Exemplar-based Word-Space Model for Compositionality Detection: Shared task system description

Siva Reddy
University of York, UK
siva@cs.york.ac.uk

Diana McCarthy
Lexical Computing Ltd, UK
diana@dianamccarthy.co.uk

Suresh Manandhar
University of York, UK
suresh@cs.york.ac.uk

Spandana Gella
University of York, UK
spandana@cs.york.ac.uk

Abstract

In this paper, we highlight the problems of polysemy in word space models of compositionality detection. Most models represent each word as a single prototype-based vector without addressing polysemy. We propose an exemplar-based model which is designed to handle polysemy. This model is tested for compositionality detection and it is found to outperform existing prototype-based models. We have participated in the shared task (Biemann and Giesbrecht, 2011) and our best performing exemplar-model is ranked first in two types of evaluations and second in two other evaluations.

1 Introduction

In the field of computational semantics, to represent the meaning of a compound word, two mechanisms are commonly used. One is based on *the distributional hypothesis* (Harris, 1954) and the other is on *the principle of semantic compositionality* (Partee, 1995, p. 313).

The distributional hypothesis (DH) states that words that occur in similar contexts tend to have similar meanings. Using this hypothesis, distributional models like the Word-space model (WSM, Sahlgren, 2006) represent a target word's meaning as a *context vector* (location in space). The similarity between two meanings is the *closeness* (proximity) between the vectors. The context vector of a target word is built from its distributional behaviour observed in a corpus. Similarly, the context vector of a compound word can be built by treating the compound as a single word. We refer to such a vector as a DH-based vector.

The other mechanism is based on the principle of semantic compositionality (PSC) which states that the meaning of a compound word is a function of, and only of, the meaning of its parts and the way in which the parts are combined. If the meaning of a part is represented in a WSM using the distributional hypothesis, then the principle can be applied to compose the distributional behaviour of a compound word from its parts without actually using the corpus instances of the compound. We refer to this as a PSC-based vector. So a PSC-based is composed of component DH-based vectors.

Both of these two mechanisms are capable of determining the meaning vector of a compound word. For a given compound, if a DH-based vector and a PSC-based vector of the compound are projected into an identical space, one would expect the vectors to occupy the same location i.e. both the vectors should be nearly the same. However the principle of semantic compositionality does not hold for non-compositional compounds, which is actually what the existing WSMs of compositionality detection exploit (Giesbrecht, 2009; Katz and Giesbrecht, 2006; Schone and Jurafsky, 2001). The DH-based and PSC-based vectors are expected to have high similarity when a compound is compositional and low similarity for non-compositional compounds.

Most methods in WSM (Turney and Pantel, 2010) represent a word as a single context vector built from merging all its corpus instances. Such a representation is called the *prototype-based* modelling (Murphy, 2002). These prototype-based vectors do not

distinguish the instances according to the senses of a target word. Since most compounds are less ambiguous than single words, there is less need for distinguishing instances in a DH-based prototype vector of a compound and we do not address that here but leave ambiguity of compounds for future work. However the constituent words of the compound are more ambiguous. When DH-based vectors of the constituent words are used for composing the PSC-based vector of the compound, the resulting vector may contain instances, and therefore contexts, that are not relevant for the given compound. These noisy contexts effect the similarity between the PSC-based vector and the DH-based vector of the compound. Basing compositionality judgements on a such a noisy similarity value is no longer reliable.

In this paper, we address this problem of polysemy of constituent words of a compound using an exemplar-based modelling (Smith and Medin, 1981). In exemplar-based modelling of WSM (Erk and Padó, 2010), each word is represented by all its corpus instances (*exemplars*) without merging them into a single vector. Depending upon the purpose, only relevant exemplars of the target word are activated and then these are merged to form a refined prototype-vector which is less-noisy compared to the original prototype-vector. Exemplar-based models are more powerful than prototype-based ones because they retain specific instance information.

