Incorporating Syntactic and Semantic Information in Word Embeddings using Graph Convolutional Networks

Shikhar Vashishth[1] **Manik Bhandari**[1*] **Prateek Yadav**[2*]
Piyush Rai[3] **Chiranjib Bhattacharyya**[1] **Partha Talukdar**[1]
[1]Indian Institute of Science, Bangalore
[2]Microsoft Research, [3]IIT Kanpur
{shikhar,manikb,chiru,ppt}@iisc.ac.in
t-pryad@microsoft.com, piyush@cse.iitk.ac.in

Abstract

Word embeddings have been widely adopted across several NLP applications. Most existing word embedding methods utilize *sequential context* of a word to learn its embedding. While there have been some attempts at utilizing *syntactic context* of a word, such methods result in an explosion of the vocabulary size. In this paper, we overcome this problem by proposing SynGCN, a flexible Graph Convolution based method for learning word embeddings. SynGCN utilizes the dependency context of a word without increasing the vocabulary size. Word embeddings learned by SynGCN outperform existing methods on various intrinsic and extrinsic tasks and provide an advantage when used with ELMo. We also propose SemGCN, an effective framework for incorporating diverse semantic knowledge for further enhancing learned word representations. We make the source code of both models available to encourage reproducible research.

1 Introduction

Representing words as real-valued vectors is an effective and widely adopted technique in NLP. Such representations capture properties of words based on their usage and allow them to generalize across tasks. Meaningful word embeddings have been shown to improve performance on several relevant tasks, such as named entity recognition (NER) (Bengio et al., 2013), parsing (Socher et al., 2013), and part-of-speech (POS) tagging (Ma and Hovy, 2016). Using word embeddings for initializing Deep Neural Networks has also been found to be quite useful (Collobert et al., 2011; Johnson et al., 2017; Strubell et al., 2018).

Most popular methods for learning word embeddings are based on the distributional hypothesis, which utilizes the co-occurrence statistics from *sequential* context of words for learning word representations (Mikolov et al., 2013a; Pennington et al., 2014). More recently, this approach has been extended to include syntactic contexts (Levy and Goldberg, 2014) derived from dependency parse of text. Higher order dependencies have also been exploited by Komninos and Manandhar (2016); Li et al. (2018). Syntax-based embeddings encode functional similarity (in-place substitutable words) rather than topical similarity (topically related words) which provides an advantage on specific tasks like question classification (Komninos and Manandhar, 2016). However, current approaches incorporate syntactic context by concatenating words with their dependency relations. For instance, in Figure 1 *scientists_subj*, *water_obj*, and *mars_nmod* needs to be included as a part of vocabulary for utilizing the dependency context of *discover*. This severely expands the vocabulary, thus limiting the scalability of models on large corpora. For instance, in Levy and Goldberg (2014) and Komninos and Manandhar (2016), the context vocabulary explodes to around 1.3 million for learning embeddings of 220k words.

Incorporating relevant signals from semantic knowledge sources such as WordNet (Miller, 1995), FrameNet (Baker et al., 1998), and Paraphrase Database (PPDB) (Pavlick et al., 2015) has been shown to improve the quality of word embeddings. Recent works utilize these by incorporating them in a neural language modeling objective function (Yu and Dredze, 2014; Alsuhaibani et al., 2018), or as a post-processing step (Faruqui et al., 2014; Mrkšić et al., 2016). Although existing approaches improve the quality of word embeddings, they require explicit modification for handling different types of semantic information.

Recently proposed Graph Convolutional Networks (GCN) (Defferrard et al., 2016; Kipf and

[*] Contributed equally to the work.

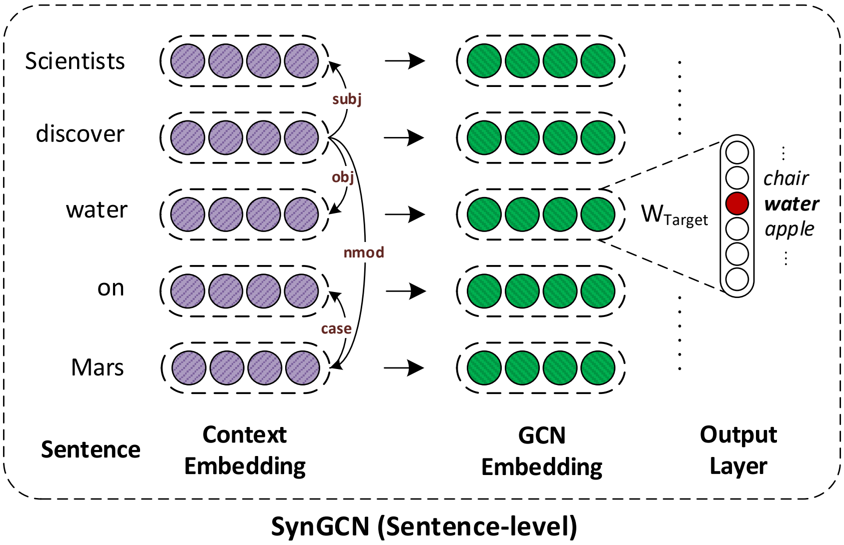

Figure 1: Overview of SynGCN: SynGCN employs Graph Convolution Network for utilizing dependency context for learning word embeddings. For each word in vocabulary, the model learns its representation by aiming to predict each word based on its dependency context encoded using GCNs. Please refer Section 5 for more details.

