Vietnam Journal of Science and Technology ***64***(2) (2026) 363–374
DOI: https://doi.org/10.15625/2525-2518/21463

Adapting knowledge graph embedding for neural machine translation

Nha Tran[1], Tri Le[1], Nam Nguyen[1], Long Nguyen[2,*]

[1]*Faculty of Information Technology, Ho Chi Minh City University of Education, No. 280 An Duong Vuong, Cho Quan ward, Ho Chi Minh city 749000, Viet Nam*
[2]*Faculty of Information Technology, University of Science, No. 227 Nguyen Van Cu, Cho Quan ward, Ho Chi Minh city 749000, Viet Nam*

[*]Email: *nhblong@fit.hcmus.edu.vn*

Received: 04 September 2024; Accepted for publication: 24 October 2024

Abstract. In the era of deep learning and the rise of Sequence to Sequence architecture, Neural Machine Translation (NMT) has significantly improved in efficiency and performance. However, NMT models still face challenges due to the need for large amounts of training data, particularly for language pairs with insufficient resources, resulting in the corpus sparsity problem. This paper explores the integration of Knowledge Graphs (KGs) into NMT models to enhance the translation of rare and out-of-vocabulary (OOV) words. Specifically, our method, KGE-NMT, leverages structured knowledge from KGs to improve the semantic representation of entities in sentences, thereby enhancing the overall translation quality. Experimental results on English-Vietnamese and English-German language pairs (i.e., IWSLT datasets) show that our KGE-NMT model significantly outperforms baseline models, confirming the benefits of incorporating external knowledge into the machine translation process.

Keywords: neural machine translation, knowledge graph embedding, graph embedding.

Classification numbers: 4.8.4, 4.7.4.

1. INTRODUCTION

With the development of deep learning and the advent of Sequence to Sequence architecture, Neural Network Machine Translation (NMT) has achieved significant improvements in efficiency and performance [1-3]. Although promising, these NMT models face challenges due to the need for large amounts of training data, while many language pairs lack sufficient resources, leading to the corpus sparsity problem. The issue of data sparsity in Machine Translation (MT), primarily caused by insufficient training data, results in poor translation of rare and out-of-vocabulary (OOV) words. This problem occurs when terms appear infrequently or are absent during the training phase. Consequently, several approaches have been developed to improve the training of sparse datasets and enhance translation performance on these datasets. Earlier studies attempted to address this by focusing on translating entities within the translation process, as the accurate translation of entities greatly impacts the overall sentence quality [4, 5].

Translating these entities remains challenging [6], and various methods have been proposed to improve their translation [7-9]. Some methods aim to incorporate knowledge graphs (KGs) to utilize their structured knowledge about entities and enhance entity translation. The KGs often describe entities as triples, including a subject (entity), a relationship (usually an attribute), and an object (another entity). This structural information can be used to enhance the semantic representation of entities in sentences [10-12], or to extract important semantic vectors using the KGs [13].

In this study, we investigate the feasibility of using contextual information from knowledge graphs to improve NMT models. Specifically, our knowledge graph integrated augmented machine translation model, called KGE-NMT, achieves significant improvements on IWSLT datasets for English-Vietnamese and English-German language pairs. To the best of our knowledge, there is no existing research that integrates knowledge graphs to enhance the translation of entities and overall translation performance. Therefore, the main contribution of this study is to examine the influence of entities and use knowledge graphs to enhance the ability to translate these entities. Furthermore, we demonstrate that improving the translation of entities by incorporating knowledge from knowledge graphs can enhance the overall translation quality of the NMT system. The main contributions in this study are as follows:

- Proposing a method to integrate information from the knowledge graph into the Transformer model to enhance the ability to translate OOV terms and words;

- Conducting experiments on two bilingual pairs (En-Vi, En-De) and performing an in-depth analysis to demonstrate the effectiveness of the proposed method.

2. RELATED WORK

In this study, we integrate an NMT model with a knowledge graph to supplement information for infrequent words (i.e., entities). The following studies are relevant to our work:

2.1. Neural machine translation

NMT aims to translate an input sentence from the source language to the target language. With the rapid development of deep learning, new ideas in machine translation continue to emerge. Currently, the NMT model typically includes an encoder and a decoder (encoder-decoder), which is a two-part architecture where the encoder reads the input sequence $x = (x_0, x_1, ..., x_n)$ and the decoder predicts the target sequence $y = (y_0, y_1, ..., y_n)$. Commonly used encoders and decoders in NMT include RNN [14], LSTM [15], and CNN [2]. The cross-attention mechanism, introduced by [16], significantly improved the encoder-decoder architecture in machine translation by allowing the model to focus on and align important parts of the source and target sentences.

