Measuring the compositionality of collocations via word co-occurrence vectors: Shared task system description

Alfredo Maldonado-Guerra and **Martin Emms**
School of Computer Science and Statistics
Trinity College Dublin
Ireland
{maldonaa, mtemms}@scss.tcd.ie

Abstract

A description of a system for measuring the compositionality of collocations within the framework of the shared task of the Distributional Semantics and Compositionality workshop (DISCo 2011) is presented. The system exploits the intuition that a highly compositional collocation would tend to have a considerable semantic overlap with its constituents (headword and modifier) whereas a collocation with low compositionality would share little semantic content with its constituents. This intuition is operationalised via three configurations that exploit cosine similarity measures to detect the semantic overlap between the collocation and its constituents. The system performs competitively in the task.

1 Introduction

Collocations or multiword expressions vary in the degree to which a native speaker is able to understand them based on the interaction of their constituents' individual meanings. The concept of compositionality of a collocation captures this notion. The shared task of the DISCo 2011 workshop (Biemann and Giesbrecht, 2011) consists in comparing systems' compositionality scores against compositionality scores based on human judgements. Systems were evaluated on the match of the compositional scores generated by the system and those based on human judgements – specifically taking the mean of the absolute difference of these scores. Additionally the organisers also classified the human-derived scores into three coarse categories of compositionality: non-compositional (*low*), somewhat compositional (*medium*) and compositional (*high*). Systems were required to produce an additional compositionality labelling into these three coarse categories and were evaluated on the precision of this labelling.

The methods used by our system for measuring compositionality take inspiration from the work of McCarthy et al. (2003), who measured the similarity between a phrasal verb (a main verb and a preposition like *blow up*) and its main verb (*blow*) by comparing the words that are closely semantically related to each, and use this similarity as an indicator of compositionality. Our method for measuring compositionality is considerably different as it instead directly compares the semantic similarity between the headword and the collocation and between the modifier and the collocation by computing a cosine similarity score between word co-occurrence vectors that represent the headword, the modifier and the collocation (see 3.2). Our system can be regarded as fully unsupervised as it does not employ any parsers in its processing or any external data other than the corpus and the collocation lists provided by the organisers.

The rest of the paper is organised as follows: Section 2 describes the corpora and the collocation list provided by the task organisers. Section 3 introduces some definitions and describes the three configurations in detail. Section 4 presents the results and concludes.

2 Data

Shared task participants were provided with a list of collocations of three grammatical forms: adjective-

Proceedings of the Workshop on Distributional Semantics and Compositionality (DiSCo'2011), pages 48–53,

Portland, Oregon, 24 June 2011.

noun collocations (**A-N**), subject-verb collocations (**S-V**) and verb-object collocations (**V-O**). Our system assumes that each collocation consists of a headword and a modifier and it interprets these constituents in each grammatical form as follows: **A-N**: adjective - modifier, noun - headword; **S-V**: subject - modifier, verb - headword; **V-O**: verb - headword, object - modifier.

As a corpus, our system uses a random sample of 500,000 documents from the plain-text, non-parsed version of the English ukWaC corpus (Baroni et al., 2009).

3 System description

Our system can be employed in three different configurations. All three rely in representing words and collocations as word co-occurrence vectors and measure semantic similarity using the cosine measure.

3.1 Preliminary definitions

These definitions are largely based on the construction of first-order context vectors, word co-occurrence vectors and second-order context vectors via global selection as described in Schütze (1998) and in Purandare and Pedersen (2004) by considering context windows of 20 words centred at a target word.

The **first-order context vector** is a vector representing a *token* of a word, or equivalently a *position* p in a document. Dimensions of the vector are word types w, and the value on dimension w *is a count of the frequency with which* w *occurs in a specified window around* p *in a given document doc*.

$$\mathbf{C^1}(p)(w) = \sum_{\substack{p' \neq p \\ p-10 \leq p' \\ p' \leq p+10}} (1 \text{ if } w = \text{doc}(p'), \text{ else } 0) \quad (1)$$

In this work the dimensions are the 2,000 non-function words that are most frequent in the corpus[1]. The **word co-occurrence vector** (or simply **word vector**) is a vector recording the co-occurrence behaviour of a particular word *type* w in a corpus. As such it can be defined by summation over first-order context vectors:

$$\mathbf{W}(w) = \sum_{p} (1 \text{ if } w = \text{doc}(p), \text{ else } 0) \cdot \mathbf{C^1}(p) \quad (2)$$

And the **second-order context vector** is a further vector representing an instance of a word. For a particular location p, it is defined to be *sum of the word vectors of words in a given window around* p

$$\mathbf{C^2}(p) = \sum_{\substack{p' \neq p \\ p-10 \leq p' \\ p' \leq p+10}} \mathbf{W}(\text{doc}(p)) \quad (3)$$

Although the above are defined for types and tokens of *words*, they can be generalised to *multiword* expressions in various ways. In this work, for any multiword expression *type* x y, its *tokens* are taken to be occurrences of the sequence $x\gamma y$, where γ can be any sequence of intervening words of length l, $0 \leq l \leq 3$. By taking the position of x as the position of the multiword token, and taking the first position after the token as position $p+1$, the definitions of $\mathbf{C^1}$, $\mathbf{W}$ and $\mathbf{C^2}$ can be carried over to multiword expressions.

