Evaluation of Simple Distributional Compositional Operations on Longer Texts

Tamara Polajnar*, Laura Rimell, Stephen Clark
Computer Laboratory
University of Cambridge
Cambridge, UK
*tamara.polajnar@cl.cam.ac.uk

Abstract

Distributional semantic models have been effective at representing linguistic semantics at the word level, and more recently research has moved on to the construction of distributional representations for larger segments of text. However, it is not well understood how the composition operators that work well on short phrase-based models scale up to full-length sentences. In this paper we test several simple compositional methods on a sentence-length similarity task and discover that their performance peaks at fewer than ten operations. We also introduce a novel sentence segmentation method that reduces the number of compositional operations.

Keywords: distributional semantics, sentence segmentation, sentence similarity

1. Introduction

Distributional semantic models Turney and Pantel (2010) have been effective at representing linguistic semantics at the word level, with comparison to human word similarity judgements being one of the most important means of evaluating such models (Deerwester et al., 1990; Landauer and Dumais, 1997; Bruni et al., 2012).

More recently, research has focused on how distributional word representations can be combined, initially producing representations of short, two-word phrases (Mitchell and Lapata, 2008; Mitchell and Lapata, 2010). Semantic similarity has continued to figure prominently in evaluation, particularly the dataset of (Mitchell and Lapata, 2010), which includes human similarity judgements for verb-object, noun-noun, and adjective-noun combinations.

In the last two years, similarity-based evaluations for longer text segments have been introduced, specifically the Semantic Textual Similarity (STS) shared tasks (Agirre et al., 2012; Agirre et al., 2013). Using naturally occurring data, these tasks represent a major increase in complexity, with systems needing to adequately represent sentences of up to 40 words in length.

A variety of methods have been applied to STS tasks, some of which make use of well-known composition operators whose effectiveness was comprehensively demonstrated on phrase similarity tasks. However, it is not known how well such operators scale up. Consider the following STS pair, which was given an average human similarity rating of 3.8 on a scale of 1-5:

> *"I know of no pressure," said Mr. Feith, the under secretary of defense for policy.*
> *"I know of nobody who pressured anybody," Douglas Feith, undersecretary of defense for policy, said at a Pentagon briefing.*

Elementwise multiplication of word vectors, for example, underestimates similarity, since the proper name Feith has an extremely sparse vector, which "zeroes out" the final sentence representation. Vector addition overestimates similarity, because of the dense vectors of the highly frequent words in the sentence. Another problem is that the stop words (*of, no, ...*) are clearly important for this sentence pair, but play no part in the vector representations of traditional models.

In this paper, we test several compositional methods familiar used in phrase similarity experiments, and show that, while they are applicable to short phrases, the multiplication-based methods degenerate quickly as the number of operations increases and that even the more stable summation operator is less accurate on STS data than the lexical overlap baseline.

We introduce a novel sentence segmentation method that re-introduces some of the discarded high-frequency terms and reduces the number of composition operations required to produce a sentence representation. We find that this method can help with certain datasets, but is also less accurate than lexical overlap. An analysis based on sentence length suggests that phrase-based methods fail to scale up to sentences longer than ten words.

2. Methods

Semantic similarity evaluations are based on a list of paired items. A method must produce a similarity score for each pair, which is typically compared indirectly against the gold standard scores using a rank correlation measure. In this paper we use Spearman's rank correlation coefficient (ρ), although we also tried Pearson and Kendall-τb and found general agreement in trends.

Since rank correlation cannot express how accurate a method is for any given sentence pair, and varies widely with the makeup of the dataset, we also examine the data using mean squared error (MSE). In order to apply MSE we normalise the gold standard scores and model output scores to $[0, 1]$ with:

$$norm(\mathbf{x}) = \left\{ \frac{\mathbf{x}_i - \min(\mathbf{x})}{\max(\mathbf{x}) - \min(\mathbf{x})} \right\}_i \quad (1)$$

where $\mathbf{x}$ is a list of scores.