We have evaluated our models on the validation data released in the shared task (Biemann and Giesbrecht, 2011). Based on the validation results, we have chosen three systems for public evaluation and participated in the shared task (Biemann and Giesbrecht, 2011).

2 Word Space Model

In this section, construction of WSM for all our experiments is described. We use Sketch Engine[1] (Kilgarriff et al., 2004) to retrieve all the exemplars for a target word or a pattern using corpus query language. Let w_1 w_2 be a compound word with constituent words w_1 and w_2. E_w denotes the set of exemplars of w. V_w is the prototype vector of the word w, which is built by merging all the exemplars in E_w

---

[1] Sketch Engine http://www.sketchengine.co.uk

For the purposes of producing a PSC-based vector for a compound, a vector of a constituent word is built using only the exemplars which *do not* contain the compound. Note that the vectors are sensitive to a compound's word-order since the exemplars of w_1 w_2 are not the same as w_2 w_1.

We use other WSM settings following Mitchell and Lapata (2008). The dimensions of the WSM are the top 2000 content words in the given corpus (along with their coarse-grained part-of-speech information). Cosine similarity *(sim)* is used to measure the similarity between two vectors. Values at the specific positions in the vector representing context words are set to the ratio of the probability of the context word given the target word to the overall probability of the context word. The context window of a target word's exemplar is the whole sentence of the target word excluding the target word. Our language of interest is English. We use the ukWaC corpus (Ferraresi et al., 2008) for producing out WSMs.

3 Related Work

As described in Section 1, most WSM models for compositionality detection measure the similarity between the true distributional vector $V_{w_1 w_2}$ of the compound and the composed vector $V_{w_1 \oplus w_2}$, where $\oplus$ denotes a compositionality function. If the similarity is high, the compound is treated as compositional or else non-compositional.

Giesbrecht (2009); Katz and Giesbrecht (2006); Schone and Jurafsky (2001) obtained the compositionality vector of w_1 w_2 using vector addition $V_{w_1 \oplus w_2} = aV_{w_1} + bV_{w_2}$. In this approach, if $sim(V_{w_1 \oplus w_2}, V_{w_1 w_2}) > \gamma$, the compound is classified as compositional, where γ is a threshold for deciding compositionality. Global values of a and b were chosen by optimizing the performance on the development set. It was found that no single threshold value γ held for all compounds. Changing the threshold alters performance arbitrarily. This might be due to the polysemous nature of the constituent words which makes the composed vector $V_{w_1 \oplus w_2}$ filled with noisy contexts and thus making the judgement unpredictable.

In the above model, if a=0 and b=1, the resulting model is similar to that of Baldwin et al. (2003). They also observe similar behaviour of the thresh-

old γ. We try to address this problem by addressing the polysemy in WSMs using exemplar-based modelling.

The above models use a simple addition based compositionality function. Mitchell and Lapata (2008) observed that a simple multiplication function modelled compositionality better than addition. Contrary to that, Guevara (2011) observed additive models worked well for building compositional vectors. In our work, we try using evidence from both compositionality functions, simple addition and simple multiplication.

Bannard et al. (2003); McCarthy et al. (2003) observed that methods based on distributional similarities between a phrase and its constituent words help when determining the compositionality behaviour of phrases. We therefore also use evidence from the similarities between each constituent word and the compound.

4   Our Approach: Exemplar-based Model

Our approach works as follows. Firstly, given a compound w_1 w_2, we build its DH-based prototype vector $V_{w_1 w_2}$ from all its exemplars $E_{w_1 w_2}$. Secondly, we remove irrelevant exemplars in E_{w_1} and E_{w_2} of constituent words and build the refined prototype vectors $V_{w_1^r}$ and $V_{w_2^r}$ of the constituent words w_1 and w_2 respectively. These refined vectors are used to compose the PSC-based vectors [2] of the compound. Related work to ours is (Reisinger and Mooney, 2010) where exemplars of a word are first clustered and then prototype vectors are built. This work does not relate to compositionality but to measuring semantic similarity of single words. As such, their clusters are not influenced by other words whereas in our approach for detecting compositionality, the other constituent word plays a major role.

