Unsupervised Lexical Substitution with a Word Space Model

Dario Pucci[1], Marco Baroni[2], Franco Cutugno[1], and Alessandro Lenci[3]

[1]University of Naples, Department of Physics - NLP Group
[2]University of Trento, CIMeC
[3]University of Pisa, Department of Linguistics
d.pucci@studenti.unina.it
marco.baroni@unitn.it
cutugno@na.infn.it
alessandro.lenci@ling.unipi.it

Abstract. We describe a system to tackle the Lexical Substitution task that exploits, as its only resource, co-occurrence statistics from a large PoS-tagged corpus. The system exploits the *word space model* formalism, and represents the word to be substituted by a composite vector that takes into account both the overall distribution of the word in the input corpus and its local context. As far as the precision and recall are concerned, the system is ranked among the highest positions in the Evalita competition, while it results winner in the mode p and mode r ranking.

Key words: word space models, composition in word space models, corpus-based semantics

1 Introduction

Word space semantic models (WSMs) represent the meaning of a word as a distributional vector recording its co-occurrence with various types of linguistic contexts (other words, syntactic constructions, lexicalized patterns, etc.). According to the so-called *Distributional Hypothesis* [3], the semantic similarity between two words can be modeled as a function of their distributional similarity, with the latter computed by measuring the distance in vector space between the word vectors (e.g., their cosine or euclidean distance). WSMs achieve very good results in many semantic tasks, often outperforming semantic approaches based on lexical resources such as WordNet [7]. In particular, WSMs have achieved very good results in the TOEFL synonym detection task, in which the system must choose the correct synonym of a target word out of four alternatives. The task was first introduced in [2] as a way of evaluating algorithms for measuring the degree of similarity between words. Nowadays WSMs almost match human performance on this task [5]. WSMs are particularly attractive because of their unsupervised and data-driven nature, besides the fact that they have been claimed to have a high cognitive plausibility as models for semantic representation and acquisition [2].

A notorious limit of WSMs is that distributional vectors provide a semantic representation for word types without distinguishing their different senses in specific contexts. This means that the various senses of an ambiguous or polysemous word are squeezed on the same vector. For instance, the vector associated with *ball* will contain distributional information related both to its sense of round concrete object and to the dancing event sense. Various models have been proposed to carve word senses out of distributional vectors and to capture meaning shifts in context. [8] introduces an algorithm for Word Sense Disambiguation with WSMs in which word senses are characterized by second-order context vectors. More recently, [4] have proposed a two-layer model for context-sensitive semantic representation in a WSM. Word types are first associated with *out-of-context vectors*, as in standard vector space semantics. Then, their representation in context (e.g., the meaning of *ball* in the context of the verb *dance*) is obtained by combining (through vector summing or component-wise vector multiplication) the out-of-context vectors of the co-occurring words.

In this paper, we propose a WSM for unsupervised lexical substitution which drives inspiration from [4]. The algorithm includes three basic steps, which will be described in detail in the following section:

1. we first build a WSM that assigns distributional *out-of-context* (ooc) vectors to word types;
2. for each word instance w appearing in context c, we build its *contextualized vector* w_c by combining the ooc vector of w with the ooc vectors of the words appearing in a window of k words to the left and to the right of w in context c;
3. we measure the distance between the context vector w_c and the ooc vectors of all the other words in the vector space that have the same PoS as w. We choose the lexical substitutes for w in context c among the top n nearest neighbors of the context vector of w_c.

2 Method

Constructing ooc vectors

We collected statistics from a dataset obtained by concatenating the *la Repubblica* corpus (`http://sslmit.unibo.it/repubblica`), the *itWaC* corpus (`http://wacky.sslmit.unibo.it/`) and a snapshot of the Italian Wikipedia (`http://it.wikipedia.org/`). The concatenated corpus was tokenized, Part-of-Speech-tagged and lemmatized using the TreeTagger (`www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger`). After processing, it contains about 2 billion tokens.

