Knowledge-Powered Deep Learning for Word Embedding

Jiang Bian, Bin Gao, and Tie-Yan Liu

Microsoft Research
{jibian,bingao,tyliu}@microsoft.com

Abstract. The basis of applying deep learning to solve natural language processing tasks is to obtain high-quality distributed representations of words, i.e., word embeddings, from large amounts of text data. However, text itself usually contains incomplete and ambiguous information, which makes necessity to leverage extra knowledge to understand it. Fortunately, text itself already contains well-defined morphological and syntactic knowledge; moreover, the large amount of texts on the Web enable the extraction of plenty of semantic knowledge. Therefore, it makes sense to design novel deep learning algorithms and systems in order to leverage the above knowledge to compute more effective word embeddings. In this paper, we conduct an empirical study on the capacity of leveraging morphological, syntactic, and semantic knowledge to achieve high-quality word embeddings. Our study explores these types of knowledge to define new basis for word representation, provide additional input information, and serve as auxiliary supervision in deep learning, respectively. Experiments on an analogical reasoning task, a word similarity task, and a word completion task have all demonstrated that knowledge-powered deep learning can enhance the effectiveness of word embedding.

1 Introduction

With rapid development of deep learning techniques in recent years, it has drawn increasing attention to train complex and deep models on large amounts of data, in order to solve a wide range of text mining and natural language processing (NLP) tasks [4, 1, 8, 13, 19, 20]. The fundamental concept of such deep learning techniques is to compute distributed representations of words, also known as word embeddings, in the form of continuous vectors. While traditional NLP techniques usually represent words as indices in a vocabulary causing no notion of relationship between words, word embeddings learned by deep learning approaches aim at explicitly encoding many semantic relationships as well as linguistic regularities and patterns into the new embedding space.

Most of existing works employ generic deep learning algorithms, which have been proven to be successful in the speech and image domains, to learn the word embeddings for text related tasks. For example, a previous study [1] proposed a widely used model architecture for estimating neural network language model; later some studies [5, 21] employed the similar neural network architecture to learn word embeddings in order to improve and simplify NLP applications. Most recently, two models [14, 15] were proposed to learn word embeddings in a similar but more efficient manner so as to capture syntactic and semantic word similarities. All these attempts fall into a common framework to leverage the power of deep learning; however, one may want to ask the following questions: *Are these works the right approaches for text-related tasks? And, what are the principles of using deep learning for text-related tasks?*

To answer these questions, it is necessary to note that text yields some unique properties compared with other domains like speech and image. Specifically, while the success of deep learning on the speech and image domains lies in its capability of discovering important signals from noisy input, the major challenge for text understanding is instead the missing information and semantic ambiguity. In other words, image understanding relies more on the information contained in the image itself than the background knowledge, while text understanding often needs to seek help from various external knowledge since text itself only reflects limited information and is sometimes ambiguous. Nevertheless, most of existing works have not sufficiently considered the above uniqueness of text. Therefore it is worthy to investigate how to incorporate more knowledge into the deep learning process.

Fortunately, this requirement is fulfillable due to the availability of various text-related knowledge. First, since text is constructed by human based on morphological and grammatical rules, it already contains well defined morphological and syntactic knowledge. Morphological knowledge implies how a word is constructed, where morphological elements could be syllables, roots, or affix (prefix and suffix). Syntactic knowledge may consist of part-of-speech (POS) tagging as well as the rules of word transformation in different context, such as the comparative and superlative of an adjective, the past and participle of a verb, and the plural form of a noun. Second, there has been a rich line of research works on mining semantic knowledge from large amounts of text data on the Web, such as WordNet [25], Freebase [2], and Probase [26]. Such semantic knowledge can indicate entity category of the word, and the relationship between words/entities, such as synonyms, antonyms, *belonging-to* and *is-a*. For example, Portland *belonging-to* Oregon; Portland *is-a* city. Given the availability of the morphological, syntactic, and semantic knowledge, the critical challenge remains as how to design new deep learning algorithms and systems to leverage it to generate high-quality word embeddings.

In this paper, we take an empirical study on the capacity of leveraging morphological, syntactic, and semantic knowledge into deep learning models. In particular, we investigate the effects of leveraging morphological knowledge to define new basis for word representation and as well as the effects of taking advantage of syntactic and semantic knowledge to provide additional input information and serve as auxiliary supervision in deep learning. In our study, we employ an emerging popular continuous bag-of-words model (CBOW) proposed in [14] as the base model. The evaluation results demonstrate that, knowledge-powered deep learning framework, by adding appropriate knowledge in a proper way, can greatly enhance the quality of word embedding in terms of serving syntactic and semantic tasks.

The rest of the paper is organized as follows. We describe the proposed methods to leverage knowledge in word embedding using neural networks in Section 2. The experimental results are reported in Section 3. In Section 4, we briefly review the related work on word embedding using deep neural networks. The paper is concluded in Section 5.

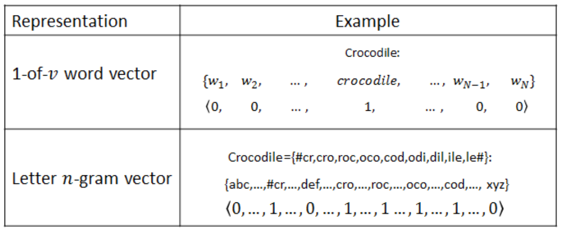
2 Incorporating Knowledge into Deep Learning

In this paper, we propose to leverage morphological knowledge to define new basis for word representation, and we explore syntactic and semantic knowledge to provide additional input information and serve as auxiliary supervision in the deep learning framework. Note that, our proposed methods may not be the optimal way to use those types of knowledge, but our goal is to reveal the power of knowledge for computing high-quality word embeddings through deep learning techniques.