Weight	WS353	MEN
tTest	**0.60**	**0.68**
tTest+RI	0.60	0.67
ML Orig	0.42	–

(a)

	tTest				ML Original		
Comb	Full	VO	NN	AN	VO	NN	AN
Prod	0.32	0.30	0.39	0.31	0.46	0.49	0.37
Sum	**0.38**	**0.33**	**0.47**	0.29	0.36	0.39	0.30
pConv	0.29	0.29	0.30	0.29	–	–	–
Conv	0.30	0.33	0.36	0.23	0.09	0.05	0.10
bigram	0.31	0.24	**0.46**	**0.40**	–	–	–

(b)

Table 1: (a) Word similarity and (b) phrasal similarity results.

2.1. Evaluation Datasets

To verify the performance of our vectors on word similarity tasks, we used WS353 (Finkelstein et al., 2002), containing 353 word pairs, and MEN (Bruni et al., 2012), containing 3,000 word pairs. To verify performance of word and bigram distributional vectors on phrase similarity, we used the phrase similarity dataset (ML2010) and evaluation technique from (Mitchell and Lapata, 2010).

The largest datasets annotated for similarity between longer texts come from the STS track of the SEMEVAL conference Agirre et al. (2012). We used the MSRpar dataset, which consists of 1,500 grammatically correct and diverse sentence pairs. We also used the SMT dataset, which contains 1,193 pairs of grammatical sentences and their (possibly ungrammatical) translations.

2.2. Distributional Vectors

We used a corpus of 450 million cleaned and lemmatised tokens from a September, 2012 snapshot of Wikipedia, and constructed vectors by using sentences as co-occurrence contexts. The set of context words C consisted of the 10,000 most frequent words occurring in this dataset, with the exception of standard stopwords and the 25 most frequent words in the corpus. Therefore, a frequency vector for a target word $w_i \in W$ is represented as $\vec{w_i} = \{f_{w_i c_j}\}_j$, where $c_j \in C$, $|C| = 10000$, W is a set of target words in a particular evaluation dataset, and $f_{w_i c_j}$ is the co-occurrence frequency between the target and context words.

We first reweight our co-occurrence vectors with tTest

$$tTest(\vec{w_i}, c_j) = \frac{p(w_i, c_j) - p(w_i)p(c_j)}{\sqrt{p(w_i)p(c_j)}} \quad (2)$$

where $p(w_i) = \frac{\sum_j f_{w_i c_j}}{\sum_k \sum_l f_{w_k c_l}}$, $p(c_j) = \frac{\sum_i f_{w_i c_j}}{\sum_k \sum_l f_{w_k c_l}}$, and $p(w_i, c_j) = \frac{f_{w_i c_j}}{\sum_k \sum_l f_{w_k c_l}}$. Then we also create a random indexed (RI) version for use with the convolution operator.[1] We follow (Jones and Mewhort, 2007) and assign each context word a random vector $\vec{e_{c_j}} = \{r_k\}_k$ where r_k are drawn from the normal distribution $\mathcal{N}(0, \frac{1}{D})$ and $|\vec{e_{c_j}}| = D = 4096$. The RI representation of a target word $RI(\vec{w_i}) = \vec{w_i}\mathbf{R}$ is constructed by multiplying the word vector $\vec{w_i}$, obtained as before, by the $|C| \times D$ matrix $\mathbf{R}$ where each column represents the vectors $\vec{e_{c_j}}$.