The transformer model [3] has effectively followed this encoder-decoder architecture and utilized the self-attention mechanism, which allow the Transformer model to effectively learn and capture these differences. The encoder-decoder structure processes the source sentence and generates the target sentence while respecting each language's syntax. The self-attention mechanism helps track dependencies between words, allowing the model to rearrange sentences according to target language rules. Additionally, positional encoding helps the model understand word order, which is crucial when dealing with languages that have different syntax. These features enable NMT models to handle syntactic differences effectively.

2.2. Knowledge graph embedding

In recent years, building and exploiting knowledge graphs (KGs) has become an important research area in artificial intelligence and natural language processing. A large number of KGs have been developed, such as YAGO (Yet Another Great Ontology) [17], Freebase [18], and Dbpedia [19]. Recently, various approaches for knowledge embedding (KGE) aim to represent concepts and relationships in KGs as vectors in a high-dimensional vector space. These approaches include embedding both entities and relationships into low-dimensional spaces, such as TransE [18], TransH [20], and TransR [21]. In a recent research, Bojanowski [22] proposed a new method called fastText, based on the Bag-of-Words (BoW) structure, to represent knowledge in knowledge graphs. This method considers each triple, consisting of a head entity (h), relationship (r), and tail entity (t), as a single semantic unit (unique discrete token). This allows the fastText model to effectively capture and represent complex semantic structures in KGs. Wang *et al.* [23] proposed a technique for embedding structured and unstructured data (such as text), which allows efficient mining and connection of semantic information from both data sources, enhancing predictive models. These methods have been successfully applied in automatic question answering systems [24] and recommendation systems [25], contributing to improved natural language understanding and more accurate recommendations

2.3. NMT augmention

Recent efforts are focused on improving NMT performance, particularly addressing data scarcity and non-vocabulary words. Hoang and his colleagues [26] presented a reverse translation method to address the problem of limited translation resources. Recently, some studies have integrated KGs into NMT. For instance, Shi and colleagues [13] proposed using knowledge from KGs to embed semantics into the NMT model. Important semantic vectors can be extracted from KGs and integrated into the encoding or decoding process in NMT, thereby improving the model's understanding and translation quality. Moussallem *et al.* [10] exploited linked entities to enhance entity translation. Lu *et al.* proposed using entity relationships in KGs to strengthen the connection between source words and their translations. Zhao *et al.* [12] proposed a method for combining KGs that includes three steps: generating new translation results for entities by converting KGs into a unified semantic space, generating pseudo-parallel sentence pairs containing induced entity pairs, and training the NMT model by combining both original and pseudo sentence pairs. The difference between our method and previous methods lies in enhancing the translation of entities contained in sentences by adding semantic information to entities, significantly improving their translation quality.

3. METHOD

We propose to integrate the knowledge graph into NMT through semantic enrichment and sub-entity granularity, which includes two steps: 1) KGE and 2) KGE into NMT. Figure 1 provides an overview of each part of the architecture. Next, we will introduce each step in the following sections.

3.1. Notation

Let X and Y be the source language domain and target language domain respectively, with $X = (x_0, x_1, \ldots, x_n)$ and $Y = (y_0, y_1, \ldots, y_n)$ representing the sets of sentences in the

corresponding languages. Additionally, we need a knowledge graph K to assist in the translation process. It is often challenging to find bilingual KG pairs, and in some low-resource languages or domains, even monolingual KGs may not be available. Therefore, we only consider the case of integrating a KG in the source language, English. We denote the knowledge graph as $K = \{(h, r, t)\}$, where h, t, and r represent the head entity, tail entity, and relation in the source language, respectively (Figure 1). Q, K, and V represent the query, key, and value, respectively. Here, Q is the d_q-dimensional vector ($d_q \in Z$), and K and V are two sets with $|K| = |V|$.