2.1 Define New Basis for Word Representation

Currently, two major kinds of basis for word representations have been widely used in the deep learning techniques for NLP applications. One of them is the 1-of-v word vector, which follows the conventional bag-of-word models. While this kind of representation preserves the original form of the word, it fails to effectively capture the similarity between words (i.e., every two word vectors are orthogonal), suffers from too expensive computation cost when the vocabulary size is large, and cannot generalize to unseen words when it is computationally constrained.

Another kind of basis is the letter n-gram [11]. For example, in letter tri-gram (or tri-letter), a vocabulary is built according to every combination of three letters, and a word is projected to this vocabulary based on the tri-letters it contains. In contrast to the first type of basis, this method can significantly reduce the training complexity and address the problem of word orthogonality and unseen words. Nevertheless, letters do not carry on semantics by themselves; thus, two words with similar set of letter n-grams may yield quite different semantic meanings, and two semantically similar words might share very few letter n-grams. Figure 1 illustrates one example for each of these two word representation methods.



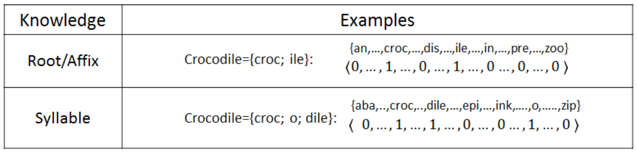
Representation	Example
1-of-v word vector	Crocodile: $\{w_1,\ w_2,\ \dots,\ \mathit{crocodile},\ \dots,\ w_{N-1},\ w_N\}$ $\langle 0,\ 0,\ \dots,\ 1,\ \dots,\ 0,\ 0\rangle$

Letter n-gram vector	Crocodile={#cr,cro,roc,oco,cod,odi,dil,ile,le#}:
	{abc,...,#cr,...,def,...,cro,...,roc,...,oco,...,cod,..., xyz}
	$\langle 0, \dots, 1, \dots, 0, \dots, 1, \dots, 1 \dots, 1, \dots, 1, \dots, 0\rangle$

Fig. 1. An example of how to use 1-of-v word vector and letter n-gram vector as basis to represent a word.

To address the limitations of the above word representation methods, we propose to leverage the morphological knowledge to define new forms of basis for word representation, in order to reduce training complexity, enhance capability to generalize to new emerging words, as well as preserve semantics of the word itself. In the following, we will introduce two types of widely-used morphological knowledge and discuss how to use them to define new basis for word representation.

Root/Affix As an important type of morphological knowledge, root and affix (prefix and suffix) can be used to define a new space where each word is represented as a vector of root/affix. Since most English words are composed by roots and affixes and both roots and affixes yield semantic meaning, it is quite beneficial to represent words using the vocabulary of roots and affixes, which may not only reduce the vocabulary size, but also reflect the semantics of words. Figure 2 shows an example of using root/affix to represent a word.



Knowledge	Examples
Root/Affix	Crocodile={croc; ile}: {an,...,croc,...,dis,...,ile,...,in,...,pre,...,zoo} ⟨0, ... , 1, ... , 0, ... , 1, ... , 0 ... , 0, ... , 0 ⟩
Syllable	Crocodile={croc; o; dile}: {aba,..,croc,..,dile,...,epi,...,ink,....,o,.....,zip} ⟨ 0, ... , 1, ... , 1, ... , 0, ... , 0 ... , 1, ... , 0 ⟩

Fig. 2. An example of how to use root/affix and syllable to represent a word.

Syllable Syllable is another important type of morphological knowledge that can be used to define the word representation. Similar to root/affix, using syllable can significantly reduce the dimension of the vocabulary. Furthermore, since syllables effectively encodes the pronunciation signals, they can also reflect the semantics of words to some extent (considering that human beings can understand English words and sentences based on their pronunciations). Meanwhile, we are able to cover any unseen words by using syllables as vocabulary. Figure 2 presents an example of using syllables to represent a word.

2.2 Provide Additional Input Information

Existing works on deep learning for word embeddings employ different types of data for different NLP tasks. For example, Mikolov *et al* [14] used text documents collected from Wikipedia to obtain word embeddings; Collobert and Weston [4] leveraged text documents to learn word embeddings for various NLP applications such as language model and chunking; and, Huang *et al* [11] applied deep learning approaches on queries and documents from click-through logs in search engine to generate word representations targeting the relevance tasks. However, those various types of text data, without extra information, can merely reflect partial information and usually cause semantic ambiguity. Therefore, to learn more effective word embeddings, it is necessary to leverage additional knowledge to address the challenges.