We use the compositionality functions, simple addition and simple multiplication to build $V_{w_1^r + w_2^r}$ and $V_{w_1^r \times w_2^r}$ respectively. Based on the similarities $sim(V_{w_1 w_2}, V_{w_1^r})$, $sim(V_{w_1 w_2}, V_{w_2^r})$, $sim(V_{w_1 w_2}, V_{w_1^r + w_2^r})$ and $sim(V_{w_1 w_2}, V_{w_1^r \times w_2^r})$, we decide if the compound is compositional or non-compositional. These steps are described in a little more detail below.

---

[2]Note that we use two PSC-based vectors for representing a compound.

4.1   Building Refined Prototype Vectors

We aim to remove irrelevant exemplars of one constituent word with the help of the other constituent word's distributional behaviour. For example, let us take the compound *traffic light*. *Light* occurs in many contexts such as quantum theory, optics, lamps and spiritual theory. In ukWaC, $light$ has 316,126 instances. Not all these exemplars are relevant to compose the PSC-based vector of *traffic light*. These irrelevant exemplars increases the semantic differences between *traffic light* and *light* and thus increase the differences between $V_{\text{traffic} \oplus \text{light}}$ and $V_{\text{traffic light}}$. $sim(V_{\text{light}}, V_{\text{traffic light}})$ is found to be 0.27.

Our intuition and motivation for exemplar removal is that it is beneficiary to choose only the exemplars of *light* which share similar contexts of *traffic* since *traffic light* should have contexts similar to both *traffic* and *light* if it is compositional. We rank each exemplar of *light* based on common co-occurrences of *traffic* and also words which are distributionally similar to *traffic*. Co-occurrences of *traffic* are the context words which frequently occur with *traffic*, e.g. car, road etc. Using these, the exemplar from a sentence such as "*Cameras capture cars running red lights* ..." will be ranked higher than one which does not have contexts related to *traffic*. The distributionally similar words to *traffic* are the words (like synonyms, antonyms) which are similar to *traffic* in that they occur in similar contexts, e.g. transport, flow etc. Using these distributionally similar words helps reduce the impact of data sparseness and helps prioritise contexts of *traffic* which are semantically related. We use Sketch Engine to compute the scores of a word observed in a given corpus. Sketch Engine scores the co-occurrences (collocations) using logDice motivated by (Curran, 2003) and distributionally related words using (Rychlý and Kilgarriff, 2007; Lexical Computing Ltd., 2007). For a given word, both of these scores are normalised in the range (0,1)

All the exemplars of *light* are ranked based on the co-occurrences of these collocations and distributionally related words of *traffic* using

$$s_{E \in E_{\text{light}}}^{\text{traffic}} = \sum_{c \in E} x_c^E \times y_c^{\text{traffic}} \qquad (1)$$

where $s_{E \in E_{\text{light}}}^{\text{traffic}}$ stands for the relevance score of the

exemplar E w.r.t. *traffic*, c for context word in the exemplar E, x_c^E is the coordinate value (contextual score) of the context word c in the exemplar E and y_c^{traffic} is the score of the context word c w.r.t. *traffic*.

A refined prototype vector of *light* is then built by merging the top n exemplars of *light*

$$V_{\text{light}^r} = \sum_{e_i \in E_{\text{light}}^{\text{traffic}}; i=0}^{n} e_i \qquad (2)$$

where $E_{\text{light}}^{\text{traffic}}$ are the set of exemplars of *light* ranked using co-occurrence information from the other constituent word *traffic*. n is chosen such that $sim(V_{\text{light}^r}, V_{\text{traffic light}})$ is maximised. This similarity is observed to be greatest using just 2286 (less than 1%) of the total exemplars of *light*. After exemplar removal, $sim(V_{\text{light}^r}, V_{\text{traffic light}})$ increased to 0.47 from the initial value of 0.27. Though n is chosen by maximising similarity, which is not desirable for non-compositional compounds, the lack of similarity will give the strongest possible indication that a compound is not compositional.