Welling, 2016) have been found to be useful for encoding structural information in graphs. Even though GCNs have been successfully employed for several NLP tasks such as machine translation (Bastings et al., 2017), semantic role labeling (Marcheggiani and Titov, 2017), document dating (Vashishth et al., 2018a) and text classification (Yao et al., 2018), they have so far not been used for learning word embeddings, especially leveraging cues such as syntactic and semantic information. GCNs provide flexibility to represent diverse syntactic and semantic relationships between words all within one framework, without requiring relation-specific special handling as in previous methods. Recognizing these benefits, we make the following contributions in this paper.

1. We propose SynGCN, a Graph Convolution based method for learning word embeddings. Unlike previous methods, SynGCN utilizes syntactic context for learning word representations without increasing vocabulary size.
2. We also present SemGCN, a framework for incorporating diverse semantic knowledge (e.g., synonymy, antonymy, hyponymy, etc.) in learned word embeddings, without requiring relation-specific special handling as in previous methods.
3. Through experiments on multiple intrinsic and extrinsic tasks, we demonstrate that our proposed methods obtain substantial improvement over state-of-the-art approaches, and also yield an advantage when used in conjunction with methods such as ELMo (Peters et al., 2018).

The source code of both the methods has been made available at `http://github.com/malllabiisc/WordGCN`.

2 Related Work

Word Embeddings: Recently, there has been much interest in learning meaningful word representations such as neural language modeling (Bengio et al., 2003) based continuous-bag-of-words (CBOW) and skip-gram (SG) models (Mikolov et al., 2013a). This is further extended by Pennington et al. (2014) which learns embeddings by factorizing word co-occurrence matrix to leverage global statistical information. Other formulations for learning word embeddings include multi-task learning (Collobert et al., 2011) and ranking frameworks (Ji et al., 2015).

Syntax-based Embeddings: Dependency parse context based word embeddings is first introduced by Levy and Goldberg (2014). They allow encoding syntactic relationships between words and show improvements on tasks where functional similarity is more relevant than topical similarity. The inclusion of syntactic context is further enhanced through second-order (Komninos and Manandhar, 2016) and multi-order (Li et al., 2018) dependencies. However, in all these existing approaches, the word vocabulary is severely expanded for incorporating syntactic relationships.

Incorporating Semantic Knowledge Sources: Semantic relationships such as *synonymy*, *antonymy*, *hypernymy*, etc. from several semantic sources have been utilized for improving the quality of word representations. Existing methods either exploit them jointly (Xu et al., 2014; Kiela et al., 2015; Alsuhaibani et al., 2018) or as a post-processing step (Faruqui et al., 2014; Mrkšić et al., 2016). SynGCN falls under the latter category and is more effective at incorporating semantic constraints (Section 9.2 and 9.3).

Graph Convolutional Networks: In this paper, we use the first-order formulation of GCNs via a layer-wise propagation rule as proposed by (Kipf and Welling, 2016). Recently, some variants of GCNs have also been proposed (Yadav et al., 2019; Vashishth et al., 2019). A detailed description of GCNs and their applications can be found in Bronstein et al. (2017). In NLP, GCNs have been utilized for semantic role labeling (Marcheggiani and Titov, 2017), machine translation (Bastings et al., 2017), and relation extraction (Vashishth et al., 2018b). Recently, Yao et al. (2018) use GCNs for text classification by jointly embedding words and documents. However, their learned embeddings are task specific whereas in our work we aim to learn task agnostic word representations.

3 Background: Graph Convolutional Networks

In this section, we will provide a brief overview of Graph Convolutional Networks (GCNs) (Defferrard et al., 2016; Kipf and Welling, 2016) and its extension to directed labeled graphs.

3.1 GCN on Directed Labeled Graphs

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{X})$ be a directed graph where $\mathcal{V}$ is the set of nodes ($|\mathcal{V}| = n$), $\mathcal{E}$ indicates the edge set, and $\mathcal{X} \in \mathbb{R}^{n \times d}$ denotes the d-dimensional *input* node features. An edge from node u to v with label l_{uv} is denoted by (u, v, l_{uv}). As the information need not always propagate only along the direction of the edge, following Marcheggiani and Titov (2017), we include inverse edges (v, u, l_{uv}^{-1}) in $\mathcal{E}$. Embedding $h_v^{k+1} \in \mathbb{R}^d$ of a node v after k-GCN layers is given as follows.

$$h_v^{k+1} = f\left(\sum_{u \in \mathcal{N}_+(v)} \left(W_{l_{uv}}^k h_u^k + b_{l_{uv}}^k\right)\right)$$

Here, $W_{l_{uv}}^k \in \mathbb{R}^{d \times d}$ and $b_{l_{uv}} \in \mathbb{R}^d$ are label specific model parameters, $\mathcal{N}_+(v) = \mathcal{N}(v) \cup \{v\}$ is the set of immediate neighbors of v (including v itself), and $h_u^k \in \mathbb{R}^d$ is hidden representation of node u after $k-1$ layers.