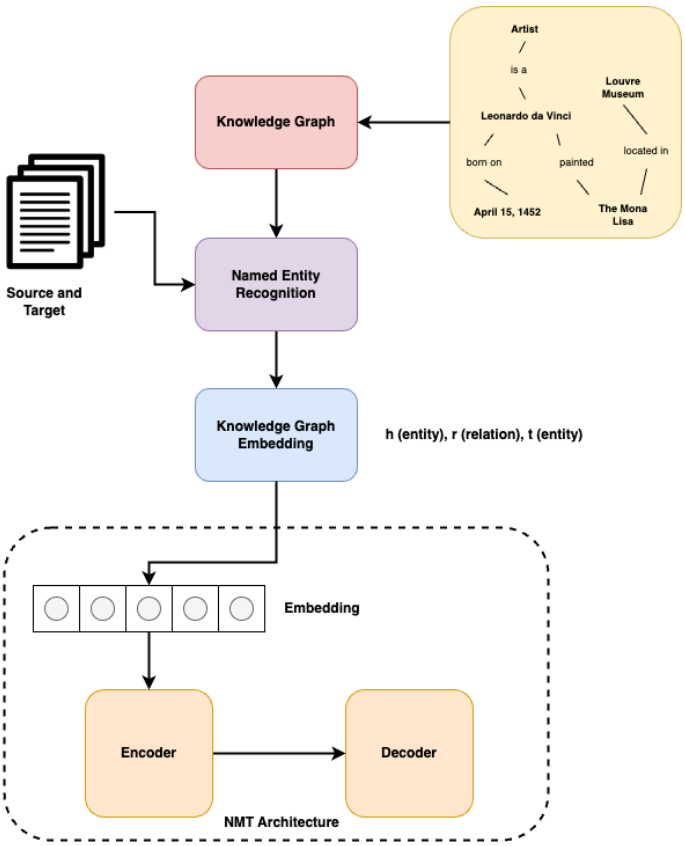

Figure 1. KGE-NMT architecture.

3.2. Knowledge graph embedding

Recent research efforts have successfully incorporated different types of knowledge into NMT models, such as linguistic information [27] and named entity tags (NE-tags) [28]. Drawing on these approaches, rather than training the NMT model on a large bilingual dataset to improve the efficiency of translating entities, our idea is to identify entities present in the text to be translated and then map them to corresponding nodes in the KG. The KG can add semantic information to entities, supporting the NMT model through its graph structure. This process includes two steps: Knowledge Query (KG-Query) and Knowledge Integration (KG-Embed). Specifically, given an input sentence $X = (x_0, x_1, \ldots, x_n)$ and a KG K, during the query process, all entities contained in the sentence X that are related to the KG will be selected to query their corresponding triples from the KG-Query. The KG-Query can be represented as (1), where E is the set of corresponding triples denoted as $E = \{(x_0, r_{01}, t_{01}), \ldots, (x_0, r\ \ , r\ \ )\}$ (Figure 2).

$$E = KG - Query(X, K) \quad (1)$$

Next, KG-Embed is responsible for integrating the queried set E into the sentence X by converting the triples in E into embeddings and adding them to their respective positions. Accordingly, the relationship r can be modeled as a displacement vector ($h + r = t$), while preserving the inherent structure of the KG.

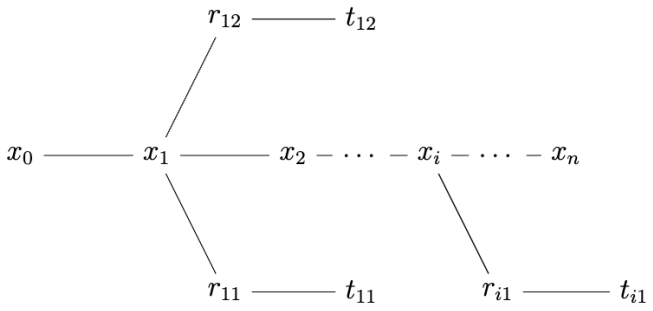

Figure 2. KGE injection.

3.3. Integrating KG to transformer

After integrating the knowledge, we use the BERT-NMT model in Figure 3 as the baseline for the NMT model. This architecture is supported by multi-head attention mechanisms, utilized in various ways including self-attention (in both the encoder and decoder) and cross-attention (in the decoder). One of the outstanding advantages of BERT is its ability to grasp the relationship between pairs of sentences, thanks to the Next Sentence Prediction (NSP) task during the pre-training process. This task requires the model to determine whether two sentences appear next to each other in the original text, helping BERT learn to understand and exploit broader contextual information. We can leverage this capability of BERT to improve the quality of document-level translation, similar to the idea proposed in the study by Miculicich [29].

