Graph-based Relation Mining for Context-free Out-of-vocabulary Word Embedding Learning

Ziran Liang and **Yuyin Lu** and **Hegang Chen** and **Yanghui Rao***
School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China
{liangzr5,luyy37,chenhg25}@mail2.sysu.edu.cn, raoyangh@mail.sysu.edu.cn

Abstract

The out-of-vocabulary (OOV) words are difficult to represent while critical to the performance of embedding-based downstream models. Prior OOV word embedding learning methods failed to model complex word formation well. In this paper, we propose a novel graph-based relation mining method, namely GRM, for OOV word embedding learning. We first build a Word Relationship Graph (WRG) based on word formation and associate OOV words with their semantically relevant words, which can mine the relational information inside word structures. Subsequently, our GRM can infer high-quality embeddings for OOV words through passing and aggregating semantic attributes and relational information in the WRG, regardless of contextual richness. Extensive experiments demonstrate that our model significantly outperforms state-of-the-art baselines on both intrinsic and downstream tasks when faced with OOV words.

1 Introduction

Pre-trained word embedding models, such as Word2Vec (Mikolov et al., 2013) and BERT (Devlin et al., 2019), can not only boost the performance of downstream tasks but also accelerate the convergence of downstream models (Kuratov and Arkhipov, 2019; Kao and Lee, 2021). However, in real-world scenarios, the pre-trained models trained with generic large-scaled corpora may encounter a lot of words never seen before in downstream tasks due to domain specificity. These out-of-vocabulary (OOV) words rarely appear, resulting in a scarcity of their contexts, while traditional word embedding methods require a large number of contexts to learn high-quality word embeddings (Herbelot and Baroni, 2017). The OOV words may cause a dramatic performance degradation in downstream tasks because of their poor word embeddings (Nayak et al., 2020; Schick and Schütze, 2020; Won et al., 2021), which leads to the OOV problem. Thus, it is vital to explore an effective way of learning high-quality OOV word embeddings in natural language processing.

Traditional methods for tackling the OOV problem injected sub-units of words into the training process of pre-trained word embedding models to get the sub-unit embeddings and then calculated the OOV word embedding as a summation of them (Bojanowski et al., 2017; Cao et al., 2018; Devlin et al., 2019). These methods require training from scratch, which are time consuming. To save computing resources, two categories of methods have been proposed. Methods in the first category attempted to fully utilize limited contextual information carried by OOV words directly without modifying the training process of background models (Garneau et al., 2018; Hu et al., 2019; Schick and Schütze, 2019). These methods are often lightweight, but they cannot deal with some frequently occurring situations where the OOV words are extremely context-less. To break the limitation of contexts, methods in the other category learned word embeddings for OOV words through fine-grained sub-units or morphemes to model word formation implicitly without using contexts (Pinter et al., 2017; Zhao et al., 2018; Chen et al., 2022). However, the word formation can be complex and highly internally structured (Anderson, 1992), rendering simple simulations cannot represent the word formation well.

In the situation of context absence, it's meaningful to utilize word formation for OOV words since most language vocabularies are derived from the creation of new words on the basis of old ones (Denison, 1997; Josefsson, Gunlög, 1997). Intuitively, humans can guess the meaning of an OOV word based on its complex word formation and association with similar words, as shown in Figure 1. However, the measures of word formation are varied, and the relationships inside word struc-

*The corresponding author.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 14133–14149

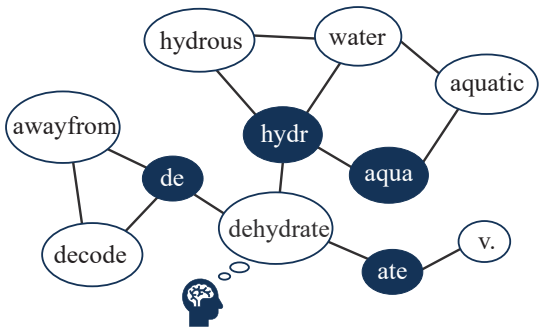

Figure 1: Human learning pattern for a new word.

tures are sophisticated. Taking this into account, we introduce a *Word Relationship Graph* (WRG) to imitate word formation and word association for better capturing the relational information of word-internal structure, which logically simulates human learning habits when facing an OOV word.

In light of these considerations, we propose a Graph-based Relation Mining (GRM) model on the basis of WRG to learn embeddings for OOV words without contexts, which can help mine the relational information about complex word formation. Our method can also explore additional semantic information by associating each OOV word with its relevant words. To achieve these, we transport and incorporate relational information and semantic attributes by *Graph Neural Network* (GNN). Noteworthy, we use the graph structure to find more reasonable positive sample pairs for contrastive learning, forcing every node embedding to be more informative. The contributions of our work can be summarized as follows:

- We develop a WRG which is built upon the rules of word formation. The graph structure can mine the relational information of word-internal structure and associate OOV words with semantically relevant words, which is in line with human study habits.
- We present a generic approach that incorporates both relational information and semantic attributes by GNN in word embedding learning. Furthermore, we select rational positive sample pairs for contrastive learning by utilizing graph structure.
- Our GRM model achieves state-of-the-art results on various evaluation metrics and largely improves the performance of static and contextual word embedding models on downstream tasks.

2 Related Work

2.1 Context-based Out-of-vocabulary Word Embedding Learning

The occurrence of OOV words is often accompanied by data scarcity of contexts. Traditional methods for OOV word embedding learning integrated the word formation information into the training process of pre-trained models and they were trained from scratch (Bojanowski et al., 2017; Cao et al., 2018; Devlin et al., 2019; Boukkouri et al., 2020; Sun et al., 2021), which consumed considerable computational resources and time costs. To address this problem, some methods attempted to make full use of the limited contextual information carried by OOV words, which is valuable for learning OOV word embeddings. Herbelot and Baroni (2017) and Kabbach et al. (2019) adopted a high-risk learning rate strategy, while Hu et al. (2019) took a few-shot learning pattern to fit the tiny data situation. Besides, some works employed the attention mechanism to emphasize important and informative contexts (Garneau et al., 2018; Schick and Schütze, 2019). These methods were often lightweight since they didn't modify the training process of original models. However, in practice, some OOV words tend to occur in extremely context-less situations, where these methods are hard to work. Furthermore, the data scarcity of contexts may introduce noise to the context-based models easily, which deteriorates their performance.

2.2 Context-free Out-of-vocabulary Word Embedding Learning

In some cases, the contextual information of OOV words will be extremely scarce. Context-free approaches can tackle this problem easily by learning the word embedding through the OOV word itself. These methods focused mainly on finding correlations between word embedding and word formation. They represented the word form information through characters (Pinter et al., 2017), sub-units (Zhao et al., 2018; Zhang et al., 2019; Sasaki et al., 2019; Fukuda et al., 2020; Chen et al., 2022), images (Chen et al., 2020a), and so forth. Generally, word formation is complex and cannot be simulated by simply cutting words or imitating the glyph of words. Although these methods try to implicitly model word formation, partial information about the relationships inside the word structures is usually lost.