Edge Label Gating Mechanism: In real-world graphs, some of the edges might be erroneous or irrelevant for the downstream task. This is predominant in automatically constructed graphs like dependency parse of text. To address this issue, we employ edge-wise gating (Marcheggiani and Titov, 2017) in GCNs. For each node v, we calculate a relevance score $g_{l_{uv}}^k \in \mathbb{R}$ for all the edges in which v participates. The score is computed independently for each layer as shown below.

$$g_{l_{uv}}^k = \sigma\left(\hat{W}_{l_{uv}}^k h_u^k + \hat{b}_{l_{uv}}^k\right)$$

Here, $\hat{W}_{l_{uv}}^k \in \mathbb{R}^{1 \times d}$ and $\hat{b}_{l_{uv}}^k \in \mathbb{R}$ are trainable parameters and $\sigma(\cdot)$ is the sigmoid function. The updated GCN propagation rule for the k^{th} layer can be written as shown below.

$$h_v^{k+1} = f\left(\sum_{u \in \mathcal{N}_+(v)} g_{l_{uv}}^k \times \left(W_{l_{uv}}^k h_u^k + b_{l_{uv}}^k\right)\right) \quad (1)$$

4 Methods Overview

The task of learning word representations in an unsupervised setting can be formulated as follows: Given a text corpus, the aim is to learn a d-dimensional embedding for each word in the vocabulary. Most of the distributional hypothesis based approaches only utilize sequential context for each word in the corpus. However, this becomes suboptimal when the relevant context words lie beyond the window size. For instance in Figure 1, a relevant context word *discover* for *Mars* is missed if the chosen window size is less than 3. On the contrary, a large window size might allow irrelevant words to influence word embeddings negatively.

Using dependency based context helps to alleviate this problem. However, all existing syntactic context based methods (Levy and Goldberg, 2014; Komninos and Manandhar, 2016; Li et al., 2018) severely expand vocabulary size (as discussed in Section 1) which limits their scalability to a large corpus. To eliminate this drawback, we propose SynGCN which employs Graph Convolution Networks to better encode syntactic information in embeddings. We prefer GCNs over other graph encoding architectures such as Tree LSTM (Tai et al., 2015) as GCNs do not restrict graphs to be trees and have been found to be more effective at capturing global information (Zhang et al., 2018). Moreover, they give substantial speedup as they do not involve recursive operations which are difficult to parallelize. The overall architecture is shown in Figure 1, for more details refer to Section 5.

Enriching word embeddings with semantic knowledge helps to improve their quality for several NLP tasks. Existing approaches are either incapable of utilizing these diverse relations or need to be explicitly modeled for exploiting them. In this paper, we propose SemGCN which automatically learns to utilize multiple semantic constraints by modeling them as different edge types. It can be used as a post-processing method similar to Faruqui et al. (2014); Mrkšić et al. (2016). We describe it in more detail in Section 6.

5 SynGCN

In this section, we provide a detailed description of our proposed method, SynGCN. Following Mikolov et al. (2013b); Levy and Goldberg (2014); Komninos and Manandhar (2016), we separately define target and context embeddings for each word in the vocabulary as parameters in the model. For a given sentence $s = (w_1, w_2, \dots, w_n)$, we first extract its dependency parse graph $\mathcal{G}_s = (\mathcal{V}_s, \mathcal{E}_s)$ using Stanford CoreNLP parser (Manning et al., 2014). Here, $\mathcal{V}_s = \{w_1, w_2, \dots, w_n\}$ and $\mathcal{E}_s$ denotes the labeled directed dependency edges of the form (w_i, w_j, l_{ij}), where l_{ij} is the dependency relation of w_i to w_j.

Similar to Mikolov et al. (2013b)'s continuous-bag-of-words (CBOW) model, which defines the context of a word w_i as $\mathcal{C}_{w_i} = \{w_{i+j} : -c \leq j \leq c, j \neq 0\}$ for a window of size c, we define the context as its neighbors in $\mathcal{G}_s$, i.e., $\mathcal{C}_{w_i} = \mathcal{N}(w_i)$. Now, unlike CBOW which takes the sum of the context embedding of words in $\mathcal{C}_{w_i}$ to predict w_i, we apply directed Graph Convolution Network (as defined in Section 3) on $\mathcal{G}_s$ with context embeddings of words in s as input features. Thus, for each word w_i in s, we obtain a representation h_i^{k+1} after k-layers of GCN using Equation 1 which we reproduce below for ease of readability (with one exception as described below).

$$h_i^{k+1} = f\left(\sum_{j \in \mathcal{N}(i)} g_{l_{ij}}^k \times \left(W_{l_{ij}}^k h_j^k + b_{l_{ij}}^k\right)\right)$$

Please note that unlike in Equation 1, we use $\mathcal{N}(i)$ instead of $\mathcal{N}_+(i)$ in SynGCN, i.e., we do not include self-loops in $\mathcal{G}_s$. This helps to avoid overfitting to the initial embeddings, which is undesirable in the case of SynGCN as it uses random initialization. We note that similar strategy has been followed by Mikolov et al. (2013b). Furthermore, to handle erroneous edges in automatically constructed dependency parse graph, we perform edge-wise gating (Section 3.1) to give importance to relevant edges and suppress the noisy ones. The embeddings obtained are then used to calculate the loss as described in Section 7.