We extracted the top 20,000 most frequent content words (nouns, verbs, adjectives and adverbs) from the corpus, and we built ooc vectors for each of them by counting their co-occurrences within the same sentence. Raw counts were transformed into log-likelihood ratio scores [1]. To make matrix manipulation easier, we compressed the resulting 20,000x20,000 sparse matrix into a

20,000x10,000 approximation, using the standard Random Indexing compression method [6].

Contextualizing vectors

In the EVALITA task, we are given a target word w in a sentential context c, and we have to produce a ranked list of potential synonyms of the word in context. To this extent, we PoS-tag and lemmatize the sentence, we remove stop words and the words not included among the 20,000 selected content words (cf. above). We then compute a composite contextualized vector w_c for the target w by summing its ooc vector with the (normalized) ooc vectors of its contextual collocates. In particular, we select $\frac{k}{2}$ words on the right and $\frac{k}{2}$ words on the left of the target w. Before summing, we down-weight the impact of collocates on the composite vector in function of their distance from w. In particular, the collocate ooc vectors are multiplied by $\frac{1}{sv * d}$, where $d \in [1, \frac{k}{2}]$ represents the distance of the considered word from the target one and sv is a static value empirically estimated on the base of the results we will present in the next section. This value is used to reduce the weight of more distant words within the context window. For instance, If we consider the following sentence with *fermare* as the target:

Walter Cerfeda, segretario confederale della Cgil, ha proposto di **fermare** *una trattativa insensata per sollecitare un rapido chiarimento tra Alitalia e Governo.*

the context window with k=4 for the target word will be [$confederale\ proporre\ trattativa\ insensato$]. The context window does not include *Cgil* because it does not belong to the set of selected words. The words therein contained will have a distance d as showed in the following example:

Word	Distance
confederale	2
proporre	1
trattativa	1
insensato	2

Finding lexical substitutes

We measure the cosine with the contextualized vector w_c of all ooc vectors representing words with the same PoS as w. Our candidate lexical substitutes are simply the words whose ooc vectors have the highest cosine with w_c.

3 Evaluation and Parameters Setting

The measures used for the evaluation in the Lexical substitution task in EVALITA are:

- **Recall** (percentage of right answers on the total of instances in the test-set/trial-set): it is the basic accuracy measurement for this type of tasks, as it shows how many correct disambiguations the system achieved;
- **Precision** (percentage of right answers in the set instance for which an answer has been given): it favors systems that are very accurate only on a small number of subsets of answers;
- **Mode precision** and **Mode recall** calculate precision and recall, respectively, against the synonym chosen by the majority of annotators (if there is a majority).

for the types of scoring:

- **Best**. Scores the best guessed synonym.
- **Out-of-ten (oot)**. Scores the best 10 guessed synonyms.

We conducted various experiments on the trial-set to tune the following parameters: the number of synonyms to evaluate for the best scoring, the size of the window of context k and the weight associated to the words therein contained. For both the best and the our-of-ten scores, the top results were obtained by taking a size for the context windows k of 4 words and a weight equal to $\frac{1}{4 * d}$, where d is the distance between the current word and the target one (cf. Table 1 and 2). Moreover, we obtained an optimal accuracy by simply providing the most probable synonym in each instance.

Tables 3 and 4 reports in boldface the results achieved by our model on the test set, with the parameters setting described above. The system is ranked in second position in the Evalita competition, while it results winner in the mode p and mode r ranking. It is worth remarking that we did not include multiword expressions among our targets. This means that all the multiword synonyms in the test answers were systematically missed by our system.