4.2 Building Compositional Vectors

We use the compositionality functions, simple addition and simple multiplication to build compositional vectors $V_{w_1^r + w_2^r}$ and $V_{w_1^r \times w_2^r}$. These are as described in (Mitchell and Lapata, 2008). In model addition, $V_{w_1 \oplus w_2} = aV_{w_1} + bV_{w_2}$, all the previous approaches use static values of a and b. Instead, we use dynamic weights computed from the participating vectors using $a = \frac{sim(V_{w_1 w_2}, V_{w_1})}{sim(V_{w_1 w_2}, V_{w_1}) + sim(V_{w_1 w_2}, V_{w_2})}$ and $b = 1 - a$. These weights differ from compound to compound.

4.3 Compositionality Judgement

To judge if a compound is compositional or non-compositional, previous approaches (see Section 3) base their judgement on a single similarity value. As discussed, we base our judgement based on the collective evidences from all the similarity values using a linear equation of the form

$$\begin{aligned}
\alpha(V_{w_1^r}, V_{w_2^r}) = {} & a_0 + a_1.sim(V_{w_1 w_2}, V_{w_1^r}) \\
& + a_2.sim(V_{w_1 w_2}, V_{w_2^r}) \qquad (3) \\
& + a_3.sim(V_{w_1 w_2}, V_{w_1^r + w_2^r}) \\
& + a_4.sim(V_{w_1 w_2}, V_{w_1^r \times w_2^r})
\end{aligned}$$

Model	APD	Acc.
Exm-Best	13.09	88.0
Pro-Addn	15.42	76.0
Pro-Mult	17.52	80.0
Pro-Best	15.12	80.0

Table 1: Average Point Difference (APD) and Average Accuracy (Acc.) of Compositionality Judgements

where the value of α denotes the compositionality score. The range of α is in between 0-100. If $\alpha \leq 34$, the compound is treated as non-compositional, $34 < \alpha < 67$ as medium compositional and $\alpha \geq 67$ as highly compositional. The parameters a_i's are estimated using ordinary least square regression by training over the training data released in the shared task (Biemann and Giesbrecht, 2011). For the three categories – adjective-noun, verb-object and subject-verb – the parameters are estimated separately.

Note that if $a_1 = a_2 = a_4 = 0$, the model bases its judgement only on addition. Similarly if $a_1 = a_2 = a_3 = 0$, the model bases its judgement only on multiplication.

We also experimented with combinations such as $\alpha(V_{w_1^r}, V_{w_2})$ and $\alpha(V_{w_1}, V_{w_2^r})$ i.e. using refined vector for one of the constituent word and the unrefined prototype vector for the other constituent word.

4.4 Selecting the best model

To participate in the shared task, we have selected the best performing model by evaluating the models on the validation data released in the shared task (Biemann and Giesbrecht, 2011). Table 1 displays the results on the validation data. The average point difference is calculated by taking the average of the difference in a model's score α and the gold score annotated by humans, over all compounds. Table 1 also displays the overall accuracy of coarse grained labels – low, medium and high.

Best performance for verb(v)-object(o) compounds is found for the combination $\alpha(V_{v^r}, V_{o^r})$ of Equation 3. For subject(s)-verb(v) compounds, it is for $\alpha(V_{s^r}, V_{v^r})$ and $a_3 = a_4 = 0$. For adjective(j)-noun(n) compounds, it is $\alpha(V_{j^r}, V_n)$. We are not certain of the reason for this difference, perhaps there may be less ambiguity of words within specific grammatical relationships or it may be simply due to

	TotPrd	Spearman ρ	Kendalls τ
Rand-Base	174	0.02	0.02
Exm-Best	169	**0.35**	**0.24**
Pro-Best	169	0.33	0.23
Exm	169	0.26	0.18
SharedTaskNextBest	174	0.33	0.23

