National University,
Quarter 6, Linh Trung Ward, Thu Duc District, Ho Chi Minh City, Vietnam

Received 27 December 2021
Revised 31 March 2022; Accepted 5 May 2022

Abstract: Natural Language Inference (NLI) is one of the critical tasks in natural language understanding which we take through the VLSP2021-NLI Shared Task competition. VLSP2021-NLI Shared Task is a competition to improve existing methods for NLI tasks, thereby enhancing the efficiency of applications. One of the challenges of the competition is the dataset in both Vietnamese and English. In this article, we report on evaluating the NLI task of the competition. We first implement the 5-fold cross-validation evaluation method. We following leverage model architectures pre-trained on cross-lingual language datasets such as XLM-RoBERTa and RemBERT to create contextual word embeddings for classification. Our final result reaches 90.00% on the test dataset of the organizers.

Keywords: Contextual Word Embedding, NLI, NLP.

1. Introduction

Natural Language Inference (NLI) or Recognizing Textual Entailment (RTE) is the task of determining whether a natural-language hypothesis can be inferred from a given premise [1]. Specifically, NLI is the task of classifying a pair of premise and hypothesis sentences into three classes: entailment, neutral, and contradiction. NLI plays an essential role in Natural Language Understanding, such as Question Answering, Text summarizing, and Relation Extraction. For example, Question Answering applies NLI to validate or re-rank candidate answers. A candidate answer is considered correct if the passage entails the corresponding hypothesized answer. The passage in the Question Answering problem is considered the premise, and the answers are considered a different hypothesis. Nowadays, the amount of information online is growing, especially textual information. The recognition and understanding of textual content are essential because textual content on the internet can be harmful, deceptive, misleading, or violent. Therefore, the NLI task is helpful for

* Corresponding author.
E-mail address: 18521006@gm.uit.edu.vn

https://doi.org/10.25073/2588-1086/vnucsce.317

many practical applications and avoids many risks from negative information.

We tackled the NLI task through the VLSP2021-NLI Shared Task competition [2]. One of the challenges of the competition is that the pairs of sentences are in Vietnamese or English or may not be in the same language, which causes difficulty in classification.

We approach this task using pre-trained models on cross-lingual languages dataset to generate contextual word embeddings for classification and evaluation based on the 5-fold cross-validation method. The results that we achieve are pretty good, with the average result of 5 models corresponding to the validation dataset being 0.963 and the result on the test dataset reaching 0.900.

This paper presents the process of performing NLI tasks of the VLSP2021-NLI competition. In Section 2 and Section 3, we briefly present the works involved and describe the tasks required to be performed, respectively. Section 4 presents the process of problem solving methods. In Section 5, we present the experimental results, error analysis, and results submitted to the organizers. Finally, we present the main contribution and conclusion in section 6.

2. Related Works

The NLI task has attracted the attention of researchers since 2005. A series of RTE conferences (RTE-1 [3] - RTE-7 [4]) have been organized every year from 2005 - 2011. The series of conferences aimed to develop methods to enhance results and enrich data for this task. Papers presenting the method in RTE-1 - RTE-5 [5] conference, whose results depend on the use of additional information about syntax and semantic interpretation from a variety of sources. Besides, the methods in the RTE-6 [6] and RTE-7 conferences focus on the context of the training dataset and the knowledge dataset.

Google introduced the Transformer architecture [7] for neural machine translation application in 2017. The core idea of the architecture is the multi-head self-attention to compute the input in parallel. Therefore, this architecture solved the long-term dependency problem that has occurred traditional sequential model. Besides, the multi-head self-attention captures different parts and aspects of the input, helping to understand the context. The Transformers sets the stage for developing contextual representation models.

The XLM-RoBERTa [8] model architecture is a variation of transformers architecture proposed by Facebook AI in 2020. This model is trained based on a Transformer based masked language model architecture on 100 languages improving cross-lingual languages' understanding of the model. This work represents that the representations learned on large-scale multilingual datasets are adequate for NLI downstream tasks via fine-tuning. The model's average accuracy across the NLI task languages reached 79.2% with 80.8% in Vietnamese and 89.1% in English.

In 2020, the RemBERT [9] model architecture, which is a larger version of the XLM-R model, trained by Google on large-scale cross-lingual languages datasets. Their work focuses on analyzing embedding size. They observed that increasing embedding output size improves performance on the fine-tuning tasks. Therefore, they decreased the number of input parameters and increased the number of output parameters of the embedding layer. The performance of the RemBERT outperforms the XLM-RoBERTa on sentence-pair classification. The average accuracy of the model across languages is higher than the XLM-RoBERTa model with 80.8%.