SynGCN utilizes *syntactic context* to learn more meaningful word representations. We validate this in Section 9.1. Note that, the word vocabulary remains unchanged during the entire learning process, this makes SynGCN more scalable compared to the existing approaches.

Note that, SynGCN is a generalization of CBOW model, as shown below.

Theorem 1. *SynGCN is a generalization of Continuous-bag-of-words (CBOW) model.*

Proof. The reduction can be obtained as follows. For a given sentence s, take the neighborhood of each word w_i in $\mathcal{G}_s$ as it sequential context, i.e., $\mathcal{N}(w_i) = \{w_{i+j} : -c \leq j \leq c, j \neq 0\}\ \forall w_i \in s$. Now, if the number of GCN layers are restricted to 1 and the activation function is taken as identity ($f(x) = x$), then Equation 1 reduces to

$$h_i = \sum_{-c \leq j \leq c, j \neq 0} \left(g_{l_{ij}} \times \left(W_{l_{ij}} h_j + b_{l_{ij}}^k\right)\right).$$

Finally, $W_{l_{ij}}^k$ and $b_{l_{ij}}^k$ can be fixed to an identity matrix ($\mathbf{I}$) and a zero vector ($\mathbf{0}$), respectively, and edge-wise gating ($g_{l_{ij}}$) can be set to 1. This gives

$$h_i = \sum_{-c \leq j \leq c, j \neq 0} (\mathbf{I} \cdot h_j + \mathbf{0}) = \sum_{-c \leq j \leq c, j \neq 0} h_j,$$

which is the hidden layer equation of CBOW model. □

6 SemGCN

In this section, we propose another Graph Convolution based framework, SemGCN, for incorporating semantic knowledge in pre-trained word

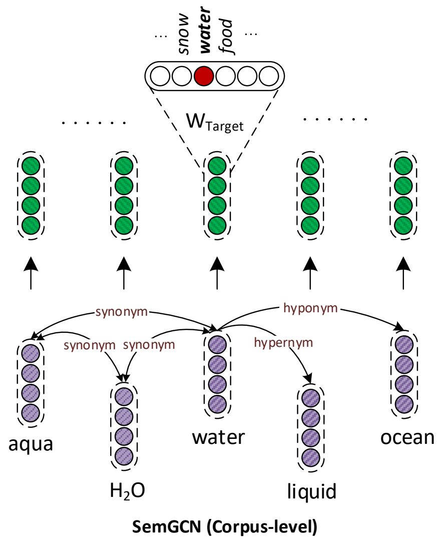

Figure 2: Overview of SemGCN, our proposed Graph Convolution based framework for incorporating diverse semantic information in learned embeddings. Double-headed edges denote two edges in both directions. Please refer to Section 6 for more details.

embeddings. Most of the existing approaches like Faruqui et al. (2014); Mrkšić et al. (2016) are restricted to handling symmetric relations like *synonymy* and *antonymy*. On the other hand, although recently proposed (Alsuhaibani et al., 2018) is capable of handling asymmetric information, it still requires manually defined relation strength function which can be labor intensive and suboptimal.

SemGCN is capable of incorporating both symmetric as well as asymmetric information jointly. Unlike SynGCN, SemGCN operates on a corpus-level directed labeled graph with words as nodes and edges representing semantic relationship among them from different sources. For instance, in Figure 2, semantic relations such as *hyponymy*, *hypernymy* and *synonymy* are represented together in a single graph. Symmetric information is handled by including a directed edge in both directions. Given the corpus level graph $\mathcal{G}$, the training procedure is similar to that of SynGCN, i.e., predict the word w based on its neighbors in $\mathcal{G}$. Inspired by Faruqui et al. (2014), we preserve the semantics encoded in pre-trained embeddings by initializing both target and context embeddings with given word representations and keeping target embeddings fixed during training. SemGCN uses Equation 1 to update node embeddings. Please note that in this case $\mathcal{N}_+(v)$ is used as the neighborhood definition to preserve the initial learned representation of the words.

7 Training Details

Given the GCN representation (h_t) of a word (w_t), the training objective of SynGCN and SemGCN is to predict the target word given its neighbors in the graph. Formally, for each method we maximize the following objective[1].

$$E = \sum_{t=1}^{|V|} \log P(w_t | w_1^t, w_2^t \ldots w_{N_t}^t)$$

where, w_t is the target word and $w_1^t, w_2^t \ldots w_{N_t}^t$ are its neighbors in the graph. The probability $P(w_t | w_1^t, w_2^t \ldots w_{N_t}^t)$ is calculated using the softmax function, defined as

$$P(w_t | w_1^t, w_2^t \ldots w_{N_t}^t) = \frac{\exp(v_{w_t}^T h_t)}{\sum_{i=1}^{|V|} \exp(v_{w_i}^T h_t)}.$$

Hence, E reduces to

$$E = \sum_{t=1}^{|V|} \left(v_{w_t}^T h_t - \log \sum_{i=1}^{|V|} \exp(v_{w_i}^T h_t) \right), \qquad (2)$$

where, h_t is the GCN representation of the target word w_t and v_{w_t} is its target embedding.

The second term in Equation 2 is computationally expensive as the summation needs to be taken over the entire vocabulary. This can be overcome using several approximations like noise-contrastive estimation (Gutmann and Hyvärinen, 2010) and hierarchical softmax (Morin and Bengio, 2005). In our methods, we use negative sampling as used by Mikolov et al. (2013b).