2.3. Composition and Segmentation

Operators To combine distributional vectors into a single-vector sentence representation, we use a representative set of methods from (Mitchell and Lapata, 2010). In particular we use vector addition, elementwise (Hadamard) product, and periodic circular convolution (Plate, 1991; Jones and Mewhort, 2007), which are defined as follows for two word vectors $\vec{w_k}, \vec{w_l}$:

Sum $\vec{w_k} + \vec{w_l} = \{\vec{w_{k}}_i + \vec{w_{l}}_i\}_i$

Prod $\vec{w_k} \odot \vec{w_l} = \{\vec{w_{k}}_i \cdot \vec{w_{l}}_i\}_i$

Conv $\vec{w_k} \circledast \vec{w_l} = \left\{\sum_{j=0}^{n} (\vec{w_k})_{j\%n} \cdot (\vec{w_l})_{(i-j)\%n}\right\}_i$

We also use **pConv**, a variation on **Conv** where one of the operand vectors is permutated to force the operation to be non-commutative and thus encode word order. We build the sentence vectors iteratively from left to right.

The standard baseline for the sentence data is lexical overlap (**Lex**). Here we calculate **Lex** as the cosine between the vectors encoding bag-of-lemmatised-words representations of sentences with stopwords removed, which improves the baseline performance.

Segmentation To consider the effect that the average sentence length of the dataset has on the operators, we can artificially reduce the sentence length by reducing the number of operations required to produce the full sentence vector. To do this we segment the sentence into n-grams and combine their distributional vectors using the above compositional operators, effectively treating the n-gram as a single word. This approach has precedent in Baroni and Zamparelli (2010), who assume that n-gram vectors are more accurate than composed vectors, given a sufficiently large number of examples in the corpus; we also tested our distributional bigram vectors on ML2010 (Table 1).

Using n-grams with $n = 1...4$ having corpus frequency greater than 50, we segment each sentence into the longest possible non-overlapping sequences. To accomplish this we employ the Viterbi segmentation algorithm (Russell and Norvig, 2003). This algorithm chooses segments according to a fitness function, which we define for an n-gram ng as:

$$Ft(ng) = e^{\frac{|ng|}{maxN+1} + \frac{\ln(f_{ng}+1)}{100}} e^{-1} \quad (3)$$

where $|ng|$ is the length of ng in words, $maxN$ is the length of the longest n-grams available to the algorithm, and f_{ng} is the frequency of the n-gram in the corpus from which we gathered its distributional vector. The function weights n-grams on a scale of $[0.2, 1]$, giving preference first to longer and then to more frequent segments.[2]

[1] Mitchell and Lapata (Mitchell and Lapata, 2010) omitted RI vectors, reducing convolution performance.

[2] We assign the default value of 0.2 to single words with no distributional vector, i.e. unseen words or stopwords.

