Knowledge-Enhanced Vietnamese Paraphrase Identification

Abstract. Paraphrase Identification (PI) is a fundamental task in natural language processing (NLP) that determines whether a pair of sentences convey the same meaning. This task plays a crucial role in various applications such as machine translation, computer-assisted translation, and question answering. While extensive research has been conducted in English and several other languages, Vietnamese PI remains relatively underexplored. Pre-trained language models (PLMs) have become the standard approach for tackling language understanding tasks, including PI. However, despite their rapid advancement, these models are still limited in their capacity to capture external knowledge. In this study, we propose a novel architecture that integrates pretrained language models with external knowledge for Vietnamese PI. Experimental results show that our approach, using mBERT as a base model, achieved an F1-score of 95.59% on a combined corpus consisting of vnPara and an additional 1498 sentence pairs enriched with diverse entities. This demonstrates the effectiveness of our model in distinguishing between different entities.

Keywords: paraphrase identification · Vietnamese · pre-trained model · external knowledge · Wikipedia2Vec

1 Introduction

Understanding whether two sentences convey the same meaning is a fundamental task in natural language understanding. PI, a task that determines whether two text fragments express the same idea, is crucial for downstream tasks, including question answering, text retrieval, text summarization, and semantic search.

In low-resource languages such as Vietnamese, accurately determining paraphrases is still a challenge due to the variety in lexical and data limitations. The terms often vary in lexical and structural features, making them challenging to identify just on the basis of surface-level semantic cues. Furthermore, detecting paraphrases from textual entailment or contradiction requires a deep semantic understanding. Recent advances in the Vietnamese PI task [1, 2] often leverage robust pretrained models, such as BERT [3] and its variants. Despite their success, these models often struggle with understanding the distinction between different named entities, reflecting the limitation of external knowledge.

In addition, Vietnamese is an *analytic, non-inflectional* language in which grammatical relations are conveyed primarily through word order and function words rather than morphology. For example, the sentence "Tôi ăn cơm" (I eat rice) differs from "Cơm ăn tôi" (Rice eats me) only by word order, since words do not inflect for case. Orthographically, words are written as sequences of syllables separated by spaces, making *word segmentation* a non-trivial preprocessing step; for instance, the single word "học sinh" (student) is written as two syllables "học" and "sinh". Furthermore, Vietnamese is a *tonal language* with six tones, where tone and context crucially determine meaning: "ma", "má", and "mà" are distinct words depending on tone. These linguistic properties, combined with frequent synonymy and flexible patterns of expression, pose distinctive challenges for paraphrase identification.

In this study, we propose a PI model that enhances a pretrained language model with external knowledge integration. Specifically, we incorporate named entity information from Wikipedia by training Wikipedia2Vec [4] on the latest Vietnamese Wikipedia dump to obtain entity embeddings. Our model is trained and evaluated on the vnPara corpus and an additional 1200 sentence pairs enriched with diverse entities. The experimental results indicate that our model outperforms the strong baseline, which includes BERT and its variants.

The rest of this paper is organized as follows. Section 2 reviews related works. Section 3 describes the corpora used in our study. Section 4 presents our proposed methodology. Section 5 reports the experimental setup, results, and discussion. Finally, Section 6 concludes the paper and outlines future directions.

2 Related Works

Transformer's architectural breakthrough led to the emergence of BERT [3] and its variants such as ALBERT [5], ELECTRA [6]... Soon, they became the state-of-the-art methods for PI. From there, the trend is to augment BERT or other similar architectures with additional structures to produce better results. One of those works is of Arase and Tsujii [7] where they introduced a transfer fine-tuning method that incorporates phrasal paraphrases into BERT. Xu *et al.* [8] proposed the Lexical, Syntactic, and Sentential Encodings (LSSE) framework for the PI, fusing BERT-based representations with information from dependency parses. Yu *et al.* [9] proposed a multi-granularity fusion model (WSTM) based on WoBERT (a word-based Chinese BERT model) that captures semantic representations at the Chinese character, word, and sentence levels with soft-attention alignment. Recently, Shi *et al.* [10] formally established a connection between PI and natural language inference (NLI) with the novel framework PI2NLI.