In particular, both syntactic and semantic knowledge can serve as additional inputs. An example is shown in Figure 3. Suppose the 1-of-v word vector is used as basis for word representations. To introduce extra knowledge beyond a word itself, we can use entity categories or POS tags as the extension to the original 1-of-v word vector. For example, given an entity knowledge graph, we can define an entity space. Then, a word will be projected into this space such that some certain elements yield non-zero values if the word belongs to the corresponding entity categories. In addition, relationship between words/entities can serve as another type of input information. Particularly, given

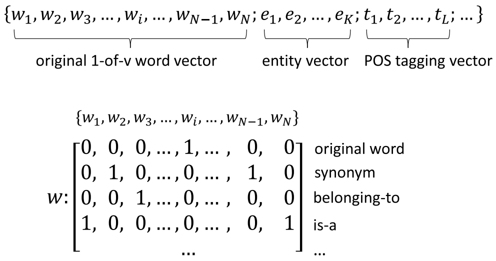

$\{w_1, w_2, w_3, \dots, w_i, \dots, w_{N-1}, w_N; e_1, e_2, \dots, e_K; t_1, t_2, \dots, t_L; \dots\}$
original 1-of-v word vector
entity vector
POS tagging vector
$\{w_1, w_2, w_3, \dots, w_i, \dots, w_{N-1}, w_N\}$
w:
original word
synonym
belonging-to
is-a
…

Fig. 3. An example of using syntactic or semantic knowledge, such as entity category, POS tags, and relationship, as additional input information.

various kinds of syntactic and semantic relations, such as *synonym*, *antonym*, *belonging-to*, *is-a*, etc., we can construct a relation matrix $\mathcal{R}_w$ for one word w (as shown in Figure 3), where each column corresponds to a word in the vocabulary, each row encodes one type of relationship, and one element $\mathcal{R}_w(i, j)$ has non-zero value if w yield the i-th relation with the j-th word.

2.3 Serve as Auxiliary Supervision

According to previous studies on deep learning for NLP tasks, different training samples and objective functions are suitable for different NLP applications. For example, some works [4, 14] define likelihood based loss functions, while some other work [11] leverages cosine similarity between queries and documents to compute objectives. However, all these loss functions are commonly used in the machine learning literature without considering the uniqueness of text.

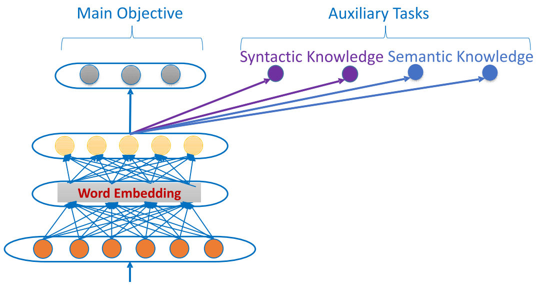


Fig. 4. Using syntactic and semantic knowledge as auxiliary objectives.

Text related knowledge can provide valuable complement to the objective of the deep learning framework. Particularly, we can create auxiliary tasks based on the knowledge to assist the learning of the main objective, which can effectively regularize the learning of the hidden layers and improve the generalization ability of deep neural networks so as to achieve high-quality word embedding. Both semantic and syntactic knowledge can serve as auxiliary objectives, as shown in Figure 4.

Note that this multi-task framework can be applied to any text related deep learning technique. In this work, we take the continuous bag-of-words model (CBOW) [14] as a specific example. The main objective of this model is to predict the center word given the surrounding context. More formally, given a sequence of training words $w_1, w_2, \cdots$, w_X, the main objective of the CBOW model is to maximize the average log probability:

$$\mathcal{L}_M = \frac{1}{X} \sum_{x=1}^{X} \log p(w_x | \mathcal{W}_x^d) \tag{1}$$

where $\mathcal{W}_x^d = \{w_{x-d}, \cdots, w_{x-1}, w_{x+1}, \cdots, w_{x+d}\}$ denotes a $2d$-sized training context of word w_x.

To use semantic and syntactic knowledge to define auxiliary tasks to the CBOW model, we can leverage the entity vector, POS tag vector, and relation matrix (as shown in Figure 3) of the center word as the additional objectives. Below, we take entity and relationship as two examples for illustration. Specifically, we define the objective for entity knowledge as

$$\mathcal{L}_E = \frac{1}{X} \sum_{x=1}^{X} \sum_{k=1}^{K} \mathbf{1}(w_x \in e_k) \log p(e_k | \mathcal{W}_x^d) \tag{2}$$

where K is the size of entity vector; and $\mathbf{1}(\cdot)$ is an indicator function, $\mathbf{1}(w_x \in e_k)$ equals 1 if w_x belongs to entity e_k, otherwise 0; note that the entity e_k could be denoted by either a single word or a phrase. Moreover, assuming there are totally R relations, i.e., there are R rows in the relation matrix, we define the objective for relation as:

$$\mathcal{L}_R = \frac{1}{X} \sum_{x=1}^{X} \sum_{r=1}^{R} \lambda_r \sum_{n=1}^{N} r(w_x, w_n) \log p(w_n | \mathcal{W}_x^d) \tag{3}$$

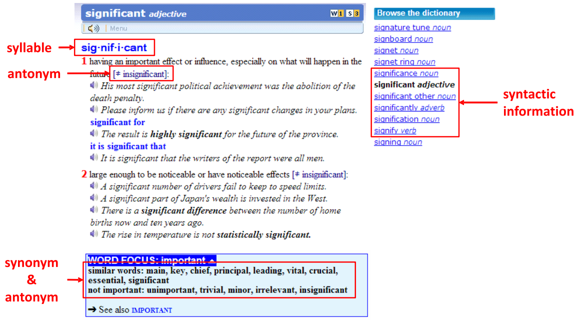
where N is vocabulary size; $r(w_x, w_n)$ equals 1 if w_x and w_n have relation r, otherwise 0; and λ_r is an empirical weight of relation r.

