An Exploration of Embeddings for Generalized Phrases

Wenpeng Yin and Hinrich Schütze
Center for Information and Language Processing
University of Munich, Germany
wenpeng@cis.lmu.de

Abstract

Deep learning embeddings have been successfully used for many natural language processing problems. Embeddings are mostly computed for word forms although lots of recent papers have extended this to other linguistic units like morphemes and word sequences. In this paper, we define the concept of *generalized phrase* that includes conventional linguistic phrases as well as skip-bigrams. We compute embeddings for generalized phrases and show in experimental evaluations on coreference resolution and paraphrase identification that such embeddings perform better than word form embeddings.

1 Motivation

One advantage of recent work in deep learning on natural language processing (NLP) is that linguistic units are represented by rich and informative embeddings. These embeddings support better performance on a variety of NLP tasks (Collobert et al., 2011) than symbolic linguistic representations that do not directly represent information about similarity and other linguistic properties. Embeddings are mostly derived for word forms although a number of recent papers have extended this to other linguistic units like morphemes (Luong et al., 2013), phrases and word sequences (Socher et al., 2010; Mikolov et al., 2013).[1] Thus, an important question is: what are the basic linguistic units that should be represented by embeddings in a deep learning NLP system? Building on the prior work in (Socher et al., 2010; Mikolov et al., 2013), we generalize the notion of phrase to include *skip-bigrams* (SkipBs) and lexicon entries, where lexicon entries can be both "continuous" and "noncontinuous" *linguistic phrases*. Examples of skip-bigrams at distance 2 in the sentence "this tea helped me to relax" are: "this*helped", "tea*me", "helped*to" ... Examples of linguistic phrases listed in a typical lexicon are continuous phrases like "cold_cuts" and "White_House" that only occur without intervening words and discontinous phrases like "take_over" and "turn_off" that can occur with intervening words. We consider it promising to compute embeddings for these phrases because many phrases, including the four examples we just gave, are noncompositional or weakly compositional, i.e., it is difficult to compute the meaning of the phrase from the meaning of its parts. We write gaps as "*" for SkipBs and "_" for phrases.

We can approach the question of what basic linguistic units should have representations from a practical as well as from a cognitive point of view. In practical terms, we want representations to be optimized for good generalization. There are many situations where a particular task involving a word cannot be solved based on *the word itself*, but it can be solved by analyzing *the context of the word*. For example, if a coreference resolution system needs to determine whether the unknown word "Xiulan" (a Chinese first name) in "he helped Xiulan to find a flat" refers to an animate or an inanimate entity, then the SkipB "helped*to" is a good indicator for the animacy of the unknown word – whereas the unknown word itself provides no clue.

From a cognitive point of view, it can be argued that many basic units that the human cognitive system uses have multiple words. Particularly convincing examples for such units are phrasal verbs in English, which often have a non-compositional meaning. It is implausible to suppose that we retrieve atomic representations for, say, "keep", "up", "on" and "from" and then combine them to

[1] Socher et al. use the term "word sequence". Mikolov et al. use the term "phrase" for word sequences that are mostly frequent continuous collocations.

Proceedings of the ACL 2014 Student Research Workshop, pages 41–47,
Baltimore, Maryland USA, June 22-27 2014. ©2014 Association for Computational Linguistics

form the meanings of the expressions “keep your head up,” “keep the pressure on,” “keep him from laughing”. Rather, it is more plausible that we recognize “keep up”, “keep on” and “keep from” as relevant basic linguistic units in these contexts and that the human cognitive systems represents them as units.

We can view SkipBs and discontinuous phrases as extreme cases of treating two words that do not occur next to each other as a unit. SkipBs are defined purely statistically and we will consider any pair of words as a potential SkipB in our experiments below. In contrast, discontinuous phrases are well motivated. It is clear that the words “picked” and “up” in the sentences “I picked it up” belong together and form a unit very similar to the word “collected” in “I collected it”. The most useful definition of discontinuous units probably lies in between SkipBs and phrases: we definitely want to include all phrases, but also some (but not all) statistical SkipBs. The initial work presented in this paper may help in finding a good “compromise” definition.

