applied sciences

MDPI

Article

Neuro-Symbolic Word Embedding Using Textual and Knowledge Graph Information

Dongsuk Oh †, Jungwoo Lim † and Heuiseok Lim *

Department of Computer Science and Engineering, Korea University, 145 Anam-ro, Seongbuk-gu, Seoul 02841, Korea

* Correspondence: limhseok@korea.ac.kr

† These authors contributed equally to this work.

Abstract: The construction of high-quality word embeddings is essential in natural language processing. In existing approaches using a large text corpus, the word embeddings learn only sequential patterns in the context; thus, accurate learning of the syntax and semantic relationships between words is limited. Several methods have been proposed for constructing word embeddings using syntactic information. However, these methods are not trained for the semantic relationships between words in sentences or external knowledge. In this paper, we present a method for improved word embeddings using symbolic graphs for external knowledge and the relationships of the syntax and semantic role between words in sentences. The proposed model sequentially learns two symbolic graphs with different properties through a graph convolutional network (GCN) model. A new symbolic graph representation is generated to understand sentences grammatically and semantically. This graph representation includes comprehensive information that combines dependency parsing and semantic role labeling. Subsequently, word embeddings are constructed through the GCN model. The same GCN model initializes the word representations that are created in the first step and trains the relationships of ConceptNet using the relationships between words. The proposed word embeddings outperform the baselines in benchmarks and extrinsic tasks.

Keywords: neuro-symbolic; graph convolutional network; word embedding; dependency parsing; semantic role labeling; ConceptNet; external knowledge

Citation: Oh, D.; Lim, J.; Lim, H. Neuro-Symbolic Word Embedding Using Textual and Knowledge Graph Information. *Appl. Sci.* **2022**, *12*, 9424. https://doi.org/10.3390/app12199424

Academic Editor: Valentino Santucci

Received: 21 August 2022
Accepted: 18 September 2022
Published: 20 September 2022

1. Introduction

Meaningful word embeddings using deep learning can effectively improve the performance of various natural language processing (NLP) tasks, such as syntax analysis [1,2], semantic analysis [3–5], and question-answering systems [6]. In previous studies, co-occurrence words were mainly learned in a sequential context. Such word embedding methods exhibit limitations because the syntactic relationships between words in the sentences are not considered. Various methods using syntactic analysis for constructing word embeddings have been proposed to overcome these limitations [7,8]. However, these methods require further investigation into understanding the relationships of the semantic roles between words [9] and external knowledge.

WordNet [10] is a set of semantic lexicons in English. Each vocabulary is divided into nouns, verbs, adjectives, and adverbs and is classified into a group of synonyms known as a synset. Furthermore, this lexical knowledge graph focuses on formal classifications such as synonyms, antonyms, hypernyms, and hyponyms. In contrast, ConceptNet [11] is a lexical knowledge graph that is constructed from various lexical resources such as WordNet [10], Wiktionary [12], and DBpedia [13]. Therefore, WordNet focuses on the formal classifications of words, whereas ConceptNet defines the richer semantic relationships between words by combining various lexical resources.

In this study, word embeddings are constructed sequentially through the neuro-symbolic learning of textual and knowledge graph information. A new graph representation is obtained to understand the syntactic and semantic relationships of the sentences. The example in Figure 1 depicts the graphs of the dependency parsing (DP) [14] and semantic role labeling (SRL) [9] of "hot spot uses technology developed in the military for tank and jet fighter tracking." The DP represents all syntactic trees of a sentence for the syntax-aware model. The SRL enables the model to understand the predicate–argument relationships within sentences. In our proposed method, these two graphs improve the word embeddings through a graph convolutional network (GCN) [15,16]. As word embeddings with symbolic information of the textual graph do not understand the relationships that are defined in the external knowledge graph, the model is trained using ConceptNet. The same architecture as that of the GCN is used in this process, and the word embeddings that are created by the first method are initialized.

