# English-Vietnamese Cross-Lingual Paraphrase Identification Using MT-DNN

Hung Vo Tran Chi
Faculty of Information Technology
VNU-HCM - University of Science
Ho Chi Minh City, Vietnam
hung.votranchi@gmail.com

Duy Lu Anh
Faculty of Information Technology
VNU-HCM - University of Science
Ho Chi Minh City, Vietnam
lu.duy.pro@gmail.com

Nguyen Le Thanh
Faculty of Information Technology
VNU-HCM - University of Science
Ho Chi Minh City, Vietnam
lethanhnguyen.vn@gmail.com

Dien Dinh
Faculty of Information Technology
VNU-HCM - University of Science
Ho Chi Minh City, Vietnam
ddien@fit.hcmus.edu.vn

**Abstract-Paraphrase identification is a crucial task in natural language understanding, especially in cross-language information retrieval. Nowadays, Multi-Task Deep Neural Network (MT-DNN) has become a state-of-the-art method that brings outstanding results in paraphrase identification [1]. In this paper, our proposed method based on MT-DNN [2] to detect similarities between English and Vietnamese sentences, is proposed. We changed the shared layers of the original MT-DNN from original the BERT [3] to other pre-trained multi-language models such as M-BERT [3] or XLM-R [4] so that our model could work on cross-language (in our case, English and Vietnamese) information retrieval. We also added some tasks as improvements to gain better results. As a result, we gained 2.3% and 2.5% increase in evaluated accuracy and F1. The proposed method was also implemented on other language pairs such as English – German and English – French. With those implementations, we got a 1.0%/0.7% improvement for English – German and a 0.7%/0.5% increase for English – French.**

***Keywords-MT-DNN; BERT; XLM-R; English; Vietnamese; cross-language; paraphrase identification***

## I. INTRODUCTION

Paraphrase Identification (PI) is a task in Natural Language Processing (NLP) that concerns detecting a pair of text fragments that has the same meaning at different textual levels [1]. PI has a relation with the way we would quantify the number of mutual semantics between two text fragments. Measuring how two text fragments are semantically related is essential. The example in Table I was taken from the MRPC (Microsoft Research Paraphrase Corpus) dataset and was translated to Vietnamese. PI applications can be found in search engines, legal matters, or, especially, plagiarism check. A few solutions have been proposed for the mentioned problem such as the Fuzzy-based method and the BabelNet semantic network [5] or Siamese LSTM (Long-Short Term Memory) [6] but they are still limited.

TABLE I. AN EXAMPLE OF TWO ENGLISH PARAPHRASED SENTENCES AND THEIR TRANSLATION INTO VIETNAMESE

| | |
|---|---|
| **Sentence A** | Singapore is already the United States' 12th-largest trading partner, with two-way trade totaling more than $ 34 billion.<br>Singapore đã là đối tác thương mại lớn thứ 12 của Hoa Kỳ, với tổng kim ngạch thương mại hai chiều hơn 34 tỷ USD. |
| **Sentence B** | Although a small city-state, Singapore is the 12th-largest trading partner of the United States, with trade volume of $ 33.4 billion last year.<br>*Mặc dù là một thành phố nhỏ, Singapore là đối tác thương mại lớn thứ 12 của Hoa Kỳ, với kim ngạch thương mại đạt 33,4 tỷ USD vào năm ngoái.* |

With the use of transfer learning by applying the pre-trained model in machine learning models, NLP tasks in general or the PI task in particular had significant improvements in their results. Among many pre-trained models, Bidirectional Encoder Representation Transformer (BERT) [3], used to be considered as a state-of-the-art model, with impressive results in many NLP tasks. In this paper, the MT-DNN model was applied which is a combination of pre-trained models like BERT [3], Multilingual BERT (M-BERT) [3], or Cross-Language Model RoBERTa (XLM-R) [4] with Multi-Task Learning (MTL). The objective of this study is to improve the PI task between pairs of multilingual documents (namely English and Vietnamese) through applying transitional learning from a better pre-trained language model, with an MTL approach, including adding new improved tasks.