In the sentence-level translation problem, the input to the model is a sequence of sentences $X = (x_0, x_1, ..., x_n)$ extracted from the same paragraph or document. These sentences are closely connected in terms of context and semantics. The main goal of this research is to translate these sentences into the target language while effectively integrating and exploiting contextual information added from an external knowledge graph. Combining knowledge from graphs with BERT's context capturing capabilities help the model produce more accurate, coherent, and natural translations at the sentence level.

The BERT-NMT architecture integrates the multi-level context information from the BERT model into both the encoder and decoder of the Transformer architecture (Figure 3). First, the input X, after being integrated with knowledge through the KGE process, is encoded by BERT into a context representation h_B, where the vector h_B, i is the representation of the ith word piece in x.

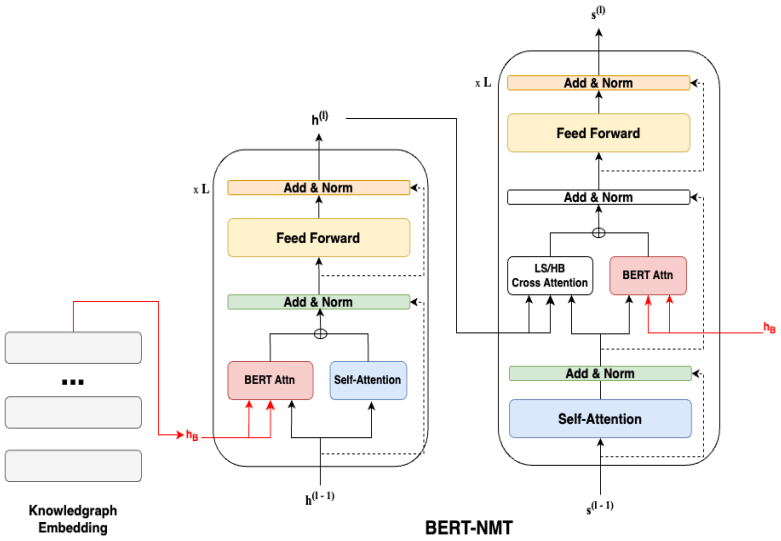

Figure 3. NMT architecture.

In the encoder, the hidden representation $h^{(l)}$ of the l-th layer is computed by combining information from the previous layer $h^{(l-1)}$ and the context from h_B through two attention mechanisms: Self-Attention and BERT-Attention. For each Q, K, and V respectively, the attention mechanism is represented as follows:

(2)

where A is the attention weight and W_q, $W_k \wedge W_v$ are the training parameters. The output of the lth layer is $h^{(l)}$, calculated by applying the function FFN(·) to each vector. This consists of two linear transformations with ReLU activation in between, similar to the approach described by [3]:

$$\text{FFN(X)} = \max(\text{x}W_1 + b_1, 0)W_2 + b_2 \quad (3)$$

Here, W_1 and W_2 are weight matrices, and b_1 and b_2 are bias vectors. In the decoder, the hidden state s^l of the lth layer is calculated based on the representation of the previous layer, the output of the encoder $h^{(l)}$, and the context from h_B through three attention mechanisms: Self-Attention, BERT-Attention, and Cross-Attention. The decoding process uses the output of the encoder $h^{(l)}$, and the hidden states of the layers in the decoder are mapped through linear and softmax transformations to predict the next word in the output sequence. The decoding process continues until the end-of-sentence token is reached. In this study, tokens were processed at the word level.

4. EXPERIMENTS

4.1. Corpora datasets

In this section, we experiment on two main datasets for Vietnamese translation (En ⇒ Vi), specifically IWSLT'15 English-Vietnamese [30]. The IWSLT'15 English-Vietnamese dataset consists of pairs of bilingual text sentences, where each English sentence is translated into Vietnamese and vice versa. These sentence pairs are collected from various sources, including

news, technical documents, and other data sources. This bilingual corpus contains about 130,000 sentence pairs, as detailed in Table 1. For the En ⇒ De translation task, we used the IWSLT'17 En-De dataset. All statistical information is presented in Table 1 and Table 2.

Table 1. Statistics of the English-Vietnamese datasets from IWSLT'15.

Dataset	#tokens		#types		#avg		#sent
	en	vi	en	vi	en	vi	
Train	2,435,771	2,867,788	44,573	21,661	20.81	24.50	117,055
Valid	27,988	34,298	3,518	2,170	18.02	22.08	1,553
Test	26,729	33,683	3,676	2,332	21.08	26.56	1,268

Table 2. Statistics of the English-German datasets from IWSLT'17.