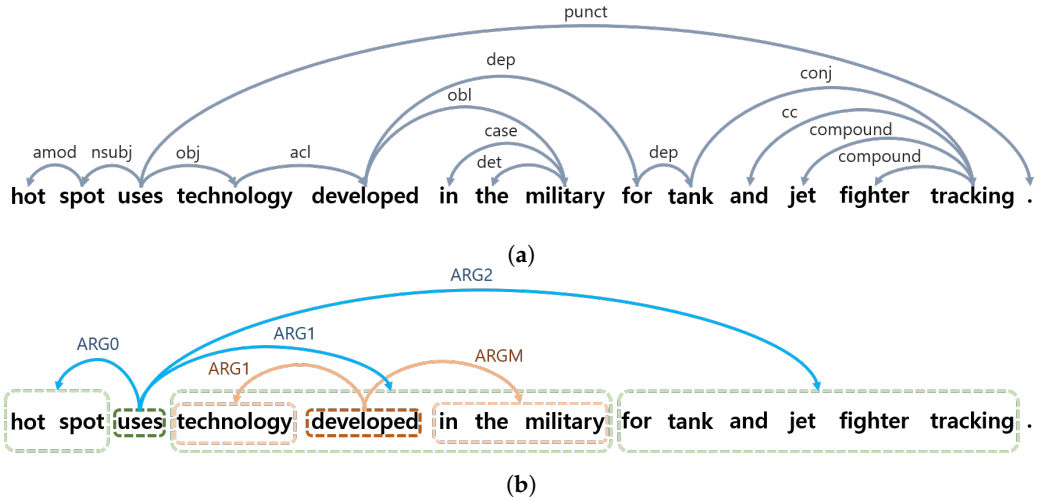

Figure 1. Examples of dependency parsing (DP) and semantic role labeling (SRL). In DP, any dependencies between words in a sentence are represented in a graph structure. In SRL, arguments are labeled with BIO tags according to predicates. For example, for the predicate "use" and the linked argument "hot spot", "hot" is B-ARG0 and "spot" is I-ARG0. (**a**) Example of dependency parsing. (**b**) Example of semantic role labeling.

The contributions of this study are summarized as follows:

- The proposed model sequentially trains on the symbolic information of the textual and external knowledge graphs to construct meaningful word embeddings.
- The proposed word embeddings (https://drive.google.com/file/d/1XXPYj47onzAx--DVvsIgib0HxzGd7wBR/view?usp=sharing, accessed on 21 August 2022) outperform those of previous methods on word embedding benchmarks.
- Our word embeddings further improve the model performance in extrinsic tasks (parts-of-speech tagging (POS) [1], named entity recognition (NER) [17], word sense disambiguation (WSD) [18–22], and the Stanford Question Answering Dataset (SQuAD) 1.1) [23].

2. Related Works

Word Embedding

Thus far, research on word embeddings has received substantial attention in NLP. The continuous bag-of-words (CBOW) and skip-gram (SG) [24] methods are based on language modeling using neural networks to represent words as real-valued vectors. However, these methods do not consider word co-occurrences in sentences. The global vectors for word representation (GloVe) [25] approach extends the use of global statistical information. FastText [26] uses subwords to solve problems such as out-of-vocabulary (OOV) words.

In recent years, language models have been proposed that represent words as different real-valued vectors according to the context. The Embeddings from Language Model (ELMo) [27] is an LSTM-based bidirectional language model, whereas bidirectional encoder representations from Transformers (BERT) [28] is a Transformer-based bidirectional language model. BERT is pretrained with a large corpus through masked-language modeling (MLM) and next sentence prediction (NSP) objectives. These contextual representation models have achieved high performance on various NLP tasks.

However, as these methods do not learn syntactic information, the understanding of the relationships between words in a sentence is limited. In a study by Levy and Goldberg [7], the encoding of syntactic relationships between words demonstrated that functional similarity exhibited higher performance than topical similarity. Another study revealed improved performance when words were represented with syntactic information using second-order [29] and multi-order [30] dependencies. It is necessary to label a large corpus automatically when employing modes that use syntactic information. Therefore, these models exhibit errors that are propagated as a result of data that are constructed in this process. To address this issue, Vashishth et al. [8] used an edge label gating mechanism to enhance the performance compared to previous models. However, models that learn only syntactic information experience difficulty in understanding semantic correlations such as external knowledge or the relations of predicate arguments among words.

Semantic Role Labeling (SRL)

SRL [9] uses the concept of the predicate–argument relation. Each predicate is connected to various arguments, and the linked arguments play a corresponding semantic role. For example, in the sentence “Tom gives Susan a book", “gives” is a predicate, whereas “Tom", “Susan", and “a book” are arguments. “Tom” is classified as an agent from the arguments, and "a book” is an argument with the theme of “give” That is, when the user asks “What is Tom giving to whom?” the answer of the model is “a book” Furthermore, the answer to the question “Where is a book?” is “Susie", so “Susie” corresponds to a goal among the types of arguments. We use the model provided by AllenNLP (https://demo.allennlp.org/semantic-role-labeling, accessed on 11 August 2019) to construct the SRL graph.