## II. LITERATURE REVIEW

### *A. Pre-Trained Model and Transfer Learning*

Transfer learning is a method of using pre-trained models and then optimize them for our purposes. A pre-trained model is a model that has been previously trained with a large dataset or with advanced methods to reduce the effort of training from scratch [8]. The model can then be further trained to fit the

Corresponding author: Hung Vo Tran Chi

actual data set or used directly in a machine learning problem. Between the pre-trained models, BERT [3] which was a state-of-the-art model in 2019, is still a solution worth considering. Specifically, BERT was constructed based on transformer (or attention's mechanism), which is a deep learning model having combined inputs and outputs where the model calculates their weights. BERT was trained on 2 main tasks: MLM (Masked Language Modeling) and NSP (Next Sentence Prediction). BERT has been applied in many applications [9, 10]. M-BERT is a single language model pre-trained from 104 languages (including English and Vietnamese). It showed abilities to not only generalize cross-lingual but also to transform scripts between many languages without having lexical overlap [3]. XLM-R was trained with MLM only, but with much more data and was based on Large-BERT. The solution used in XLM-R to deal with the burden of multilinguality was to increase model capacity by learning much more data than before. Authors in [4] showed that for the first time we can have a single large model for all languages without losing performance for any language.

### B. Paraphrase Identification Methods

Before the rising of deep learning, the most common solutions were lexical, syntactic, semantic, or hybrid techniques. After that, some sophisticated approaches have been applied by supervised or unsupervised learning.

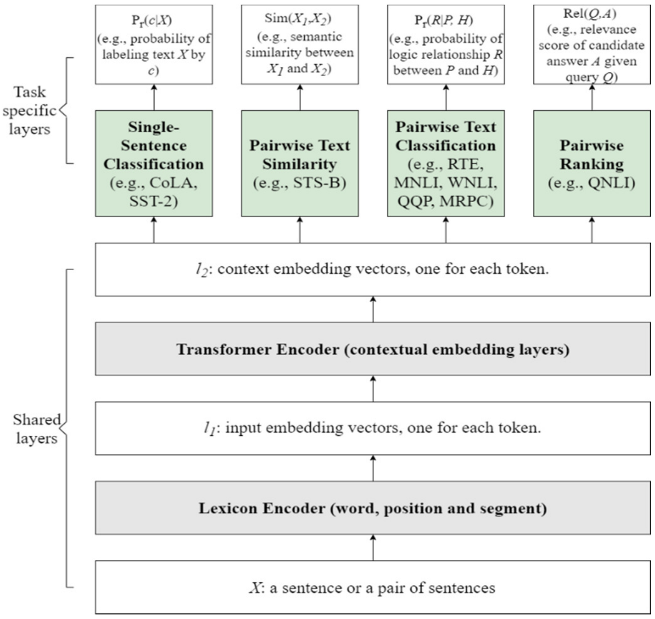
P_r(c|X)
(e.g., probability of labeling text X by c)
Sim(X_1,X_2)
(e.g., semantic similarity between X_1 and X_2)
P_r(R|P, H)
(e.g., probability of logic relationship R between P and H)
Rel(Q,A)
(e.g., relevance score of candidate answer A given query Q)
Task specific layers
Single-Sentence Classification (e.g., CoLA, SST-2)
Pairwise Text Similarity (e.g., STS-B)
Pairwise Text Classification (e.g., RTE, MNLI, WNLI, QQP, MRPC)
Pairwise Ranking (e.g., QNLI)
l_2: context embedding vectors, one for each token.
Transformer Encoder (contextual embedding layers)
Shared layers
l_1: input embedding vectors, one for each token.
Lexicon Encoder (word, position and segment)
X: a sentence or a pair of sentences

Fig. 1. The original architecture of the MT-DNN.

By using supervised learning, many traditional techniques were applied [10-13] or more advanced, using pre-trained models, specifically BERT and its improved models such as M-BERT, XLM-R, etc. MTL is an approach that achieves the generalization of results by using the inductive transfer method [14]. With MTL, we assume that knowledge gained from previous tasks could help us achieve better results when learning a new task. The main benefits from MTL are: It helps us build a platform from the previous task in order to gain better results and it does not need a big dataset with labeled data (which is hard to get) as the Deep Neural Networks (DNNs) do [15]. Furthermore, this approach creates a universal presentation without getting bias in any particular tasks. It could be achieved by regularized effects thus minimize overfitting error [15]. In the original MT-DNN architecture, we have two main layers: Shared and Task-specific. The first layer type takes the input and encodes them using Lexicon Encoder and Transformer Encoder (through the Attention mechanism) to create embedding context vectors. These vectors are shared through all tasks. After that, the Task-specific layer (depended on the tasks we use) will take these embedding vectors and process them to get the final results (e.g. with the PI problem, MT-DNN's task-specific layers will use the formula $Sim(X_1, X_2) = w_{STS}^T.x$ with $X_1, X_2$ being the input pair and $w_{STS}$ the weight matrix)

## III. THE PROPOSED METHOD

### A. Schematic Overview

The overview of the proposed method can be seen in Figure 2.