8 Experimental Setup

8.1 Dataset and Training

In our experiments, we use Wikipedia[2] corpus for training the models. After discarding too long and too short sentences, we get an average sentence length of nearly 20 words. The corpus consists of 57 million sentences with 1.1 billion tokens and 1 billion syntactic dependencies.

[1] We also experimented with joint SynGCN and SemGCN model but our preliminary experiments gave suboptimal performance as compared to the sequential model. This can be attributed to the fact that syntactic information is orders of magnitude greater than the semantic information available. Hence, the semantic constraints are not effectively utilized. We leave the analysis of the joint model as a future work.

[2] https://dumps.wikimedia.org/enwiki/20180301/

8.2 Baselines

For evaluating SynGCN (Section 5), we compare against the following baselines:

- **Word2vec** is continuous-bag-of-words model originally proposed by Mikolov et al. (2013b).
- **GloVe** (Pennington et al., 2014), a log-bilinear regression model which leverages global co-occurrence statistics of corpus.
- **Deps** (Levy and Goldberg, 2014) is a modification of skip-gram model which uses dependency context in place of sequential context.
- **EXT** (Komninos and Manandhar, 2016) is an extension of Deps which utilizes second-order dependency context features.

SemGCN (Section 6) model is evaluated against the following methods:

- **Retro-fit** (Faruqui et al., 2014) is a post-processing procedure which uses similarity constraints from semantic knowledge sources.
- **Counter-fit** (Mrkšić et al., 2016), a method for injecting both antonym and synonym constraints into word embeddings.
- **JointReps** (Alsuhaibani et al., 2018), a joint word representation learning method which simultaneously utilizes the corpus and KB.

8.3 Evaluation method

To evaluate the effectiveness of our proposed methods, we compare them against the baselines on the following intrinsic and extrinsic tasks[3]:

- **Intrinsic Tasks:**
  Word Similarity is the task of evaluating closeness between semantically similar words. Following Komninos and Manandhar (2016); Pennington et al. (2014), we evaluate on Simlex-999 (Hill et al., 2015), WS353 (Finkelstein et al., 2001), and RW (Luong et al., 2013) datasets.
  Concept Categorization involves grouping nominal concepts into natural categories. For instance, *tiger* and *elephant* should belong to *mammal* class. In our experiments, we evalute on AP (Almuhareb, 2006), Battig (Baroni and Lenci, 2010), BLESS (Baroni and Lenci, 2011), ESSLI (Baroni et al., 2008) datasets.
  Word Analogy task is to predict word b_2, given three words a_1, a_2, and b_1, such that the relation $b_1 : b_2$ is same as the relation $a_1 : a_2$. We compare methods on MSR (Mikolov et al., 2013c) and SemEval-2012 (Jurgens et al., 2012).
- **Extrinsic Tasks:**
  Named Entity Recognition (NER) is the task of locating and classifying entity mentions into categories like *person*, *organization* etc. We use Lee et al. (2018)'s model on CoNLL-2003 dataset (Tjong Kim Sang and De Meulder, 2003) for evaluation.
  Question Answering in Stanford Question Answering Dataset (**SQuAD**) (Rajpurkar et al., 2016) involves identifying answer to a question as a segment of text from a given passage. Following Peters et al. (2018), we evaluate using Clark and Gardner (2018)'s model.
  Part-of-speech (POS) tagging aims at associating with each word, a unique tag describing its syntactic role. For evaluating word embeddings, we use Lee et al. (2018)'s model on Penn Treebank POS dataset (Marcus et al., 1994).
  Co-reference Resolution (Coref) involves identifying all expressions that refer to the same entity in the text. To inspect the effect of embeddings, we use Lee et al. (2018)'s model on CoNLL-2012 shared task dataset (Pradhan et al., 2012).

9 Results

In this section, we attempt to answer the following questions.

Q1. Does SynGCN learn better word embeddings than existing approaches? (Section 9.1)

Q2. Does SemGCN effectively handle diverse semantic information as compared to other methods? (Section 9.2)

Q3. How does SemGCN perform compared to other methods when provided with the same semantic constraints? (Section 9.3)

Q4. Does dependency context based embedding encode complementary information compared to ELMo? (Section 9.4)

9.1 SynGCN Evaluation

The evaluation results on intrinsic tasks – word similarity, concept categorization, and analogy – are summarized in Table 1. We report Spearman correlation for word similarity and analogy tasks and cluster purity for concept categorization task. Overall, we find that SynGCN, our proposed method, outperforms all the existing word embedding approaches in 9 out of 10 settings. The inferior performance of SynGCN and other depen-

[3] Details of hyperparameters are in supplementary.