Dataset	#tokens		#types		#avg		#sent
	en	de	en	de	en	de	
Train	4,261,095	4,020,066	69,207	140,914	20.52	19.36	207,667
Valid	30,706	29,229	3,829	5,075	18.06	17.19	1,700
Test	20,846	19,738	3,128	3,892	19.30	18.28	1,080

4.2. Knowledge graph datasets

KGs are represented in a "triples" structure, which includes a relationship and two related entities. In this structure, the relationship acts as a link between two entities, with the first entity called the "head" and the second entity called the "tail". For example, a triple could be (Ha Noi, Viet Nam). This structure allows for modeling concepts and the relationships between them in a clear and understandable way, providing a foundation for effective querying, inference, and knowledge exploitation (see Figure 4).

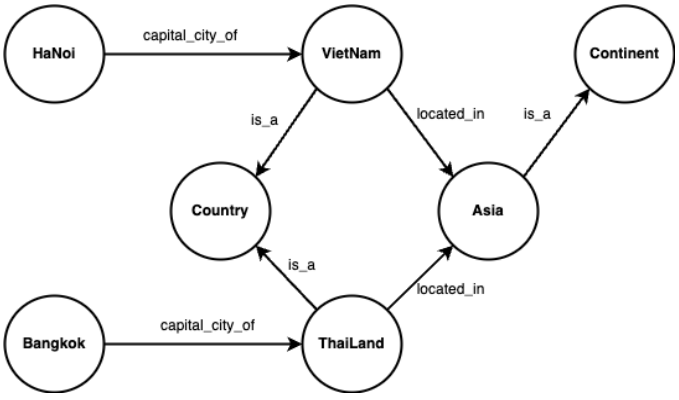

Figure 4. Knowledge graph example.

WordNet18 is a knowledge graph with 18 relationships taken from WordNet, encompassing about 41,000 synsets and resulting in 141,442 triples. In this study, we refine the

original WN18 by converting synset-ids into triples to suit the knowledge processing step of the model.

YAGO is a knowledge graph that enhances WordNet with common knowledge facts extracted from Wikipedia, transforming WordNet from a primary linguistic resource into a general knowledge base. YAGO initially included more than 1 million entities and 5 million facts describing the relationships between these entities.

4.3. Training settings

We implement the NMT model using 6 layers for both the encoder and decoder. The word embeddings are set to 512 dimensions, and the Feed-Forward Network (FFN) has 1024 dimensions, resulting in a total of 158 million model parameters. During the training phase, the Adam optimizer [31] with a fixed learning rate of 0.001 was used, and the batch size was set to 4000 tokens per batch. All models were trained for 30 epochs. Testing was performed on Google Colab Pro. The configuration in Google Colab includes:

- Processor: Intel® Xeon® CPU @ 2.20 GHz;
- Memory: 56 GB RAM;
- GPU: Tesla A100 16 GB.

4.4. Results

In this section, we conduct experiments and compare the proposed model with well-known baseline models such as Transformer and ConvS2S. The results in Table 3 show that without using KG, the BERT-NMT model achieved a higher BLEU score than baseline models like Transformer, with an increase of 2.28 points (26.99 vs. 24.71), and a slightly lower score than ConvS2S, with a decrease of 0.83 points (26.99 vs. 27.82). True to our theory, the knowledge graph integration model yields higher results than the baseline models. Specifically, with the YAGO knowledge graph, the proposed model (i.e. KGE-NMT) achieved 28.59 BLEU points, an increase of 1.60 points compared to the model without using KG (i.e. BERT-NMT). Using the WordNet18 knowledge graph, the proposed model (i.e. KGE-NMT) achieved 28.70 BLEU points, an increase of 1.71 points compared to not using KG (i.e. BERT-NMT). Notably, with the support of WordNet18, the proposed model outperforms the ConvS2S model by a margin of 0.88 points (28.70 vs. 27.82). This improvement highlights the significant role of incorporating external knowledge into the machine translation process.

Table 3. English ⟹ Vietnamese translation results.