3 Experiments

To evaluate the effectiveness of the knowledge-powered deep learning for word embedding, we compare the quality of word embeddings learned with incorporated knowledge to those without knowledge. In this section, we first introduce the experimental settings, and then we conduct empirical comparisons on three specific tasks: a public analogical reasoning task, a word similarity task, and a word completion task.

3.1 Experimental Setup

Baseline Model In our empirical study, we use the continuous bag-of-words model (CBOW) [14] as the baseline method. The code of this model has been made publicly available[1]. We use this model to learn the word embeddings on the above dataset. In the following, we will study the effects of different methods for adding various types of knowledge into the CBOW model. To ensure the consistency among our empirical studies, we set the same number of embedding size, i.e. 600, for both the baseline model and those with incorporated knowledge.


significant *adjective* W1 S3

Menu

sig·nif·i·cant

1 having an important effect or influence, especially on what will happen in the future [≠ insignificant]:

His most significant political achievement was the abolition of the death penalty.

Please inform us if there are any significant changes in your plans.

significant for

*The result is **highly significant** for the future of the province.*

it is significant that

It is significant that the writers of the report were all men.

2 large enough to be noticeable or have noticeable effects [≠ insignificant]:

A significant number of drivers fail to keep to speed limits.

A significant part of Japan's wealth is invested in the West.

*There is a **significant difference** between the number of home births now and ten years ago.*

*The rise in temperature is not **statistically significant.***

WORD FOCUS: important

similar words: main, key, chief, principal, leading, vital, crucial, essential, significant

not important: unimportant, trivial, minor, irrelevant, insignificant

➔ See also IMPORTANT

Browse the dictionary

signature tune *noun*

signboard *noun*

signet *noun*

signet ring *noun*

significance *noun*

significant *adjective*

significant other *noun*

significantly *adverb*

signification *noun*

signify *verb*

signing *noun*

Fig. 5. Longman Dictionaries provide several types of morphological, syntactic, and semantic knowledge.

Table 1. Knowledge corpus used for our experiments (Type: MOR-morphological; SYN-syntactic; SEM-semantic).

Corpus	Type	Specific knowledge	Size
Morfessor	MOR	root, affix	200K
Longman	MOR/SYN /SEM	syllable, POS tagging, synonym, antonym	30K
WordNet	SYN/SEM	POS tagging, synonym, antonym	20K
Freebase	SEM	entity, relation	1M

Applied Knowledge For each word in the Wikipedia dataset as described above, we collect corresponding morphological, syntactic, and semantic knowledge from four data sources: Morfessor [23], Longman Dictionaries[2], WordNet [25], and Freebase[3].

Morfessor provides a tool that can automatically split a word into roots, prefixes, and suffixes. Therefore, this source allows us to collect morphological knowledge for each word existed in our training data.

Longman Dictionaries is a large corpus of words, phrases, and meaning, consisting of rich morphological, syntactic, and semantic knowledge. As shown in Figure 5, Longman Dictionaries provide word's syllables as morphological knowledge, word's syntactic transformations as syntactic knowledge, and word's synonym and antonym as semantic knowledge. We collect totally 30K words and their corresponding knowledge from Longman Dictionaries.

WordNet is a large lexical database of English. Nouns, verbs, adjectives, and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations. Note that WordNet interlinks not just word forms (syntactic information) but also specific senses of words (semantic information). WordNet also labels the semantic relations among words. Therefore, WordNet provides us with another corpus of rich semantic and syntactic knowledge. In our experiments, we sample 15K words with 12K synsets, and there are totally 20K word-senses pairs.

Freebase is an online collection of structured data harvested from many online sources. It is comprised of important semantic knowledge, especially the entity and relation in-

[1] http://code.google.com/p/word2vec/

[2] http://www.longmandictionariesonline.com/

[3] http://www.freebase.com/

formation (e.g., categories, *belonging-to*, *is-a*). We crawled 1M top frequent words and corresponding information from Freebase as another semantic knowledge base.
We summarize these four sources in Table 1[4].

3.2 Evaluation Tasks

We evaluate the quality of word embeddings on three tasks.

1. Analogical Reasoning Task:

The analogical reasoning task was introduced by Mikolov *et al* [16, 14], which defines a comprehensive test set that contains five types of semantic analogies and nine types of syntactic analogies[5]. For example, to solve semantic analogies such as *Germany* : *Berlin* = *France* : *?*, we need to find a vector x such that the embedding of x, denoted as vec(x) is closest to vec(*Berlin*) - vec(*Germany*) + vec(*France*) according to the cosine distance. This specific example is considered to have been answered correctly if x is *Paris*. Another example of syntactic analogies is *quick* : *quickly* = *slow* : *?*, the correct answer of which should be *slowly*. Overall, there are 8,869 semantic analogies and 10,675 syntactic analogies.

In our experiments, we trained word embeddings on a publicly available text corpus[6], a dataset about the first billion characters from Wikipedia. This text corpus contains totally 123.4 million words, where the number of unique words, i.e., the vocabulary size, is about 220 thousand. We then evaluated the overall accuracy for all analogy types, and for each analogy type separately (i.e., semantic and syntactic).

2. Word Similarity Task:

A standard dataset for evaluating vector-space models is the WordSim-353 dataset [7], which consists of 353 pairs of nouns. Each pair is presented without context and associated with 13 to 16 human judgments on similarity and relatedness on a scale from 0 to 10. For example, (*cup*, *drink*) received an average score of 7.25, while (*cup*, *substance*) received an average score of 1.92. Overall speaking, these 353 word pairs reflect more semantic word relationship than syntactic relationship.