3 Proposed Method

Within the existing methods, Mimick (Pinter et al., 2017) used a lightweight post-processing learning paradigm, which attempted to mimic the vector space of a background embedding model for OOV words and can therefore be applied to different types of embedding models. The mimick paradigm sought to maximize the similarity between the inferred embeddings produced by the OOV word embedding model and the original embeddings derived from the background embedding model. We follow this mimick learning paradigm to mine the relational information about word formation. Compared to other data structures, graph structure can model complex data compositions well. Therefore, we construct a WRG to model word formation rules and associate other semantically related words. The relational information and relevant semantic attributes can be transported and aggregated on WRG by GNN. Besides, we utilize graph structure in the process of positive sample pairs selection for contrastive learning, which can provide the flexibility to obtain more reasonable positive sample words.

3.1 Word Relationship Graph Construction

To better represent word formation rules, we construct a WRG around each OOV word. Firstly, we tokenize all words into sub-units by WordPiece tokenizer (Wu et al., 2016), which allows a sub-unit to retain its entire semantics in the smallest possible unit like a morpheme. We denote the sub-units produced by WordPiece tokenizer as wordpieces in the following. Then, we connect words with the corresponding wordpieces. This connected edge carries position information, which is the position of the wordpiece in the associated word. Finally, we construct a two-layer undirected graph around an OOV word, with its wordpiece in the first layer and relevant words that have the same wordpiece in the second layer. In this way, we simulate the lexical rules of word formation and naturally associate OOV words with the learned semantic relevant words via common morphemes, which allows us to better model word formation in a human learning mindset. To make full use of the graph structure, we treat a word or a wordpiece as a common node n_i in the graph and treat the corresponding node attribute $h_i \in \mathbb{R}^d$ as its embedding, where d denotes the dimension of embedding. Besides, we add a self-loop to the OOV word node itself to include its attributes.

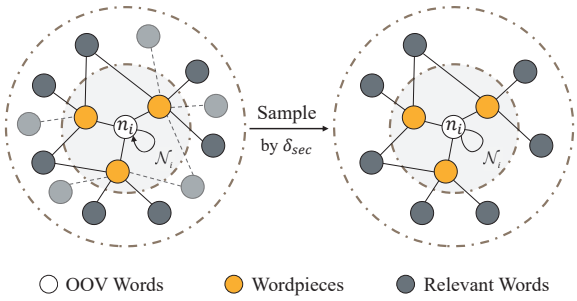

Figure 2: The construction of the WRG.

After constructing the WRG around each OOV word, we keep all the nodes in the first layer to maintain the entire wordpiece information. As for the second layer, we only sample a fixed number of nodes for training, because a wordpiece node with a lot of neighbors may be noisy. For example, the morpheme *ly* mainly plays a syntactic role instead of having sufficient semantic information. We therefore set a threshold δ_{sec} to limit the number of neighbor nodes in the second layer, which also saves training costs and prevents over-fitting. For simplicity, we just sample the words to leave in the second layer randomly. We show the WRG construction in Figure 2, where $\mathcal{N}_i$ denotes the set of neighbor nodes of node n_i.

3.2 Model Architecture

To exploit information contained in the WRG, we choose GNN as the basic learning method. GNN can transmit the attributes of neighbor nodes to node n_i via the topology structure, which can act as a low-pass filter to emphasize the connectivity between nodes in the neighborhood field (NT and Maehara, 2019). Following the transmission routes, the attributes of the pre-trained wordpiece nodes and other in-vocabulary word nodes, as well as topological information about the relationships inside the word structures can be fused and passed to the OOV word nodes. It is worth noting that, in the construction of WRG, we connect relevant words with OOV words indirectly via the same wordpiece nodes rather than directly. Thus, GNN primarily uncovers and transports the relationship of word-internal structure.

To extract the most important information and reduce the impact of noise neighbor nodes, we choose *Graph Attention Network* (GAT) (Velickovic et al., 2018) as the backbone in this part, which can assign different learning weights to dif-

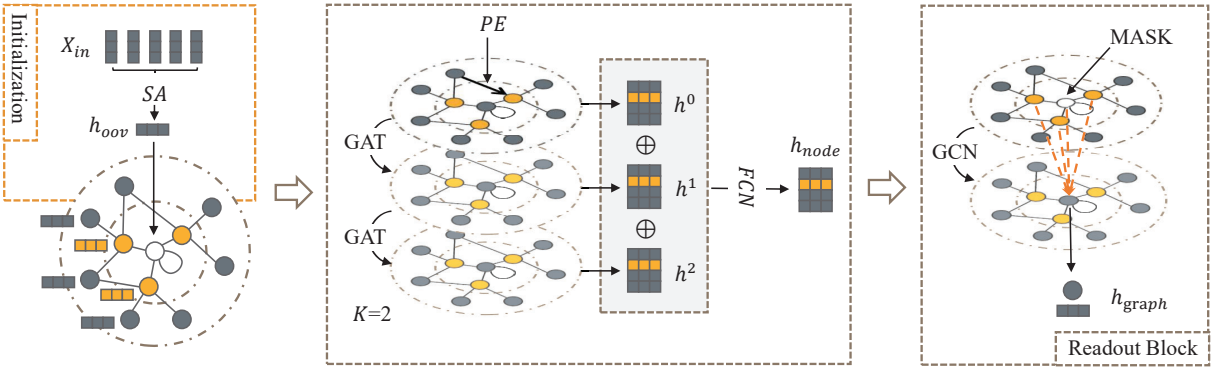

Figure 3: The architecture of our GRM model.

ferent neighbors according to the attention mechanism. The attention coefficients α_{ij} between node n_i and its neighbor $n_j \in \mathcal{N}_i$ is normalized by the softmax function and can be computed as follows:

$$e_{ij} = a(Wh_i, Wh_j). \tag{1}$$

$$\alpha_{ij} = softmax_j(e_{ij}) = \frac{exp(e_{ij})}{\sum_{n_k \in \mathcal{N}_i} exp(e_{ik})}, \tag{2}$$

where e_{ij} denotes the attention coefficients, a is a shared attentional mechanism, W is a learnable weight matrix of GAT. Noteworthy, the graph structure will ignore the sequence information of word formation. To alleviate this problem, we add the position embeddings PE_{ij} of the position information carried by the link between n_i and n_j proposed by Devlin et al. (2019) to the message passing routes of the basic GAT, as follows:

$$h_i^l = \sigma(\sum_{n_j \in \mathcal{N}_i} \alpha_{ij}(W^l h_j^{l-1} + PE_{ij})), \tag{3}$$

where $h_i^l \in \mathbb{R}^d$ means the hidden embedding of node n_i in layer l, $\sigma(\cdot)$ denotes the sigmoid activation function at the end of each GAT layer. Then, we can get the node-level representation $h_{node_i} \in \mathbb{R}^d$ of node n_i by concatenating the initial input with the hidden embedding of each layer and fusing them using a fully connected network $FCN(\cdot)$, which can prevent information loss between network layers, i.e.,

$$h_{node_i} = FCN(h_i^0 \oplus ... \oplus h_i^K), \tag{4}$$

where K means the total number of network layers. In our model, $K = 2$, which is consistent with the number of layers in the WRG.