	Word Similarity				Concept Categorization				Word Analogy	
Method	WS353S	WS353R	SimLex999	RW	AP	Battig	BLESS	ESSLI	SemEval2012	MSR
Word2vec	71.4	52.6	38.0	30.0	63.2	43.3	77.8	63.0	18.9	44.0
GloVe	69.2	**53.4**	36.7	29.6	58.0	41.3	80.0	59.3	18.7	45.8
Deps	65.7	36.2	39.6	33.0	61.8	41.7	65.9	55.6	22.9	40.3
EXT	69.6	44.9	43.2	18.6	52.6	35.0	65.2	66.7	21.8	18.8
SynGCN	**73.2**	45.7	**45.5**	**33.7**	**69.3**	**45.2**	**85.2**	**70.4**	**23.4**	**52.8**

Table 1: **SynGCN Intrinsic Evaluation:** Performance on word similarity (Spearman correlation), concept categorization (cluster purity), and word analogy (Spearman correlation). Overall, SynGCN outperforms other existing approaches in 9 out of 10 settings. Please refer to Section 9.1 for more details.

Method	POS	SQuAD	NER	Coref
Word2vec	95.0±0.1	78.5±0.3	89.0±0.2	65.1±0.3
GloVe	94.6±0.1	78.2±0.2	89.1±0.1	64.9±0.2
Deps	95.0±0.1	77.8±0.3	88.6±0.3	64.8±0.1
EXT	94.9±0.2	**79.6±0.1**	88.0±0.1	64.8±0.1
SynGCN	**95.4±0.1**	**79.6±0.2**	**89.5±0.1**	**65.8±0.1**

Table 2: **SynGCN Extrinsic Evaluation:** Comparison on parts-of-speech tagging (POS), question answering (SQuAD), named entity recognition (NER), and co-reference resolution (Coref). SynGCN performs comparable or outperforms all existing approaches on all tasks. Refer Section 9.1 for details.

Method	POS	SQuAD	NER	Coref
X = SynGCN	95.4±0.1	79.6±0.2	**89.5±0.1**	65.8±0.1
Retro-fit (X,1)	94.8±0.1	79.6±0.1	88.8±0.1	66.0±0.2
Counter-fit (X,2)	94.7±0.1	79.8±0.1	88.3±0.3	65.7±0.3
JointReps (X,4)	95.4±0.1	79.4±0.3	89.1±0.3	65.6±0.1
SemGCN (X,4)	**95.5±0.1**	**80.4±0.1**	**89.5±0.1**	**66.1±0.1**

Table 3: **SemGCN Extrinsic Evaluation:** Comparison of different methods for incorporating diverse semantic constraints in SynGCN embeddings on all extrinsic tasks. Refer Section 9.2 for details.

dency context based methods on WS353R dataset compared to sequential context based methods is consistent with the observation reported in Levy and Goldberg (2014); Komninos and Manandhar (2016). This is because the syntactic context based embeddings capture functional similarity rather than topical similarity (as discussed in Section 1). On average, we obtain around 1.5%, 5.7% and 7.5% absolute increase in performance on word similarity, concept categorization and analogy tasks compared to the best performing baseline. The results demonstrate that the learned embeddings from SynGCN more effectively capture semantic and syntactic properties of words.

We also evaluate the performance of different word embedding approaches on the downstream tasks as defined in Section 8.3. The experimental results are summarized in Table 2. Overall, we find that SynGCN either outperforms or performs comparably to other methods on all four tasks. Compared to the sequential context based methods, dependency based methods perform superior at question answering task as they effectively encode syntactic information. This is consistent with the observation of Peters et al. (2018).

9.2 Evaluation with Diverse Semantic Information

In this section, we compare SemGCN against the methods listed in Section 8.2 for incorporating diverse semantic information in pre-trained embeddings. We use *hypernym*, *hyponym*, and *antonym* relations from WordNet, and *synonym* relations from PPDB as semantic information. For each method, we provide the semantic information that it can utilize, e.g., Retro-fit can only make use of *synonym* relation[4]. In our results, **M(X, R)** denotes the fine-tuned embeddings obtained using method M while taking X as initialization embeddings. R denotes the types of semantic information used as defined below.

- **R=1:** Only synonym information.
- **R=2:** Synonym and antonym information.
- **R=4:** Synonym, antonym, hypernym and hyponym information.

For instance, Counter-fit (GloVe, 2) represents GloVe embeddings fine-tuned by Counter-fit using synonym and antonym information.

Similar to Section 9.1, the evaluation is performed on the three intrinsic tasks. Due to space limitations, we report results on one representative dataset per task. The results are summarized

[4] Experimental results controlling for semantic information are provided in Section 9.3.

Init Embeddings (=X)	Word2vec			GloVe			Deps			EXT			SynGCN		
Datasets	WS353	AP	MSR	WS353	AP	MSR	WS353	AP	MSR	WS353	AP	MSR	WS353	AP	MSR
Performance of X	63.0	63.2	44.0	58.0	60.4	45.8	55.6	64.2	40.3	59.3	53.5	18.8	61.7	69.3	52.8
Retro-fit (X,1)	63.4	**67.8**	**46.7**	58.5	61.1	**47.2**	54.8	64.7	41.0	61.6	55.1	40.5	61.2	67.1	51.4
Counter-fit (X,2)	60.3	62.9	31.4	53.7	62.5	29.6	46.9	60.4	33.4	52.0	54.4	35.8	55.2	66.4	31.7
JointReps (X,4)	60.9	61.1	28.5	59.2	55.5	37.6	54.8	58.7	38.0	58.8	54.8	20.6	60.9	68.2	24.9
SemGCN (X,4)	**64.8**	**67.8**	36.8	**63.3**	**63.2**	44.1	**62.3**	$\boxed{69.3}$	**41.1**	**62.9**	**67.1**	**52.1**	$\boxed{65.3}$	$\boxed{69.3}$	$\boxed{54.4}$

Table 4: **SemGCN Intrinsic Evaluation:** Evaluation of different methods for incorporating diverse semantic constraints initialized using various pre-trained embeddings (X). M(X, R) denotes the fine-tuned embeddings using method M taking X as initialization embeddings. R denotes the type of semantic relations used as defined in Section 9.2. SemGCN outperforms other methods in 13 our of 15 settings. SemGCN with SynGCN gives the best performance across all tasks (highlighted using $\boxed{\cdot}$). Please refer Section 9.2 for details.