In our experiments, similar to the Analogical Reasoning Task, we also learned the word embeddings on the same Wikipedia dataset. To evaluate the quality of learned word embedding, we compute Spearman's ρ correlation between the similarity scores computed based on learned word embeddings and human judgments.

3. Sentence Completion Task:

Another advanced language modeling task is Microsoft Sentence Completion Challenge [27]. This task consists of 1040 sentences, where one word is missing in each sentence and the goal is to select word that is the most coherent with the rest of the sentence, given a list of five reasonable choices. In general, accurate sentence completion requires better understanding on both the syntactic and semantics of the context.

In our experiments, we learn the 600-dimensional embeddings on the 50M training data provided by [27], with and without applied knowledge, respectively. Then, we compute score of each sentence in the test set by using each of the sliding windows

[4] We plan to release all the knowledge corpora we used in this study after the paper is published.

[5] http://code.google.com/p/word2vec/source/browse/trunk/ questions-words.txt

[6] http://mattmahoney.net/dc/enwik9.zip

(window size is consistent with the training process) including the unknown word at the input, and predict the corresponding central word in a sentence. The final sentence score is then the sum of these individual predictions. Using the sentence scores, we choose the most likely sentence to answer the question.

3.3 Experimental Results

Effects of Defining Knowledge-Powered Basis for Word Representation As introduced in Section 2.1, we can leverage morphological knowledge to design new basis for word representation, including root/affix-based and syllable-based bases. In this experiment, we separately leverage these two types of morphological basis, instead of the conventional 1-of-v word vector and letter n-gram vector, in the CBOW framework (as shown in Figure 6). Then, we compare the quality of the newly obtained word embeddings with those computed by the baseline models. Note that, after using root/affix, syllable, or letter n-gram as input basis, the deep learning framework will directly generate the embedding for each root/affix, syllable, or letter n-gram; the new embedding of a word can be obtained by aggregating the embeddings of this word's morphological elements.

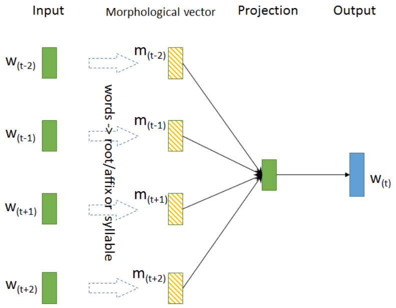


Fig. 6. Define morphological elements (root, affix, syllable) as new bases in CBOW.

Table 2. The accuracy of analogical questions by using word embeddings learned with different bases for word representation.

Representation	Dimensionality	Semantic Accuracy	Syntactic Accuracy	Overall Accuracy	Overall Relative Gain
Original words	220K	16.62%	34.98%	26.65%	-
Root/affix	24K	14.27%	44.15%	30.59%	14.78%
Syllable	10K	2.67%	18.72%	11.44%	-57.07%
Letter 3-gram	13K	0.18%	9.12%	5.07%	-80.98%
Letter 4-gram	97K	17.29%	32.99%	26.89%	0.90%
Letter 5-gram	289K	16.03%	34.27%	26.00%	-2.44%

Table 2 shows the accuracy of analogical questions by using baseline word embeddings and by using those learned from morphological knowledge-powered bases, respectively. As shown in Table 2, different bases yield various dimensionalities; and, using root/affix to represent words can significantly improve the accuracy with about 14% relative gain, even with a much lower input dimensionality than the original 1-of-v representation.

However, syllable and letter 3-gram lead to drastically decreasing accuracy, probably due to their low dimensionalities and high noise levels. In addition, as the average word length of the training data is 4.8, using letter 4-gram and 5-gram is very close to using 1-of-V as basis. Therefore, as shown in Table 2, letter 4-gram and 5-gram can perform as good as baseline.

Table 3 illustrate the performance for the word similarity task by using word embeddings trained from different bases. From the table, we can find that, letter 4-gram and 5-gram yields similar performances to the baseline; however, none of root/affix, syllable, and letter tri-gram can benefit word similarity task.

Table 3. Spearman's ρ correlation on WordSim-353 by using word embeddings learned with different bases.

Model	$\rho \times 100$	Relative Gain
Original words	60.1	-
Root/affix	60.6	0.83%
Syllable	17.9	-70%
3-gram	14.2	-76%
4-gram	60.3	0.33%
5-gram	60.0	-0.17%

For the sentence completion task, Table 4 compares the accuracy by using word embeddings trained with different bases. Similar to the trend of the first task, except Root/affix that can raise the accuracy by 3-4%, other bases for word representation have little or negative influence on the performance.

Table 4. Accuracy of different models on the Microsoft Sentence Completion Challenge.

Model	Accuracy	Relative gain
Original words	41.2%	-
Root/affix	42.7%	3.64%
Syllable	40.0%	-2.91%
3-gram	41.3%	0.24%
4-gram	40.8%	-0.97%
5-gram	41.0%	-0.49%

Effects of Providing Additional Knowledge-Augmented Input Information In this experiment, by using the method described in Section 2.2, we add syntactic and semantic knowledge of each input word as additional inputs into the CBOW model (as shown in Figure 7). Then, we compare the quality of the newly obtained word embeddings with the baseline.