3. Task Description

VLSP2021-NLI Shared Task requires candidates to predict a given pair of sentences whether they semantically agree, disagree, or are neutral with each other. Overview of the training dataset provided by VLSP2021-NLI Shared Task includes a JSON file containing 16185 data points, each data point consists of 6 attributes: id, lang_1, lang_2, sentence_1, sentence_2, and label. The label attribute includes three types of "agree", "neutral", and "disagree" equally

distributed. The test dataset consists of 4177 data points with 3 attributes id, sentence_1, sentence_2, and label, the number of labels on the test set is also equally distributed. The number of Vietnamese and English sentences is shown in Table 1. The organizers use the F1-score to evaluate the prediction results.

Table 1. Count the number of Vietnamese and Englishsentences

	Training dataset		Test dataset	
	sentence_1	sentence_2	sentence_1	sentence_2
vi	8685	16185	2118	4177
en	7500	0	2059	0

4. Methodology

4.1. Data Processing

Because there is no validation dataset provided, we decided to fine-tune and evaluate the performance of models using the k-fold cross-validation method. The goal of this method is to maximize the use of the data set for training and evaluation. In the folds k-fold cross-validation method, the provided training dataset is randomly divided into k without replacement, k 1 folds used fine-tuning, and one fold for performance evaluation. This process is repeated for k iterations shown in Figure 1. In this method, we chose k = 5, the number of labels for each model corresponding to each iteration is accounted in Table 2.

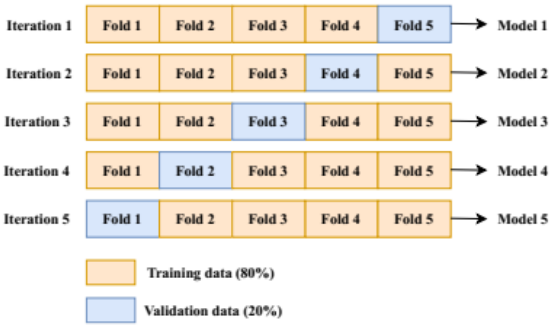


Figure 1. Visualization of 5-fold cross-validation method.

To prepare for the fine-tuning process, two attributes sentence_1 and sentence_2 are tokenized pairwise into vectors using the built-in SentencePiece model [10] in the HuggingFace library 1, each vector has an equal fixed length of 128. Besides, the label attribute containing the labels: “agree”, “neutral”, and “disagree” is also encoded as positive integers 0, 1, and 2 respectively.

Table 2. The number of labels over five folds

	Agree		Neutral		Disagree	
	Train	Validation	Train	Validation	Train	Validation
Iteration 1	4304	1096	4347	1053	4309	1091
Iteration 2	4276	1124	4321	1079	4363	1037
Iteration 3	4367	1033	4286	1114	4307	1093
Iteration 4	4307	1093	4341	1059	4312	1088
Iteration 5	4346	1054	4305	1095	4309	1091

4.2. Feature Extraction

BERT was one of the earliest variations of the transformers architecture using only the encoder architecture. BERT model learns the representation on large-scale linguistic dataset through two strategies Masked Language Modeling and Next Sentence Prediction, the details of the BERT model from [11]. Each layer of BERT captures the different features of the

input text. The BERT model has significantly increased accuracy for the NLI task via fine-tuning compared to traditional deep learning methods. In addition, the authors of BERT demonstrated that BERT could also be used to generate contextual word embeddings with good results in some tasks, such as named-entity recognition. There are six outstanding options given for creating contextual word embeddings, of which the combination of the last layer of the BERT model gives the best results 2. Since XLM-RoBERTa and RemBERT models are upgraded versions of BERT, we also applied this strategy to these models for the NLI task.

4.2.1. XLM-RoBERTa

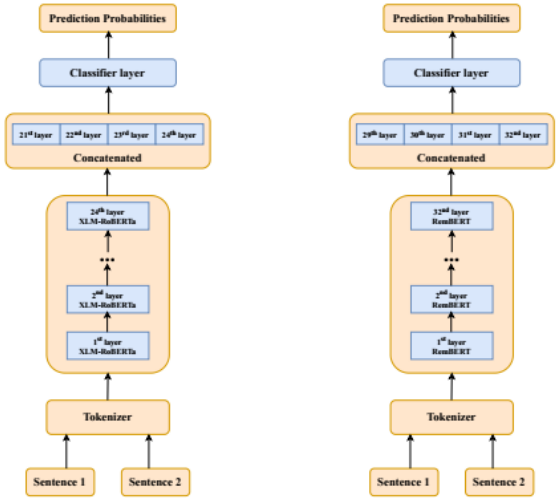
The XLM-RoBERTa model consists of 24 encoder layers stacked on top of each other; the number of hidden states of the model is 1024, 16 attention heads per layer, with a 250k vocabulary trained on data covering 100 languages. The total number of parameters of the model is about 550M. We have leveraged the XLM-RoBERTa model to create contextual word embedding according to the strategy mentioned above. Therefore, the last four layers of the XLM-RoBERTa model were concatenated to obtain different semantic information.