This paper contributes to a preliminary investigation of generalized phrase embeddings and shows that they are better suited than word embedding for a coreference resolution classification task and for paraphrase identification. Another contribution lies in that the phrase embeddings we release[2] could be a valuable resource for others.

The remainder of this paper is organized as follows. Section 2 and Section 3 introduce how to learn embeddings for SkipBs and phrases, respectively. Experiments are provided in Section 4. Subsequently, we analyze related work in Section 5, and conclude our work in Section 6.

2 Embedding learning for SkipBs

With English Gigaword Corpus (Parker et al., 2009), we use the *skip-gram model* as implemented in word2vec[3] (Mikolov et al., 2013) to induce embeddings. Word2vec skip-gram scheme is a neural network language model, using a given word to predict its context words within a window size. To be able to use word2vec directly without code changes, we represent the corpus as a sequence of sentences, each consisting of two tokens: a SkipB and a word that occurs between the two enclosing words of the SkipB. The distance k between the two enclosing words can be varied. In our experiments, we use either distance $k = 2$ or distance $2 \leq k \leq 3$. For example, for $k = 2$, the trigram $w_{i-1}\ w_i\ w_{i+1}$ generates the single sentence “w_{i-1}*$w_{i+1}\ w_i$”; and for $2 \leq k \leq 3$, the fourgram $w_{i-2}\ w_{i-1}\ w_i\ w_{i+1}$ generates the four sentences “w_{i-2}*$w_i\ w_{i-1}$”, “w_{i-1}*$w_{i+1}\ w_i$”, “w_{i-2}*$w_{i+1}\ w_{i-1}$” and “w_{i-2}*$w_{i+1}\ w_i$”.

In this setup, the middle context of SkipBs are kept (i.e., the second token in the new sentences), and the surrounding context of words of original sentences are also kept (i.e., the SkipB in the new sentences). We can run word2vec without any changes on the reformatted corpus to learn embeddings for SkipBs. As a baseline, we run word2vec on the original corpus to compute embeddings for words. Embedding size is set to 200.

3 Embedding learning for phrases

3.1 Phrase collection

Phrases defined by a lexicon have not been deeply investigated before in deep learning. To collect canonical phrase set, we extract two-word phrases defined in Wiktionary[4], and two-word phrases defined in Wordnet (Miller and Fellbaum, 1998) to form a collection of size 95218. This collection contains phrases whose parts always occur next to each other (e.g., “cold cuts”) and phrases whose parts more often occur separated from each other (e.g., “take (something) apart”).

3.2 Identification of phrase continuity

Wiktionary and WordNet do not categorize phrases as continuous or discontinous. So we need a heuristic for determining this automatically.

For each phrase “A_B”, we compute $[c_1, c_2, c_3, c_4, c_5]$ where $c_i, 1 \leq i \leq 5$, indicates there are c_i occurrences of A and B in that order with a distance of i. We compute these statistics for a corpus consisting of Gigaword and Wikipedia. We set the maximal distance to 5 because discontinuous phrases are rarely separated by more than 5 tokens.

If c_1 is 10 times higher than $(c_2+c_3+c_4+c_5)/4$, we classify “A_B” as *continuous*, otherwise as *discontinuous*. Taking phrase “pick_off” as an example, it gets vector [1121, 632, 337, 348, 4052], c_1 (1121) is smaller than the average 1342.25, so

[2] http://www.cis.lmu.de/pub/phraseEmbedding.txt.bz2

[3] https://code.google.com/p/word2vec/

[4] http://en.wiktionary.org/wiki/Wiktionary:Main_Page

"pick_off" is set as "discontinuous". Further consider "Cornell University" which gets [14831, 16, 177, 331, 3471], satisfying above condition, hence it is treated as a continuous phrase.