Initialization for OOV Nodes At the beginning, we need to assign a node attribute to the corresponding node. The node attributes of in-vocabulary words or wordpieces are initialized as their pre-trained embeddings. However, as for the OOV word nodes, we cannot know their embeddings in advance. Assigning random initialization or all-zero vectors to OOV words may lead to confusion in the attention mechanism, and thus the performance of the networks will deteriorate. To avoid that, we represent an OOV word as a set of characters and get the initial value by character-level embeddings. Instead of a simple summation, we use a self-attention network $SA(\cdot)$ (Vaswani et al., 2017) to emphasize the important character components. This operation not only provides a good initialization for the OOV word nodes but also replenishes the serialized textual information of the OOV words. Notably, it provides sufficient information on word formation even in extreme cases where splitting words is unfeasible. Given a series of n characters, $\{x_1, x_2, ..., x_n\}$, forming a matrix $X_{in} = \{x_1, x_2, ..., x_n\} \in \mathbb{R}^{n \times d}$, the representation of the OOV node $h_{oov} \in \mathbb{R}^d$ can be computed as follows:

$$h_{oov} = SA(X_{in}). \tag{5}$$

Readout Block At this stage, we can get node-level representations of all nodes, but it is not enough to obtain a node-level representation for modeling word formation. The formation of a word is composed of its sub-units and the relationships between sub-units and itself. According to the structure of WRG, the wordpiece nodes in the first layer and the connections between wordpiece nodes and OOV word nodes can represent the internal structure of the OOV word. The graph-level representation can summarize and represent the

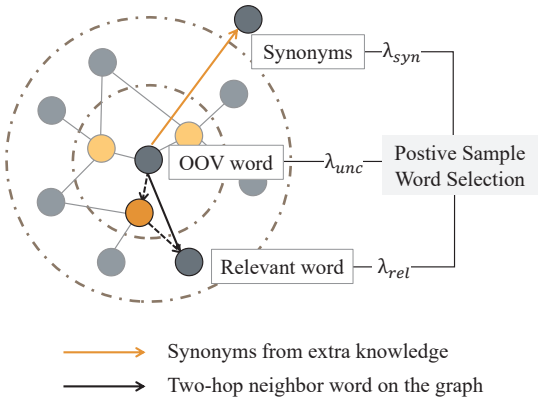

Figure 4: The postive sample word selection strategy.

information of the entire WRG by aggregating the node-level representation with a readout function. A simple one-layer *Graph Convolutional Network* (GCN) (Kipf and Welling, 2017) can satisfy our needs for representing word formation. In addition, based on the theory of Hou et al. (2022), we mask the OOV word node embedding to force GCN to uncover deeper relationships with neighboring nodes. The operation in the readout block can be described as follows:

$$h_{graph_i} = \sigma(\sum_{n_j \in \mathcal{N}_i} \frac{1}{c_{ij}} W_{gcn} h_{node_i})), \quad (6)$$

where $h_{graph_i} \in \mathbb{R}^d$ means a graph-level representation of node n_i, W_{gcn} is a learnable weight matrix of GCN, c_{ij} is the normalization factor, $\sigma(\cdot)$ denotes the sigmoid activation function at the end of the single GCN layer. Noteworthy, the mask operation in the readout block will not discard all information of the node-level representation h_{node}, since our graph WRG is an undirected one, which means the information of h_{node} will be passed to its neighbors, especially for the wordpiece nodes. And then in the readout block, this information can be "reawakened" by a layer of GCN. The complete GRM model architecture is illustrated in Figure 3.

3.3 Loss Function

Mean square error (MSE), a traditional loss function, is fragile and prone to over-fitting (Hou et al., 2022). To avoid this issue, we introduce a contrastive learning loss NT-Xent (Chen et al., 2020b) for the final output, which focuses on two indicators, alignment and uniformity (Wang and Isola, 2020). Alignment makes positive pairs more similar, while uniformity spreads word embeddings out in space. Except for drawing the positive pair (x, y) closer, we treat the other $2(N-1)$ pairs in the same batch as negative examples and try to keep a distance from them, where N is the batch size. The loss can be calculate as follows:

$$l(x,y) = -log \frac{exp(sim(x,y)/\tau)}{\sum_{z=1[z \neq x]}^{2N} exp(sim(x,z)/\tau)}, \quad (7)$$

where $sim(\cdot,\cdot)$ is a function that measures the similarity between two samples. We choose the cosine similarity function here. τ denotes a temperature coefficient.

In order to make the node embeddings more semantically informative, we propose a strategy to select positive sample pair (x, y) through WRG for contrastive learning. If we take the inferred embedding h_{graph_i} of OOV word node n_i generated by GRM as sample x, then sample y can be the original embedding of the positive sample word from the background model vocabulary, which is what we are trying to mimic. The positive sample word can be selected from the following three options: (1) The relevant words, namely two-hop neighbor words of OOV words in the WRG, since they share the same wordpiece nodes with OOV words. (2) The synonyms of each OOV word, which can further improve the learning ability for semantics. (3) The OOV word itself. The proportions of these three choices are λ_{rel}, λ_{syn}, and λ_{unc}, respectively. We show details about the selection strategy in Figure 4.

4 Experiments

In this section, we carry out extensive experiments on several widely-used text datasets varying in scale to test different methods, which can be categorized into intrinsic and extrinsic evaluators. Furthermore, we plug GRM into static and contextual word embedding models to show the gains brought by GRM. Finally, we conduct qualitative analysis and ablation study on GRM.

4.1 Datasets and Experimental Settings

Datasets We evaluate our work on two types of intrinsic evaluators: word similarity and word analogy. For the word similarity task, we follow the setting in Chen et al. (2022) to conduct evaluations on six benchmark datasets: RareWord (Luong et al., 2013), SimLex (Hill et al., 2015), MTurk (Halawi et al., 2012), MEN (Bruni et al., 2014), Rel353 (Agirre et al., 2009), and simverb (Agirre et al., 2009). For the word analogy task,

Model	Params	Word Similarity (Spearman's ρ)							Word Analogy (Acc)
		RareWord	MEN	SimLex	Rel353	simverb	muturk	**AVG**	Google
Mimick (2017)	9M	13.29	3.84	-7.25	1.10	-1.95	-0.57	1.41	0.05
BoS (2018)	500M	**40.41**	48.99	14.32	39.15	15.02	40.22	33.02	39.78
KVQ-FH (2019)	12M	38.91	53.06	8.84	41.12	12.13	46.26	33.39	33.12
LOVE (2022)	9M	38.38	56.00	**26.51**	43.87	**26.65**	49.13	40.09	34.27
GRM (Ours)	1.8M	35.57	**68.24**	24.20	**50.40**	23.83	**58.94**	**43.53**	**54.73**

Table 1: Overall experimental results of the context-free models on the word similarity and word analogy tasks.