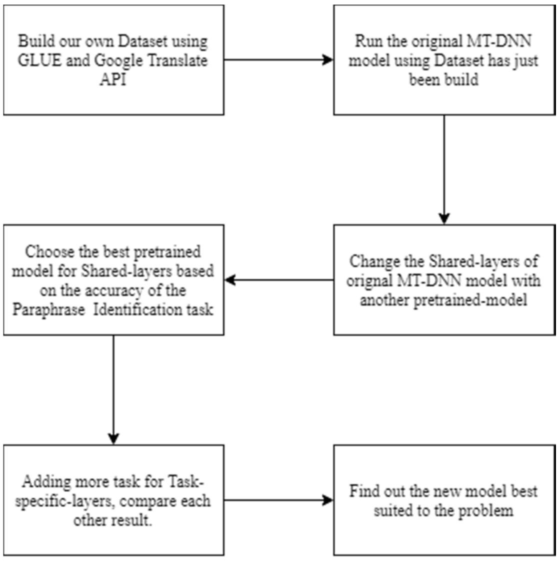
Build our own Dataset using GLUE and Google Translate API
Run the original MT-DNN model using Dataset has just been build
Change the Shared-layers of orignal MT-DNN model with another pretrained-model
Choose the best pretrained model for Shared-layers based on the accuracy of the Paraphrase Identification task
Adding more task for Task-specific-layers, compare each other result.
Find out the new model best suited to the problem

Fig. 2. The proposed solution.

### B. Updating the Pretrained Model

With the change of subject in our model from a single language to cross-language, we need to change the shared layers of the original MT-DNN in Figure 1 due to two main reasons:

- The original BERT used in MT-DNN only can be used for English but for no other languages.
- With many other multi-language pre-trained models, we can achieve not only better embedding but also have a positive effect on our results from the transfer learning method. We replaced BERT with M-BERT and XLM-R to be able to work on the English-Vietnamese language pair. M-BERT was not trained specifically for having shared presentations through languages.

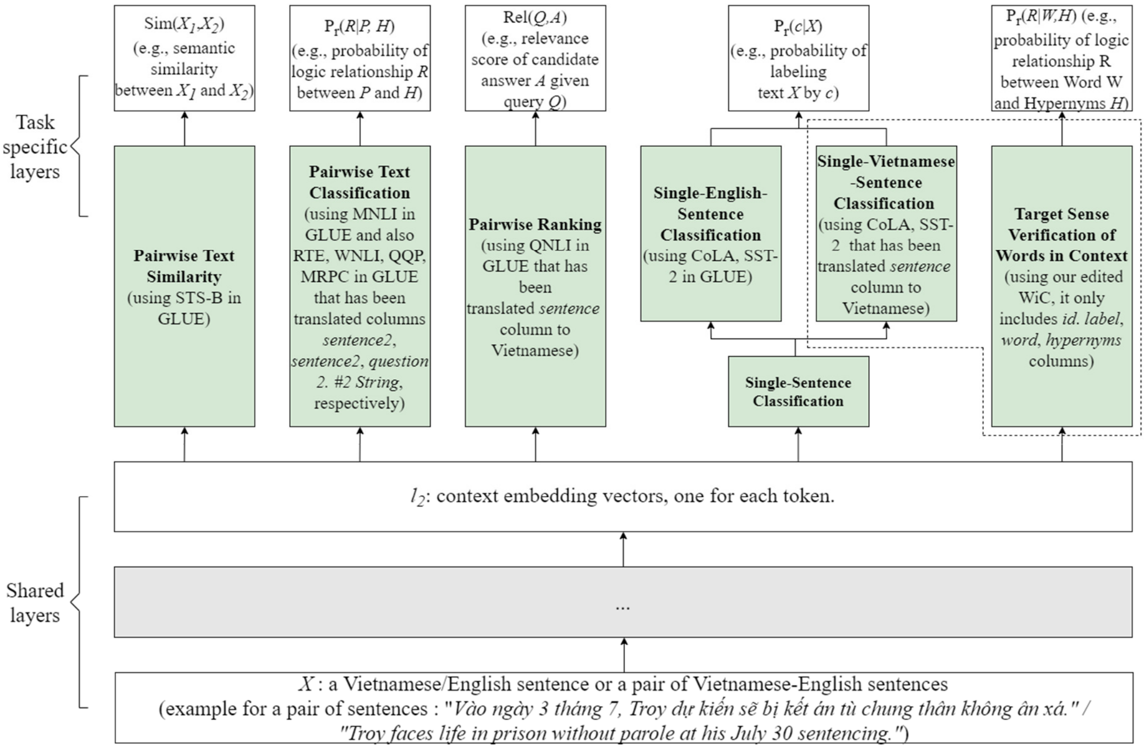
Sim($X_1$,$X_2$) (e.g., semantic similarity between $X_1$ and $X_2$)
P$_r$($R$|$P$, $H$) (e.g., probability of logic relationship $R$ between $P$ and $H$)
Rel($Q$,$A$) (e.g., relevance score of candidate answer $A$ given query $Q$)
P$_r$($c$|$X$) (e.g., probability of labeling text $X$ by $c$)
P$_r$($R$|$W$,$H$) (e.g., probability of logic relationship R between Word W and Hypernyms $H$)
Task specific layers
Pairwise Text Similarity (using STS-B in GLUE)
Pairwise Text Classification (using MNLI in GLUE and also RTE, WNLI, QQP, MRPC in GLUE that has been translated columns *sentence2*, *sentence2*, *question 2*, *#2 String*, respectively)
Pairwise Ranking (using QNLI in GLUE that has been translated *sentence* column to Vietnamese)
Single-English-Sentence Classification (using CoLA, SST-2 in GLUE)
Single-Vietnamese-Sentence Classification (using CoLA, SST-2 that has been translated *sentence* column to Vietnamese)
Target Sense Verification of Words in Context (using our edited WiC, it only includes *id. label, word, hypernyms* columns)
Single-Sentence Classification
$l_2$: context embedding vectors, one for each token.
...
Shared layers
$X$ : a Vietnamese/English sentence or a pair of Vietnamese-English sentences
(example for a pair of sentences : "*Vào ngày 3 tháng 7, Troy dự kiến sẽ bị kết án tù chung thân không ân xá.*" / "*Troy faces life in prison without parole at his July 30 sentencing.*")