In parallel, researchers also considered the knowledge problem in PI. There are embedding models with knowledge integration in mind such as Wikipedia2Vec [4], E-BERT [11], etc. Our method of PI with knowledge is primarily inspired by Knowing, a model proposed by Wang *et al.* [12]. Knowing enriches BERT-like models with Wikipedia knowledge by retrieving relevant outlines with BM25 [13] and making predictions based on attention and gated mechanisms.

Unfortunately, all research above is only well-experimented on English, Chinese, or other languages except Vietnamese. As far as we studied, Bach *et al.* [14] was the first to plunge into the Vietnamese PI. Their method utilized string similarity measures, as they used 9 of them together with 7 representations of different abstraction levels of two input sentences. Using strong machine learning methods, they trained a model based on formed features. In the end, the highest result was obtained from the SVM classifier. Bach *et al.* [14] also constructed vnPara, the first Vietnamese paraphrase corpus containing 3000 sentence pairs, which they used to evaluate on. The following work is done by Nguyen-Son *et al.* [22]. They presented a method using matching duplicate phrases and similar words. First, words in each sentence are separated using a Vietnamese WordNet [23]. After removal of stop words, they proposed SimVN similarity to match words. Then, SimMat metric is calculated for prediction. They also contributed a more diverse but unbalanced Vietnamese News Paraphrase Corpus (VNPC) with 3134 sentence pairs. As Dinh and Thanh [1] pointed out, these methods still relied heavily on the string-based approach. So they developed a hybrid Siamese architecture combining PhoBERT [20], with WordNet and POS information. This hybrid method achieved strong performance, outperforming all previous models. Another usage of BERT can be seen in the work of Phan *et al.* [2], where they developed a Vietnamese Sentence BERT model for sentence embeddings, using PhoBERT as their main transformer for token embeddings. They then fine-tuned a Siamese Sentence-BERT architecture and tested it on vnPara.

Each of these attempts has contributed greatly to improving the performance on the Vietnamese PI task. However, as far as we know, none of them have explicitly integrated external knowledge into the process. In contrast, our work, inspired by Knowing [12], will systematically fuse external knowledge into the PI model, enabling the model to reason over richer semantic meanings.

3 Corpora

This section describes the corpora used in our experiments. We also present some investigations on their characteristics.

3.1 vnPara

This is a corpus designed specifically for the Vietnamese PI task. Bach *et al.* [14] have worked to annotate the corpus, resulting in 3000 sentence pairs with nearly balanced labels. As pointed out by Dinh and Thanh [1], main source of vnPara came from various texts collected from news websites such as news dantri.com.vn, vnexpress.net, and thanhnien.com.vn...

3.2 Augmented vnPara

This is an augmented version of vnPara. We annotated about 1500 new sentence pairs, which were chosen for their enriched named entity mentions. The main motive of such a movement is the lack of named entities in the vnPara corpus, which prevented our exploration of model performance. We ensured the corpus is labeled evenly with votes from 2 annotators. The source to construct the augmentation is Vietnamese Wikipedia, which contains a great amount of entities.

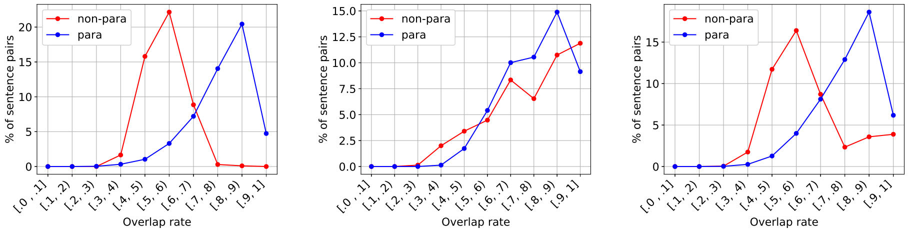
3.3 Some statistics on the corpora

For further clarification, vnPara consists of exactly 3083 sentence pairs, 1577 of which are paraphrase pairs, the other 1506 are non-paraphrase pairs. The augmented part we created contains evenly 749 paraphrase pairs and 749 non-paraphrase pairs. To monitor the improvement over named entities in the corpora, we counted entities and calculated the average number of entities per sentence (e.p.s.) as in Table 1.