3.3 Sentence reformatting

Given the continuity information of phrases, sentence "$\cdots A \cdots B \cdots$" is reformated into "$\cdots A\_B \cdots A\_B \cdots$" if "A_B" is a discontinuous phrase and is separated by maximal 4 words, and sentence "$\cdots AB \cdots$" into "$\cdots A\_B \cdots$" if "A_B" is a continuous phrase.

In the first case, we use phrase "A_B" to replace each of its component words for the purpose of making the context of both constituents available to the phrase in learning. For the second situation, it is natural to combine the two words directly to form an independent semantic unit.

Word2vec is run on the reformatted corpus to learn embeddings for both words and phrases. Embedding size is also set to 200.

3.4 Examples of phrase neighbors

Usually, compositional methods for learning representations of multi-word text suffer from the difficulty in integrating word form representations, like word embeddings. To our knowledge, there is no released embeddings which can directly facilitate measuring the semantic affinity between linguistic units of arbitrary lengths. Table 1 attempts to provide some nearest neighbors for given typical phrases to show the promising perspective of our work. Note that discontinuous phrases like "turn_off" have plausible single word nearest neighbors like "unplug".

4 Experiments

Our motivation for generalized phrases in Section 1 was that they can be used to infer the attributes of the context they enclose and that they can capture non-compositional semantics. Our hypothesis was that they are more suitable for this than word embeddings. In this section we carry out two experiments to test this hypothesis.

4.1 Animacy classification for markables

A *markable* in coreference resolution is a linguistic expression that refers to an entity in the real world or another linguistic expression. Examples of markables include noun phrases ("the man"), named entities ("Peter") and nested nominal expressions ("their"). We address the task of *animacy classification* of markables: classifying them as animate/inanimate. This feature is useful for coreference resolution systems because only animate markables can be referred to using masculine and feminine pronouns in English like "him" and "she". Thus, this is an important clue for automatically clustering the markables of a document into correct coreference chains.

To create training and test sets, we extract all 39,689 coreference chains from the CoNLL2012 OntoNotes corpus.[5] We label chains that contain an animate pronoun markable ("she", "her", "he", "him" or "his") and no inanimate pronoun markable ("it" or "its") as animate; and chains that contain an inanimate pronoun markable and no animate pronoun markable as inanimate. Other chains are discarded.

We extract 39,942 markables and their contexts from the 10,361 animate and inanimate chains. The context of a markable is represented as a SkipB: it is simply the pair of the two words occurring to the left and right of the markable. The gold label of a markable and its SkipB is the animacy status of its chain: either animate or inanimate. We divide all SkipBs having received an embedding in the embedding learning phase into a training set of 11,301 (8097 animate, 3204 inanimate) and a balanced test set of 4036.

We use LIBLINEAR (Fan et al., 2008) for classification, with penalty factors 3 and 1 for inanimate and animate classes, respectively, because the training data are unbalanced.

4.1.1 Experimental results

We compare the following representations for animacy classification of markables. (i) Phrase embedding: Skip-bigram embeddings with skip distance $k = 2$ and $2 \leq k \leq 3$; (ii) Word embedding: concatenation of the embeddings of the two enclosing words where the embeddings are either standard word2vec embeddings (see Section 2) or the embeddings published by (Collobert et al., 2011);[6] (iii) the one-hot vector representation of a SkipB: the concatentation of two one-hot vectors of dimensionality V where V is the size of the vocabulary. The first (resp. second) vector

[5] http://conll.cemantix.org/2012/data.html

[6] http://metaoptimize.com/projects/wordreprs/

turn_off	**caught_up**	**take_over**	**macular_degeneration**	**telephone_interview**
switch_off	mixed_up	take_charge	eye_disease	statement
unplug	entangled	replace	diabetic_retinopathy	interview
turning_off	involved	take_control	cataracts	conference_call
shut_off	enmeshed	stay_on	periodontal_disease	teleconference
block_out	tangled	retire	epilepsy	telephone_call
turned_off	mired	succeed	glaucoma	told
fiddle_with	engaged	step_down	skin_cancer	said