Table 1. Best score results on the trial set, with different parameter settings

Results						
Setting			**Best Score**			
Window k/2	**sv**	**Synonyms**	**Precision**	**Recall**	**Mode Precision**	**Mode Recall**
1	3	1	3.28	3.14	5.22	5.13
2	3	1	3.28	3.14	5.22	5.13
3	3	1	2.93	2.81	4.35	4.27
1	4	1	3.28	3.14	5.22	5.13
2	4	1	**3.28**	**3.14**	**5.22**	**5.13**
3	4	1	3.17	3.03	5.22	5.13

Table 2. Out-of-ten score results on the trial set, with different parameter settings

Results					
Setting		**Out-of-ten (oot) Score**			
Window k/2	**sv**	**Precision**	**Recall**	**Mode Precision**	**Mode Recall**
0	1	9.95	9.53	14.78	14.53
1	2	9.74	9.34	15.65	15.38
2	2	9.92	9.50	16.52	16.24
3	2	9.63	9.23	16.52	16.24
4	2	9.46	9.07	15.65	15.38
1	3	9.82	9.41	15.65	15.38
2	3	11.39	10.92	17.39	17.09
3	3	11.39	10.92	17.39	17.09
4	3	10.43	10.00	15.65	15.38
1	4	9.82	9.41	15.65	15.38
2	4	**11.39**	**10.92**	**17.39**	**17.09**
3	4	10.87	10.42	16.52	16.24

Table 3. Best score results on the test set. The scores achieved by our model are in boldface

Results Best Score			
Precision	**Recall**	**Mode Precision**	**Mode Recall**
8.16	7.18	10.58	10.58
6.26	**6.01**	**11.28**	**10.84**
6.8	5.53	8.9	8.9
6.28	5.46	8.13	8.13
3.95	3.21	6.58	6.58
3.9	3.17	6.71	6.71
3.16	3.16	6.97	6.97
3.52	2.8	5.03	5.03

Table 4. Out-of-ten score results on the test set. The scores achieved by our model are in boldface

Results Out-of-ten (oot) Score			
Precision	**Recall**	**Mode Precision**	**Mode Recall**
41.46	36.5	47.23	47.23
37.74	30.69	34.84	34.84
28.54	24.79	34.58	34.58
20.09	20.09	27.74	27.74
23.48	19.11	26.58	26.58
23	18.72	26.32	26.32
16.65	**16**	**24.97**	**24**
18.62	14.78	20.52	20.52

4 Conclusion

The overall encouraging results obtained by our approach to the EVALITA Lexical Substitution Task must be interpreted in light of the fact that we only used as input a PoS-tagged corpus, and relied on a fully unsupervised algorithm. This is even more significant once we take into consideration i.) the inherent difficulty of the Lexical Substitution Task, and ii.) the specific way in which it was implemented in EVALITA, with very general and highly polysemous test words, whose senses were often linked to specific collocational constructions. While there is of course much room for improvement, and we plan in particular to explore different ways to define context and to construct composite vectors (including component-wise multiplication instead of summing), the current results suggest that WSM-based approaches that only require a corpus as input and do not rely on supervision can tackle even an advanced, open-ended, context-dependent semantic task such as lexical substitution.

References

1. Dunning, T.: Accurate methods for the statistics of surprise and coincidence. Computational Linguistics, vol. 19, issue 1, pp. 61–74 (1993)
2. Landauer, T.K., Dumais, S.T.: A solution to Plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. Psychological Review, vol. 104, issue 2, pp. 211–240 (1997)
3. Miller, G. A., Charles, W. G.: Contextual Correlates of Semantic Similarity. Language and Cognitive Processes, vol. 6, pp. 1–28 (1991)
4. Mitchell J., Lapata, M.: Vector-based Models of Semantic Composition. In: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 236–244. Columbus, Ohio (2008)
5. Rapp, R.: A freely available automatically generated thesaurus of related words. In: Proceedings of LREC 2004 , pp. 395–398 (2004)
6. Sahlgren, M.: An Introduction to Random Indexing. In: Proceedings of the Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering. Copenhagen, Denmark (2005)
7. Sahlgren, M.: The Word-Space Model. PhD Dissertation, Stockholm University (2006)
8. Schütze, H.: Automatic word sense discrimination. Computational Linguistics, vol. 24, issue 1, pp. 97–123 (1998)