Saarland: Vector-based models of semantic textual similarity

Georgiana Dinu
Center of Mind/Brain Sciences
University of Trento
georgiana.dinu@unitn.it

Stefan Thater
Dept. of Computational Linguistics
Universität des Saarlandes
stth@coli.uni-saarland.de

Abstract

This paper describes our system for the Semeval 2012 Sentence Textual Similarity task. The system is based on a combination of few simple vector space-based methods for word meaning similarity. Evaluation results show that a simple combination of these unsupervised data-driven methods can be quite successful. The simple vector space components achieve high performance on short sentences; on longer, more complex sentences, they are outperformed by a surprisingly competitive word overlap baseline, but they still bring improvements over this baseline when incorporated into a mixture model.

1 Introduction

Vector space models are widely-used methods for word meaning similarity which exploit the so-called *distributional hypothesis*, stating that semantically similar words tend to occur in similar contexts. Word meaning is represented by the contexts in which a word occurs, and similarity is computed by comparing these contexts in a high-dimensional vector space (Turney and Pantel, 2010). Distributional models of word meaning are attractive because they are simple, have wide coverage, and can be easily acquired at virtually no cost in an unsupervised way. Furthermore, recent research has shown that, at least to some extent, these models can be generalized to capture similarity beyond the (isolated) word level, either as lexical meaning modulated by context, or as vectorial meaning representations for phrases and sentences. In this paper we evaluate the use of some of these models for the Semantic Textual Similarity (STS) task, which measures the degree of semantic equivalence between two sentences.

In recent work Mitchell and Lapata (2008) has drawn the attention to the question of building vectorial meaning representations for sentences by combining individual word vectors. They propose a family of simple "compositional" models that compute a vector for a phrase or a sentence by combining vectors of the constituent words, using different operations such as vector addition or component-wise multiplication. More refined models have been proposed recently by Baroni and Zamparelli (2010) and Grefenstette and Sadrzadeh (2011).

Thater et al. (2011) and others take a slightly different perspective on the problem: Instead of computing a vector representation for a complete phrase or sentence, they focus on the problem of "disambiguating" the vector representation of a target word based on distributional information about the words in the target's context. While this approach is not "compositional" in the sense described above, it still captures some meaning of the complete phrase in which a target word occurs.

In this paper, we report on the system we used in the Semeval 2012 Sentence Textual Similarity shared task and describe an approach that uses a combination of few simple vector-based components. We extend the model of Thater et al. (2011), which has been shown to perform well on a closely related paraphrase ranking task, with an additive composition operation along the lines of Mitchell and Lapata (2008), and compare it with a simple alignment-based approach which in turn uses vector-based similarity scores. Results show that in particular the alignment-based approach can achieve good performance on the Microsoft Research Video Description dataset. On the other datasets, all vector-based components are outperformed by a surprisingly competitive word

*First Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 603–607,
Montréal, Canada, June 7-8, 2012.

overlap baseline, but they still bring improvements over this baseline when incorporated into a mixture model. On the test dataset, the mixture model ranks 10th and 13th on the Microsoft Research Paraphrase and Video Description datasets, respectively, which we take this to be a quite promising result given that we use only few relatively simple vector based components to compute similarity scores for sentences.

The rest of the paper is structured as follows: Section 2 presents the individual vector-based components used by our system. In Section 3 we present detailed evaluation results on the training set, as well as results for our system on the test set, while Section 4 concludes the paper.

2 Systems for Sentence Similarity

Our system is based on four different components: We use two different vector space models to represent word meaning—a basic bag-of-words model and a slightly simplified variant of the contextualization model of Thater et al. (2011)—and two different methods to compute similarity scores for sentences based on these two vector space models—one "compositional" method that computes vectors for sentences by summing over the vectors of the constituent words, and one alignment-based method that uses vector-based similarity scores for word pairs to compute an alignment between the words in the two sentences.