Table 1: Phrases and their nearest neighbors

is the one-hot vector for the left (resp. right) word of the SkipB. Experimental results are shown in Table 2.

representation		accuracy
phrase embedding	$k = 2$	0.703
	$2 \leq k \leq 3$	0.700
word embedding	word2vec	$0.668^{*\dagger}$
	Collobert et al.	$0.662^{*\dagger}$
one-hot vectors		$0.638^{*\dagger}$

Table 2: Classification accuracy. Mark "*" means significantly lower than "phrase embedding", $k = 2$; "†" means significantly lower than "phrase embedding", $2 \leq k \leq 3$. As significance test, we use the test of equal proportion, p $<$ .05, throughout.

The results show that phrase embeddings have an obvious advantage in this classification task, both for $k = 2$ and $2 \leq k \leq 3$. This validates our hypothesis that learning embeddings for discontinuous linguistic units is promising.

In our error analysis, we found two types of frequent errors. (i) **Unspecific SkipBs.** Many SkipBs are equally appropriate for animate and inanimate markables. Examples of such SkipBs include "take*in" and "then*goes". (ii) **Untypical use of specific SkipBs.** Even SkipBs that are specific with respect to what type of markable they enclose sometimes occur with the "wrong" type of markable. For example, most markables occurring in the SkipB "of*whose" are animate because "whose" usually refers to an animate markable. However, in the context "…the southeastern area of Fujian whose economy is the most active" the enclosed markable is Fujian, a province of China. This example shows that "whose" occasionally refers to an inanimate entity even though these cases are infrequent.

4.1.2 Nearest neighbors of SkipBs

Table 3 shows some SkipBs and their nearest neighbors in descending order, where similarity is computed with cosine measure.

A general phenomenon is that phrase embeddings capture high degree of consistency in inferring the attributes of enclosed words. Considering the neighbor list in the first column, we can estimate that a *verb* probably appears as the middle token. Furthermore, *noun*, *pronoun*, *adjective* and *adverb* can roughly be inferred for the remaining columns, respectively.[7]

4.2 Paraphrase identification task

Paraphrase identification depends on semantic analysis. Standard approaches are unlikely to assign a high similarity score to the two sentences "he started the machine" and "he turned the machine on". In our approach, embedding of the phrase "turned on" can greatly help us to infer correctly that the sentences are paraphrases. Hence, phrase embeddings and in particular embeddings of discontinuous phrases seem promising in paraphrase detection task.

We use the Microsoft Paraphrase Corpus (Dolan et al., 2004) for evaluation. It consists of a training set with 2753 true paraphrase pairs and 1323 false paraphrase pairs, along with a test set with 1147 true and 578 false pairs. After discarding pairs in which neither sentence contains phrases, 3027 training pairs (2123 true vs. 904 false) and 1273 test pairs (871 true vs. 402 false) remain.

[7] A reviewer points out that this is only a suggestive analysis and that corpus statistics about these contexts would be required to establish that phrase embeddings can predict part-of-speech with high accuracy.

who*afghanistan,	some*told	women*have	with*responsibility	he*worried
had*afghanistan	other*told	men*have	of*responsibility	she*worried
he*afghanistan	two*told	children*have	and*responsibility	was*worried
who*iraq	–*told	girls*have	“*responsibility	is*worried
have*afghanistan	but*told	parents*have	that*responsibility	said*worried
fighters*afghanistan	one*told	students*have	’s*responsibility	that*worried
who*kosovo	because*told	young*have	the* responsibility	they*worried
was*afghanistan	and*told	people*have	for*responsibility	’s*worried

Table 3: SkipBs and their nearest neighbors

We tackle the paraphrase identification task via supervised binary classification. Sentence representation equals to the addition over all the token embeddings (words as well as phrases). A slight difference is that when dealing with a sentence like “$\cdots A\_B \cdots A\_B \cdots$” we only consider “A_B” embedding once. The system “word embedding” is based on the embeddings of single words only. Subsequently, pair representation is derived by concatenating the two sentence vectors. This concatentation is then classified by LIBLINEAR as “paraphrase” or “no paraphrase”.