Table 1. Entity count on each corpus

Corpus	**Total sentences**	**Total entities**	**e.p.s.**
vnPara	6166	6075	0.9852
Additional data	2996	7524	2.5113
Augmented vnPara	9162	13599	1.4843

Dinh and Thanh [1] also noticed some important special cases: non-trivial paraphrase, and non-trivial non-paraphrase. The chosen measure for triviality is the Jaccard index, which indicates how much overlapping characters are from two sentences (overlap rate). Inspired by this piece, we conducted an investigation on vnPara and our augmented corpus (see Fig. 1).



(a) Distribution of overlap rate in vnPara corpus

(b) Distribution of overlap rate in additional corpus

(c) Distribution of overlap rate in augmented corpus

Fig. 1. Distribution of sentences across overlap rate. The x-axis is the overlap rate measured by Jaccard index divided into 10 intervals. The y-axis is the percentage of sentence pairs that is non-paraphrase (in red) or paraphrase (in blue).

Fig. 1a shows that the vnPara corpus consists of many trivial paraphrases, and non-paraphrases are distributed around the center (40% - 70% overlap rate) indicating that there are few non-trivial paraphrases. Our augmentation will try to balance the amount of trivial cases in the corpus. In Fig. 1b, although our augmentation has many non-trivial non-paraphrases, collecting non-trivial paraphrases is still a great challenge as we get a high percentage of trivial paraphrases. This results in a slight shift to the right in non-paraphrase distribution, but also a spiking number of paraphrases at the high overlap rate in Fig. 1c.

4 Methodology

This section describes our proposed method. For clarification, we define our notation here. Given a sentence pair $(S_i^1, S_i^2) \in \mathcal{X}$ and an external knowledge base $\mathcal{B}$, our goal is to learn a function $F(S_i^1, S_i^2, \mathcal{B}) \rightarrow \{0, 1\}$ to determine whether two sentences S_i^1 and S_i^2 have the same or different meaning, where $\mathcal{X} = \{(S_1^1, S_1^2), (S_2^1, S_2^2), \ldots, (S_n^1, S_n^2)\}$ denotes the training dataset, and the corresponding labels are $\mathcal{Y} = \{y_1, y_2, \ldots, y_n\}$. We adopted a common label convention: 0 means that S_i^1 and S_i^2 are not paraphrases, and 1 means otherwise.

4.1 Overview

To integrate external knowledge into BERT-like models, we proposed a PI model consisting of 6 key modules: BERT-like PLM, Entity Extractor, Wikipedia2Vec, Entity Attention Encoder, Entity Interaction Encoder, and Classifier (see Fig. 2).

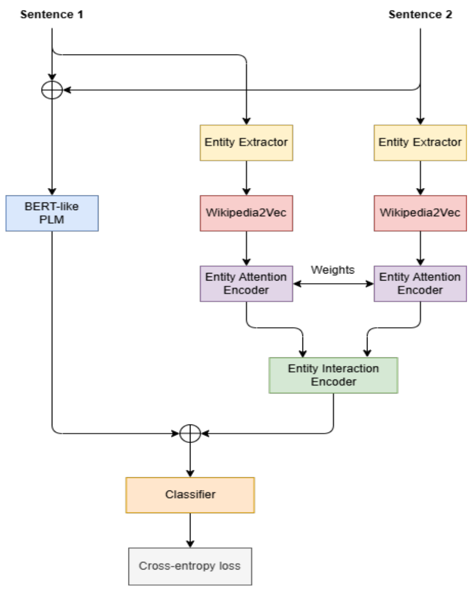


Fig. 2. Schema of the proposed knowledge-enhanced model. The schema highlights the interplay between 6 key modules: BERT-like PLM, Entity Extractor, Wikipedia2Vec, Entity Attention Encoder, Entity Interaction Encoder, and Classifier.