Model	Params	Named Entity Recognition (F1-score)					POS Tagging (Acc)		
		CoNLL	BC2GM	BC4Chemd	BC5CDR	NCBI	UD	ARK	Ritter
HiCE (2019)	5M	76.69	50.41	62.17	46.93	54.63	88.82	71.14	68.31
AM (2020)	52M	80.57	65.60	75.34	70.63	66.55	92.44	75.29	72.06
Mimick (2017)	9M	66.00	41.41	48.44	56.45	33.09	87.23	65.39	60.28
BoS (2018)	500M	76.72	63.59	60.61	72.84	**78.39**	92.03	75.22	72.76
KVQ-FH (2019)	12M	54.33	34.26	47.90	46.86	28.50	89.36	67.03	58.57
LOVE (2022)	9M	80.82	64.57	74.80	73.81	63.62	93.39	79.64	76.25
GRM (Ours)	1.8M	**83.76**	**71.41**	**81.97**	**83.08**	77.81	**93.90**	**85.85**	**82.89**

Table 2: Overall experimental results of GRM and baselines on NER and POS tagging tasks.

we conduct evaluations on the Google benchmark dataset (Mikolov et al., 2013). And we evaluate our work on two types of extrinsic evaluators: Named Entity Recognition (NER), and Part-Of-Speech (POS) tagging. For the NER task, we conduct evaluations on five datasets: CoNLL (Sang and Meulder, 2003) , BC2GM (Smith et al., 2008), BC4Chemd (Krallinger et al., 2015), BC5CDR (Wei et al., 2016), and NCBI-DISEASE (Dogan et al., 2014). For the POS tagging task, we conduct evaluations on three datasets: Universal Dependencies (UD) scheme version 1.4 (Marneffe et al., 2014), Twitter POS ARK (Gimpel et al., 2011), and Ritter POS (Ritter et al., 2011). These datasets are all English datasets and most of them have high OOV rates. More details about intrinsic and extrinsic datasets are shown in Appendix A.

Experimental Settings Our GRM model requires tokenizing words to construct the WRG, and we choose the wordpiece vocabulary from (Chen et al., 2022). The vocabulary is more fine-grained than the vocabulary of BERT, which allows us to discover the relationships of word-internal structure conveniently. We choose a Word2Vec model trained from a Wikipedia snapshot of 2019 as the pre-trained background word embedding model for the quantitative evaluation by following a previous work (Kabbach et al., 2019). And we use synonyms from WordNet[1], which are all in the background vocabulary. For a comprehensive and fair comparison, we select two classes of baseline models, which are all proposed for OOV word embedding learning. One class of models don't need any additional contextual information for training, including Mimick (Pinter et al., 2017), BoS (Zhao et al., 2018), KVQ-FH (Sasaki et al., 2019), and LOVE (Chen et al., 2022). And the other class takes contexts into consideration, including HiCE (Hu et al., 2019) and AM (Schick and Schütze, 2020). We train these baseline models according to their published optimal settings. More information about experimental settings is detailed in Appendix B.

[1] https://wordnet.princeton.edu/

4.2 Quantitative Evaluation

Intrinsic Evaluation Intrinsic evaluators measure the quality of word embeddings by directly checking whether the word embedding vectors match the semantic relationships between words. The words in the intrinsic datasets have no contextual information, we only compare GRM with the baselines trained without contexts. Table 1 shows all experimental results of intrinsic evaluations. The performance of Mimick was limited because it only considers the information of characters, which is difficult to find semantic relationships between words. Our model achieved the best average score and superior results on most tasks, which demonstrates that GRM can model word formation better than other context-free models. But GRM performed slightly worse on the RareWord, SimLex, and simverb datasets.

Model	Named Entity Recognition										POS Tagging					
	CoNLL		BC2GM		BC4Chemd		BC5CDR		NCBI		UD		ARK		Ritter	
	ALL	OOV	ALL	OOV	ALL	OOV	ALL	OOV	ALL	OOV	ALL	OOV	ALL	OOV	ALL	OOV
Static Embedding Model																
Word2Vec (2013)	61.19	61.40	66.00	70.95	72.48	58.93	77.73	61.23	67.89	63.47	88.51	58.87	70.04	35.63	69.40	33.71
+HiCE (2019)	79.05	79.53	68.67	73.97	77.09	67.56	77.53	61.39	74.68	77.72	91.95	75.65	76.68	53.40	76.08	54.13
+AM (2020)	81.85	82.16	68.50	75.38	78.05	70.08	80.12	69.50	72.61	72.13	93.27	81.20	77.70	54.90	75.99	53.01
+Mimick (2017)	71.53	71.71	69.11	74.71	74.83	64.04	80.31	68.67	70.08	69.65	92.35	76.72	76.43	53.52	76.39	55.66
+BoS (2018)	78.06	78.43	67.61	72.87	77.30	68.29	82.03	**72.05**	69.75	66.39	92.54	77.74	76.26	52.27	75.47	51.05
+KVQ-FH (2019)	64.50	64.72	66.29	70.41	72.97	59.98	77.80	62.40	66.85	61.36	91.15	69.92	71.55	39.15	70.19	39.16
+LOVE (2022)	81.55	81.84	69.93	75.33	78.51	69.91	81.23	69.82	71.89	71.17	94.11	84.32	81.95	68.14	78.70	61.68
+GRM (Ours)	**86.10**	**86.29**	**71.48**	**81.86**	**82.53**	**82.02**	**82.30**	65.44	**76.48**	**80.68**	**94.64**	**87.25**	**86.74**	**79.96**	**83.59**	**76.36**
Contextual Embedding Model																
BERT (2019)	91.18	92.17	87.95	90.20	91.95	92.74	92.23	93.07	91.22	91.77	**96.14**	**74.68**	77.84	60.91	71.00	**38.20**
+HiCE (2019)	88.15	87.82	75.57	79.03	79.94	79.53	75.76	75.72	78.46	79.17	93.92	31.08	58.27	7.69	62.41	5.36
+AM (2020)	93.23	95.95	88.63	92.66	**93.06**	**94.63**	92.38	94.44	**92.35**	**94.75**	94.92	67.26	71.83	56.59	69.42	29.31
+Mimick (2017)	91.43	93.82	87.63	91.88	91.57	93.38	92.58	94.20	90.25	93.25	94.20	66.36	74.58	51.71	69.73	28.42
+BoS (2018)	92.55	94.94	87.40	91.57	91.29	92.81	92.29	94.19	91.20	93.34	93.37	61.09	69.07	41.46	67.53	28.72
+KVQ-FH (2019)	91.51	93.85	86.46	90.72	90.38	92.02	89.28	91.72	89.08	92.54	92.44	61.15	66.90	37.75	66.65	25.18
+LOVE (2022)	91.87	93.99	87.71	91.77	91.94	93.66	91.55	93.38	90.88	93.31	95.06	70.33	76.68	57.50	71.57	31.54
+GRM (Ours)	**93.29**	**96.00**	**88.71**	**92.76**	92.84	94.49	**93.19**	**95.04**	92.19	94.33	95.34	72.46	**78.17**	**62.12**	**72.86**	35.38

Table 3: The experimental results of extending Word2Vec and BERT to NER and POS tagging tasks. We measure the performance by F1-score, except reporting accuracy in the background of Word2Vec model.

These datasets provide some superficially unrelated but semantically similar word pairs, especially for the RareWord dataset. Our GRM model is sensitive to word formation, which leads to overfitting in these datasets. Notably, our model outperformed on word analogy tasks due to the superiority of graph structure, which means GRM can uncover the semantic relationship information of word formation.