Table 2: Correlation Scores

	All	ADJ-NN	V-SUBJ	V-OBJ
Rand-Base	32.82	34.57	29.83	32.34
Zero-Base	23.42	24.67	17.03	25.47
Exm-Best	**16.51**	15.19	15.72	18.6
Pro-Best	16.79	14.62	18.89	18.31
Exm	17.28	15.82	18.18	18.6
SharedTaskBest	**16.19**	14.93	21.64	14.66

Table 3: Average Point Difference Scores

	All	ADJ-NN	V-SUBJ	V-OBJ
Rand-Base	0.297	0.288	0.308	0.30
Zero-Base	0.356	0.288	0.654	0.25
Most-Freq-Base	0.593	0.673	0.346	0.65
Exm-Best	**0.576**	0.692	0.5	0.475
Pro-Best	0.567	0.731	0.346	0.5
Exm	0.542	0.692	0.346	0.475
SharedTaskBest	**0.585**	0.654	0.385	0.625

Table 4: Coarse Grained Accuracy

the actual compounds in those categories. We leave analysis of this for future work. We combined the outputs of these category-specific models to build the best model *Exm-Best*.

For comparison, results of standard models prototype addition *(Pro-Addn)* and prototype-multiplication *(Pro-Mult)* are also displayed in Table 1. *Pro-Addn* can be represented as $\alpha(V_{w_1}, V_{w_2})$ with $a_1 = a_2 = a_4 = 0$. *Pro-Mult* can be represented as $\alpha(V_{w_1}, V_{w_2})$ with $a_1 = a_2 = a_3 = 0$. *Pro-Best* is the best performing model in prototype-based modelling. It is found to be $\alpha(V_{w_1}, V_{w_2})$. (Note: Depending upon the compound type, some of the a_i's in *Pro-Best* may be 0).

Overall, exemplar-based modelling excelled in both the evaluations, average point difference and coarse-grained label accuracies. The systems *Exm-Best*, *Pro-Best* and *Exm* $\alpha(V_{w_1^r}, V_{w_2^r})$ were submitted for the public evaluation in the shared task. All the model parameters were estimated by regression on the task's training data separately for the 3 compound types as described in Section 4.3. We perform the regression separately for these classes to maximise performance. In the future, we will investigate whether these settings gave us better results on the test data compared to setting the values the same regardless of the category of compound.

5 Shared Task Results

Table 2 displays Spearman ρ and Kendalls τ correlation scores of all the models. TotPrd stands for the total number of predictions. Rand-Base is the baseline system which randomly assigns a compositionality score for a compound. Our model Exm-Best was the best performing system compared to all other systems in this evaluation criteria. Shared-TaskNextBest is the next best performing system apart from our models. Due to lemmatization errors in the test data, our models could only predict judgements for 169 out of 174 compounds.

Table 3 displays average point difference scores. Zero-Base is a baseline system which assigns a score of 50 to all compounds. SharedTaskBest is the overall best performing system. Exm-Best was ranked second best among all the systems. For ADJ-NN and V-SUBJ compounds, the best performing systems in the shared task are Pro-Best and Exm-Best respectively. Our models did less well on V-OBJ compounds and we will explore the reasons for this in future work.

Table 4 displays coarse grained scores. As above, similar behaviour is observed for coarse grained accuracies. Most-Freq-Base is the baseline system which assigns the most frequent coarse-grained label for a compound based on its type (ADJ-NN, V-SUBJ, V-OBJ) as observed in training data. Most-Freq-Base outperforms all other systems.

6 Conclusions

In this paper, we examined the effect of polysemy in word space models for compositionality detection. We showed exemplar-based WSM is effective in dealing with polysemy. Also, we use multiple evidences for compositionality detection rather than basing our judgement on a single evidence. Overall, performance of the Exemplar-based models of compositionality detection is found to be superior to prototype-based models.

References

Baldwin, T., Bannard, C., Tanaka, T., and Widdows, D. (2003). An empirical model of multiword expression decomposability. In *Proceedings of the ACL 2003 workshop on Multiword expressions: analysis, acquisition and treatment - Volume 18*, MWE '03, pages 89–96, Stroudsburg, PA, USA. Association for Computational Linguistics.