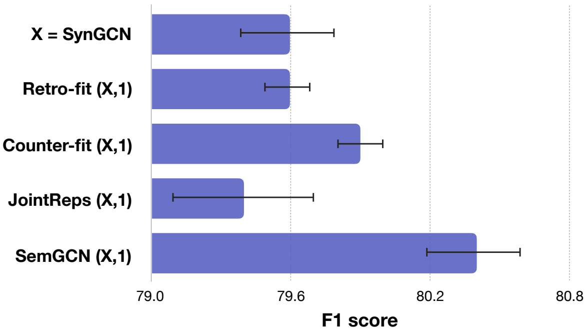

Figure 3: Comparison of different methods when provided with the same semantic information (synonym) for fine tuning SynGCN embeddings. Results denote the F1-score on SQuAD dataset. SemGCN gives considerable improvement in performance. Please refer Section 9.3 for details.

in Table 4. We find that in 13 out of 15 settings, SemGCN outperforms other methods. Overall, we observe that SemGCN, when initialized with SynGCN, gives the best performance on all the tasks (highlighted by $\boxed{\cdot}$ in Table 4).

For comparing performance on the extrinsic tasks, we first fine-tune SynGCN embeddings using different methods for incorporating semantic information. The embeddings obtained by this process are then evaluated on extrinsic tasks, as in Section 9.1. The results are shown in Table 3. We observe that while the other methods do not always consistently give improvement over the baseline SynGCN, SemGCN is able to improve upon SynGCN in all settings (better or comparable). Overall, we observe that SynGCN along with SemGCN is the most suitable method for incorporating both syntactic and semantic information.

9.3 Evaluation with Same Semantic Information

In this section, we compare SemGCN against other baselines when provided with the same semantic information: synonyms from PPDB. Similar to Section 9.2, we compare both on intrinsic and extrinsic tasks with different initializations. The evaluation results of fine-tuned SynGCN embeddings by different methods on SQuAD are shown in the Figure 3. The remaining results are included in the supplementary (Table S1 and S2). We observe that compared to other methods, SemGCN is most effective at incorporating semantic constraints across all the initializations and outperforms others at both intrinsic and extrinsic tasks.

Method	POS	SQuAD	NER	Coref
ELMo (E)	96.1±0.1	81.8±0.2	90.3±0.3	67.8±0.1
E+SemGCN(SynGCN, 4)	**96.2±0.1**	**82.4±0.1**	**90.9±0.1**	**68.3±0.1**

Table 5: Comparison of ELMo with SynGCN and SemGCN embeddings on multiple extrinsic tasks. For each task, models use a linear combination of the provided embeddings whose weights are learned. Results show that our proposed methods encode complementary information which is not captured by ELMo. Please refer Section 9.4 for more details.

9.4 Comparison with ELMo

Recently, ELMo (Peters et al., 2018) has been proposed which fine-tunes word embedding based on sentential context. In this section, we evaluate SynGCN and SemGCN when given along with pre-trained ELMo embeddings. The results are reported in Table 5. The results show that dependency context based embeddings encode complementary information which is not captured by ELMo as it only relies on sequential context. Hence, our proposed methods serves as an effective combination with ELMo.

10 Conclusion

In this paper, we have proposed SynGCN, a graph convolution based approach which utilizes syntactic context for learning word representations. SynGCN overcomes the problem of vocabulary explosion and outperforms state-of-the-art word embedding approaches on several intrinsic and extrinsic tasks. We also propose SemGCN, a framework for jointly incorporating diverse semantic information in pre-trained word embeddings. The combination of SynGCN and SemGCN gives the best overall performance. We make the source code of both models available to encourage reproducible research.

Acknowledgments

We thank the anonymous reviewers for their constructive comments. This work is supported in part by the Ministry of Human Resource Development (Government of India) and Google PhD Fellowship.

References

Abdulrahman Almuhareb. 2006. Attributes in lexical acquisition.

Mohammed Alsuhaibani, Danushka Bollegala, Takanori Maehara, and Ken-ichi Kawarabayashi. 2018. Jointly learning word embeddings using a corpus and a knowledge base. *PLOS ONE*, 13(3):1–26.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The berkeley framenet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*, ACL '98, pages 86–90.

Marco Baroni, Stefan Evert, and Alessandro Lenci. 2008. Esslli 2008 workshop on distributional lexical semantics. Association for Logic, Language and Information.

Marco Baroni and Alessandro Lenci. 2010. Distributional memory: A general framework for corpus-based semantics. *Comput. Linguist.*, 36(4):673–721.