Extrinsic Evaluation Extrinsic evaluators measure word embeddings by their performance on the downstream tasks. Table 2 shows all experimental results of extrinsic evaluations. The performance of baseline models was degraded with varying degrees in these datasets, because it is hard to understand the meaning of OOV words by contexts for downstream models in the datasets with high OOV rates. In contrast, GRM generally performed the best among these models, even in tasks with high OOV rates. This verifies the superior quality of the word embeddings inferred by our model. Besides, our model achieved excellent results even when compared to models that use context, which demonstrates that word formation is indeed valid for learning OOV word embeddings. It's worth noting that our model requires the fewest parameters among these models. More details about the efficiency analysis are described in Appendix E.

4.3 Model Adaptability

To investigate the effectiveness our model brings to static and contextual models in downstream tasks, we plug our model into Word2Vec and BERT respectively. In order to explore the improvement of our model on the OOV problem, we add metrics on OOV words when conducting experiments on NER and POS tagging tasks. It is easy to extend static word embedding models by directly adding new words and embeddings into the background models. We choose the Word2Vec model mentioned before as the static pre-trained embedding models. And for the contextual pre-trained embedding models, we choose the uncased BERT-base model (Wolf et al., 2020) as the background model. Note that the word embeddings in contextual word embedding models are diverse because of their contextual training method. Inspired by Chen et al. (2022), we use the whole words in BERT pre-trained embedding for model training, and infer reasonable embeddings for the words which were tokenized into pieces. Besides, the AM model introduces an one-token approximation (OTA) component to support its application in BERT, which represents a sequence of piece embeddings as an one-token embedding according to contexts (Schick and Schütze, 2020). Obviously, the reason why BERT cannot cope with the OOV problem is that BERT fails to assign a reasonable semantic meaning for those words that are over-divided by general embeddings of wordpieces (Chen et al., 2022). To prevent the phenomenon of over-division, we infer embeddings for the words that should be segmented and fed the embeddings into BERT.

Table 3 shows the experimental results of plugging different baseline models. Our GRM model

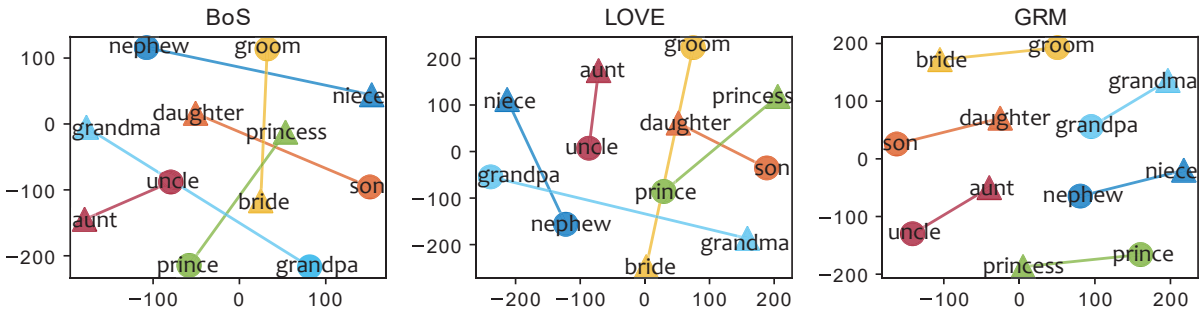

Figure 5: Visualization results of word embeddings generated by BoS, LOVE, and GRM on the word analogy task, where "△" denotes the female word and "◯" denotes the male word in the word pair.

brought the most significant improvement over the tasks with Word2Vec as the background model, not only in the evaluation of OOV words but also in the overall metric. This demonstrates the rationale for learning high-quality OOV word embeddings through modeling word formation. Furthermore, the mimic learning paradigm allows us to augment the Word2Vec model. The performance of GRM and AM were comparable on the NER tasks with BERT as the background model. However, AM requires initialisation via OTA first, which consumes additional 6 days of GPU time on all these datasets. Besides, plugging BERT with any baseline models led to performance dips on the POS tagging tasks. GRM improved the performance of BERT on some POS tagging tasks, but slipped on the UD dataset, which has a low OOV rate. We conjecture this is because if contextual information is adequate and correct enough, it is easy to tag POS labels over the OOV words. In addition, the word division process in BERT will highlight the syntactic part of words, which also makes tagging OOV words easier.

4.4 Qualitative Analysis

To better illustrate the quality of word embeddings inferred by GRM, we select six pairs of words from the family part of the Google dataset for the word analogy task and visualize the results by reducing dimension through t-SNE (van der Maaten and Hinton, 2008). Due to space limitations, we only show the top three models that work best on the word analogy task, the rest model results are represented in Appendix D.1. Figure 5 shows the visualization result of different models. The results of BoS and LOVE were inconsistent with the semantics of the words. Although they try to model word formation implicitly, they ignore relational information inside word structures. Our GRM model achieved the best visual result, where the word pairs are almost parallel and uniformly distributed in their linear concatenations, with only two word pairs having opposite gender positions. This shows that GRM can preserve semantic relational information through graphs, which is consistent with human cognition. Furthermore, selecting positive sample pairs by WRG for contrastive learning makes the word embeddings more reasonable.

4.5 Ablation Study

In this section, we conduct ablation experiment of each component in our GRM model to validate their effectiveness. GRM w/o Readout refers to the GRM model without the readout block. GRM w/o mask denotes the GRM model without mask operations in the readout block. GRM w/o relevant refers to the GRM model having no relevant words in the second layer of the WRG, which means δ_{sec} is set as 0. GRM w/o SA refers to the GRM model without the initialization part for OOV nodes. GRM w/o PE denotes that removing the position embeddings to the message passing route in the GAT part. GRM w/o data aug means the GRM model only take the OOV words themselves as the option in positive samples selection, in other words, the value of λ_{unc} is set as 1 under the condition of $\lambda_{sim} = \lambda_{syn} = 0$.

Figure 6 shows ablation experimental results in different tasks, as we can see, the absence of any component will affect the performance of GRM. The effect of GRM w/o Readout slipped dramatically on all tasks, which validates the importance of obtaining a graph-level representation instead of a node-level one. But it is worth to mention that the quality of the node-level representation is also convincing, since GRM w/o Readout achieved a quite competitive result when com-

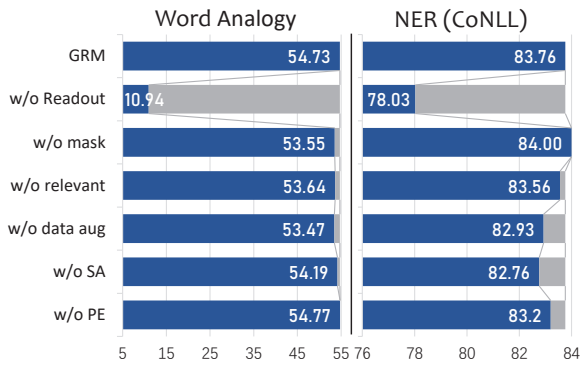

Figure 6: The experimental results of ablation study.

pared to baselines on the NER task. In addition, the mask operations brought some gains on word analogy, which illustrates the mask operation can force GRM model to uncover deeper relationships inside word structures. The GRM model without relevant words had a slight dip in performance, which indicates that associating related words can compensate for some semantic information. More results about the threshold δ_{sec} settings are presented in Appendix C.1. The performance of GRM w/o SA and GRM w/o PE were slightly lower than the GRM model because the SA component provides a reasonable initialization for GRM, while PE makes up some sequential information lost by graph structure. Besides, the GRM w/o data aug also caused a drop in performance in all tasks, which illustrates the strategy of utilizing graph structure for positive pairs selection makes the embeddings more semantically.