All the configurations described below use the cosine measure between vectors, defined in the standard way

$$\cos(\mathbf{v}, \mathbf{w}) = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2 \sum_{i=1}^{N} w_i^2}} \quad (4)$$

3.2 System configurations

For each collocation in the test set, the **first configuration** of our system starts off by building word vectors for the collocation, its headword and its modifier.

The first configuration of the system outputs the average of two cosine similarity measures as the compositionality score for the collocation:

$$c_1 = \frac{1}{2} \begin{bmatrix} \cos(\mathbf{W}(xy), \mathbf{W}(x)) \\ +\cos(\mathbf{W}(xy), \mathbf{W}(y)) \end{bmatrix} \quad (5)$$

where $\mathbf{W}(xy)$ is the word vector representing the collocation whose constituents are x and y, and $\mathbf{W}(x)$ and $\mathbf{W}(y)$ are the word vectors representing each constituent x and y, respectively.

[1] We employ a modified version of the stop word list supplied with Ted Pedersen's Text-NSP package (http://www.d.umn.edu/~tpederse/nsp.html)

The **second configuration** of our system considers the occurrences of the headword when accompanied by the modifier forming the collocation separately from occurrences of the headword appearing on its own and compares them. If y is the headword of a collocation and $\text{coll}(p)$ is a Boolean function that determines whether the word at position p forms a collocation with x, let

$$\mathbf{W}^{x}(y) = \sum_{p} (1 \text{ if } \begin{matrix}\text{doc}(p) = y \\ \text{coll}(p,x)\end{matrix}, \text{ else } 0) \cdot \mathbf{C^1}(p) \quad (6)$$

be the word vector computed from all the occurrences of the headword y that form a collocation with x and conversely, let

$$\mathbf{W}^{\bar{x}}(y) = \sum_{p} (1 \text{ if } \begin{matrix}\text{doc}(p) = y \\ \neg\text{coll}(p,x)\end{matrix}, \text{ else } 0) \cdot \mathbf{C^1}(p) \quad (7)$$

be the word vector representing the occurrences of y not engaging in a collocation with x. In this configuration, the compositionality score is then computed by

$$c_2 = \cos\left(\mathbf{W}^{x}(y), \mathbf{W}^{\bar{x}}(y)\right) \quad (8)$$

The intuition behind this configuration is that if the headword tends to co-occur with more or less the same words in both cases (producing a high cosine score), then the meaning of the headword is similar regardless of whether the collocation's modifier is present or not, implying a high degree of compositionality. If on the other hand, the headword co-occurs with somewhat differing words in the two cases (a low cosine score), then we assume that the presence of the collocation's modifier is markedly changing the meaning of the headword, implying a low degree of compositionality.

In its **third configuration**, our system employs clustering techniques in order to exploit semantic differences that may naturally emerge from each context in which the collocation and its constituents are used. Different senses of a collocation might have different compositionality measures as can be seen in these two example sentences employing the collocation *great deal*:

1. Two cans of soup for the price of one is such a *great deal*!

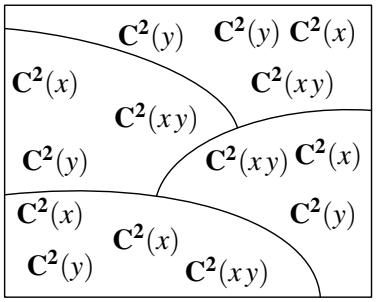

Figure 1: *Example of a clustered second-order context vector space.*

2. The tsunami caused a *great deal* of damage to the country's infrastructure.

In Word Sense Induction, clustering is used to group occurrences of a target word according to its sense or usage in context (see e.g. Pedersen (2010)) as it is expected that each cluster will represent a different sense or usage of the target word. However, since the contexts that human annotators referred to when judging the compositionality of the collocations was not provided, our system employs a workaround that uses a weighted average when measuring compositionality. This workaround is explained in what follows.