Fig. 3. The proposed model.

### *C. Adding More Tasks*

#### *1) CoLA-Vie (Corpus of Linguistic Acceptability in Vietnamese)*

The existing CoLA [16] dataset was added to our model. Translating the original dataset to Vietnamese and training our model on this dataset could create a mapping in grammaticality between Vietnamese and English. The CoLA dataset in the original language (English) helped the model to learn and get many unacceptable sentences for various reasons such as pragmatical anomalies, unavailable meanings, syntactic or semantic violations, etc. [16]. By translating the original CoLA dataset to Vietnamese, we want our model to be able to learn when to accept a sentence in Vietnamese with the same reasoning, which creates mappings in how we include and exclude sentences between Vietnamese and English. With our evaluation task, which is MRPC, if two sentences in Vietnamese and English are both acceptable, they will be more likely to be similar or paraphrased.

#### *2) SST-Vie (Stanford Sentiment Treebank in Vietnamese)*

Contrary to the original SST, this dataset instead of using 5 classes (very negative, negative, neutral, positive, very positive), only uses 2 classes (positive and negative) which allows us to create easier and more accurate mapping. It does not only look for specific words such as love or hate (which can be wrong in cases of sarcasm or negative words being positive) but checks the whole sentence. With this, we can make use of pre-trained models presenting the context much better than only considering some words. The dataset was translated to Vietnamese because:

- It can help the model learn the presentation of context and vocabulary in Vietnamese in order to improve accuracy when doing tasks that involved Vietnamese.
- To create class mapping in order to make our model able to differentiate negative or positive in Vietnamese. Thus, if two sentences in Vietnamese and English are both in the same class, the chance they are paraphrased is higher.

#### *3) NER-Vie (Named Entity Recognition in Vietnamese)*

We used the datasets from CoNLL (English) [17] and VLSP (Vietnamese) [18] in order to recognize the named entities – which can help us extract information and get tags for words in a sentence - in both languages and create the mapping between them. When the classes between CoNLL and VLSP datasets were compared, it was found that they had the same labels. With that, we added this task to:

- Help our model to learn how to recognize the named entities in both Vietnamese and English so it can extract information more efficiently (which is the original purpose of this task).
- Create mappings of the named entities between two languages. If two sentences in cross-language have several same-named entities, they are more likely to be relevant than those that do not.