5 Model Feasibility for Other Languages

In this section, we discuss the feasibility of our GRM model to other languages. In the foregoing section, we introduced our GRM model that splits OOV words into wordpieces, then constructs WRG around OOV words, which can associate relevant words through wordpieces. Due to the design of the model, our GRM matches with the properties of an agglutinative language, such as Japanese or Korean, which forms words by stringing morphemes together directly. Fusional language is more difficult to process than agglutinative one because the morphemes are usually linked together. The language explored in our paper, English, is a fusional language with some agglutinative properties. It can be observed that GRM performs quite well on the fusional language by reasonable segmentation of words, which indicates that the application effectiveness of GRM to other languages depends on the rationality of word decomposition only. Theoretically, the graph structure of WRG in GRM can cope with various complex word formations, thus GRM can infer high-quality embeddings for OOV words though capturing the relational information inside the word structures and associating other relevant words.

6 Conclusion

In this paper, we present a graph-based method named GRM for OOV word embedding learning. We creatively propose to model word formation through using WRG which can help to mine relationships inside word structures and associate relevant words. We demonstrate our superiority over baseline models through word similarity, word analogy, NER, and POS tagging tasks. Besides, our GRM model can be easily incorporated into static and contextual pre-trained embedding models, and help them alleviate the OOV problem effectively. Furthermore, on the qualitative analysis, we observe that GRM can discover the semantic relational information between words, which validates the ability of GRM to recover relationship information between words. Our code and supplementary materials are available in public at: https://github.com/liangzrtvjivo/GRM.

Limitation

The GRM model still has some limitations. Even though our model brings some performance improvement to the contextual word embedding model (i.e., BERT), this improvement is relatively small compared to the static model. In some cases, GRM may hurt the performance of BERT slightly, because the primary objective of context-based word embedding models is to infer word meaning from contexts. The approach set forward in our study enhances their initial input word embeddings through word formation, and the benefits brought by this method are modest. How to efficiently improve the performance of contextual word embedding models when faced with OOV words remains to be explored.

Acknowledgements

We sincerely appreciate all reviewers for their constructive comments and suggestions. This work has been supported by the National Natural Science Foundation of China (61972426).

References

Eneko Agirre, Enrique Alfonseca, Keith B. Hall, Jana Kravalova, Marius Pasca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 19–27.

Stephen R. Anderson. 1992. *A-Morphous Morphology*. Cambridge University Press.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomás Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Hicham El Boukkouri, Olivier Ferret, Thomas Lavergne, Hiroshi Noji, Pierre Zweigenbaum, and Jun'ichi Tsujii. 2020. CharacterBERT: Reconciling ELMo and BERT for word-level open-vocabulary representations from characters. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6903–6915.

Elia Bruni, Nam-Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, 49:1–47.

Shaosheng Cao, Wei Lu, Jun Zhou, and Xiaolong Li. 2018. Cw2vec: Learning Chinese word embeddings with stroke n-gram information. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 5053–5061.

Hong-You Chen, Sz-Han Yu, and Shou-de Lin. 2020a. Glyph2Vec: Learning Chinese out-of-vocabulary word embedding from glyphs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2865–2871.

Lihu Chen, Gaël Varoquaux, and Fabian M. Suchanek. 2022. Imputing out-of-vocabulary embeddings with LOVE makes language models robust with little cost. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3488–3504.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020b. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 1597–1607.

David Denison. 1997. The cambridge encyclopedia of the english language. *Journal of Linguistics*, 33(1):171–212.

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

Rezarta Islamaj Dogan, Robert Leaman, and Zhiyong Lu. 2014. NCBI disease corpus: A resource for disease name recognition and concept normalization. *Journal of Biomedical Informatics*, 47:1–10.

Nobukazu Fukuda, Naoki Yoshinaga, and Masaru Kitsuregawa. 2020. Robust backed-off estimation of out-of-vocabulary embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4827–4838.

Nicolas Garneau, Jean-Samuel Leboeuf, and Luc Lamontagne. 2018. Predicting and interpreting embeddings for out of vocabulary words in downstream tasks. In *Proceedings of the Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 331–333.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. 2011. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 42–47.

Guy Halawi, Gideon Dror, Evgeniy Gabrilovich, and Yehuda Koren. 2012. Large-scale learning of word relatedness with constraints. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1406–1414.

Aurélie Herbelot and Marco Baroni. 2017. High-risk learning: Acquiring new word vectors from tiny data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 304–309.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. SimLex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, Chunjie Wang, and Jie Tang. 2022. GraphMAE: Self-supervised masked graph autoencoders. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 594–604.

Ziniu Hu, Ting Chen, Kai-Wei Chang, and Yizhou Sun. 2019. Few-shot representation learning for out-of-vocabulary words. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 4102–4112.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991.

Josefsson, Gunlög. 1997. *On the principles of word formation in Swedish*. Ph.D. thesis, Lund University.

Alexandre Kabbach, Kristina Gulordava, and Aurélie Herbelot. 2019. Towards incremental learning of word embeddings using context informativeness. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 162–168.

Wei-Tsung Kao and Hung-yi Lee. 2021. Is BERT a cross-disciplinary knowledge learner? A surprising finding of pre-trained models' transferability. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2195–2208.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations*.

Martin Krallinger, Florian Leitner, Obdulia Rabal, Miguel Vazquez, Julen Oyarzabal, and Alfonso Valencia. 2015. CHEMDNER: The drugs and chemical names extraction challenge. *Journal of Cheminformatics*, 7(S-1):S1.

Yuri Kuratov and Mikhail Y. Arkhipov. 2019. Adaptation of deep bidirectional multilingual transformers for russian language. *CoRR*, abs/1905.07213.

Thang Luong, Richard Socher, and Christopher D. Manning. 2013. Better word representations with recursive neural networks for morphology. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 104–113.

Marie-Catherine De Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D. Manning. 2014. Universal stanford dependencies: A cross-linguistic typology. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, pages 4585–4592.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *5th International Conference on Learning Representations*.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations*.

Anmol Nayak, Hariprasad Timmapathini, Karthikeyan Ponnalagu, and Vijendran Gopalan Venkoparao. 2020. Domain adaptation challenges of BERT in tokenization and sub-word representations of out-of-vocabulary words. In *Proceedings of the First Workshop on Insights from Negative Results in NLP*, pages 1–5.

Hoang NT and Takanori Maehara. 2019. Revisiting graph neural networks: All we have is low-pass filters. *CoRR*, abs/1905.09550.

Yuval Pinter, Robert Guthrie, and Jacob Eisenstein. 2017. Mimicking word embeddings using subword rnns. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 102–112.

Radim Řehůřek and Petr Sojka. 2010. Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50.

Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. 2011. Named entity recognition in tweets: An experimental study. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning*, pages 142–147.

Shota Sasaki, Jun Suzuki, and Kentaro Inui. 2019. Subword-based compact reconstruction of word embeddings. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3498–3508.

Timo Schick and Hinrich Schütze. 2019. Attentive mimicking: Better word embeddings by attending to informative contexts. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 489–494.