In this configuration, the system first builds word vectors for the 20,000 most frequent words in the corpus (equation 2), and then uses these to compute the second-order context vectors for each occurrence of the collocation and its constituents in the corpus (equation 3). After context vectors for all occurrences have been computed, they are clustered using CLUTO's repeated bisections algorithm[2]. The vectors are clustered across a small number K of clusters (we employed $K = 4$). We expect that each cluster will represent a different contextual usage of the collocation, its headword and its modifier. Figure 1 depicts how a context vector space could be partitioned with $K = 4$.

The system then for each cluster k builds the word vectors (equation 2) $\mathbf{W}_k(x\,y)$, $\mathbf{W}_k(x)$, and $\mathbf{W}_k(y)$ for the collocation, its headword and its modifier, from the contexts grouped within the cluster k. The compositionality measure for the third configuration is then basically a weighted average over the clusters

[2]http://glaros.dtc.umn.edu/gkhome/views/cluto/

of the c_1 score using each cluster, that is:

$$c_3 = \sum_{k=1}^{K} \frac{\|k\|}{N} \frac{1}{2} \begin{bmatrix} \cos(\mathbf{W}_k(x\,y), \mathbf{W}_k(x)) \\ +\cos(\mathbf{W}_k(x\,y), \mathbf{W}_k(y)) \end{bmatrix} \quad (9)$$

where $\|k\|$ is the number of contexts in cluster k and N is the total number of contexts across all clusters.

For all three configurations, the value reported as the numeric compositionality score was the corresponding value obtained from equations (5), (8) or (9), multiplied by 100. Each configuration's numeric scores c_i were binned into the three coarse compositionality classes by comparing them with the configuration's maximum value through equation (10).

$$\text{coarse}(c_i) = \begin{cases} high & \text{if } \frac{2}{3}max \leq c_i \\ medium & \text{if } \frac{1}{3}max < c_i < \frac{2}{3}max \\ low & \text{if } c_i \leq \frac{1}{3}max \end{cases} \quad (10)$$

4 Results and conclusion

Table 1 shows the evaluation results for the three system configurations and two baselines. The left-hand side of the table shows the average difference between the gold-standard numeric score and each configuration's numeric score. The right-hand side reports the precision on binning the numeric scores into the coarse classes. Evaluation scores are reported on all collocations and on the collocation subtypes separately. Row **R** is the baseline suggested by the workshop organisers, assigning random numeric scores, in turn binned into the coarse categories. Row **A** shows the performance of a constant output baseline, assigning all collocations the *mean* gold-standard numeric score from the training set: 66.45, and then applying the binning strategy of equation (10) to this – which always assigns the coarse category *high*.

The first thing to note from this table is that configurations 1 and 2 generally outperform configuration 3, both on the mean difference and coarse scores. Configuration 1 slightly outperforms configuration 2 on the mean numeric difference scores, whilst configuration 2 is very close to and slightly

C	Average differences (numeric)				Precision (coarse)			
	ALL	A-N	S-V	V-O	ALL	A-N	S-V	V-O
1	**17.95**	**18.56**	20.80	**15.58**	53.4	**63.5**	19.2	62.5
2	18.35	19.62	**20.20**	15.73	**54.2**	**63.5**	19.2	**65.0**
3	25.59	24.16	32.04	23.73	44.9	40.4	**42.3**	52.5
R	32.82	34.57	29.83	32.34	29.7	28.8	30.0	30.8
A	16.86	17.73	15.54	16.52	58.5	65.4	34.6	65.0

Table 1: *Evaluation results of the three system configurations and two baselines on the test dataset. Best system scores on each grammatical subtype highlighted in bold.*

better than configuration 1 on the coarse precision scores. The exception is that configuration 3 was the best performer on the coarse precision scoring for the **S-V** subtype.

The R baseline is outperformed by configurations 1, 2 and 3; roughly speaking where 1 and 2 outperform R by d, configuration 3 outperforms R by around $d/2$. The A baseline generally outperforms all our system configurations. It seems to be also a quite competitive baseline for other systems participating in the shared task.

The other trend apparent from the table is that performance on the **V-O** and **A-N** subtypes tends to exceed that on the the **S-V** subtype.

An examination of the gold standard test files shows that the distribution over the *low*/*medium*/*high* categories is similar for both **V-O** and **A-N**, in both cases close to 0.08/0.27/0.65, with *high* covering nearly two-thirds of cases, whilst for **S-V** the distribution is quite different: 0.0/0.654/0.346, with *medium* covering nearly two-thirds of cases. This is reflected in the A baseline precision scores, as for each subtype these will necessarily be the proportion of gold-standard *high* cases. This explains for example why the A baseline is much poorer on the **S-V** cases (34.6) than on the other cases (65.0, 65.4).