The base prediction module employs a PLM to encode sentence pairs. However, this base predictor overlooks the contribution of external knowledge. To address this limitation, we introduce the Entity Extractor, which identifies entities from each sentence. These extracted entities are then individually embedded using Wikipedia2Vec. The Entity Attention Encoder aggregates the entity embeddings of each sentence, assigning attention weights to the entities to learn the importance between them. The Entity Interaction Encoder simulates the semantic relationships between entities across the two sentences. Finally, the representations from the BERT-like PLM and the Entity Interaction Encoder are concatenated and passed through the Classifier for final prediction.

4.2 BERT-like Model

BERT-like models such as PhoBERT [20] are PLMs designed to capture contextual semantic information. Given two sentences, their combined embedding is computed as follows:

$$\mathbf{e}_S = \mathrm{BERT}(S) \tag{1}$$

where $S = S_i^1 \oplus S_i^2$ denotes the concatenation of the two sentences. This embedding $\mathbf{e}_S$ captures the contextual information of the sentence pair.

4.3 Entity Extractor

We employ the `ner-vietnamese-electra-base` model developed by NlpHUST based on vELECTRA [16] to extract named entities from input sentences. The set of entities extracted from sentence S_i^1 and S_i^2 is denoted as:

$$\mathcal{E}_i^1 = \mathrm{NER}(S_i^1) \tag{2}$$

$$\mathcal{E}_i^2 = \mathrm{NER}(S_i^2) \tag{3}$$

Since the extracted entities will be embedded using Wikipedia2Vec in the subsequent module, we filter out entities that do not appear in the set of Wikipedia entities, denoted by $\mathcal{E}_{\text{wiki}}$. The filtered entity sets are defined as follows:

$$\tilde{\mathcal{E}}_i^1 = \{e \in \mathcal{E}_i^1 \mid e \in \mathcal{E}_{\text{wiki}}\} \tag{4}$$

$$\tilde{\mathcal{E}}_i^2 = \{e \in \mathcal{E}_i^2 \mid e \in \mathcal{E}_{\text{wiki}}\} \tag{5}$$

4.4 Wikipedia2Vec

Wikipedia2Vec [4] is a toolkit that jointly learns vector representations of words and entities from Wikipedia, allowing them to reside in a shared continuous vector space. This enables semantic similarity computations across both lexical and encyclopedic concepts.

Given entity $e_j^1 \in \tilde{\mathcal{E}}_i^1$ and $e_j^2 \in \tilde{\mathcal{E}}_i^2$, their corresponding embeddings are:

$$\mathbf{e}_j^1 = \mathrm{Wikipedia2Vec}(e_j^1) \in \mathbb{R}^{1\times d} \tag{6}$$

$$\mathbf{e}_j^2 = \mathrm{Wikipedia2Vec}(e_j^2) \in \mathbb{R}^{1\times d} \tag{7}$$

where d is the embedding dimension in Wikipedia2Vec space.

All m embeddings $\mathbf{e}_j^1$ are stacked along the first dimension to form a matrix $\mathbf{E}_1 \in \mathbb{R}^{m \times d}$. Similarly, all n embeddings $\mathbf{e}_j^2$ are stacked into $\mathbf{E}_2 \in \mathbb{R}^{n \times d}$.