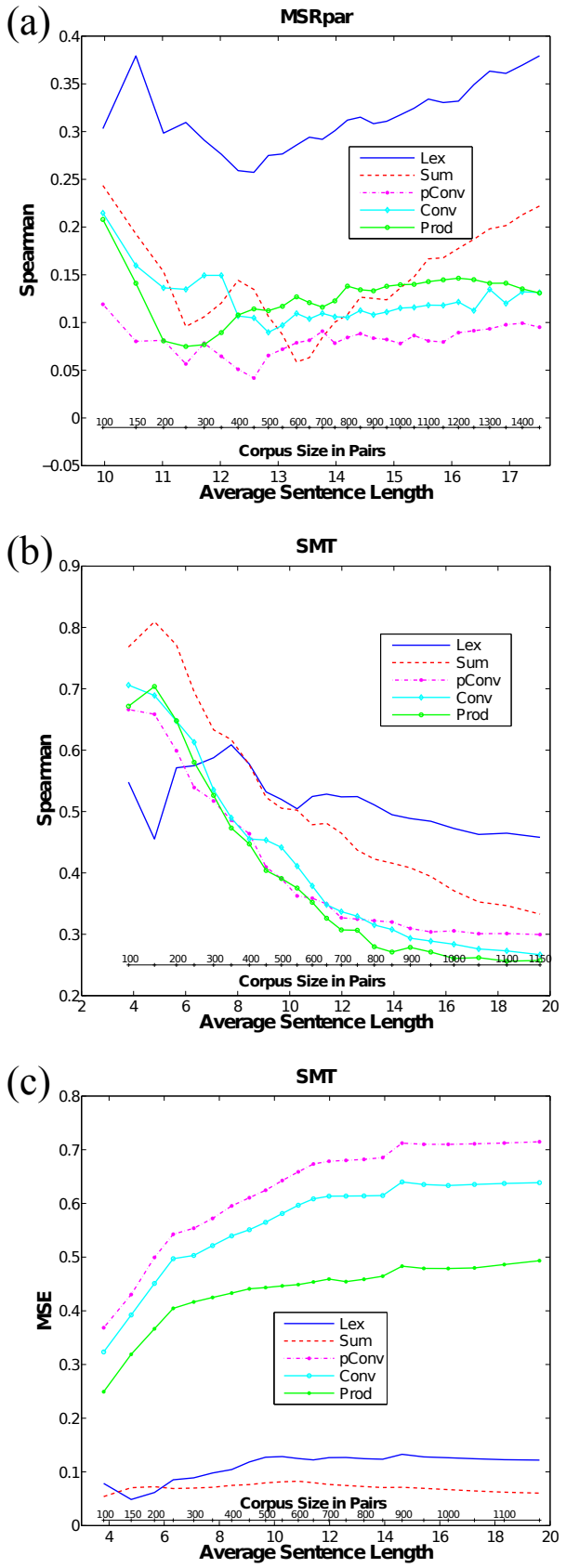

Figure 1: Spearman and MSE analysis of sentence datasets (tTest) ordered by sentence length.

	MSRpar		SMT	
Comb	MaxN	ρ	MaxN	ρ
Prod	2	0.21	4	0.31
Sum	1	**0.23**	4	**0.38**
pConv	2	0.10	3	0.25
Conv	3	0.16	4	0.29
Lex	1	0.38	1	0.46

Table 2: Sentence similarity results.

3. Experiments and Results

Table 1(a) shows that our distributional vectors outperformed the vectors of Mitchell and Lapata (2010) on WS353, and performed equally well on MEN, indicating they are of sufficient quality to use in experiments with longer phrases. RI produced a statistically insignificant reduction in performance on word similarity test.

Table 1(b) shows our results on phrase similarity, which are somewhat lower than those of Mitchell and Lapata (2010), except for the convolution results which are higher due to the use of RI-encoded vectors. Together the results in Tables 1(a) and 1(b) indicate that better performance of the word vectors does not translate directly to compositional tasks. Overall, **Prod** and **Sum** perform slightly better than **Conv** and **pConv**. Interestingly, **bigram**, which refers to using the distributional vectors for bigrams directly, has the highest correlation of all our methods for noun-noun and adjective-noun phrases. Correlation is lower on verb-object phrases, because verb-object bigrams (e.g. *ask man*) are rare.

Table 2 shows our results on the two sentence similarity datasets. For each composition method, the MaxN column shows the maximum size n-gram available to the segmentation algorithm; i.e. MaxN = 1 corresponds to word-by-word composition. While all of the methods underperform **Lex** by a wide margin, the best method is **Sum**, followed by **Prod**. The overall results are higher on SMT than MSRpar. In general, the use of n-grams was beneficial, especially for the SMT dataset where 4-grams performed the highest overall.

To further examine the effect of sentence length, we calculate sentence length in words for each pair in the SMT and MSRpar datasets, and then order the datasets by increasing average pair length. We create the first subset by taking the top 100 shortest pairs, subsequently adding 50 more pairs at a time. At each iteration we test for correlation (higher is better) and MSE (lower is better). Figures 1 (a-b) show that in general **Sum** is overall the most stable composition method and often follows the trends of **Lex**, probably because it retains the full vectors of all the words in the sentence. **Prod, Conv** and **pConv** are grouped together. This is further supported by MSE analysis in Figure 1 (c), which shows that normalised **Sum** similarity values are overall closer to the gold standard than **Lex** on SMT data, although for MSRpar data (not pictured) the general trend was the same but **Lex** was consistently better.