Timo Schick and Hinrich Schütze. 2020. Rare words: A major problem for contextualized embeddings and how to fix it by attentive mimicking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8766–8774.

Larry Smith, Lorraine K. Tanabe, Rie Johnson nee Ando, Cheng-Ju Kuo, I-Fang Chung, Chun-Nan Hsu, Yu-Shi Lin, Roman Klinger, Christoph M. Friedrich, Kuzman Ganchev, Manabu Torii, Hongfang Liu, Barry Haddow, Craig A. Struble, Richard J. Povinelli, Andreas Vlachos, William A. Baumgartner, Lawrence Hunter, Bob Carpenter, Richard Tzong-Han Tsai, Hong-Jie Dai, Feng Liu, Yifei Chen, Chengjie Sun, Sophia Katrenko, Pieter Adriaans, Christian Blaschke, Rafael Torres, Mariana Neves, Preslav Nakov, Anna Divoli, Manuel Maña-López, Jacinto Mata, and W. John Wilbur. 2008. Overview of BioCreative II gene mention recognition. *Genome Biology*, 9(S-2):S2.

Zijun Sun, Xiaoya Li, Xiaofei Sun, Yuxian Meng, Xiang Ao, Qing He, Fei Wu, and Jiwei Li. 2021. ChineseBERT: Chinese pretraining enhanced by glyph and pinyin information. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 2065–2075.

Laurens van der Maaten and Geoffrey E. Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pages 5998–6008.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *6th International Conference on Learning Representations*.

Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 9929–9939.

Chih-Hsuan Wei, Yifan Peng, Robert Leaman, Allan Peter Davis, Carolyn J. Mattingly, Jiao Li, Thomas C. Wiegers, and Zhiyong Lu. 2016. Assessing the state of the art in biomedical relation extraction: Overview of the biocreative V chemical-disease relation (CDR) task. *Database - The Journal of Biological Databases and Curation*, 2016.

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

Min-Sub Won, YunSeok Choi, Samuel Kim, Cheol-Won Na, and Jee-Hyong Lee. 2021. An embedding method for unseen words considering contextual information and morphological information. In *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, pages 1055–1062.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Ye Zhang and Byron C. Wallace. 2017. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing*, pages 253–263.

Yun Zhang, Yongguo Liu, Jiajing Zhu, Ziqiang Zheng, Xiaofeng Liu, Weiguang Wang, Zijie Chen, and Shuangqing Zhai. 2019. Learning Chinese word embeddings from stroke, structure and pinyin of characters. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1011–1020.

Jinman Zhao, Sidharth Mudgal, and Yingyu Liang. 2018. Generalizing word embeddings using bag of subwords. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 601–606.

A Dataset Statistics

In this section, we illustrate specific information about the intrinsic and extrinsic datasets used in our experiments. Table A1 shows the number of word pairs in intrinsic datasets and Table A2 shows a summary of extrinsic datasets, including the size of texts and the rate of OOV words, in which, the calculation of OOV rates is based on the Word2Vec model with 397,585 words. For the extrinsic datasets, the CoNLL and UD datasets are normal and widely-used datasets, which have some OOV words, while the other datasets have high OOV rates. The BC2GM, BC4Chemd, BC5CDR, and NCBI datasets are biomedical datasets that contain many domain-special words in the biological field. The ARK and Ritter datasets are Twitter datasets, for which, people tend to coin many new words in the Twitter application.

Datasets	RareWord	MEN	SimLex	Rel353	simverb	muturk	Google
#Word Pairs	2,034	3,000	999	252	3,500	771	19,544

Table A1: Statistics of intrinsic datasets.

B Experimental Settings

In this section, we present more detailed experimental settings about downstream models, word-piece embeddings, and training details.

B.1 Downstream Models

We use the gensim (Řehůřek and Sojka, 2010) package for intrinsic tasks. In the situation where static word embedding models are employed as background model, we use a convolutional neural

Datasets	#Train	#Val	#Test	OOV%	
				word	type
CoNLL	14,986	3,466	3,684	32.09%	57.99%
BC2GM	12,574	2,519	5,038	15.68%	55.85%
BC4Chemd	30,682	30,639	26,364	15.03%	63.07%
BC5CDR	4,560	4,581	4,797	12.97%	39.35%
NBCI	5,432	923	940	15.99%	38.25%
UD	12,543	2,002	2,077	17.13%	43.22%
ARK	1,000	327	500	38.95%	53.69%
Ritter	551	118	118	30.51%	62.12%

Table A2: Statistics of extrinsic datasets. The number associated with each dataset is the number of sentences used for training, validation, or testing. The word OOV% represents the OOV word rate in each dataset. The type OOV% represents the OOV word rate in vocabulary of each dataset.

network (Zhang and Wallace, 2017) for text classification tasks, a BiLSTM model with one CRF layer on top (Huang et al., 2015) for NER tasks, and a two-layer LSTM model (Pinter et al., 2017) for POS tagging tasks. And in the situation where contextual word embedding models are employed as background model, we use BERT (Wolf et al., 2020) with a CRF layer[2] on top for the NER evaluation tasks and BERT with a token classification layer on top (Wolf et al., 2020) for POS tagging evaluation tasks.

B.2 Pre-training Wordpiece Embeddings

To obtain the pre-trained embeddings of wordpiece nodes, we tokenized the corpus of background model using WordPiece (Wu et al., 2016) and put the processed corpus into the skipgram model (Mikolov et al., 2013) in the case where the background word embedding model is Word2Vec. We trained the skipgram model with gensim (version 4.1.2) (Řehůřek and Sojka, 2010). The experimental setting of the skipgram model is the same as the background Word2Vec model. And for the case where the background word embedding model is BERT, we directly used the pre-trained token embedding of BERT as our pre-trained embeddings of wordpieces. For the lack of tokens, we generated them by summing the sub-token embeddings to represent their composition.

B.3 Training Details

We use the same background model and synonyms as our GRM model to train the context-free baselines. However, the context-based baselines need an extra training corpus. HiCE is trained with

[2]https://pypi.org/project/TorchCRF/

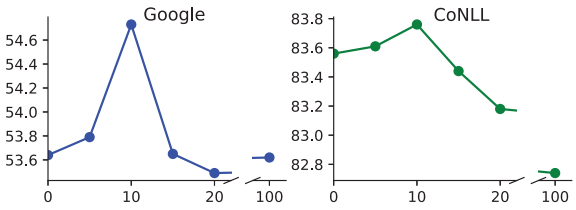

Figure A1: The different results of various δ_{sec} in word analogy and NER tasks.

WikiText-103 (Merity et al., 2017), which is used in their published experimental setting, while AM was trained with the Wikipedia snapshot of 2019, which is the original corpus of Word2Vec. The embedding dimension of our model depends on the word embedding dimension of the background model. Particularly, the embedding dimension of the Word2Vec background model is 400, and that of the BERT background model is 768. We conduct extensive experiments on several widely-used text datasets varying in scale to evaluate our work. All results are reported with a fixed seed.

C Parameter Settings of GRM

In this section, we discuss the influence of parameter settings and the selection of parameters.