For the analogical reasoning task, Table 5 reports the accuracy by using wording embeddings learned from the baseline model and that with knowledge-augmented inputs, respectively. From the table, we can find that using syntactic knowledge as additional input can benefit syntactic analogies significantly but drastically hurt the semantic accuracy, while semantic knowledge gives rise to an opposite result. This table also illustrates that using both semantic and syntactic knowledge as additional inputs can lead to about 24% performance gain.

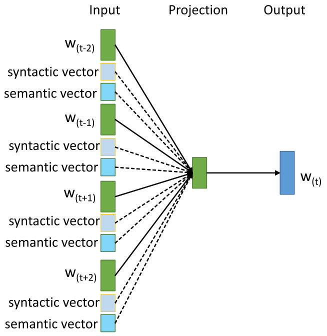


Fig. 7. Add syntactic and semantic knowledge of input word as additional inputs in CBOW.

Table 5. The accuracy of analogical questions by using word embeddings learned with different additional inputs.

Raw Data	Semantic Accuracy	Relative Gain	Syntactic Accuracy	Relative Gain	Total Accuracy	Relative Gain
Original words	16.62%		34.98%		26.65%	
+ Syntactic knowledge	6.12%	-63.18%	46.84%	33.90%	28.67%	7.58%
+ Semantic knowledge	49.16%	195.78%	17.96%	-48.66%	31.38%	17.74%
+ both knowledge	27.37%	64.68%	36.33%	3.86%	33.22%	24.65%

Furthermore, Table 6 illustrates the performance of the word similarity task on different models. From the table, it is clear to see that using semantic knowledge as additional inputs can cause a more than 4% relative gain while syntactic knowledge brings little influence on this task.

Table 6. Spearman's ρ correlation on WordSim-353 by using word embeddings learned with different additional input.

Model	$\rho \times 100$	Relative Gain
Original words	60.1	-
+ Syntactic knowledge	60.6	0.83%
+ Semantic knowledge	62.6	4.16%
+ both knowledge	60.9	1.33%

Moreover, Table 7 shows the accuracy of the sentence completion task by using models with different knowledge-augmented inputs. From the table, we can find that using either semantic or syntactic knowledge as additional inputs can benefit the performance, with more than 6% and 7% relative gains, respectively.

Effects of Serving Knowledge as Auxiliary Supervision As introduced in Section 2.3, in this experiment, we use either separate or combined syntactic and semantic knowledge as auxiliary tasks to regularize the training of the CBOW framework (as shown in Figure 8). Then, we compare the quality of the newly obtained word embeddings with those computed by the baseline model.

Table 8 illustrates the accuracy of analogical questions by using word embeddings learned from the baseline model and from those with knowledge-regularized objectives, respectively. From the table, we can find that leveraging either semantic or syntactic knowledge as auxiliary objectives results in quite little changes to the accuracy, and using both of them simultaneously can yield 1.39% relative improvement.

Table 7. Accuracy of different models on the Microsoft Sentence Completion Challenge.

Model	Accuracy	Relative Gain
Original words	41.2%	-
+ Syntactic knowledge	43.7%	6.07%
+ Semantic knowledge	44.1%	7.04%
+ Both knowledge	43.8%	6.31%

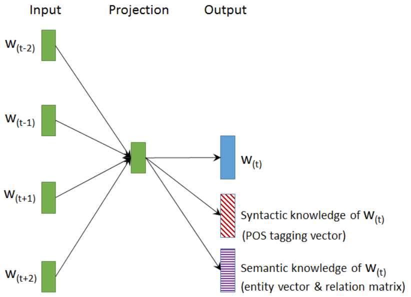

Input
Projection
Output
$W_{(t-2)}$
$W_{(t-1)}$
$W_{(t+1)}$
$W_{(t+2)}$
$W_{(t)}$
Syntactic knowledge of $W_{(t)}$
(POS tagging vector)
Semantic knowledge of $W_{(t)}$
(entity vector & relation matrix)

Fig. 8. Use syntactic and semantic knowledge as auxiliary objectives in CBOW.

Furthermore, Table 9 compares different models' performance on the word similarity task. From the table, we can find that using semantic knowledge as auxiliary objective can result in a significant improvement, with about 5.7% relative gain, while using syntactic knowledge as auxiliary objective cannot benefit this task. And, using both knowledge can cause more than 3% improvement.

Moreover, for the sentence completion task, Table 10 shows the accuracy of using different knowledge-regularized models. From the table, we can find that, while syntactic knowledge does not cause much accuracy improvement, using semantic knowledge as auxiliary objectives can significantly increase the performance, with more than 9% relative gain. And, using both knowledge as auxiliary objectives can lead to more than 7% improvement.

3.4 Discussions

In a summary, our empirical studies investigate three ways (i.e., new basis, additional inputs, and auxiliary supervision) of incorporating knowledge into three different text related tasks (i.e., analogical reasoning, word similarity, and sentence completion), and we explore three specific types of knowledge (i.e., morphological, syntactic, and semantic). Figure 9 summarizes whether and using which method each certain type of knowledge can benefit different tasks, in which a tick indicates a relative gain of larger than 3% and a cross indicates the remaining cases. In the following of this section, we

Table 8. The accuracy of analogical questions by using word embeddings learned from baseline model and those with knowledge-regularized objectives.