	Model	BLEU⇑	TER⇓	METEOR⇑
1	Transformer	24.71	0.616	0.517
2	ConvS2S	27.82	0.551	0.545
3	BERT-NMT	26.99	0.530	0.567
4	KGE-NMT (YAGO)	28.59	0.515	**0.633**
5	**KGE-NMT (WN18)**	**28.70**	**0.529**	0.593

Similarly, we experimented with the En-De bilingual corpus. The results in Table 4 show that without using the knowledge graph (KG), the BERT-NMT model achieved a BLEU score of 25.40, which is 3.12 points higher than the basic Transformer model (25.40 compared to 22.28), but 0.75 points lower than the ConvS2S model (25.40 compared to 26.15). When integrating the YAGO knowledge graph, the performance of KGE-NMT improved to 26.22 BLEU points, an increase of 0.82 points compared to not using KG (i.e. KGE-NMT). When combined with the WN18 knowledge graph, KGE-NMT achieved the highest score of 26.40 BLEU, an increase of 1.0 points compared to the version without KG (i.e. KGE-NMT). This highlights that integrating additional knowledge from different graph sources into the machine translation model has positive effects, leading to performance improvements.

Table 4. English ⟹ German translation results.

	Model	BLEU⇑	TER⇓	METEOR⇑
1	Transformer	24.71	0.647	0.518
2	ConvS2S	27.82	0.619	0.526
3	BERT-NMT	26.99	0.565	0.521
4	KGE-NMT (YAGO)	28.59	0.548	0.549
5	KGE-NMT (WN18)	28.70	0.553	0.541

4.5. Analysis on enitity translation quality

Because we aim to improve the entity translation of the NMT model, to analyze the results across different entities, we randomly selected entities from the test set and randomly dropped these entities at rates of 10 %, 30 %, 50 %, and the entire 100 %. This analysis was performed on the model using knowledge graphs for both WN18 and YAGO in the En-Vi translation task. The results, shown in Figure 5, indicate that when 100 % of entities are dropped, the BLEU score of the KGE-NMT model is 26.85 for the WN18 set and 26.73 for the YAGO set. These scores are close to those of the BERT-NMT model that does not use a knowledge graph. However, when additional information about entities is available from the knowledge graph, the KGE-NMT model achieves significantly higher BLEU scores, such as 28.61 and 28.51 on the two datasets when only 10 % of entities are dropped. This result demonstrates that for sentences with more entities supplemented with semantic information, our method can improve translation quality, as reflected by the BLEU score. We believe this improvement come from our method enhancing the semantic representation of sub-entities, which in turn improves the translation of the entire sentence.

4.6. Analysis on sentence length

To comprehensively evaluate the effectiveness of the KGE-NMT model that combines knowledge from graphs in machine translation, we also analyze the translation results on sentences of different lengths using both the English-Vietnamese and English-German bilingual sets. Specifically, we classify sentences by length into three categories: short sentences (1-30 words), medium sentences (31-60 words), and long sentences (61-100 words).

Table 5. Analyses on sentence length on English ⇒ Vietnamese.

Sent length	Model	
	KGE-NMT (WN18)	KGE-NMT (YAGO)
Short	**30.80**	**30.02**
Medium	27.70	27.73
Long	23.42	23.90

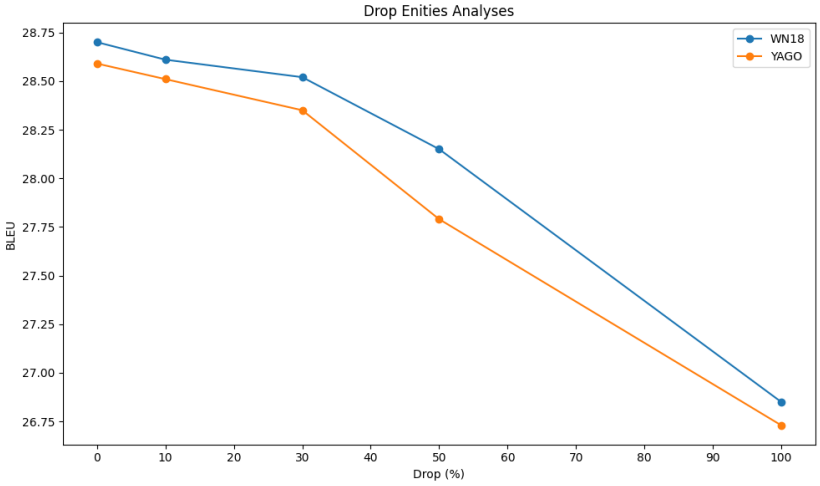

Figure 5. Drop entities analyses.