4.2.2. RemBERT

The RemBERT model is larger than the XLM-RoBERTa model; the RemBERT model includes 32 encoder layers, the number of hidden states of the model is 1152, 18 attention heads per layer, with a 250k vocabulary. Significantly, the number of input dimensions of 256 is smaller than the number of output dimensions of 1536 in the embedding layer. The total number of parameters of the model is about 559M. The RemBERT model was trained on the dataset containing 110 languages. Similarly, the last four layers of the model were also concatenated to create contextual word embedding.

4.3. Prediction Model

In this section, we describe the steps in predicting the output submitted to the organizers. The first step is to fine-tune the model on the provided training dataset. Ten models were fine-tuned and evaluated by the 5-fold cross-validation method mentioned in Section 4.1, corresponding to five models based on XLM-RoBERTa architecture and five models based on RemBERT architecture. The main components of these models are the contextual word embedding layer presented in Section 4.2 and the classifier layer. This classifier is simply a Dropout class followed by a Dense class. The architecture of these models is shown in Figure 2. All models have been inherited from the modules of the Hugging Face library. The details of hyperparameters, hardware, and timing for the fine-tuning and evaluation process of all the models are shown in Table 3.



(a) Model based XLM-RoBERTa. (b) Model based RemBERT.

Figure 2. Visualization of the fine-tuned models.

The next step is to create an ensemble model for the prediction on the test dataset; we combined models based on XLM-RoBERTa and RemBERT in pairs shown in Figure 3a that had been fine-tuned and evaluated. There are five ensemble models for prediction; these ensemble models made predictions based on the sum of the weighted prediction probabilities of the two models. We experimented on different weights and observed that a proportion of approximately 6 : 4 corresponds to the prediction probabilities

of the two models based on XLM-RoBERTa and RemBERT, respectively, giving the best results.

In the last step, the prediction probabilities of the five ensemble models are averaged, from which the output labels are calculated (Figure 3b). This output is the final result submitted to the organizers

Table 3. Configuration of all models

	Models based XLM-RoBERTa	Models based RemBERT
Epoch	5	5
Batch size	16	12
Learning rate	2e-5	2e-5
Weight decay	0.01	0.01
Warmup step	405	540
Optimizer	AdamW (β_1 = 0.9, β_2 = 0.999)	AdamW (β_1 = 0.9, β_2 = 0.999)
Training time	approximately 8h	approximately 12h
Test time	1m20s	1m40s
Hardware	1 GPU Tesla P100-PCIE (16GB) on Google Colab	1 GPU Tesla P100-PCIE (16GB) on Google Colab

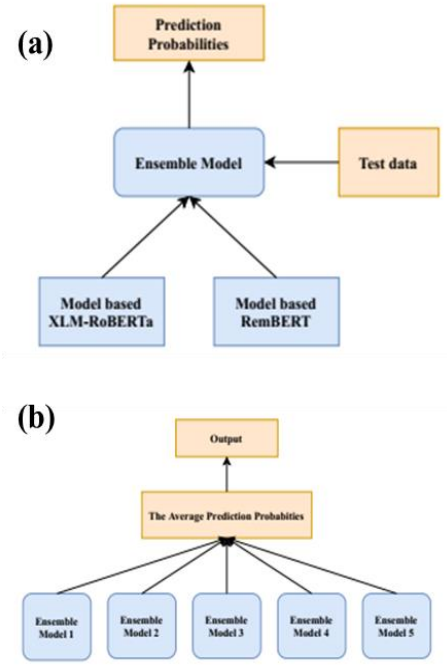


Figure 3. To illustrate the prediction process: (a) Visualization of the ensemble model and (b) how to calculate the output.

5. Results

Each layer of the XLM-RoBERTa and RemBERT captures different features of the input text. Therefore, we have explored different fine-tuning strategies, such as using the concatenation of the last four layers (concat), the average of the last four layers (mean), and only one last layer (last), thereby comparing the effects of these refining strategies on model performance. The results (Table 4) show that the performance of all models on the validation dataset gives high results and are approximately the same; this proves there is no bias between the models. In addition, these results show that the performance of the ensemble models increases by approximately 1% compared to the individual models. Besides, the results on the validation dataset show that the fine-tuning strategy by concatenating the last four layers gives about 0.1-0.5% higher results than the remaining strategies.