#### *4) PoS-Vie (Part-of-Speech Tagging in Vietnamese)*

With the same idea of NER, we tried to make our model recognize PoS tags. When checking the dataset of CoNLL and VLSP, we saw differences in quantity (49 to 35) and the number of similar tags (approximately 80%). The reasons behind this utilization are:

- Learning how to tag PoS for sentences in both Vietnamese and English could not only help our model to extract information but also to check the grammar in both languages.
- If we have more identical part-of-speech tags in both sentences, the chance of being relevant will be higher. Because the number of similar tags is little, we want our model to be able to create mappings between different tags in both languages.

### *5) WiC (Word in Corpus) [19]*

The dataset of this task included 3,832 samples with 4 attributes. Originally, this task is a binary classification problem that wants to define whether some words are the hypernyms of another word or not. We only used the labels combined with the hypernyms and the original words and removed the definitions of that word. As well as extending our vocabulary in English, we assume that if our model can learn the connection and the hierarchy between words, so it will not only be able to detect the link between a word in English and Vietnamese but also to expand to its hypernyms in English as well. Consequently, when we compare two sentences in Vietnamese and English, not only words having the same meaning are checked but also any hypernyms, if they exist in English sentences, will influence the result.

### *6) SemEval Task 8 – 2010 [20]*

This dataset includes 8,000 samples with 3 attributes. By adding this dataset, we tried to expand MNLI (Multi-Genre Natural Language Inference) and SNLI (Single Natural Language Inference). While MNLI only has some simple relationships (contradiction, neutral, and entailment), this task will specify the relationships with more details by increasing the number of relationships (19 classes in the dataset which includes 9 pairs of active or passive relationships and 1 class for others) and the positions of relationship's subjects in the sentences. By using this task in the proposed model, we are able:

- To use the benefit from NER and PoS tasks if they improve our result, especially the specified pair of subjects in sentences so we can specify the relationship between them.
- To expand the benefits from MNLI and SNLI for a more specific classification of relationships. If two words belong to the same classification, they will be more likely to be paraphrased.

TABLE II. THE CONFIGURATION OF MT-DNN ARCHITECTURE OF BOTH XLM-R AND M-BERT

| | | **num_embeddings** | **embedding_dim** | **padding_idx** | |
|---|---|---|---|---|---|
| **Embeddings** | **Word_embeddings** | 250002 | 768 | 1 | adding NormLayer: size(768,) and Dropout(p=0.1) |
| | **Position_embeddings** | 514 | 768 | 1 | |
| | **Token_type_embeddings** | 1 | 768 | 1 | |
| | | **in_features** | **out_features** | **bias** | |
| **Encoder: Include 11 linear BERT layers** | **BertSelfAttention** | 768 | 768 | TRUE | 3 linear layers for query + key + value and dropout layers (p=0.1) |
| | **BertSelfOutput** | 768 | 768 | TRUE | adding NormLayer: size(768,) and Dropout(p=0.1) |
| | **BertIntermediate** | 768 | 3072 | TRUE | |
| | **BertOutput** | 3072 | 768 | TRUE | adding NormLayer: size(768,) and Dropout(p=0.1) |
| **Pooler** | **Dense layer** | 768 | 768 | TRUE | With activation function = Tanh |
| **Scoring list: Number of layers = number of tasks** | **Linear** | 768 | number of classes of this task | TRUE | |

## IV. IMPLEMENTATION

### *A. Preparing the Dataset*

We started with the original GLUE (General Language Understanding Evaluation) dataset [21]. We used 9 out of 10 tasks in our dataset (excluding the AX dataset for format reasons). These 9 datasets represent 4 tasks that will be learned in our model. To make it more appropriate with our problems, we translated the dataset as follows:

- Single-Sentence Classification (CoLA and SST-2): We translated all to Vietnamese and used both versions (Vietnamese and English) to create Single-Vietnamese-Sentence Classification (SVSC).
- Pairwise Text Classification (MNLI, RTE, WNLI, QQP, and MRPC dataset) and Text Similarity (STS-B - Question-Answering Natural Language Inference): We kept the question in English and translated the answer to Vietnamese. The AX (diagnostic in GLUE) dataset was excluded because we should create labels by using a model trained on MNLI.
- Relevance Ranking (QLNI - Question-Answering Natural Language Inference): We kept the question in English and translated the answer to Vietnamese.

Then, NER and POS from CoNLL for English and from VLSP for Vietnamese were added to make a mapping in entities from English to Vietnamese and backwards.

In the end, we added the WiC-TSV and SemEval-Task8 2010 datasets to enhance the connection of words and relationships in English. We also translated SemEval-Task 8 2010 to Vietnamese for learning relationships in this language.