Objective	Semantic Accuracy	Relative Gain	Syntactic Accuracy	Relative Gain	Total Accuracy	Relative Gain
Original words	16.62%		34.98%		26.65%	
+ Syntactic knowledge	17.09%	2.83%	34.74%	−0.69%	26.73%	0.30%
+ Semantic knowledge	16.43%	−1.14%	35.33%	1.00%	26.75%	0.38%
+ both knowledge	17.59%	5.84%	34.86%	−0.34%	27.02%	1.39%

Table 9. Spearman's ρ correlation on WordSim-353 by using baseline model and the model trained by knowledge-regularized objectives.

Model	$\rho \times 100$	Relative Gain
Original words	60.1	-
+ Syntactic knowledge	59.8	-0.50%
+ Semantic knowledge	63.5	5.66%
+ both knowledge	62.1	3.33%

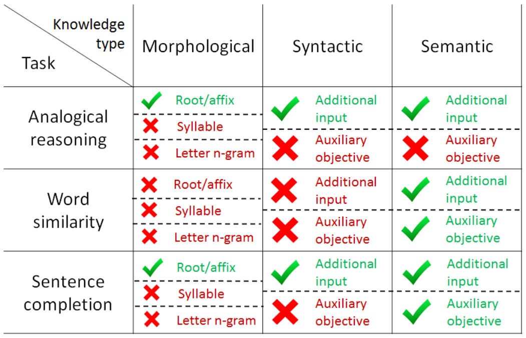
Table 10. Accuracy of different models on the Microsoft Sentence Completion Challenge.

Model	Accuracy	Relative Gain
Original words	41.2%	-
+ Syntactic knowledge	41.9%	1.70%
+ Semantic knowledge	45.2%	9.71%
+ both knowledge	44.2%	7.28%

will take further discussions to generalize some guidelines for incorporating knowledge into deep learning.

Different Tasks Seek Different Knowledge According to the task descriptions in Section 3.2, it is clear to see that the three text related tasks applied in our empirical studies are inherently different to each other, and such differences further decide each task's sensitivity to different knowledge.

Specifically, the analogical reasoning task consists of both semantic questions and syntactic questions. As shown in Figure 9, it is beneficial to applying both syntactic and semantic knowledge as additional input into the learning process. Morphological knowledge, especially root/affix, can also improve the accuracy of this task, because root/affix plays a key role in addressing some of the syntactic questions, such as *adj* : *adv*, *comparative* : *superlative*, the evidence of which can be found in Table 2 that illustrates using root/affix as basis can improve syntactic accuracy more than semantic accuracy.



Knowledge type Task	Morphological	Syntactic	Semantic
Analogical reasoning	✓ Root/affix ✗ Syllable ✗ Letter n-gram	✓ Additional input ✗ Auxiliary objective	✓ Additional input ✗ Auxiliary objective
Word similarity	✗ Root/affix ✗ Syllable ✗ Letter n-gram	✗ Additional input ✗ Auxiliary objective	✓ Additional input ✓ Auxiliary objective
Sentence completion	✓ Root/affix ✗ Syllable ✗ Letter n-gram	✓ Additional input ✗ Auxiliary objective	✓ Additional input ✓ Auxiliary objective

Fig. 9. A summary of whether and using which method each certain type of knowledge can benefit different tasks.

As aforementioned, the goal of the word similarity task is to predict the semantic similarity between two words without any context. Therefore only semantic knowledge can enhance the learned word embeddings for this task. As shown in Table 6 and 9, it is clear to see that using semantic knowledge as either additional input or auxiliary supervision can improve the word similarity task.

As a sentence is built to represent certain semantics under human defined morphological and syntactic rules, sentence completion task requires accurate understanding on the semantics of the context, the syntactic structure of the sentence, and the morphological rules for key words in it. Thus, as shown in Figure 9, all three types of knowledge can improve the accuracy of this task if used appropriately.

Effects of How to Incorporate Different Knowledge According to our empirical studies, syntactic knowledge is effective to improve analogical reasoning and sentence completion only when it is employed as additional input into the deep learning framework, which implies that syntactic knowledge can provide valuable input information but may not be suitable to serve as regularized objectives. Our empirical studies also demonstrate that, using semantic knowledge as either additional input or regularized objectives can improve the performance of the word similarity task and sentence completion tasks. Furthermore, comparing Table 9 and 10 with Table 6 and 7, we can find that applying semantic knowledge as auxiliary objectives can achieve slightly better performance than using it as additional input. However, for the analogical reasoning task, semantic knowledge is effective only when it is applied as additional input.

4 Related Work

Obtaining continious word embedding has been studied for a long time [9]. With the progress of deep learning, deep neural network models have been applied to obtain word embeddings. One of popular model architectures for estimating neural network language model (NNLM) was proposed in [1], where a feed-forward neural network with a linear projection layer and a non-linear hidden layer was used to learn jointly the word embedding and a statistical language model. Many studies follow this approach to improve and simplify text mining and NLP tasks [4–6, 8, 11, 19, 22, 20, 17, 10]. In these studies, estimation of the word embeddings was performed using different model architectures and trained on various text corpora.

For example, Collobert *et al* [5] proposed a unified neural network architecture to learn adequate internal representations on the basis of vast amounts of mostly unlabeled training data, to deal with various natural language processing tasks. In order to adapt the sequential property of language modeling, a recurrent architecture of NNLM was present in [13], referred as RNNLM, where the hidden layer at current time will be recurrently used as input to the hidden layer at the next time. Huang *et al* [11] developed a deep structure that project queries and documents into a common word embedding space where the query-document similarity is computed as the cosine similarity. The word embedding model is trained by maximizing the conditional likelihood of the clicked documents for a given query using the click-through data. Mikolov *et al* [14, 15] proposed the continuous bag-of-words model (CBOW) and the continuous skip-gram model (Skip-gram) for learning distributed representations of words from large amount of unlabeled text data. Both models can map the semantically or syntactically similar words to close positions in the learned embedding space, based on the principal that the context of the similar words are similar.