Dependence Parsing (DP)

DP [14] is a method of representation in the syntax tree structure that defines the relations between words in a sentence. This method presupposes that a correlation exists between the words of a sentence that influence one another. The modifier is known as the “head” or “governor", and the word that receives the modifier is known as the “dependent” or “modifier” The sentence is represented by a tree structure when the relationship between words is established. We used CoreNLP (https://stanfordnlp.github.io/CoreNLP/depparse.html, accessed on 11 August 2019) provided by the Stanford NLP Group to construct the DP graph.

ConceptNet

ConceptNet [11] represents a factual assertion in the real world using two nodes (concepts) and a directional edge (relation). The nodes define words or phrases that appear in natural language sentences. As ConceptNet performs construction by collecting various knowledge bases, the nodes are represented differently according to the knowledge. The edges (relations) include lexical relation information and general common-sense information of humans. ConceptNet may also define multiple edges between two nodes. For example, the edges of “person” and “eat” are labeled as CapableOf and Desires, respectively. Therefore, knowledge information is used in the models for various ambiguities in each sentence.

3. Proposed Method

In this study, the GCN [15,16] is exploited to learn the graph representation for word embedding. The proposed model consists of two stages. First, the model trains a new graph representation that combines the DP and SRL graphs. Second, the model initializes the word representations that are constructed in the first process and learns the inner correlations within ConceptNet on the same structure.

The graph is defined as $G = (V, E, X)$, where the node set is $|V| = n$, and E represents the edge set. $X \in \mathbb{R}^{n \times d}$ means an input node with d-dimensionality. When nodes u and v exist, $E(Edge)$ from u to v is labeled as (u, v, l_{uv}). Moreover, as information does not always need to propagate in one direction, the reverse direction (u, v, l_{uv}^{-1}) is included in $E(Edge)$. The proposed model is depicted in Figure 2. The sentence s is composed of a set of words such as $s = w_0, w_1, \ldots, w_n$.

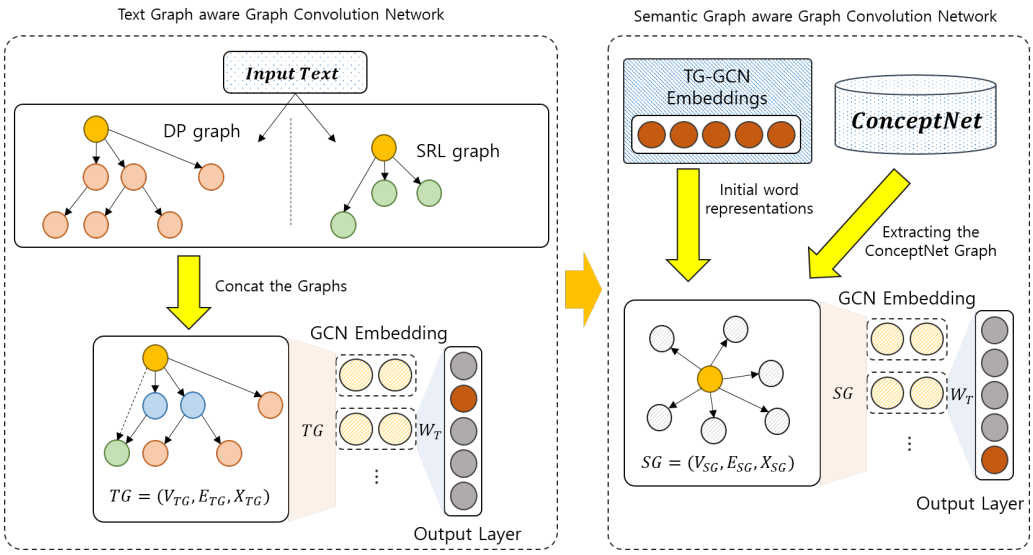

Figure 2. Overview of the proposed method. The model sequentially trains symbolic graphs of different properties in two steps.