After translation, we used preprocessing scripts to encode our data with XLM-R (or M-BERT) to make input for our model.

So, our model inputs and outputs were:

- Inputs: A pair of sentences in English - Vietnamese
- Output: Whether they are paraphrased or not

Example: Inputs: There are 103 Democrats in the Assembly and 47 Republicans - Đảng Dân chủ chiếm ưu thế trong Quốc hội trong khi Đảng Cộng hòa kiểm soát Thượng viện. Output: Not paraphrased (True – in original methods they are marked as paraphrased)

### *B. Training*

For each time of traning, we selected the task we wanted to add to our original model and the hyperparameters. After many epochs, we got the checkpoint model which had best results on our MPRC dataset

### *C. Fine-Tuning*

In the fine-tuning stage, we striped the task-specific layers (included other tasks which are not MRPC) and trained our model (which has the best checkpoint from training) with the MRPC dataset again to achieve as best results as possible. The reason for our fine-tuning step is that it improves the result significantly with a little consumption of time and resources. As in the training step, we got the checkpoint model which had the best results on the MPRC dataset after many epochs.

### *D. Configuration*

We used Google Colaboratory Pro configuration with: Intel Xeon R (2 cores) 2.20 Ghz CPU, NVIDIA V100-SXM2 (16GB VRAM HBM2) GPU, 12.72GB RAM, and 150GB Hard Disk.

TABLE III. RESULTS OF VIETNAMESE-ENGLISH PAIRS

| **Task** | **No finetuning** | | **With finetuning** | |
|---|---|---|---|---|
| | **Accuracy** | **F1 Score** | **Accuracy** | **F1 Score** |
| Original Architecture (XLM-R) – Standard result | 82.8 | 87.6 | 84.3 | 88.5 |
| Original Architecture (XLM-R) + SVSC | 80.1 | 86.2 | 85.7 | 89.6 |
| Original Architecture (XLM-R) + SVSC, NER (English), POS (English) | 79.1 | 85.2 | 84.8 | 88.8 |
| Original Architecture (XLM-R) + SVSC, NER (English) | 77.9 | 84.3 | 81.6 | 86.3 |
| Original Architecture (XLM-R) + SVSC, POS (English) | 80.6 | 86.4 | 83.8 | 88.3 |
| Original Architecture (XLM-R) + SVSC, NER (English, Vietnamese), POS (English, Vietnamese) | 80.6 | 86.3 | 85.5 | 89.6 |
| Original Architecture (XLM-R) + SVSC, POS (English, Vietnamese) | 82.1 | 87.3 | 83.3 | 87.3 |
| Original Architecture (XLM-R) + SVSC, NER (English, Vietnamese) | 79.4 | 85.4 | 85.2 | 89.2 |
| Original Architecture (XLM-R) + SVSC, WiC-TSV | **83** | **88.1** | **87** | **91** |
| Original Architecture (XLM-R) + SVSC, WiC-TSV, NER (English, Vietnamese) | 80.6 | 86.7 | 86 | 89.7 |

When configuring the pre-trained models, we used M-BERT and XLM-R with the following hyperparameter specifications:

- Batch-size: 8
- Learning rate: $5\times10^{-5}$
- Gradient accumulation step: 4 (for making training more stable and faster)
- While they have a different number of parameters, both XLM-R and M-BERT have the same structure as presented in Table II.

TABLE IV. SPECIFICATIONS OF M-BERT AND XLM-R

| **Name** | **Number of layers** | **Detail name (in Huggingface)** |
|---|---|---|
| M-BERT | 177865744 | bert-base-multilingual-cased |
| XLM-R | 278060566 | xlm-roberta-base |

## V. RESULT AND DISCUSSION

The results from Table V indicate that XLM-R shows an outstanding performance when compared to M-BERT with a 9% and 6% increase in Accuracy and F1 Score respectively before the finetuning stage. After the finetuning stage, that gap expanded to 11% and 7%. That result could be explained by the way M-BERT was trained, since it did not have shared presentations through languages specifically.