Looking further into the differences between the three subtypes, Figure 2 shows the gold standard numeric score distribution across the three collocation subtypes (**Test GS**), and the corresponding distributions for scores from the system's first configuration (**Conf 1**). This shows in more detail the nature of the poorer performance on **S-V**, with the gold standard having a peak around 50-60, and the system having a peak around 70-80. For the other subtypes

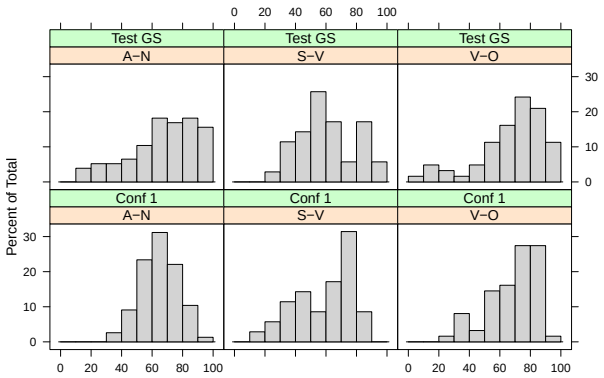

Figure 2: *The distribution of the gold standard numeric score vs. the distribution of the system's first configuration numeric scores.*

	A-N	S-V	V-O
Instances	177254	11092	121317
Avg intervening	0.0684	0.3867	0.4612

Table 2: *Some corpus statistics: the number of matched collocations per subtype (**Instances**) and the average number of intervening words per subtype (**Avg intervening**).*

the contrast in the distributions seems broadly consistent with the mean numeric difference scores of Table 1.

One can speculate on the reasons for the system's poorer performance on the **S-V** subtype. The system treats intervening words in a collocation in a particular way, namely by ignoring them. This is one option, and another would be to include them as features counted in the vectors. Table 2 shows the average intervening words in the occurrences of the collocations. **S-V** and **V-O** are alike in this respect, both being much more likely to present intervening words than collocations of the **A-N** subtype. So the explanation of the poorer performance on **S-V** cannot lie there. Also because the average number of intervening words is low, we believe it is unlikely that including them as features will impact performance significantly.

Table 2 also gives the number of matched collocations per subtype. The number for the **S-V** collocations is an order of magnitude smaller than for the other subtypes. Although the collocations supplied by the organisers are in their base form, the system attempts to match them 'as is' in the unlemmatised version of the corpus. Whilst for **A-N** and **V-O** the base-form sequences relatively frequently do double service as inflected forms, this is far less frequently the case for the **S-V** sequences (e.g. *user see* (**S-V**) is far less common than *make money* (**V-O**)). This much smaller number of occurrences for **S-V** cases, or the fact that they are drawn from syntactically special contexts, may be a factor in the relatively poorer performance. This perhaps is also a factor in the earlier noted fact that although configuration 3 was generally outperformed, on the **S-V** subtype the reverse occurs.

The unlemmatised version of the corpus was used because initial experimentation with the validation set produced slightly better results when employing raw words as features rather than lemmas. A possibility for future work would be to to refer to lemmas for matching collocations in the corpus, but to continue to use unlemmatised words as features.

Other areas for future investigation involve the effects of weighting schemes (such as IDF) and the use of similarity measures other than cosine, as well as alternatives in configurations 2 and 3. For example, configuration 2 could involve the modifier in the computation of the compositionality score, and configuration 3 could create separate clustering spaces for collocation, headword and modifier and compute similarity scores based on vectors representing these clusters.

In sum, the simplest configuration of a totally unsupervised system yielded surprisingly good results at measuring compositionality of collocations in raw corpora, and whereas there is scope for further development and refinement, the system as it is constitutes a robust baseline to compare against more elaborate systems.

5 Acknowledgements

We would like to thank our anonymous reviewers for their insightful comments and ideas. This research is supported by the Science Foundation Ireland (Grant 07/CE/I1142) as part of the Centre for Next Generation Localisation (www.cngl.ie) at Trinity College Dublin.

References

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The WaCky wide web: a collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 43(3):209–226, February.

Chris Biemann and Eugenie Giesbrecht. 2011. Distributional Semantics and Compositionality 2011: Shared Task Description and Results. In *Proceedings of the Distributional Semantics and Compositionality workshop (DISCo 2011) in conjunction with ACL 2011*, Portland, Oregon.

Diana McCarthy, Bill Keller, and John Carroll. 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*, pages 73–80, Sapporo. Association for Computational Linguistics.

Ted Pedersen. 2010. Duluth-WSI: SenseClusters applied to the sense induction task of SemEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, number July, pages 363–366, Uppsala, Sweden. Association for Computational Linguistics.

Amruta Purandare and Ted Pedersen. 2004. Word sense discrimination by clustering contexts in vector and similarity spaces. *Proceedings of the Conference on Computational Natural Language Learning*, pages 41–48.

Hinrich Schütze. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123.