TG-GCN

The text graph GCN (TG-GCN) performs training by combining the two graphs. Unlike in DP, the SRL information is predicted with a BIO tag on the sentence. Therefore, the defined tags are replaced with a new graph representation. For example, in Figure 1, the "hot spot" that is connected to "uses" is defined as hot (B-ARG0) and spot (I-ARG0). When this is represented as an edge, ("uses", "hot", B-ARG0), ("uses", "spot", I-ARG0) is expressed. Following the combination of the two graphs, the graph information is defined as $TG = (V_{TG}, E_{TG}, X_{TG})$. In Equation (1), h_v^{k+1} is represented through the k-GCN layer with the newly created graph representation as the input, where h_v^{k+1} is $h_v^{k+1} \in \mathbb{R}^d$.

$$h_v^{k+1} = f\Big(\sum_{u \in N_+(v)} (W_{l_{uv}}^k h_u^k + b_{l_{uv}}^k)\Big), \tag{1}$$

where $W_{l_{uv}}^k \in \mathbb{R}^{d \times d}$ and $b_{l_{uv}} \in \mathbb{R}^d$ denotes the model parameters for the relationship between nodes. Furthermore, the adjacency set of $N_+(v)$ (including v itself) and $h_v^k \in \mathbb{R}^d$ is the representation of node v through layer $k-1$.

When using an external model to generate graph structures from the corpus automatically, it is possible to include incorrect edges and labels within the graph structure. Error propagation occurs during the training process of the models because manual tagging is not performed. To address this limitation, our method calculates the edgewise gating score, as proposed by Marcheggiani and Titov [31]. The edge-label gating mechanism score is applied to the node representation in Equation (1) of the GCN. The score is calculated

independently for all connected edges of each node v, and the difference between the edges is distinguished. The score $g_{l_{uv}}^k \in \mathbb{R}$ is determined as indicated in Equation (2).

$$g_{l_{uv}}^k = \sigma(\hat{W}_{l_{uv}}^k h_u^k + \hat{b}_{l_{uv}}^k), \tag{2}$$

where $\hat{W}_{l_{uv}}^k \in \mathbb{R}^{1\times d}$ and $\hat{b}_{l_{uv}}^k \in \mathbb{R}$ is a learnable parameter. Furthermore, $\sigma(\cdot)$ is a sigmoid function. Finally, h_v^{k+1} with the edgewise gating mechanism is expressed by Equation (3).

$$h_v^{k+1} = f(\sum_{u\in N+(v)} g_{l_{uv}}^k \times (W_{l_{uv}}^k h_u^k + b_{l_{uv}}^k)) \tag{3}$$

CN-GCN

The word embeddings that are constructed by the TG-GCN only understand limited semantic relationships within the context. For example, the model merely understands whether the word "apple" is a company or food through the surrounding contexts "eat" or "computer", respectively. However, the model does not know whether "apple" is a hyponym for "fruit" The model can only understand such factual information using structured external knowledge. Therefore, our method creates meaningful word embeddings by learning the semantic relationships of ConceptNet. The ConceptNet GCN (CN-GCN) learns ConceptNet graphs to understand the semantic relationships of the words in a sentence with external knowledge. The inputs of the ConceptNet graphs are defined as $CN = (V_{CN}, E_{CN}, X_{CN})$. The model structure is the same as that of the TG-GCN, and it initializes the word embeddings that are constructed in the first step.

Output Layer

The output layers of TG-GCN and CN-GCN are updated in the same manner as in continuous bag-of-words (CBOW). The training objective L is defined by Equation (4):

$$L = \sum_{t=1}^{|V|} logP(w_t|w_1^t, \ldots, w_{N_t}^t), \tag{4}$$

where w_t is the target word and $w_1^t, \ldots, w_{N_t}^t$ is connected to neighboring nodes N in the graph.

$$P(w_t|w_1^t, \ldots, w_{N_t}^t) = \frac{exp(v_{w_t}^T h_t)}{\sum_{i=1}^{|V|} exp(v_{w_t}^T h_t)} \tag{5}$$

In Equation (5), $P(w_t|w_1^t, \ldots, w_{N_t}^t)$ is calculated using the softmax function. Furthermore, h_t is the GCN representation of the target word w_t, and v_{w_t} is the target embedding. However, L updates the weights to

$$L = \sum_{t=1}^{|V|} (v_{w_t}^T h_t - log \sum_{i=1}^{|V|} exp(v_{w_i}^T h_t)) \tag{6}$$

The second term in Equation (6) is computationally expensive because it requires summing up the total vocabulary V. The proposed method described by Mikolov et al. [32] uses negative sampling to reduce the computational cost.