The results in Table 5 show that for short sentences (1-30 words), the model achieves very high BLEU scores of 30.80 on the WN18 set and 30.02 on the YAGO set. This indicates that the model can effectively translate short sentences with less complex contexts due to the support of knowledge from graphs. For medium sentences (31-60 words), the BLEU scores drop to 27.70 on WN18 and 27.73 on YAGO. This decrease reflects the need for the model to synthesize more complex knowledge and context for translating longer, more informative sentences. For long sentences (61-100 words), considered the most challenging to translate, the BLEU scores are 23.42 on WN18 and 23.90 on YAGO. These results show that despite the integration of knowledge graphs, translating long, complex sentences remains a significant challenge for the model. Although the proposed method does not entirely overcome this challenge, it does improve the overall translation quality. For the En-De corpus, the results are presented in Table 6. For short sentences (1-30 words), the model scores 30.23 BLEU (WN18) and 30.12 BLEU (YAGO) on the English-German set. For medium sentences (31-60 words), the BLEU scores are 23.46 (WN18) and 23.34 (YAGO) on the English-German collection. For long sentences (61-100 words), the BLEU scores are 20.07 (WN18) and 20.19 (YAGO) on the English-German set.

Table 6. Analyses on sentence length on English ⇒ German.

Sent length	Model	
	KGE-NMT (WN18)	KGE-NMT (YAGO)
Short	**30.23**	**30.12**

Medium	23.46	23.34
Long	20.07	20.19

5. CONCLUSIONS

In conclusion, this study demonstrates the feasibility and effectiveness of integrating Knowledge Graphs into Neural Machine Translation models to enhance the translation of entities and overall translation performance. Our proposed KGE-NMT model, which incorporates semantic information from KGs, shows significant improvements in translation quality for both English-Vietnamese and English-German language pairs. The experimental results confirm that using the KGs to supplement the translation process leads to higher BLEU scores compared to traditional NMT models. The integration of KGs addresses the data sparsity problem by enriching the semantic representation of entities, thereby improving the translation of rare and OOV words. One limitation of our method is the restriction of considering Knowledge Graphs as static, where no new entities, relationships, or facts are integrated during the translation process. This limits the system's ability to incorporate real-time or evolving knowledge, which could further enhance translation accuracy, especially in domains where new information is frequently introduced.

Future work will focus on further refining the integration process and exploring additional applications of the KGs in other language pairs and translation tasks. Additionally, addressing the limitation of integrating new entities, relationships, or facts to improve adaptability and performance in Neural Machine Translation is also a good direction.

Acknowledgements. This research is funded by Ho Chi Minh City University of Education Foundation for Science and Technology under grant number CS.2023.19.20.

CRediT authorship contribution statement. Nha Tran: Methodology, Visualization. Tri Le: Implementation, Software. Nam Nguyen: Methodology, Writing – Original draft. Long Nguyen: Methodology, Writing – Review & Editing, Validation, Supervision.

Declaration of competing interest. The authors state there is no conflict of interest.

REFERENCES

1. Luong M. T., Pham H., Manning C. D. - Conference on Empirical Methods in Natural Language Processing, (2015) 1412-1421. https://doi.org/10.18653/v1/D15-1166.
2. Gehring J., Auli M., Grangier D., Yarats D., Dauphin Y. - International Conference on Machine Learning, (2017)
3. Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser Ł., Polosukhin I. - Advances in Neural Information Processing Systems, (2017)
4. Koehn P., Hoang H. - Conference on Empirical Methods in Natural Language Processing, (2007) 868-876.
5. Wang Y., Wang L., Zeng X., Wong D. F., Chao L. S., Lu Y. - Conference on Computational Natural Language Learning Shared Task, (2014) 83-90. https://doi.org/10.3115/v1/W14-1711.
6. Moussallem D., Ngonga Ngomo A. C., Buitelaar P., Arčan M. - International Conference on Knowledge Capture, (2019) 139-146. https://doi.org/10.1145/3360901.3364423.
7. Ugawa A., Tamura A., Ninomiya T., Takamura H., Okumura M. - International Conference on Computational Linguistics, (2018) 3240-3250.