2.1 Vector Space Models

For the basic vector-space model, we assume a set W of words, and represent the meaning of a word $w \in W$ by a vector in the vector space V spanned by the set of basis vectors $\{\vec{e}_{w'} \mid w' \in W\}$ as follows:

$$v_{basic}(w) = \sum_{w' \in W} f(w, w')\, \vec{e}_{w'}$$

where f is a function that assigns a co-occurrence value to the word pair (w, w'). In the experiments reported below, we use pointwise mutual information estimated on co-occurrence frequencies for words within a 5-word window around the target word on either side.[1]

[1] We use a 5-word window here as this setting has been shown to give best results on a closely related task in the literature (Mitchell and Lapata, 2008)

This basic "bag of words" vector space model represents word meaning by summing over all contexts in which the target word occurs. Since words are often ambiguous, this means that context words pertaining to different senses of the target word are mixed within a single vector representation, which can lead to "noisy" similarity scores. The vector for the noun *coach*, for instance, contains context words like *teach* and *tell* (person sense) as well as *derail* and *crash* (vehicle sense).

To address this problem, Thater et al. (2011) propose a "contextualization" model in which the individual components of the target word's vector are reweighted, based on distributional information about the words in the target's context. Let us assume that the context consist of a single word c. The vector for a target w in context c is then defined as:

$$v(w, c) = \sum_{w' \in W} \alpha(c, w')\, f(w, w')\, \vec{e}_{w'}$$

where α is some similarity score that quantifies to what extent the vector dimension that corresponds to w' is compatible with the observed context c. In the experiments reported below, we take α to be the cosine similarity of c and w'; see Section 3 for details.

In the experiments reported below, we use all words in the syntactic context of the target word to contextualize the target:

$$v_{ctx}(w) = \sum_{c \in C(w)} v(w, c)$$

where $C(w)$ is the context in which w occurs, i.e. all words related to w by a dependency relation such as subject or object, including inverse relations.

Remark. The contextualization model presented above is a slightly simplified version of the original model of Thater et al. (2011): it uses standard bag-of-words vectors instead of syntax-based vectors. This simplified version performs better on the training dataset. Furthermore, the simplified model has been shown to be equivalent to the models of Erk and Padó (2008) and Thater et al. (2010) by Dinu and Thater (2012), so the results reported below carry over directly to these other models as well.

2.2 Vector Composition and Alignment

The two vector space models sketched above represent the meaning of *words,* and thus cannot be applied

directly to model similarity of phrases or sentences. One obvious and straightforward way to extend these models to the sentence level is to follow Mitchell and Lapata (2008) and represent sentences by vectors obtained by summing over the individual vectors of the constituent words. These "compositional" models can then be used to compute similarity scores between sentence pairs in a straightforward way, simply by computing the cosine of the angle between vectors (or some other similarity score) for the two sentences:

$$sim_{add}(S,S') = cos\big(\sum_{w\in S} v(w), \sum_{w'\in S'} v(w')\big) \qquad (1)$$

where $v(w)$ can be instantiated either with *basic* or with *ctx* vectors.

In addition to the compositional models, we also experimented with an alignment-based approach: Instead of computing vectors for complete sentences, we compute an alignment between the words in the two sentences. To be more precise, we compute cosine similarity scores between all possible pairs of words (tokens) of the two sentences; based on these similarity scores, we then compute a one-to-one alignment between the words in the two sentences[2], using a greedy search strategy (see Fig. 1). We assign a weight to each link in the alignment which is simply the cosine similarity score of the corresponding word pair and take the sum of the link weights, normalized by the maximal length of the two sentences to be the corresponding similarity score for the two sentences. The final score is then:

$$sim_{align}(S,S') = \frac{\sum_{(w,w')\in \textsc{Align}(S,S')} cos(v(w),v(w'))}{max(|S|,|S'|)}$$

where $v(w)$ is the vector for w, which again can be either the basic or the contextualized vector.