4. Experimental Results and Analysis

4.1. Experimental Setup

The training corpus was obtained from the Wikipedia dump corpus provided by Vashishth et al. [8]. This corpus includes 57 million sentences with 1.1 billion tokens (https://dumps.wikimedia.org/enwiki/, accessed on 11 August 2019). The GCN proposed in this

study has the same parameters (https://github.com/malllabiisc/WordGCN, accessed on 11 August 2019) as that of Vashishth et al. [8]. The hardware used was a Geforce RTX 2080 Ti, and the learning time took about a week. The baseline on word embedding benchmarks evaluates by published word embeddings. However, we re-implemented the unpublished SemGCN. The effectiveness of the constructed word embeddings was evaluated using word embedding benchmarks and extrinsic tasks (WSD, SQuAD 1.1, POS, and NER).

4.1.1. Word Embedding Benchmarks

Word Similarity

Word similarity is defined as the closeness between words with similar meanings. The performance of the model was evaluated using Spearman's rank correlation on the tasks of Word Similarity-343 (WS353) [33], SimLex999 [34], Rare Word (RW) [35], and MEN3K [36].

Word Analogy

Given three words w_1, w_2, and c_1, word analogy analysis predicts c_2 for c_1 as the relationship between w_1 and w_2. We validated the model on the MSR [32] and SemEval-2012 [37] tasks.

4.1.2. Extrinsic Tasks

Word Sense Disambiguation (WSD)

WSD is a semantic analysis task in NLP, which involves classifying ambiguous words in a document or sentence correctly. We used the standard evaluation dataset of Raganato et al. [38], which consists of SenseEval2 (SE2) [18], SenseEval3 (SE3) [19], SemEval2007 (SE7) [20], SemEval2013 (SE13) [21], and SemEval2015 (SE15) [22]. The model was assessed using the F1-score.

SQuAD 1.1

SQuAD [23] is a question-answering task in NLP. Given a question and passage, the answers to all questions are a segment of text or span. SQuAD version 1.1, which is an extension of Rajpurkar et al. [23], was used in this study.

Part-of-Speech (POS) Tagging

POS is the task of labeling words in a sentence with parts-of-speech that play a syntactic role. We used the Penn Treebank POS dataset in the experiments [1].

Named Entity Recognition (NER)

The NER task involves classifying words that refer to entities in a sentence into categories such as PERSON, ORGANIZATION, LOCATION, and TIME. We evaluated the model using the CoNLL-2003 dataset [17].

4.2. Analysis of Word Embedding Benchmarks

In this section, the baseline models are described and the performance of the proposed method on the word embedding benchmarks is analyzed.

4.2.1. Baseline

Skip-Gram (SG)

In SG, which was proposed by Mikolov et al. [24], the representation of the target word is used to predict the context.

Continuous-Bag-of-Word (CBOW)

CBOW was proposed by Mikolov et al. [24]. The representations of the surrounding words are combined to predict the target word.

GloVe

GloVe [25] performs training with the word co-occurrence probability as an objective function.

FastText

FastText [26] considers subwords based on character n-grams and learns using the SG method to solve the OOV problem.

Deps

Deps [7] trains the dependencies between words into SG using the relationships of the DP tree as the input.

SynGCN

The SynGCN [8] performs learning as in the GCN to understand the relationships between words using the DP tree as the input. The output layer is the same as that in the CBOW method.

SemGCN

The SemGCN [8] initializes the word embeddings that are constructed from the SynGCN and learns the four relationships of WordNet (synonym, hypernym, hyponym, and antonym). The structure of the model is the same as that of the SynGCN.

BERT

BERT [28] is a pre-trained language model that uses the Transformer encoder. The model trains on a large corpus using training objectives such as MLM and NSP for pre-training. The MLM replaces (MASK) tokens with a 15% probability among the words in the input sentence and predicts the appropriate target word using the surrounding context. NSP determines whether the second sentence follows the first sentence. The model learns the relevance of two sentences through these training objectives. Base and large models are available according to the number of parameters.

4.3. Analysis

Table 1 presents the performance on word similarity and word analogy. The suggested word embeddings achieved the highest average score in both domains. The CN-GCN exhibited higher performance than the other models except for FastText in all word similarity tasks. FastText outperformed the CN-GCN in WS353, WS353R, RW, and MEN3K. These results demonstrate the advantages of the subwords used by FastText for solving the OOV problem. BERT, which uses subwords as a contextual representation model, exhibited low performance in the word embedding benchmarks when the dataset did not include sentences as the input.

Table 1. Task performance in the word similarity and word analogy domains on word embedding benchmarks. The overall performance for each domain is also presented as an average. The highest performance among the models is indicated in bold.