TABLE V. RESULT OF THE ORIGINAL MODEL WITH M-BERT AND XLM-R

| **Dataset** | **Pretrained model** | **No finetuning** | | **With finetuning** | |
|---|---|---|---|---|---|
| | | **Accuracy** | **F1 Score** | **Accuracy** | **F1 Score** |
| All Dataset in Glue | M-BERT | 73.2 | 81.4 | 73.7 | 81.3 |
| All Dataset in Glue | XLM-R | 82.8 | 87.6 | **84.3** | **88.5** |

On the other hand, XLM-R was trained with Common Crawl [4] which helps in gaining data from languages with low resources. With the result of 84.3% in Accuracy and 88.5% in F1 score, we chose that result as our standard to compare when adding the improvement tasks. In Table III, when we look at the Accuracy and F1 score in adding CoLA-vie and SST-vie, the results were all decreased by approximately 1-4% (Accuracy) and 0.5-3% (F1 score). Only the model with the additional WiC task had a slightly higher result. After the finetuning stage, the results improved. The accuracy increased by 0.5-2.7% and the F1 score by 0.3-2.5% (with the highest result firmly belonging to the model which the WiC task). In the final model, we had 83% and 88.1% before finetuning (Accuracy and F1 score respectively). After the finetuning stage, those numbers increased to 87% and 91%. Besides this increase, adding NER-en and POS-en tasks made our model's performance decrease.

While the change of the evaluation's result after the finetuning stage was proved in the paper, adding PoS-en and PoS-vie did not improve the result. When looking at the data from CoNLL (English) and VLSP (Vietnamese), they have differences in the number of total and identical tags. Due to this, we must transfer those data to a universal PoS tag. Because of the differences in sentence structure, accurate mappings could not be created between the two languages and the context of learning was different, which incommodated our model. When we added NER-en and NER-vie, we also found that the result did not improved because the units in the vocabulary of English and Vietnamese have differences. In English, a word is a single unit while in Vietnamese, a word can be formed by many units (e.g in English we have "toothbrush" whereas in Vietnamese will be "bàn-chải"). Adding the WiC task made the difference in terms of Accuracy and F1 score. Because this task did not care about the way words are built like NER or POS but concerned more about the meaning of the sentence or the connection between words, which improved our model significantly. In Table VI, by applying the above improvements (SVSC + WiC-TSV) in German and French, we proved that adding this task not only worked for the original pair but also for other language pairs. More specifically, the result for English – German increased by 1.0% in terms of Accuracy (from 86.2% to 87.2%) and 0.7% in F1 score (from 90.1% to 90.8%). In English – French, those numbers were 0.7% (from 85.5% to 86.2%) and 0.5% (from 89.7% to 90.2%).

TABLE VI. RESULTS OF VIETNAMESE-ENGLISH PAIRS

| **Language pair** | **No finetuning** | | **With finetuning** | |
|---|---|---|---|---|
| | **Accuracy** | **F1 Score** | **Accuracy** | **F1 Score** |
| English - French | 83.5 | 88.4 | 85.5 | 89.7 |
| | 80.3 | 86.4 | **86.2** | **90.2** |
| English - German | 81.8 | 87 | 86.2 | 90.1 |
| | 82.1 | 87.3 | **87.2** | **90.8** |

For further improvements, many pre-trained models with better results could be applied to the model. Another improvement is to standardize the dataset in the translation step so it could make the mappings more accurate.

## VI. CONCLUSION

In this paper, the application of MT-DNN with transfer learning (by using a pre-trained model improved from BERT, M-BERT, and XLM-R), combined with modified MTL for the cross-language English - Vietnamese pair to achieve competitive performance in paraphrase identification task, was studied and presented. The result evaluation stage confirmed the suitability of the proposed model which includes XLM-R, SVSP, and TSV of WiC which helped obtaining better results, such as 87% and 91% in Accuracy and F1 Score. Besides the original language pair, our proposed method also had a good performance for English – German and English – French pairs with Accuracy and F1 Score of 86.2% and 90.2% and 87.2% and 90.8% respectively. Our model can be improved by changing the pre-trained models with a state-of-the-art model in the future.

## ACKNOWLEDGEMENT

This research is partially funded by the Faculty of Information Technology, University of Science, Ho Chi Minh City, Vietnam.

## REFERENCES

[1] A. Amaral, "Paraphrase Identification and Applications in Finding Answers in FAQ Databases." 2013, [Online]. Available: https://fenix.tecnico.ulisboa.pt/downloadFile/395145918749/resumo.pdf.

[2] X. Liu, P. He, W. Chen, and J. Gao, "Multi-Task Deep Neural Networks for Natural Language Understanding," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, Jul. 2019, pp. 4487–4496, https://doi.org/10.18653/v1/P19-1441.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv:1810.04805 [cs]*, May 2019, Accessed: Aug. 26, 2021. [Online]. Available: http://arxiv.org/abs/1810.04805.

[4] A. Conneau *et al.*, "Unsupervised Cross-lingual Representation Learning at Scale," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, Jul. 2020, pp. 8440–8451, https://doi.org/10.18653/v1/2020.acl-main.747.