4.5 Entity Attention Encoder

The Entity Attention Encoder aggregates entity embeddings for each sentence by assigning higher attention weights to more informative entities. To compute the importance of each entity embedding, we employed a reduced form of additive attention introduced by Bahdanau, Cho, and Bengio [15] as follows:

$$\boldsymbol{\alpha}_1 = \text{softmax}\left(\mathbf{v}^\top \tanh(\mathbf{W}_1 \mathbf{E}_1^\top)\right) \in \mathbb{R}^{1 \times m} \tag{8}$$

$$\boldsymbol{\alpha}_2 = \text{softmax}\left(\mathbf{v}^\top \tanh(\mathbf{W}_1 \mathbf{E}_2^\top)\right) \in \mathbb{R}^{1 \times n} \tag{9}$$

where $\mathbf{W}_1 \in \mathbb{R}^{d_a \times d}$ and $\mathbf{v} \in \mathbb{R}^{d_a}$ are learnable parameters, d_a is the hidden dimension for attention calculation.

The aggregated embeddings $\mathbf{e}_1$ and $\mathbf{e}_2$ for two sentences are obtained via weighted sums:

$$\mathbf{e}_1 = \boldsymbol{\alpha}_1 \mathbf{E}_1 \in \mathbb{R}^{1 \times d} \tag{10}$$

$$\mathbf{e}_2 = \boldsymbol{\alpha}_2 \mathbf{E}_2 \in \mathbb{R}^{1 \times d} \tag{11}$$

4.6 Entity Interaction Encoder

The Entity Interaction Encoder is designed to capture semantic interactions between the entity representations of the two input sentences. It explicitly models both similarities and differences by combining multiple element-wise operations.

We can model the interaction between these embeddings with a simple concatenation operation proposed by Chen *et al.* [17] between $\mathbf{e}_1$ and $\mathbf{e}_2$ from the previous step. In particular, we construct a joint interaction vector as follows:

$$\mathbf{s} = \mathbf{W}\left[\mathbf{e}_1 \oplus \mathbf{e}_2 \oplus |\mathbf{e}_1 - \mathbf{e}_2| \oplus \mathbf{e}_1 \odot \mathbf{e}_2\right]^\top \in \mathbb{R}^{d'} \tag{12}$$

where $\oplus$ denotes vector concatenation, $|\cdot|$ is the element-wise absolute difference, $\odot$ denotes element-wise multiplication, and $\mathbf{W} \in \mathbb{R}^{d' \times 4d}$ is a learnable weight matrix that projects the concatenated vector into a shared semantic space of dimension d'.

4.7 Classifier

The Classifier is a multi-layer perceptron (MLP) which is responsible for producing the final output that is ready to be converted into probabilities. It takes as input the concatenation of the sentence-pair embedding vector $\mathbf{e}_S$ and the aggregated entity interaction vector $\mathbf{s}$:

$$o = \text{Classifier}(\mathbf{e}_S \oplus \mathbf{s}) \tag{13}$$

5 Experiment

5.1 Experiment Setup

Evaluation Mehthod. We'll substitute multiple PLMs such as PhoBERT [20], mBERT [3], XML-R [18], and ViDeBERTa [21] in place of the embedding model. The evaluation is conducted with metrics such as accuracy and F1-score. The system will be trained on our augmented vnPara, whose splits are as follows:

- Training set: 3263 sentence pairs.
- Validation set: 409 sentence pairs.
- Testing set: 909 sentence pairs.

Configurations of Optimizer. Our proposed model will be trained with AdamW [19] optimizer. We'll try different combinations of hyper-parameters. They are restricted to certain range of values as follows:

- Learning rate: [1×10^{-5}, 2×10^{-5}, 3×10^{-5}, 4×10^{-5}, 5×10^{-5}]
- Weight decay: [0.01, 0.02, 0.05, 0.1]
- Batch size: [8, 16, 32]
- Epochs: [2, 3, 4, 5, 6, 7, 8, 9, 10]

For more control, we employed early stopping with patience set to 3 and minimum change of 0.01. The loss here is the standard cross-entropy.

Configurations of MLP. We used the same MLP across all models. A 4-layer feed-forward network with decreasing dimensions from 2048, 1024, 256 to 2 with ReLU activations and normalizations in between each layer is utilized.

5.2 Experimental Results

In this section, we'll present experimental results of PI using different baseline PLMs. The most effective set of hyper-parameters for each tested model can be seen in Table 2 below.