4.2.1 Experimental results and analysis

Table 4 shows the performance of two methods. Phrase embeddings are apparently better. Most work on paraphrase detection has devised intricate features and achieves performance numbers higher than what we report here (Ji and Eisenstein, 2013; Madnani et al., 2012; Blacoe and Lapata, 2012). Our objective is only to demonstrate the superiority of considering phrase embedding over merely word embedding in this standard task.

We are interested in how phrase embeddings make an impact on this task. To that end, we perform an analysis on test examples where word embeddings are better than phrase embeddings and vice versa.

Table 5 shows four pairs, of which “phrase embedding” outperforms “word embedding” in the first two examples, “word embedding” defeats “phrase embedding” in the last two examples. In the first pair, successful phrase detection enables to split sentences into better units, thus the generated representation can convey the sentence meaning more exactly.

Methods	Accuracy	F1
baseline	0.684	0.803
word embedding	0.695	0.805
phrase embedding	**0.713**	**0.812**

Table 4: Paraphrase task results.

The meaning difference in the second pair originates from the synonym substitution between “take over as chief financial officer” and “fill the position”. The embedding of the phrase “take_over” matches the embedding of the single word “fill” in this context.

“Phrase embedding” in the third pair suffers from wrong phrase detection. Actually, “in” and “on” can not be treated as a sound phrase in that situation even though “in_on” is defined by Wiktionary. Indeed, this failure, to some extent, results from the shortcomings of our method in discovering true phrases. Furthermore, figuring out whether two words are a phrase might need to analyse syntactic structure in depth. This work is directly based on naive intuitive knowledge, acting as an initial exploration. Profound investigation is left as future work.

Our implementation discovers the contained phrases in the fourth pair perfectly. Yet, “word embedding” defeats “phrase embedding” still. The pair is not a paraphrase partly because the numbers are different; e.g., there is a big difference between “5.8 basis points” and “50 basis points”. Only a method that can correctly treat numerical information can succeed here. However, the appearance of phrases “central_bank”, “interest_rates” and “basis_points” makes the non-numerical parts more expressive and informative, leading to less dominant for digital quantifications. On the contrary, though “word embedding” fails to split the sen-

GWP	sentence 1	sentence 2
1 0 1	Common **side_effects** include **nasal_congestion**, **runny_nose**, **sore_throat** and cough, the FDA said .	The most common **side_effects** after getting the nasal spray were **nasal_congestion**, **runny_nose**, **sore_throat** and cough .
1 0 1	Douglas Robinson, a senior **vice_president** of finance, will **take_over** as chief financial officer on an interim basis .	Douglas Robinson, CA senior **vice_president**, finance, will fill the position in the interim .
1 1 0	They were being held Sunday in the Camden County Jail on $ 100,000 bail each .	The Jacksons remained **in_on** Camden County jail $ 100,000 bail .
0 0 1	The **interest_rate** sensitive two year Schatz yield was down 5.8 **basis_points** at 1.99 percent .	The Swedish **central_bank** cut **interest_rates** by 50 **basis_points** to 3.0 percent .

Table 5: Four typical sentence pairs in which the predictions of word embedding system and phrase embedding system differ. G = gold annotation, W = prediction of word embedding system, P = prediction of phrase embedding system. The formatting used by the system is shown. The original word order of sentence 2 of the third pair is "· · · in Camden County jail on $ 100,000 bail".

tences into better units, it weakens unexpectedly the expressiveness of subordinate context. This example demonstrates the difficulty of paraphrase identification. Differing from simple similarity tasks, two sentences are often not paraphrases even though they may contain very similar words.