C.1 Impact of Parameter δ_{sec}

We finetune the value of threshold δ_{sec} in the second layer of WRG to check the influence of word association. As shown in Figure A1, we can find that the performance of our GRM model on the Google dataset and CoNLL dataset gradually enhances as the value of δ_{sec} increases, and it reaches the peak when the threshold $\delta_{sec} = 10$. That means, the semantically related words in the second layer do provide information for OOV words. The method utilizing lexical rules to associate relevant words complements the semantic information of OOV words. Then the performance decreases when the threshold δ_{sec} is greater than 10, especially for the CoNLL dataset. This phenomenon proves that with the increase of threshold, some wordpiece nodes that contain less semantic information will include some irrelevant words, which introduce noise to our GRM model and have a negative impact on extrinsic evaluations. Notably, the noise introduced by word nodes in the second layer does not significantly affect the overall performance since GAT can reduce the influence of noise to some extent.

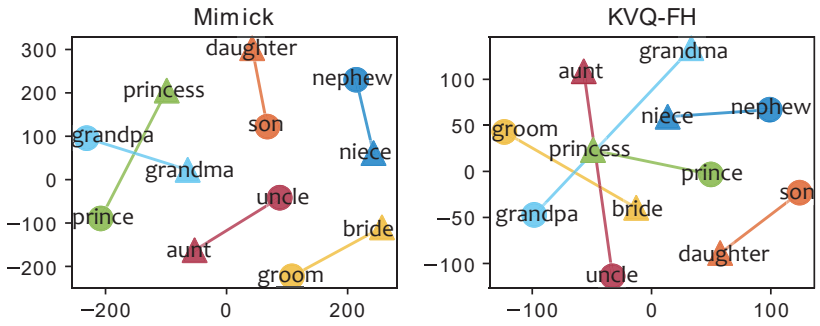

Figure A2: Visualization results of word embeddings generated by Mimick and KVQ-FH on the word analogy task, where "△" denotes the female word and "◯" denotes the male word in the word pair.

C.2 Hyper-parameters of GRM

We train the GRM model for 5 epochs in total. For the other hyper-parameters of GRM, we use grid search to determine the best value, the result is shown in Table A3. In the tasks with Word2Vec as the background model, we train GRM with five learning rates $\{5e-3, 3e-3, 1e-3, 8e-4, 5e-4\}$ and select the best one to report results. And in the tasks with BERT as the background model, we train GRM with five learning rates $\{1e-3, 8e-4, 5e-4, 3e-4, 1e-4\}$ and select the best results to report.

Hyper-parameter	Word2Vec-based		BERT-based NER		BERT-based POS	
	Range	Value	Range	Value	Range	Value
$\|\mathcal{B}\|$	[64,128,256]	256	[64,128,256]	128	[64,128,256]	64
λ_{syn}	[0.1,0.2,0.3]	0.2	[0.1,0.2,0.3]	0.2	[0.1,0.2,0.3]	0.2
λ_{unc}	[0.1,0.2,0.3]	0.2	[0.1,0.2,0.3]	0.2	[0.1,0.2,0.3]	0.2

Table A3: Grid search results of all hyper-parameters in GRM on different background models, i.e., Word2Vec and BERT. $|\mathcal{B}|$ is the batch size of the GRM model. We fix the values of λ_{rel} as 0.7, and adjust the values of λ_{syn} and λ_{unc} to find the best choice of positive samples proportions.

D Additional Results

D.1 Qualitative Analysis

Due to the limited space, we provide qualitative analysis on the rest of baselines (i.e., Mimick and KVQ-FH) in this section. As shown in Figure A2, the result of the Mimick model seems to be overlapping, but two parallel pairs have opposite distributions. For example, the gender positions of "aunt-uncle" and "groom-bride" are opposite. The result of the KVQ-FH model is apparently inconsistent with the word semantics. This demonstrates that modelling word formation by characters or sub-units cannot achieve good results in the word analogy task, because of the lost relationship information inside word structures.

D.2 Visualization of GAT Weights

As mentioned before, the GAT can emphasis the most important information and reduce the impact of noise wordpiece nodes for OOV words. To illustrate the action of two-layer GAT, we show an OOV example *insulinomimetic* chosen from the BC2GM dataset and visualize the attention weights on each layer. As shown in Figure A3, we can find that the *insulin* wordpiece which contains the most important semantic information is not assigned a high weight in the first layer, while the proportion of *insulin* increased by more than half in the second layer. Understandably, the OOV word node *insulinomimetic* didn't have a reasonable embedding yet in the first layer, even though we provide a better initialization for it using a self-attention block. It is worth noting that, the *##ic* wordpiece accounts for the least part overall, which means our model can reduce noise caused by syntactic wordpieces.

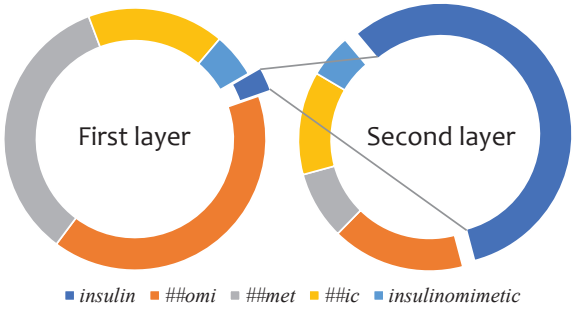

Figure A3: A case study of attention weights in different layers of GAT.

E Efficiency Analysis

In this section, we demonstrate the running time of GRM. We train the GRM model for 5 epochs in to-

tal when the Word2Vec model is taken as the background word embedding model. The word vocabulary size of the background Word2Vec model is 397,585. Each epoch consumes 1.8 hours on a workstation equipped with an Intel(R) Xeon(R) Silver 4214R CPU @ 2.40GHz and an Nvidia RTX 1080-Ti GPU. Besides, almost half of the time is spent on sampling the second layer of nodes in WRG, which consumes CPU time instead of GPU time.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
We discuss the limitations of our work in the Limitation section.

☑ A2. Did you discuss any potential risks of your work?
We treat the potential risks as limitations and discuss them in the Limitation section.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
We summarize the main claims of our work in the Abstract and Introduction sections.

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

We novelly propose a new scientific artifact described in Section 3. And we use some scientific artifacts in experiments, which are discussed and cited in Section 4.

☑ B1. Did you cite the creators of artifacts you used?
We cite the creators of the used artifacts in Section 4.1.

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
We will discuss the license or terms of the artifacts in the ReadME file of our code, which will be released upon publication.

☒ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
We will discuss the intended use of the artifacts in the ReadME file of our code, which will be released upon publication.

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
The datasets we use are all public datasets and there are no relevant sensitive information issues, thus we didn't discuss this problem in our work.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
We report the language and basic information about the artifacts in Section 4, Appendix A, and Appendix B.1.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
We report relevant statistics in detail in Appendix A.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

C ☑ Did you run computational experiments?

We report the setting and results of computational experiments in Section 4, Appendix B, Appendix C, and Appendix D.

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
We report the number of parameters in the models used in Section 4.2. And we report the details about the total computational time and the computing infrastructure in Appendix E.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
We discuss the experiment settings in Section 4.1 and Appendix B. And we discuss the parameter settings including hyperparameters in Appendix C.

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
We report summary statistics from sets of experiments in Section 4 and report experimental settings in Appendix B, which is about the details of reporting results.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
We report the used existing packages for preprocessing and evaluation in Section 4.1 and Appendix B.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.