Model	**Word Similarity**							**Word Analogy**		
	WS353	**WS353S**	**WS353R**	**SimLex999**	**RW**	**MEN3K**	**Avg**	**MSR**	**SemEval12**	**Avg**
SG	61.0	68.9	53.7	34.9	34.5	67.0	53.3	30.6	20.5	25.6
CBOW	62.7	70.7	53.9	38.0	30.0	68.6	54.0	44.0	18.9	31.5
GloVe	54.2	64.3	50.2	31.6	29.9	68.3	49.8	**61.4**	16.9	39.2
FastText	**68.3**	74.6	**61.6**	38.2	**37.3**	**74.8**	59.1	53.2	19.5	36.4
Deps	60.6	73.1	46.8	39.6	33.0	60.5	52.3	40.3	22.9	31.6
BERT-base	51.8	60.1	37.3	48.1	26.7	42.8	44.5	52.7	20.8	36.8
BERT-large	57.8	65.8	46.4	47.5	27.6	51.8	49.5	57.6	20.7	39.2
OurModel(CN-GCN)	65.9	**78.8**	54.7	**57.0**	36.5	71.0	**60.7**	54.6	**24.4**	**39.5**

Our word embeddings achieved a higher average score than that of GloVe, but lower performance in the MSR of word analogy tasks. This demonstrates the limitation in understanding the relationships between words as the word co-occurrence probability was not reflected.

Table 2 displays the experimental results of the use of a symbolic graph. The TG-GCN, which learned a new graph representation in combination with the SRL graph, exhibited higher performance than the SynGCN. Furthermore, the CN-GCN achieved higher performance than the SemGCN, which was trained only on four relations of WordNet.

Table 2. Performance comparison of different methods using the symbolic graph on word embedding benchmarks. Bold indicates high performance.

Model	WS353	MSR
SynGCN	60.9	52.8
SemGCN	65.3	54.4
TG-GCN	62.7	54.0
Our model (CN-GCN)	**65.9**	**54.6**

4.4. Analysis of Extrinsic Tasks

This section presents a comparison of four tasks to analyze whether the proposed word embeddings were effective in the downstream task. The comparison models for each task are briefly described, and the performance when using the proposed embeddings is discussed.

4.4.1. Baselines

ELMo

ELMo [27] is a feature-based pretrained bidirectional language model based on LSTM. This model provides separate forward and backward language models and trains these using a weighted sum.

GAS

GAS [4] is a word disambiguation model. The model introduces a framework that uses reasoning to train dictionary information for ambiguous words.

BiDAF

BiDAF [6] is the most commonly used baseline model for question-answering tasks. This model was initially developed to perform reading comprehension tasks. It uses a bi-attention network between the passage and question for reading comprehension.

Baseline of POS and NER

We used the model of Lee et al. [5], which proves the efficiency of the word embeddings proposed by Vashishth et al. [8]. As this study also constructed word embeddings using symbolic information, we compared the performance on the same model.

4.5. Analysis

Table 3 presents the experimental results for the word disambiguation. SE7 data were used to validate the model during training, and the remaining tasks were used as the test data. All results were evaluated using the test set, including SE7. GAS achieved the highest performance when using the suggested word embeddings in all tasks.

Table 3. F1-score on fine-grained English all-word word sense disambiguation (WSD). Bold indicates high performance. The SE7 task was considered for all performances.

Model	Test Datasets				Concatenation of Test Datasets
	SE2	SE3	SE13	SE15	All
GAS (linear) with GloVe	72.0	70.0	66.7	71.6	70.1
GAS (concat) with GloVe	72.1	70.2	67.0	71.8	70.3
GAS_{ext} (linear) with GloVe	72.4	70.1	67.1	72.1	70.4
GAS_{ext} (concat) with GloVe	72.2	70.5	67.2	72.6	70.6
GAS_{ext} (concat) with CN-GCN	**75.6**	**73.5**	**71.4**	**73.5**	**71.0**

Table 4 displays the experimental results for SQuAD 1.1. Higher performance was achieved when using the proposed word embeddings.

Table 4. Performance evaluation on SQuAD 1.1 with BiDAF. The performance is compared for initialization using GloVe or CN-GCN embedding. Bold indicates high performance.