Marco Baroni and Alessandro Lenci. 2011. How we blessed distributional semantic evaluation. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, GEMS '11, pages 1–10, Stroudsburg, PA, USA. Association for Computational Linguistics.

Joost Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Simaan. 2017. Graph convolutional encoders for syntax-aware neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1957–1967. Association for Computational Linguistics.

Y. Bengio, A. Courville, and P. Vincent. 2013. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155.

M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. 2017. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42.

Christopher Clark and Matt Gardner. 2018. Simple and effective multi-paragraph reading comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 845–855. Association for Computational Linguistics.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. *CoRR*, abs/1606.09375.

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard H. Hovy, and Noah A. Smith. 2014. Retrofitting word vectors to semantic lexicons. *CoRR*, abs/1411.4166.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context: The concept revisited. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 406–414, New York, NY, USA. ACM.

Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 297–304, Chia Laguna Resort, Sardinia, Italy. PMLR.

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

Shihao Ji, Hyokun Yun, Pinar Yanardag, Shin Matsushima, and S. V. N. Vishwanathan. 2015. Wordrank: Learning word embeddings via robust ranking. *CoRR*, abs/1506.02761.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

David A. Jurgens, Peter D. Turney, Saif M. Mohammad, and Keith J. Holyoak. 2012. Semeval-2012 task 2: Measuring degrees of relational similarity. In *Proceedings of 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 '12, pages 356–364, Stroudsburg, PA, USA. Association for Computational Linguistics.

Douwe Kiela, Felix Hill, and Stephen Clark. 2015. Specializing word embeddings for similarity or relatedness. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2044–2048. Association for Computational Linguistics.

Thomas N. Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907.

Alexandros Komninos and Suresh Manandhar. 2016. Dependency based embeddings for sentence classification tasks.

Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. 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)*, pages 687–692. Association for Computational Linguistics.

Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308. Association for Computational Linguistics.

Chen Li, Jianxin Li, Yangqiu Song, and Ziwei Lin. 2018. Training and evaluating improved dependency-based word embeddings. In *AAAI*.

Minh-Thang Luong, Richard Socher, and Christopher D. Manning. 2013. Better word representations with recursive neural networks for morphology. In *CoNLL*, Sofia, Bulgaria.

Xuezhe Ma and Eduard H. Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *CoRR*, abs/1603.01354.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1506–1515. Association for Computational Linguistics.

Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. The penn treebank: Annotating predicate argument structure. In *Proceedings of the Workshop on Human Language Technology*, HLT '94, pages 114–119, Stroudsburg, PA, USA. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 3111–3119, USA. Curran Associates Inc.

Tomas Mikolov, Scott Wen-tau Yih, and Geoffrey Zweig. 2013b. 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 (NAACL-HLT-2013)*. Association for Computational Linguistics.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013c. 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*, pages 746–751. Association for Computational Linguistics.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

Frederic Morin and Yoshua Bengio. 2005. Hierarchical probabilistic neural network language model. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics, AISTATS 2005, Bridgetown, Barbados, January 6-8, 2005*.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 142–148. Association for Computational Linguistics.

Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2015. Ppdb 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *In Proceedings of the Association for Computational Linguistics (ACL-2015)*, pages 425–430. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, CoNLL '12, pages 1–40, Stroudsburg, PA, USA. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392. Association for Computational Linguistics.

Richard Socher, John Bauer, Christopher D. Manning, and Andrew Y. Ng. 2013. Parsing With Compositional Vector Grammars. In *ACL*.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038. Association for Computational Linguistics.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566. Association for Computational Linguistics.

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 at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 142–147, Stroudsburg, PA, USA. Association for Computational Linguistics.

Shikhar Vashishth, Shib Sankar Dasgupta, Swayambhu Nath Ray, and Partha Talukdar. 2018a. Dating documents using graph convolution networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1605–1615. Association for Computational Linguistics.

Shikhar Vashishth, Rishabh Joshi, Sai Suman Prayaga, Chiranjib Bhattacharyya, and Partha Talukdar. 2018b. Reside: Improving distantly-supervised neural relation extraction using side information. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1257–1266. Association for Computational Linguistics.

Shikhar Vashishth, Prateek Yadav, Manik Bhandari, and Partha Talukdar. 2019. Confidence-based graph convolutional networks for semi-supervised learning. In *Proceedings of Machine Learning Research*, volume 89 of *Proceedings of Machine Learning Research*, pages 1792–1801. PMLR.

Chang Xu, Yalong Bai, Jiang Bian, Bin Gao, Gang Wang, Xiaoguang Liu, and Tie-Yan Liu. 2014. Rc-net: A general framework for incorporating knowledge into word representations.

Prateek Yadav, Madhav Nimishakavi, Naganand Yadati, Shikhar Vashishth, Arun Rajkumar, and Partha Talukdar. 2019. Lovasz convolutional networks. In *Proceedings of Machine Learning Research*, volume 89 of *Proceedings of Machine Learning Research*, pages 1978–1987. PMLR.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2018. Graph Convolutional Networks for Text Classification. *ArXiv e-prints*, page arXiv:1809.05679.

Mo Yu and Mark Dredze. 2014. Improving lexical embeddings with semantic knowledge. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 545–550. Association for Computational Linguistics.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2205–2215. Association for Computational Linguistics.