Table 5 represents the model performance of different fine-tuning strategies on the test dataset. It can be seen that the results on the test set decrease about 7% compared to the corresponding validation set for each model. However, the results on this table also show that using 5-fold cross-validation helps avoid overfitting or underfitting. The fine-tuning strategy using the concatenation of the last four-layer gives better results than the strategy using

the average of the last-four layers and only one last layer. Therefore, we use the concatenation of the last four-layers strategy to predict the output for submission to the organizers.

Figure 4 shows the trend of the ensemble models on the validation dataset and the result submitted on the test dataset of the concatenation strategy. The error trend of the models is the same, in which the label "disagree" accounts for the most errors, followed by the labels "neutral" and "agree" corresponding to each model. Confusion matrices (Figure 5) represents explicitly the proportion of labels that are wrongly predicted into the other labels. One of the main reasons why the model predicts wrong is because of the length of the sentence, there are 20 pairs of encoded sentences whose length exceeds 128. If the length of the encoded pairs exceeds 128, it can cause information loss.

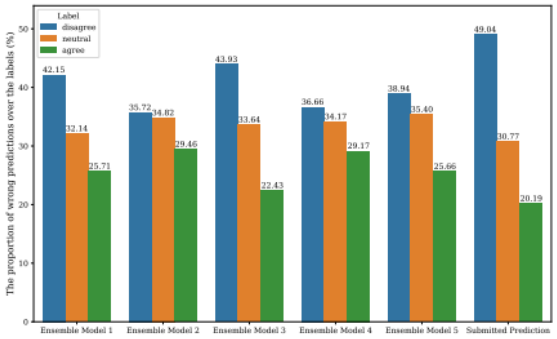


Figure 4. Visualization of the proportion of wrong predictions over labels of the concatenation strategy.

Table 4. Metric summary of fine-tuned models on the validation dataset. All models were measured in f1-score

	Model based XLM-RoBERTa			**Model based RemBERT**			**Ensemble Model**		
	Concat	Mean	Last	Concat	Mean	Last	Concat	Mean	Last
Model 1	**0.958**	0.957	0.956	**0.955**	0.959	0.953	**0.965**	0.964	0.963
Model 2	**0.956**	0.954	0.955	**0.959**	0.954	0.954	**0.956**	0.961	0.962
Model 3	**0.955**	0.953	0.954	**0.958**	0.949	0.954	**0.961**	0.959	0.960
Model 4	**0.961**	0.950	0.951	**0.953**	0.954	0.952	**0.962**	0.961	0.961
Model 5	**0.960**	0.956	0.958	**0.952**	0.958	0.953	**0.968**	0.964	0.963

Table 5. Metric summary of fine-tuned models on the test dataset

	Concat				Mean				Last			
	Accuracy	Precision	Recalll	F1-score	Accuracy	Precision	Recalll	F1-score	Accuracy	Precision	Recalll	F1-score
Ensemble 1	0.895	0.895	0.895	0.895	0.891	0.892	0.891	0.891	0.889	0.889	0.889	0.889
Ensemble 2	0.892	0.893	0.892	0.892	0.895	0.895	0.895	0.895	0.882	0.882	0.882	0.882
Ensemble 3	0.897	0.897	0.897	0.896	0.897	0.897	0.897	0.896	0.893	0.893	0.893	0.893
Ensemble 4	0.896	0.896	0.895	0.895	0.893	0.893	0.892	0.892	0.890	0.890	0.890	0.890
Ensemble 5	0.895	0.896	0.895	0.895	0.895	0.895	0.894	0.894	0.894	0.893	0.894	0.893
Average	0.900	0.900	0.900	0.900	0.898	0.898	0.898	0.898	0.897	0.897	0.896	0.897

For example, a pair of sentences consisting of sentence_1: “The US Centers for Disease Control and Prevention removed instructions from its website for doctors on how to prescribe two anti-malarial drugs that President Donald Trump says have the potential to stop the new coronavirus yesterday.” and sentence_2: “Guidance for doctors on prescribing two antimalarial drugs that President Donald Trump considers a potential cure for SARs-CoV2 was posted on the Centers for Disease Control website and America’s Disease Prevention and Control yesterday.” predicted to label “agree” while the correct label is “disagree”. It can be seen that it is quite a long sentence pair exceeding 128, in

which important information is located near the end of sentence 2, after encoding with a fixed length of 128, it is cut off, leading to incorrect prediction results.