Bannard, C., Baldwin, T., and Lascarides, A. (2003). A statistical approach to the semantics of verb-particles. In *Proceedings of the ACL 2003 workshop on Multiword expressions: analysis, acquisition and treatment - Volume 18*, MWE '03, pages 65–72, Stroudsburg, PA, USA. Association for Computational Linguistics.

Biemann, C. and Giesbrecht, E. (2011). Distributional semantics and compositionality 2011: Shared task description and results. In *Proceedings of DISCo-2011 in conjunction with ACL 2011*.

Curran, J. R. (2003). From distributional to semantic similarity. Technical report, PhD Thesis, University of Edinburgh.

Erk, K. and Padó, S. (2010). Exemplar-based models for word meaning in context. In *Proceedings of the ACL 2010 Conference Short Papers*, ACLShort '10, pages 92–97, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ferraresi, A., Zanchetta, E., Baroni, M., and Bernardini, S. (2008). Introducing and evaluating ukwac, a very large web-derived corpus of english. In *Proceedings of the WAC4 Workshop at LREC 2008*, Marrakesh, Morocco.

Giesbrecht, E. (2009). In search of semantic compositionality in vector spaces. In *Proceedings of the 17th International Conference on Conceptual Structures: Conceptual Structures: Leveraging Semantic Technologies*, ICCS '09, pages 173–184, Berlin, Heidelberg. Springer-Verlag.

Guevara, E. R. (2011). Computing semantic compositionality in distributional semantics. In *Proceedings of the Ninth International Conference on Computational Semantics*, IWCS '2011.

Harris, Z. S. (1954). Distributional structure. *Word*, 10:146–162.

Katz, G. and Giesbrecht, E. (2006). Automatic identification of non-compositional multi-word expressions using latent semantic analysis. In *Proceedings of the Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties*, MWE '06, pages 12–19, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kilgarriff, A., Rychly, P., Smrz, P., and Tugwell, D. (2004). The sketch engine. In *Proceedings of EURALEX*.

Lexical Computing Ltd. (2007). Statistics used in the sketch engine.

McCarthy, D., Keller, B., and Carroll, J. (2003). Detecting a continuum of compositionality in phrasal verbs. In *Proceedings of the ACL 2003 workshop on Multiword expressions: analysis, acquisition and treatment - Volume 18*, MWE '03, pages 73–80, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mitchell, J. and Lapata, M. (2008). Vector-based Models of Semantic Composition. In *Proceedings of ACL-08: HLT*, pages 236–244, Columbus, Ohio. Association for Computational Linguistics.

Murphy, G. L. (2002). *The Big Book of Concepts*. The MIT Press.

Partee, B. (1995). Lexical semantics and compositionality. *L. Gleitman and M. Liberman (eds.) Language, which is Volume 1 of D. Osherson (ed.) An Invitation to Cognitive Science (2nd Edition)*, pages 311–360.

Reisinger, J. and Mooney, R. J. (2010). Multi-prototype vector-space models of word meaning. In *Proceedings of the 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-2010)*, pages 109–117.

Rychlý, P. and Kilgarriff, A. (2007). An efficient algorithm for building a distributional thesaurus (and other sketch engine developments). In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 41–44, Stroudsburg, PA, USA. Association for Computational Linguistics.

Sahlgren, M. (2006). *The Word-Space Model: Using distributional analysis to represent syntag-*

matic and paradigmatic relations between words in high-dimensional vector spaces. PhD thesis, Stockholm University.

Schone, P. and Jurafsky, D. (2001). Is knowledge-free induction of multiword unit dictionary headwords a solved problem? In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '01.

Smith, E. E. and Medin, D. L. (1981). *Categories and concepts / Edward E. Smith and Douglas L. Medin*. Harvard University Press, Cambridge, Mass. :.

Turney, P. D. and Pantel, P. (2010). From frequency to meaning: vector space models of semantics. *J. Artif. Int. Res.*, 37:141–188.