3 Evaluation

In this section we present our experimental results. In addition to the models described in Section 2, we define a baseline model which simply computes the word overlap between two sentences as:

$$sim_{overlap}(S,S') = \frac{|S\cap S'|}{|S\cup S'|} \qquad (2)$$

[2]Note that this can result in some words not being aligned

function ALIGN(S_1, S_2)
  alignment $\leftarrow \emptyset$
  marked $\leftarrow \emptyset$
  pairs $\leftarrow \{\langle w,w'\rangle \mid w \in S_1, w' \in S_2\}$
  while pairs not empty **do**
    $\langle w,w'\rangle \leftarrow$ highest cosine pair in pairs
    if $w \notin$ marked and $w' \notin$ marked **then**
      alignment $\leftarrow \langle w,w'\rangle \cup$ alignment
      marked $\leftarrow \{w,w'\} \cup$ marked
    end if
    pairs $\leftarrow$ pairs $\setminus \{\langle w,w'\rangle\}$
  end while
  return alignment
end function

Figure 1: The alignment algorithm

The score assigned by this method is simply the number of words that the two sentences have in common divided by their total number of words. Finally, we also propose a straightforward mixture model which combines all of the above methods. We use the training data to fit a degree two polynomial over these individual predictors using least squares regression. We report cross-validation scores.

3.1 Evaluation setup

The vector space used in all experiments is a bag-of-words space containing word co-occurrence counts. We use the GigaWord (1.7 billion tokens) as input corpus and extract word co-occurrences within a symmetric 5-word context window. Co-occurrence counts smaller than three are set to 0 and we further apply (positive) pmi weighting.

3.2 Training results

The training data results are shown in Figure 2. The best performance on the video dataset is achieved by the alignment method using a basic vector representation to compute word-level similarity. All vector-space methods perform considerably better than the simple word overlap baseline on this dataset, the alignment method achieving almost 20% gain over this baseline. This indicates that information about the meaning of the words is very beneficial for this type of data, consisting of small, well-structured sentences.

Using the alignment method with contextualized

Component	MSRvid	MSRpar	SMTeur
basic/add	70.9	33.3	31.8
ctx/add	65.7	23.0	30.4
basic/align	**74.6**	40.5	32.1
overlap	56.8	**59.5**	**50.0**
mixture	**78.1**	**61.8**	**54.1**

Figure 2: Results on the training set.

Component	MSRvid	MSRpar	SMTeur
basic/add	−2.1	−0.1	−1.5
ctx/add	−0.6	+1.3	+0.4
basic/align	−4.1	−1.9	−2.6
overlap	−0.1	−17.0	−23.0

Figure 3: Results on the training set when removing individual components from the mixture model.

vector representations (omitted in the table) does not bring any improvement and it performs similarly to the *ctx/add* method. This suggests that aligning similar words in the two sentences does not benefit from further meaning disambiguation through contextualized vectors and that some level of disambiguation may be implicitly performed.

On the paraphrase and europarl datasets, the overlap baseline outperforms, by a large margin, the vector space models. This is not surprising, as it is known that word overlap baselines can be very competitive on Recognizing Textual Entailment datasets, to which these two datasets bare a large resemblance. In particular this indicates that the methods proposed for combining vector representations of words do not provide, in the current state, accurate models for modeling the meaning of larger sentences.

We also report 10-fold cross-validation scores obtained with the mixture model. On all datasets, this outperforms the individual methods, improving by a margin of 2%-4% the best single methods. In particular, on the paraphrase and europarl datasets, this shows that despite the considerably inferior performance of the vector-based methods, these can still help improve the overall performance.

This is also reflected in Table 3, where we evaluate the performance of the mixture method when, in turn, one of the individual components is excluded: with few exceptions, all components contribute to the performance of the mixtures.

3.3 Test results

We have submitted as our official runs the best single vector space model, performing alignment with basic vector similarity, as well as the mixture methods. The mixture method uses weights individually learned for each of the datasets made available during training. For the two surprise datasets we carry over the weights of what we have considered to be the most similar training-available sets: video weights of ontonotes and paraphrase weights for news.