Model	EM-Dev (%)	F1-Dev (%)	EM-Test (%)	F1-Test (%)
BiDAF with GloVe	62.0	73.5	65.1	75.3
BiDAF with CN-GCN	**63.0**	**74.2**	**65.9**	**75.9**

The experimental results for POS and NER are presented in Table 5. In this experiment, the performance when using only ELMo embeddings and that when concatenating word embeddings using symbolic graphs were compared. Higher performance was achieved when embeddings using symbolic information were concatenated and the highest performance was exhibited when the proposed word embeddings were used.

Table 5. Performance when initialization was performed with each word embedding in parts-of-speech (POS) and named entity recognition (NER) tasks. Bold indicates high performance.

Embedding	POS	NER
ELMo	96.1	90.3
w/ Concat SemGCN embedding	96.2	90.9
w/ Concat CN-GCN embedding	**97.0**	**91.2**

5. Conclusions and Future Work

A method for constructing word embeddings using symbolic graphs with two properties was proposed. Our approach sequentially trains the syntactic and semantic relationships, as well as external knowledge using a GCN. The proposed method outperformed baseline models in word embedding benchmarks. Moreover, the performance of the model was improved when using the suggested word embeddings in each model for the extrinsic tasks. However, the proposed word embeddings still appeared to exhibit OOV problems. In future work, we will develop a graph representation containing nodes at the subword level and an efficient graph-based model.

Author Contributions: Conceptualization, software, investigation, methodology, writing—review and editing, writing—original draft: D.O., J.L.; investigation, validation, supervision, resources, project administration, and funding acquisition: H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by an Institute of Information & communications Technology Planning & Evaluation (IITP) grant, funded by the Korean government Ministry of Science and ICT (MSIT) (No. 2020-0-00368, A Neural-Symbolic Model for Knowledge Acquisition and Inference

Techniques) and by the MSIT, Korea, under the ICT Creative Consilience program (IITP-2022-2020-0-01819) supervised by the IITP. Furthermore, this research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF), funded by the Ministry of Education (NRF-2021R1A6A1A03045425).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: ConceptNet: https://conceptnet.io/, accessed on 11 August 2019.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Marcinkiewicz, M.A. Building a large annotated corpus of English: The Penn Treebank. In *Using Large Corpora*; MIT Press: Cambridge, MA, USA, 1994; p. 273.
2. Manning, C.D.; Surdeanu, M.; Bauer, J.; Finkel, J.R.; Bethard, S.; McClosky, D. The Stanford CoreNLP natural language processing toolkit. In Proceedings of the 52nd Annual Meeting of The Association for Computational Linguistics: System Demonstrations, Baltimore, MD, USA, 23–25 June 2014; pp. 55–60.
3. Shi, P.; Lin, J. Simple bert models for relation extraction and semantic role labeling. *arXiv* **2019**, arXiv:1904.05255.
4. Luo, F.; Liu, T.; Xia, Q.; Chang, B.; Sui, Z. Incorporating Glosses into Neural Word Sense Disambiguation. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Melbourne, Australia, 15–20 July 2018; pp. 2473–2482.
5. Lee, K.; He, L.; Zettlemoyer, L. Higher-Order Coreference Resolution with Coarse-to-Fine Inference. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), New Orleans, LA, USA, 1–6 June 2018; pp. 687–692.
6. Seo, M.; Kembhavi, A.; Farhadi, A.; Hajishirzi, H. Bidirectional attention flow for machine comprehension. *arXiv* **2016**, arXiv:1611.01603.
7. Levy, O.; Goldberg, Y. Dependency-based word embeddings. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Baltimore, MD, USA, 22–27 June 2014; pp. 302–308.
8. Vashishth, S.; Bhandari, M.; Yadav, P.; Rai, P.; Bhattacharyya, C.; Talukdar, P. Incorporating Syntactic and Semantic Information in Word Embeddings using Graph Convolutional Networks. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 3308–3318.
9. Palmer, M.; Gildea, D.; Xue, N. Semantic role labeling. *Synth. Lect. Hum. Lang. Technol.* **2010**, *3*, 1–103.
10. Miller, G.A. WordNet: A lexical database for English. *Commun. ACM* **1995**, *38*, 39–41. [CrossRef]
11. Speer, R.; Chin, J.; Havasi, C. Conceptnet 5.5: An open multilingual graph of general knowledge. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
12. Meyer, C.M.; Gurevych, I. *Wiktionary: A New Rival for Expert-Built Lexicons? Exploring the Possibilities of Collaborative Lexicography*; Oxford University Press: Oxford, UK, 2012.
13. Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; Ives, Z. Dbpedia: A nucleus for a web of open data. In *The Semantic Web*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 722–735.
14. Kübler, S.; McDonald, R.; Nivre, J. Dependency parsing. *Synth. Lect. Hum. Lang. Technol.* **2009**, *1*, 1–127.
15. Defferrard, M.; Bresson, X.; Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 3844–3852.
16. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
17. Sang, E.F.T.K.; De Meulder, F. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. *Development* **1837**, *922*, 1341.
18. Palmer, M.; Fellbaum, C.; Cotton, S.; Delfs, L.; Dang, H.T. English tasks: All-words and verb lexical sample. In Proceedings of the International Workshop on Evaluating Word Sense Disambiguation Systems, Toulouse, France, 5–6 July 2001; pp. 21–24.
19. Snyder, B.; Palmer, M. The English all-words task. In Proceedings of the Proceedings the International Workshop on the Evaluation of Systems for the Semantic Analysis of Text, Barcelona, Spain, 25–26 July 2004; pp. 41–43.
20. Pradhan, S.; Loper, E.; Dligach, D.; Palmer, M. Semeval-2007 task-17: English lexical sample, srl and all words. In Proceedings of the International Workshop on Semantic Evaluations, Prague, Czech Republic, 23–24 June 2007; pp. 87–92.
21. Navigli, R.; Jurgens, D.; Vannella, D. Semeval-2013 task 12: Multilingual word sense disambiguation. In Proceedings of the International Workshop on Semantic Evaluation, Atlanta, GA, USA, 14–15 June 2013; Volume 2, pp. 222–231.
22. Moro, A.; Navigli, R. Semeval-2015 task 13: Multilingual all-words sense disambiguation and entity linking. In Proceedings of the International Workshop on Semantic Evaluation, Denver, CO, USA, 4–5 June 2015; pp. 288–297.
23. Rajpurkar, P.; Zhang, J.; Lopyrev, K.; Liang, P. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Austin, TX, USA, 1–5 November 2016.
24. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.

25. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
26. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 135–146. [CrossRef]
27. Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep Contextualized Word Representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, LA, USA, 1–6 June 2018; pp. 2227–2237.
28. Kenton, J.D.M.W.C.; Toutanova, L.K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT), Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186.
29. Komninos, A.; Manandhar, S. Dependency based embeddings for sentence classification tasks. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 12–17 June 2016; pp. 1490–1500.
30. Li, C.; Li, J.; Song, Y.; Lin, Z. Training and evaluating improved dependency-based word embeddings. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
31. Marcheggiani, D.; Titov, I. Encoding Sentences with Graph Convolutional Networks for Semantic Role Labeling. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 9–11 September 2017; pp. 1506–1515.
32. Mikolov, T.; Yih, W.t.; Zweig, G. Linguistic regularities in continuous space word representations. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Atlanta, GA, USA, 9–14 June 2013; pp. 746–751.
33. Finkelstein, L.; Gabrilovich, E.; Matias, Y.; Rivlin, E.; Solan, Z.; Wolfman, G.; Ruppin, E. Placing search in context: The concept revisited. In Proceedings of the 10th International Conference on World Wide Web, Hong Kong, China, 1–5 May 2001; pp. 406–414.
34. Kiela, D.; Hill, F.; Clark, S.; et al. Specializing Word Embeddings for Similarity or Relatedness. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Lisbon, Portugal, 17–21 September 2015; pp. 2044–2048.
35. Luong, M.T.; Socher, R.; Manning, C.D. Better word representations with recursive neural networks for morphology. In Proceedings of the Seventeenth Conference On Computational Natural Language Learning, Sofia, Bulgaria, 8–9 August 2013; pp. 104–113.
36. Bruni, E.; Boleda, G.; Baroni, M.; Tran, N.K. Distributional semantics in technicolor. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Stroudsburg, PA, USA, 8–14 July 2012; pp. 136–145.
37. Jurgens, D.; Mohammad, S.; Turney, P.; Holyoak, K. Semeval-2012 task 2: Measuring degrees of relational similarity. In Proceedings of the SEM 2012: The First Joint Conference on Lexical and Computational Semantics–Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012), Stroudsburg, PA, USA, 7–8 June 2012; pp. 356–364.
38. Raganato, A.; Camacho-Collados, J.; Navigli, R. Word sense disambiguation: A unified evaluation framework and empirical comparison. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, Valencia, Spain, 3–7 April 2017; pp. 99–110.