5 Related work

To date, approaches to extend embedding (or more generally "representation") beyond individual words are either *compositional* or *holistic* (Turney, 2012).

The best known work along the first line is by (Socher et al., 2010; Socher et al., 2011; Socher et al., 2012; Blacoe and Lapata, 2012), in which distributed representations of phrases or even sentences are calculated from the distributed representations of their parts. This approach is only plausible for units that are compositional, i.e., whose properties are systematically predictable from their parts. As well, how to develop a robust composition function still faces big hurdles; cf. Table 5.1 in (Mitchell and Lapata, 2010). Our approach (as well as similar work on continuous phrases) makes more sense for noncompositional units.

Phrase representations can also be derived by methods other than deep learning of embeddings, e.g., as vector space representations (Turney, 2012; Turney, 2013; Dinu et al., 2013). The main point of this paper – generalizing phrases to discontinuous phrases and computing representations for them – is orthogonal to this issue. It would be interesting to evaluate other types of representations for generalized phrases.

6 Conclusion and Future Work

We have argued that generalized phrases are part of the inventory of linguistic units that we should compute embeddings for and we have shown that such embeddings are superior to word form embeddings in a coreference resolution task and standard paraphrase identification task.

In this paper we have presented initial work on several problems that we plan to continue in the future: (i) How should the inventory of continuous and discontinous phrases be determined? We used a purely statistical definition on the one hand and dictionaries on the other. A combination of the two methods would be desirable. (ii) How can we distinguish between phrases that only occur in continuous form and phrases that must or can occur discontinuously? (iii) Given a sentence that contains the parts of a discontinuous phrase in correct order, how do we determine that the cooccurrence of the two parts constitutes an instance of the discontinuous phrase? (iv) Which tasks benefit most significantly from the introduction of generalized phrases?

Acknowledgments

This work was funded by DFG (grant SCHU 2246/4). We thank Google for a travel grant to support the presentation of this paper.

References

William Blacoe and Mirella Lapata. 2012. A comparison of vector-based representations for semantic composition. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 546–556. Association for Computational Linguistics.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

Georgiana Dinu, Nghia The Pham, and Marco Baroni. 2013. General estimation and evaluation of compositional distributional semantic models. In *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*, pages 50–58.

Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of the 20th international conference on Computational Linguistics*, page 350. Association for Computational Linguistics.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874.

Yangfeng Ji and Jacob Eisenstein. 2013. Discriminative improvements to distributional sentence similarity. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 891–896.

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

Nitin Madnani, Joel Tetreault, and Martin Chodorow. 2012. Re-examining machine translation metrics for paraphrase identification. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 182–190. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. *arXiv preprint arXiv:1310.4546*.

George Miller and Christiane Fellbaum. 1998. Wordnet: An electronic lexical database.

Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive science*, 34(8):1388–1429.

Robert Parker, Linguistic Data Consortium, et al. 2009. *English gigaword fourth edition*. Linguistic Data Consortium.

Richard Socher, Christopher D Manning, and Andrew Y Ng. 2010. Learning continuous phrase representations and syntactic parsing with recursive neural networks. In *Proceedings of the NIPS-2010 Deep Learning and Unsupervised Feature Learning Workshop*, pages 1–9.

Richard Socher, Jeffrey Pennington, Eric H Huang, Andrew Y Ng, and Christopher D Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 151–161.

Richard Socher, Brody Huval, Christopher D Manning, and Andrew Y Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1201–1211.

Peter D Turney. 2012. Domain and function: A dual-space model of semantic relations and compositions. *Journal of Artificial Intelligence Research*, 44:533–585.

Peter D Turney. 2013. Distributional semantics beyond words: Supervised learning of analogy and paraphrase. *Transactions of the Association for Computational Linguistics*, 1:353–366.