Table 2. Effective hyper-parameters per model

Model	**Learning rate**	**Weight decay**	**Batch size**	**Epochs**
Baseline				
PhoBERT-base	4×10^{-5}	0.02	32	6
ViDeBERTa-xsmall	3×10^{-5}	0.02	32	8
ViDeBERTa-base	5×10^{-5}	0.05	8	8
mBERT	5×10^{-5}	0.01	32	6
XLM-R	5×10^{-5}	0.02	32	6
Ours with				
PhoBERT-base	2×10^{-5}	0.01	16	6
ViDeBERTa-xsmall	2×10^{-5}	0.05	8	10
ViDeBERTa-base	2×10^{-5}	0.01	8	6
mBERT	2×10^{-5}	0.01	16	10
XLM-R	2×10^{-5}	0.1	16	8

Table 3 demonstrates the testing results such as accuracy, precision, recall, and F1-score obtained from evaluating trained models on the testing set. The highest value is highlighted in boldface.

Table 3. Summarized evaluation results with different baselines

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**
Baseline				
PhoBERT-base	0.7844	0.7905	0.7844	0.7840
ViDeBERTa-xsmall	0.6744	0.7032	0.6744	0.6814
ViDeBERTa-base	0.7140	0.7136	0.7140	0.7137
mBERT	0.7316	0.7415	0.7316	0.7314
XLM-R	0.6733	0.8036	0.6733	0.6895
Ours with				
PhoBERT-base	0.9197	0.9205	0.9197	0.9198
ViDeBERTa-xsmall	0.9032	0.9071	0.9032	0.9030
ViDeBERTa-base	0.9285	0.9285	0.9285	0.9285
mBERT	**0.9560**	**0.9572**	**0.9560**	**0.9559**
XLM-R	0.9461	0.9462	0.9461	0.9461

5.3 Discussion

Evidenced by Table 3, our proposed method unequivocally achieved significant enhancements for the baseline PLMs. Improvements are yielded across all metrics compared to their respective baselines.

However, the amount of enhancements is different for different architectures. Let's discuss monolingual models (PhoBERT and ViDeBERTa) first. When integrating with PhoBERT-base, we got a relatively high F1-score of 91.98%, a substantial gain of 13.58% from 78.40% at baseline. We observed even stronger improvements with ViDeBERTa as ViDeBERTa-xsmall achieved 90.30% (22.16% gain) and ViDeBERTa-base achieved 92.85% (21.48% gain) in F1-score. Notably, the highest improvement came from multilingual models (mBERT and XLM-R). Our model with XLM-R gained the most with 94.61% in F1-score (improved by 25.66%). But the highest results belonged to mBERT, which reached 95.59% in F1-score (22.45% gain). We observed that our approach yields better performance for multilingual models than monolingual models.

Despite good results, there are still persistent limitations with our method. The entity representations did not strongly expand their contexts into the whole sentence. The only occurrence of the interaction between entities and sentences lies in the feed-forward network near the output layer of our model. It wasted the positional information of the entities and the contexts around them.

For detecting entities, a NER-specific PLM was employed. The accuracy of finding entities relies heavily on the accuracy of the NER model. Even if the NER model was decent at its job, the Wikipedia2Vec model could still cause problems. In detail, all entities that are not appearing in Wikipedia2Vec space are discarded. However, Wikipedia2Vec is too sensitive to the form of the words. Rewriting a name with different forms can lead to non-existing entities.

Another notable limitation came from the corpora. We augmented vnPara with additional data, but as Fig. 1b shows, the non-trivial paraphrases are still very limited since they cluster at the high overlap rate, producing lots of similar sentences that are paraphrases. Our final augmentation hardly shifts this fact.