Recent studies have explored knowledge related word embedding, the purpose of of which is though quite different. For example, [3] focused on learning structured embeddings of knowledge bases; [18] paid attention to knowledge base completion; and [24] investigated relation extraction from free text. They did not explicitly study how to use knowledge to enhance word embedding. Besides, Luong *et al* [12] proposed to apply morphological information to learn better word embedding. But, it did not explore other ways to leverage various types of knowledge.

5 Conclusions and Future Work

In this paper, we take an empirical study on using morphological, syntactic, and semantic knowledge to achieve high-quality word embeddings. Our study explores these types of knowledge to define new basis for word representation, provide additional input information, and serve as auxiliary supervision in deep learning framework. Evaluations on three text related tasks demonstrated the effectiveness of knowledge-powered deep learning to produce high-quality word embeddings in general, and also reveal the best way of using each type of knowledge for a given task.

For the future work, we plan to explore more types of knowledge and apply them into the deep learning process. We also plan to study the co-learning of high-quality word embeddings and large-scale reliable knowledge.

6 Acknowledgement

We would like to thank Taifeng Wang, Qing Cui, Hanjun Dai, Xiang Li, Xuan Hu, Hongfei Xue, and Rui Zhang for their contribution to the data preparation and the experimental evaluation.

References

1. Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin. A neural probabilistic language model. In *The Journal of Machine Learning Research*, pages 3:1137–1155, 2003.
2. K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. ACM, 2008.
3. A. Bordes, J. Weston, R. Collobert, Y. Bengio, et al. Learning structured embeddings of knowledge bases. In *AAAI*, 2011.
4. R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 160–167, New York, NY, USA, 2008. ACM.
5. R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537, Nov. 2011.
6. L. Deng, X. He, and J. Gao. Deep stacking networks for information retrieval. In *ICASSP*, pages 3153–3157, 2013.

7. L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin. Placing search in context: The concept revisited. In *ACM Transactions on Information Systems*, 2002.
8. X. Glorot, A. Bordes, and Y. Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *In Proceedings of the Twenty-eight International Conference on Machine Learning, ICML*, 2011.
9. G. E. Hinton, J. L. McClelland, and D. E. Rumelhart. Distributed representations. In *Parallel distributed processing: Explorations in the microstructure of cognition*, pages 3:1137–1155. MIT Press, 1986.
10. E. Huang, R. Socher, C. Manning, and A. Ng. Improving word representations via global context and multiple word prototypes. In *Proc. of ACL*, 2012.
11. P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22Nd ACM International Conference on Conference on Information &#38; Knowledge Management*, CIKM '13, pages 2333–2338, New York, NY, USA, 2013. ACM.
12. M.-T. Luong, R. Socher, and C. D. Manning. Better word representations with recursive neural networks for morphology. *CoNLL-2013*, 104, 2013.
13. T. Mikolov. *Statistical Language Models Based on Neural Networks*. PhD thesis, Brno University of Technology, 2012.
14. T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.
15. T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, Z. Ghahramani, and K. Q. Weinberger, editors, *NIPS*, pages 3111–3119, 2013.
16. T. Mikolov, W.-T. Yih, and G. Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of NAACL-HLT*, pages 746–751, 2013.
17. A. Mnih and G. E. Hinton. A scalable hierarchical distributed language model. In *NIPS*, pages 1081–1088, 2008.
18. R. Socher, D. Chen, C. D. Manning, and A. Ng. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems*, pages 926–934, 2013.
19. R. Socher, C. C. Lin, A. Y. Ng, and C. D. Manning. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 26th International Conference on Machine Learning (ICML)*, 2011.
20. G. Tur, L. Deng, D. Hakkani-Tur, and X. He. Towards deeper understanding: Deep convex networks for semantic utterance classification. In *ICASSP*, pages 5045–5048, 2012.
21. J. P. Turian, L.-A. Ratinov, and Y. Bengio. Word representations: A simple and general method for semi-supervised learning. In *ACL*, pages 384–394, 2010.
22. P. D. Turney. Distributional semantics beyond words: Supervised learning of analogy and paraphrase. *Transactions of the Association for Computational Linguistics (TACL)*, pages 353–366, 2013.
23. S. Virpioja, P. Smit, S. Grnroos, and M. Kurimo. Morfessor 2.0: Python implementation and extensions for morfessor baseline. In *Aalto University publication series SCIENCE + TECHNOLOGY*, 2013.
24. J. Weston, A. Bordes, O. Yakhnenko, and N. Usunier. Connecting language and knowledge bases with embedding models for relation extraction. *arXiv preprint arXiv:1307.7973*, 2013.
25. WordNet. "about wordnet", princeton university, 2010.
26. W. Wu, H. Li, H. Wang, and K. Zhu. Probase: A probabilistic taxonomy for text understanding. In *Proc. of SIGMOD*, 2012.
27. G. Zweig and C. Burges. The microsoft research sentence completion challenge. In *Microsoft Research Technical Report MSR-TR-2011-129*, 2011.