The test data results are given in 4. We report the results for the individual datasets as well as the mean Pearson correlation, weighted by the sizes of the datasets. The table also shows the performance of the official task baseline as well as the top three runs accoring to the overall weighted mean score.

As expected, the mixture method outperforms by a large margin the alignment model, achieving rank 10 and rank 13 on the video and paraphrase datasets. Overall the mixture method ranks 43 according to the weighted mean measure (rank 22 if correcting our official submission which contained the wrong output file for the europarl dataset). The other more controversial measures rank our official, *not* corrected, submission at position 13 (RankNrm) and 71 (Rank), overall. This is an encouraging result, as the individual components we have used are all unsupervised, obtained solely from large amounts of unlabeled data, and with no other additional resources. The training data made available has only been used to learn a set of weights for combining these individual components.

4 Conclusions

This paper describes an approach that combines few simple vector space-based components to model sentence similarity. We have extended the state-of-the-art model for contextualized meaning representations of Thater et al. (2011) with an additive composition operation along the lines of Mitchell and Lapata (2008). We have combined this with a simple alignment-based method and a word overlap baseline into a mixture model.

Our system achieves promising results in particular

Dataset	basic/align	mixture	baseline	Run1	Run2	Run3
MSRvid	77.1	83.1	30.0	87.3	88.0	85.6
MSRpar	40.4	63.1	43.3	68.3	73.4	64.0
SMTeur	26.8	13.9 (37.1*)	45.4	52.8	47.7	51.5
OnWN	57.2	59.6	58.6	66.4	67.9	71.0
SMTnews	35.0	38.0	39.1	49.3	39.8	48.3
ALL	49.5	45.4	31.1	82.3	81.3	73.3
Rank	65	71	87	1	3	15
ALLNrm	78.7	82.5	67.3	85.7	86.3	85.2
RankNrm	50	13	85	2	1	5
Mean	50.6	56.6 (60.0*)	43.5	67.7	67.5	67.0
RankMean	60	43 (22*)	70	1	2	3

Figure 4: Results on the test set. * – corrected score (official results score wrong prediction file we have submitted for the europarl dataset). Official baseline and top three runs according to the weighted mean measure.

on the Microsoft Research Paraphrase and Video Description datasets, on which it ranks 13th and 10th, respectively. We take this to be a promising result, given that our focus has not been the development of a highly-competitive complex system, but rather on investigating what performance can be achieved when using only vector space methods.

An interesting observation is that the methods for combining word vector representations (the vector addition, or the meaning contextualization) can be beneficial for modeling the similarity of the small, well-structured sentences of the video dataset, however they do not perform well on comparing longer, more complex sentences. In future work we plan to further investigate methods for composition in vector space models using the STS datasets, in addition to the small, controlled datasets that have been typically used in this line of research.

Acknowledgments. This work was supported by the Cluster of Excellence "Multimodal Computing and Interaction," funded by the German Excellence Initiative.

References

Marco Baroni and Roberto Zamparelli. 2010. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, Cambridge, MA, October. Association for Computational Linguistics.

Georgiana Dinu and Stefan Thater. 2012. A comparison of models of word meaning in context. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Short paper, to appear.

Katrin Erk and Sebastian Padó. 2008. A structured vector space model for word meaning in context. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, Honolulu, HI, USA.

Edward Grefenstette and Mehrnoosh Sadrzadeh. 2011. Experimental support for a categorical compositional distributional model of meaning. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *Proceedings of ACL-08: HLT*, Columbus, OH, USA.

Stefan Thater, Hagen Fürstenau, and Manfred Pinkal. 2010. Contextualizing semantic representations using syntactically enriched vector models. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Uppsala, Sweden.

Stefan Thater, Hagen Fürstenau, and Manfred Pinkal. 2011. Word meaning in context: A simple and effective vector model. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1134–1143, Chiang Mai, Thailand, November. Asian Federation of Natural Language Processing.

Peter D. Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space modes of semantics. *Journal of Artificial Intelligence Research*, 37:141–188.