6 Conclusion

In this paper, we have explored the knowledge problem in Vietnamese PI. We proposed a model utilizing a NER model and Wikipedia2Vec to extract knowledge through named entities. We trained Wikipedia2Vec on the latest Vietnamese Wikipedia dump for this goal. Our model aggregates the entity embeddings with an additive attention mechanism. The entities are then compared through a simple concatenate operation to produce the final prediction. In the end, we have successfully enhanced PLMs such as PhoBERT, ViDeBERTa, mBERT, and XLM-R, achieving a 13%–26% gain in F1-score on our augmented vnPara corpus. Most notably, our model with mBERT as a base model attained the highest F1-score of 95.59%.

Despite its performance, our model still faces many problems that warrant further focus in the future. To reduce the dependency on the NER model, we plan to replace it with other extraction techniques such as retrieval with BM25 or vectorized databases. We also plan to solve the Wikipedia2Vec misalignment by using BERT-like models themselves to encode the knowledge. Finally, instead of only Wikipedia, we intend to incorporate other knowledge bases such as WordNet into our method in the near future.

Acknowledgments. We would like to thank Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for the support of time and facilities for this study.

References

1. Dinh, D., Thanh, N.L.: Vietnamese Sentence Paraphrase Identification using Pre-trained Model and Linguistic Knowledge. International Journal of Advanced Computer Science and Applications **12**(8) (2021). https://doi.org/10.14569/IJACSA.2021.0120891
2. Phan, Q.L., Doan, T.H.P., Le, N.H., Tran, N.B.D., Huynh, T.N.: Vietnamese Sentence Paraphrase Identification Using Sentence-BERT and PhoBERT. In: Nguyen, N.-T., Dao, N.-N., Pham, Q.-D., Le, H.A. (eds.) Intelligence of Things: Technologies and Applications, pp. 416–423. Springer International Publishing, Cham (2022)
3. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Burstein, J., Doran, C., Solorio, T. (eds.) 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), pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (2019). https://doi.org/10.18653/v1/N19-1423
4. Yamada, I., Asai, A., Sakuma, J., Shindo, H., Takeda, H., Takefuji, Y., Matsumoto, Y.: Wikipedia2Vec: An Efficient Toolkit for Learning and Visualizing the Embeddings of Words and Entities from Wikipedia. In: Liu, Q., Schlangen, D. (eds.) Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 23–30. Association for Computational Linguistics, Online (2020). https://doi.org/10.18653/v1/2020.emnlp-demos.4
5. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R.: ALBERT: A Lite BERT for Self-supervised Learning of Language Representations, (2020). arXiv: 1909.11942 [cs.CL]. https://arxiv.org/abs/1909.11942.
6. Clark, K., Luong, M.-T., Le, Q.V., Manning, C.D.: ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators, (2020). arXiv: 2003.10555 [cs.CL]. https://arxiv.org/abs/2003.10555.
7. Arase, Y., Tsujii, J.: Transfer fine-tuning of BERT with phrasal paraphrases. Computer Speech & Language **66**, 101164 (2021). https://doi.org/10.1016/j.csl.2020.101164
8. Xu, S., Shen, X., Fukumoto, F., Li, J., Suzuki, Y., Nishizaki, H.: Paraphrase Identification with Lexical, Syntactic and Sentential Encodings. Applied Sciences **10**(12) (2020). https://doi.org/10.3390/app10124144
9. Yu, H., Pan, W., Fan, X., Li, H.: Multi-Granularity Fusion Text Semantic Matching Based on WoBERT. In: Calzolari, N., Kan, M.-Y., Hoste, V., Lenci, A., Sakti, S., Xue, N. (eds.) Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), pp. 11766–11775. ELRA and ICCL, Torino, Italia (2024). https://aclanthology.org/2024.lrec-main.1027/
10. Shi, N., Hauer, B., Riley, J., Kondrak, G.: Paraphrase Identification via Textual Inference. In: *SEM@NAACL, pp. 133–141 (2024). https://doi.org/10.18653/v1/2024.starsem-1.11
11. Poerner, N., Waltinger, U., Schütze, H.: E-BERT: Efficient-Yet-Effective Entity Embeddings for BERT. In: Cohn, T., He, Y., Liu, Y. (eds.) Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 803–818. Association for Computational Linguistics, Online (2020). https://doi.org/10.18653/v1/2020.findings-emnlp.71