Table 6. Metric summary on the test dataset atlanguage level of the concatenation strategy

	Accuracy		Precision		Recall		F1-score	
	en-vi	vi-vi	en-vi	vi-vi	en-vi	vi-vi	en-vi	vi-vi
Ensemble Model 1	0.887	0.892	0.888	0.891	0.888	0.891	0.887	0.891
Ensemble Model 2	0.890	0.894	0.891	0.894	0.891	0.893	0.890	0.893
Ensemble Model 3	0.894	0.892	0.894	0.892	0.894	0.892	0.894	0.892
Ensemble Model 4	0.881	0.893	0.882	0.893	0.882	0.892	0.881	0.892
Ensemble Model 5	0.888	0.889	0.888	0.889	0.888	0.889	0.888	0.888
Submitted Prediction	**0.896**	**0.905**	**0.896**	**0.904**	**0.897**	**0.904**	**0.896**	**0.904**

Table 6 presents the prediction results of the ensemble models and the results submitted at the language level. It can be seen that the level of understanding between two sentences of the same Vietnamese language is higher than that of two sentences of different languages, but the difference is not too high, about 1%.

5. Conclusion

Figure 5. Confusion matrices for the ensemble models on validation dataset and the predicted results submitted to the organizers on the test dataset of the concatenation strategy.

Our main contribution to the VLSP2021-NLI Shared Task is to leverage the architecture of pre-trained models on cross-lingual language datasets to create contextual word embeddings that are efficient for classification. In addition, we have also investigated different fine-tuning strategies to compare the impact on model performance. In our experiment, we use the 5-fold cross-validation method to make the evaluation process more accurate, avoiding overfitting or underfitting.

The weakness of the models is that the ability to capture information in pairs of long sentences is not good, causing information loss. In addition, the model's generalization is not high, making it easy to misunderstand when predicting. In the future, we propose to combine with semantic models to improve the generalization of the model [13].

Acknowledgments

This research was supported by The VNUHCM-University of Information Technology's Scientific Research Support Fund.

References

[1] B. MacCartney, C. D. Manning, Modeling Semantic Containment and Exclusion in Natural Language Inference, in: Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), Coling 2008 Organizing Committee,

Manchester, UK, 2008, pp. 521–528.

[2] N. T. Quyen, H. T. Anh, N. T. M. Huyen, N. Lien, VLSP 2021 - vnNLI Challenge: Vietnamese and English-Vietnamese Textual Entailment, VLSP 2021.

[3] I. Dagan, O. Glickman, B. Magnini, The Pascal Recognising Textual Entailment Challenge, in: Machine Learning Challenges Workshop, Springer, 2005, pp. 177–190.

[4] L. Bentivogli, P. Clark, I. Dagan, D. Giampiccolo, The Seventh Pascal Recognizing Textual Entailment Challenge, in: TAC, Citeseer, 2011.

[5] L. Bentivogli, I. Dagan, H. T. Dang, D. Giampiccolo, B. Magnini, The Fifth Pascal Recognizing Textual Entailment Challenge, in: In Proc Text Analysis Conference (TAC'09), 09.

[6] L. Bentivogli, P. Clark, I. Dagan, D. Giampiccolo, The Sixth Pascal Recognizing Textual Entailment Challenge, in: TAC, 2009.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is All You Need, in: Advances in neural information processing systems, 2017, pp. 5998–6008.

[8] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised Cross-Lingual Representation Learning at Scale, arXiv preprint arXiv:1911.02116.

[9] H. W. Chung, T. Févry, H. Tsai, M. Johnson, S. Ruder, Rethinking Embedding Coupling in Pre-Trained Language Models, arXiv preprint arXiv:2010.12821.

[10] T. Kudo, J. Richardson, Sentencepiece: A Simple and Language Independent Subword Tokenizer and Detokenizer for Neural Text Processing, arXiv preprint arXiv:1808.06226.

[11] J. Devlin, M. W. Chang, K. Lee, K. Toutanova, Bert: Pre-Training of Deep Bidirectional Transformers for Language Understanding, arXiv preprint arXiv:1810.04805.

[12] J. Howard, S. Ruder, Universal Language Model Fine-Tuning for Text Classification, arXiv preprintarXiv:1801.06146.

[13] Z. Zhang, Y. Wu, H. Zhao, Z. Li, S. Zhang, X. Zhou, X. Zhou, Semantics-Aware Bert for Language Understanding, 2020, arXiv:1909.02209.