Although product-based measures appear to occasionally outperform **Lex**, this is a byproduct of the use of correlation measures. In fact, with tTest weighting, these measures tend to produce zero similarity for between 30-60%

of the pairs, increasing with sentence length. This makes it difficult to correctly judge at which point **Lex** starts outperforming compositional methods, although Figure 1(a-b) together indicate that the crossover point is certainly at a length under ten words. While on MSRpar data **Lex** shows lower MSE than **Sum**, on SMT data (pictured) the MSE shows a reversal of performance demonstrated by correlation, i.e. for **Sum** both the correlation and error decrease, when we expect them to be inversely proportional. that often, **Sum** is able to account for difficult pairs that lead to the peaked behaviour in **Lex** correlation scores. The segmentation experiments in Table 2 support the graph data with methods generally preferring more operations (shorter n-grams) on MSRpar data and fewer operations (longer n-grams) on SMT data.

4. Conclusion

This is the first paper to systematically examine the extension of simple compositional operators to longer sentences. While the product-based measures are adequate for comparison of short phrases they degrade as the number of operations increases. The **Sum** operator is more consistent, although this is not visible from correlation evaluation which is flawed due to the treatment of items tied at zero. We have also introduced a method for segmenting a sentence into frequent high-quality n-grams that can provide a baseline for more complex compositional frameworks. It reduces the number of operations and encodes some of the useful high-frequency words that are otherwise discarded.

5. Acknowledgements

Tamara Polajnar is supported by ERC Starting Grant DisCoTex (306920). Laura Rimell is supported by EPSRC grant EP/I037512/1. Stephen Clark is supported by ERC Starting Grant DisCoTex (306920) and EPSRC grant EP/I037512/1.

6. References

Agirre, E., Cer, D., Diab, M., and Gonzalez-Agirre, A. (2012). Semeval-2012 task 6: A pilot on semantic textual similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics*, pages 385–393, Montréal, Canada, 7-8 June. Association for Computational Linguistics.

Agirre, E., Cer, D., Diab, M., Gonzalez-Agirre, A., and Guo, W. (2013). *SEM 2013 shared task: Semantic textual similarity. In *The Second Joint Conference on Lexical and Computational Semantics (*SEM 2013)*, Atlanta, GA.

Baroni, M. and Zamparelli, R. (2010). Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2010)*, pages 1183–1193, East Stroudsburg PA.

Bruni, E., Boleda, G., Baroni, M., and Tran, N. K. (2012). Distributional semantics in technicolor. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 136–145, Jeju Island, Korea, July. Association for Computational Linguistics.

Deerwester, S., Dumais, S. T., Landauer, T. K., Furnas, G. W., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the Society for Information Science*, 41(6):391–407.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., and Ruppin, E. (2002). Placing search in context: The concept revisited. *ACM Transactions on Information Systems*, 20:116–131.

Jones, M. N. and Mewhort, D. J. K. (2007). Representing word meaning and order information in a composite holographic lexicon. *Psychological Review*, 114:1–37.

Landauer, T. and Dumais, S. (1997). A solution to Plato's problem: the latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 104:211–240.

Mitchell, J. and Lapata, M. (2008). Vector-based models of semantic composition. In *Proceedings of ACL-HLT*, Columbus, OH.

Mitchell, J. and Lapata, M. (2010). Composition in distributional models of semantics. *Cognitive Science*, 34(8):1388–1429.

Plate, T. A. (1991). Holographic reduced Representations: Convolution algebra for compositional distributed representations. In Mylopoulos, J. and Reiter, R., editors, *Proceedings of the 12th International Joint Conference on Artificial Intelligence, Sydney, Australia, August 1991*, pages 30–35, San Mateo, CA. Morgan Kauffman.

Russell, S. and Norvig, P. (2003). *Artificial Intelligence, A Modern Approach, Second Edition*.

Turney, P. and Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37:141–188.