12. Wang, H., Ma, F., Wang, Y., Gao, J.: Knowledge-Guided Paraphrase Identification. In: Moens, M.-F., Huang, X., Specia, L., Yih, S.W.-t. (eds.) Findings of the Association for Computational Linguistics: EMNLP 2021, pp. 843–853. Association for Computational Linguistics, Punta Cana, Dominican Republic (2021). https://doi.org/10.18653/v1/2021.findings-emnlp.72
13. Robertson, S., Zaragoza, H.: The Probabilistic Relevance Framework: BM25 and Beyond. Found. Trends Inf. Retr. **3**(4), 333–389 (2009). https://doi.org/10.1561/1500000019
14. Bach, N.X., Oanh, T.T., Hai, N.T., Phuong, T.M.: Paraphrase Identification in Vietnamese Documents. In: 2015 Seventh International Conference on Knowledge and Systems Engineering (KSE), pp. 174–179 (2015). https://doi.org/10.1109/KSE.2015.37
15. Bahdanau, D., Cho, K., Bengio, Y.: Neural Machine Translation by Jointly Learning to Align and Translate, (2016). arXiv: 1409.0473 [cs.CL]. https://arxiv.org/abs/1409.0473.
16. Bui, T.V., Tran, T.O., Le-Hong, P.: Improving Sequence Tagging for Vietnamese Text using Transformer-based Neural Models. In: Nguyen, M.L., Luong, M.C., Song, S. (eds.) Proceedings of the 34th Pacific Asia Conference on Language, Information and Computation, pp. 13–20. Association for Computational Linguistics, Hanoi, Vietnam (2020)
17. Chen, Q., Zhu, X., Ling, Z.-H., Wei, S., Jiang, H., Inkpen, D.: Enhanced LSTM for Natural Language Inference. In: Barzilay, R., Kan, M.-Y. (eds.) Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1657–1668. Association for Computational Linguistics, Vancouver, Canada (2017). https://doi.org/10.18653/v1/P17-1152
18. Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., Stoyanov, V.: Unsupervised Cross-lingual Representation Learning at Scale. In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J. (eds.) Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 8440–8451. Association for Computational Linguistics, Online (2020). https://doi.org/10.18653/v1/2020.acl-main.747
19. Loshchilov, I., Hutter, F.: Decoupled Weight Decay Regularization, (2019). arXiv: 1711.05101 [cs.LG]. https://arxiv.org/abs/1711.05101.
20. Nguyen, D.Q., Tuan Nguyen, A.: PhoBERT: Pre-trained language models for Vietnamese. In: Cohn, T., He, Y., Liu, Y. (eds.) Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 1037–1042. Association for Computational Linguistics, Online (2020). https://doi.org/10.18653/v1/2020.findings-emnlp.92
21. Tran, C.D., Pham, N.H., Nguyen, A.T., Hy, T.S., Vu, T.: ViDeBERTa: A powerful pre-trained language model for Vietnamese. In: Vlachos, A., Augenstein, I. (eds.) Findings of the Association for Computational Linguistics: EACL 2023, pp. 1071–1078. Association for Computational Linguistics, Dubrovnik, Croatia (2023). https://doi.org/10.18653/v1/2023.findings-eacl.79
22. Nguyen-Son, H.-Q., Tran, N.-P., Pham, N.-V., Tran, M.-T., Echizen, I.: Vietnamese Paraphrase Identification Using Matching Duplicate Phrases and Similar Words. In: Dang, T.K., Küng, J., Wagner, R., Thoai, N., Takizawa, M. (eds.) Future Data and Security Engineering, pp. 172–182. Springer International Publishing, Cham (2018)
23. Miller, G.A.: WordNet: a lexical database for English. Commun. ACM **38**(11), 39–41 (1995). https://doi.org/10.1145/219717.219748