8. Mota P., Cabarrao V., Farah E. - European Association for Machine Translation Conferences/Workshops, (2022)
9. Li X., Yan J., Zhang J., Zong C. - China Workshop on Machine Translation, (2018) 93-100. https://doi.org/10.1007/978-981-13-3083-4_9.
10. Moussallem D., Arčan M., Ngonga Ngomo A. C., Buitelaar P. - Augmenting Neural Machine Translation with Knowledge Graphs. arXiv preprint, (2019)
11. Lu Y., Zhang J., Zong C. - China Workshop on Machine Translation, (2018) 27-38. https://doi.org/10.1007/978-981-13-3083-4_3.
12. Zhao Y., Xiang L., Zhu J., Zhang J., Zhou Y., Zong C. - International Conference on Computational Linguistics, (2020) 4495-4505. https://doi.org/10.18653/v1/2020.coling-main.397.
13. Shi C., Liu S., Ren S., Feng S., Li M., Zhou M., Sun X., Wang H. - Annual Meeting of the Association for Computational Linguistics, (2016) 2245-2254. https://doi.org/10.18653/v1/P16-1212.
14. Cho K., van Merriënboer B., Gulcehre C., Bahdanau D., Bougares F., Schwenk H., Bengio Y. - Conference on Empirical Methods in Natural Language Processing, (2014) 1724-1734. https://doi.org/10.3115/v1/D14-1179.
15. Hochreiter S., Schmidhuber J. - Long short-term memory. Neural Comput., **9**(8) (1997) 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735.
16. Chorowski J., Bahdanau D., Serdyuk D., Cho K., Bengio Y. - Advances in Neural Information Processing Systems, (2015) 577-585.
17. Suchanek F. M., Kasneci G., Weikum G. - YAGO: A Large Ontology from Wikipedia and WordNet. J. Web Semant., **6**(3) (2008) 203-217. https://doi.org/10.1016/j.websem.2008.06.001.
18. Bordes A., Usunier N., Garcia-Duran A., Weston J., Yakhnenko O. - Advances in Neural Information Processing Systems, (2013) 2787-2795.
19. Bizer C., Lehmann J., Kobilarov G., Auer S., Becker C., Cyganiak R., Hellmann S. - DBpedia - A crystallization point for the Web of Data. J. Web Semant., **7**(3) (2009) 154-165. https://doi.org/10.1016/j.websem.2009.07.002.
20. Wang Z., Zhang J., Feng J., Chen Z. - AAAI Conference on Artificial Intelligence, (2014) 1112-1119. https://doi.org/10.1609/aaai.v28i1.8870.
21. Lin Y., Liu Z., Sun M., Liu Y., Zhu X. - AAAI Conference on Artificial Intelligence, (2015) 2181-2187. https://doi.org/10.1609/aaai.v29i1.9491.
22. Bojanowski P., Grave E., Joulin A., Mikolov T. - Enriching Word Vectors with Subword Information. Trans. Assoc. Comput. Linguist., **5** (2017) 135-146. https://doi.org/10.1162/tacl_a_00051.
23. Wang Z., Zhang J., Feng J., Chen Z. - Conference on Empirical Methods in Natural Language Processing, (2014) 1591-1601. https://doi.org/10.3115/v1/D14-1167.
24. Bordes A., Usunier N., Chopra S., Weston J. - Large-scale Simple Question Answering with Memory Networks. arXiv preprint, (2015)
25. Zhang F., Yuan N. J., Lian D., Xie X., Ma W. Y. - ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, (2016) 353-362. https://doi.org/10.1145/2939672.2939673.
26. Hoang V. C. D., Koehn P., Haffari G., Cohn T. - Workshop on Neural Machine Translation and Generation, (2018) 18-24. https://doi.org/10.18653/v1/W18-2703.
27. Sennrich R., Haddow B., Birch A. - Conference on Machine Translation, (2016) 83-91. https://doi.org/10.18653/v1/W16-2209.
28. Gu J., Lu Z., Li H., Li V. O. K. - Annual Meeting of the Association for Computational Linguistics, (2016) 1631-1640. https://doi.org/10.18653/v1/P16-1154.
29. Miculicich L., Ram D., Pappas N., Henderson J. - Conference on Empirical Methods in Natural Language Processing, (2018) 2947-2954. https://doi.org/10.18653/v1/D18-1325.
30. Cettolo M., Niehues J., Stüker S., Bentivogli L., Cattoni R., Federico M. - International Workshop on Spoken Language Translation, (2015).
31. Kingma D. P., Ba J. - Adam: A method for stochastic optimization. arXiv preprint, (2014).