[5] L. T. Nguyen and D. Dien, "English- Vietnamese Cross-Language Paraphrase Identification Method," in *Proceedings of the Eighth International Symposium on Information and Communication Technology*, New York, NY, USA, Dec. 2017, pp. 42–49, https://doi.org/10.1145/3155133.3155187.

[6] D. Dinh and N. Le Thanh, "English–Vietnamese cross-language paraphrase identification using hybrid feature classes," *Journal of Heuristics*, Apr. 2019, https://doi.org/10.1007/s10732-019-09411-2.

[7] M. Mohamed and M. Oussalah, "A hybrid approach for paraphrase identification based on knowledge-enriched semantic heuristics," *Language Resources and Evaluation*, vol. 54, no. 2, pp. 457–485, Jun. 2020, https://doi.org/10.1007/s10579-019-09466-4.

[8] U. Khan, K. Khan, F. Hassan, A. Siddiqui, and M. Afaq, "Towards Achieving Machine Comprehension Using Deep Learning on Non-GPU Machines," *Engineering, Technology & Applied Science Research*, vol. 9, no. 4, pp. 4423–4427, Aug. 2019, https://doi.org/10.48084/etasr.2734.

[9] S. Mandava, S. Migacz, and A. F. Florea, "Pay Attention when Required," *arXiv:2009.04534 [cs]*, May 2021, Accessed: Aug. 26, 2021. [Online]. Available: http://arxiv.org/abs/2009.04534.

[10] B. Ahmed, G. Ali, A. Hussain, A. Baseer, and J. Ahmed, "Analysis of Text Feature Extractors using Deep Learning on Fake News," *Engineering, Technology & Applied Science Research*, vol. 11, no. 2, pp. 7001–7005, Apr. 2021, https://doi.org/10.48084/etasr.4069.

[11] R. Mihalcea, C. Corley, and C. Strapparava, "Corpus-based and knowledge-based measures of text semantic similarity," in *Proceedings of the 21st national conference on Artificial intelligence*, Boston, MA, USA, Jul. 2006, vol. 1, pp. 775–780.

[12] W. Yin and H. Schütze, "Convolutional Neural Network for Paraphrase Identification," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Denver, CO, USA, May 2015, pp. 901–911, https://doi.org/10.3115/v1/N15-1091.

[13] H. Shahmohammadi, M. Dezfoulian, and M. Mansoorizadeh, "Paraphrase detection using LSTM networks and handcrafted features," *Multimedia Tools and Applications*, vol. 80, no. 4, pp. 6479–6492, Feb. 2021, https://doi.org/10.1007/s11042-020-09996-y.

[14] R. Caruana, "Multitask Learning," *Machine Learning*, vol. 28, no. 1, pp. 41–75, Jul. 1997, https://doi.org/10.1023/A:1007379606734.

[15] M. Crawshaw, "Multi-Task Learning with Deep Neural Networks: A Survey," *arXiv:2009.09796 [cs, stat]*, Sep. 2020, Accessed: Aug. 26, 2021. [Online]. Available: http://arxiv.org/abs/2009.09796.

[16] A. Warstadt, A. Singh, and S. R. Bowman, "Neural Network Acceptability Judgments," *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 625–641, Mar. 2019, https://doi.org/10.1162/tacl_a_00290.

[17] E. F. Tjong Kim Sang and F. De Meulder, "Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition," in *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, 2003, pp. 142–147.

[18] H. T. M. Nguyen, Q. T. Ngo, L. X. Vu, V. M. Tran, and H. T. T. Nguyen, "VLSP Shared Task: Named Entity Recognition," *Journal of Computer Science and Cybernetics*, vol. 34, no. 4, pp. 283–294, 2018, https://doi.org/10.15625/1813-9663/34/4/13161.

[19] A. Breit, A. Revenko, K. Rezaee, M. T. Pilehvar, and J. Camacho-Collados, "WiC-TSV: An Evaluation Benchmark for Target Sense Verification of Words in Context," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, Online, Apr. 2021, pp. 1635–1645.

[20] I. Hendrickx *et al.*, "SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations between Pairs of Nominals," in *Proceedings of the 5th International Workshop on Semantic Evaluation*, Uppsala, Sweden, Jul. 2010, pp. 33–38.

[21] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman, "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding," in *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, Brussels, Belgium, Nov. 2018, pp. 353–355, https://doi.org/ 10.18653/v1/W18-5446.