10.12871/CLICIT201419

Combining Distributional Semantic Models and Sense Distribution for Effective Italian Word Sense Disambiguation

Pierpaolo Basile and **Annalina Caputo** and **Giovanni Semeraro**
Department of Computer Science
University of Bari Aldo Moro
Via, E. Orabona, 4 - 70125 Bari (Italy)
{firstname.surname}@uniba.it

Abstract

English. Distributional semantics approaches have proven their ability to enhance the performance of overlap-based Word Sense Disambiguation algorithms. This paper shows the application of such a technique to the Italian language, by analysing the usage of two different Distributional Semantic Models built upon ItWaC and Wikipedia corpora, in conjunction with two different functions for leveraging the sense distributions. Results of the experimental evaluation show that the proposed method outperforms both the most frequent sense baseline and other state-of-the-art systems.

Italiano. *Gli approcci di semantica distribuzionale hanno dimostrato la loro capacità nel migliorare le prestazioni degli algoritmi di Word Sense Disambiguation basati sulla sovrapposizione di parole. Questo lavoro descrive l'applicazione di questa tipologia di tecniche alla lingua italiana, analizzando l'utilizzo di due diversi Modelli di Semantica Distribuzionale costruiti sui corpora ItWaC e Wikipedia, in combinazione con due diverse funzioni che sfruttano le distribuzioni dei significati. I risultati della valutazione sperimentale mostrano la capacità di questo metodo di superare le prestazioni sia della baseline rappresentata dal senso più comune che di altri sistemi a stato dell'arte.*

1 Introduction

Given two words to disambiguate, Lesk (1986) algorithm selects those senses which maximise the overlap between their definitions (i.e. glosses), then resulting in a pairwise comparison between all the involved glosses. Since its original formulation, several variations of this algorithm have been proposed in an attempt of reducing its complexity, like the *simplified* Lesk (Kilgarriff and Rosenzweig, 2000; Vasilescu et al., 2004), or maximizing the chance of overlap, like in the *adapted* version (Banerjee and Pedersen, 2002). One of the limitations of Lesk approach relies on the exact match between words in the sense definitions. Semantic similarity, rather than word overlap, has been proposed as a method to overcome such a limitation. Earlier approaches were based on the notion of semantic relatedness (Patwardhan et al., 2003) and tried to exploit the relationships between synsets in the WordNet graph. More recently, Distributional Semantic Models (DSM) have stood up as a way for computing such semantic similarity. DSM allow the representation of concepts in a geometrical space through word vectors. This kind of representation captures the semantic relatedness that occurs between words in paradigmatic relations, and enables the computation of semantic similarity between whole sentences. Broadening the definition of semantic relatedness, Patwardhan and Pedersen (2006) took into account WordNet contexts: a gloss vector is built for each word sense using its definition and those of related synsets in WordNet. A distributional thesaurus is used for the expansion of both glosses and the context in Miller et al. (2012), where the overlap is computed as in the original Lesk algorithm. More recently, Basile et al. (2014) proposed a variation of Lesk algorithm based on both the simplified and the adapted version. This method combines the enhanced overlap, given by the definitions of related synsets, with the reduced number of matching that are limited to the contextual words in the simplified version. The evaluation was conducted on the SemEval-2013 Multilingual Word Sense Disambiguation task (Navigli et al., 2013), and involved the use of BabelNet

(Navigli and Ponzetto, 2012) as sense inventory. While performance for the English task was above the other task participants, the same behaviour was not reported for the Italian language.

This paper proposes a deeper investigation of the algorithm described in Basile et al. (2014) for the Italian language. We analyse the effect on the disambiguation performance of the use of two different corpora for building the distributional space. Moreover, we introduce a new sense distribution function (SDfreq), based on synset frequency, and compare its capability in boosting the distributional Lesk algorithm with respect to the one proposed in Basile et al. (2014).

The rest of the paper is structured as follows: Section 2 provides details about the *Distributional* Lesk algorithm and DSM, and defines the two above mentioned sense distribution functions exploited in this work. The evaluation, along with details about the two corpora and how the DSM are built, is presented in Section 3, which is followed by some conclusions about the presented results.

2 Distributional Lesk Algorithm

The distributional Lesk algorithm (Basile et al., 2014) is based on the simplified version (Vasilescu et al., 2004) of the original method. Let $w_1, w_2, ...w_n$ be a sequence of words, the algorithm disambiguates each target word w_i by computing the semantic similarity between the glosses of the senses associated to the target word and its context. This similarity is computed by representing in a DSM both the gloss and the context as the sum of the words they are composed of; then this similarity takes into account the co-occurrence evidences previously collected through a corpus of documents. The corpus plays a key role since the richer it is the higher is the probability that each word is fully represented in all its contexts of use. Finally, the correct sense for a word is selected by choosing the one whose gloss maximizes the semantic similarity. Despite the use of a *SemanticSpace* for computing the similarity, still the sense description can be too short for a meaningful comparison with the word context. Following this observation, we adopted an approach inspired by the adapted Lesk (Banerjee and Pedersen, 2002), and we decided to enrich the gloss of the sense with those of related meanings, duly weighted to reflect their distances with respect to the original sense. As sense inventory we choose BabelNet 1.1, a huge multilingual semantic network which comprises both WordNet and Wikipedia. The algorithm consists of the steps described as follows.

Building the glosses. We retrieve the set $S_i = \{s_{i1}, s_{i2}, ..., s_{ik}\}$ of senses associated to w_i by firstly looking up to the WordNet portion of BabelNet, then if no sense is found we seek for senses from Wikipedia, since probably the word is a named entity. This strategy was selected after tuning our system. For each sense s_{ij}, the algorithm builds the extended gloss representation g^*_{ij} by adding to the original gloss g_{ij} the glosses of related meaning retrieved through the BabelNet function "getRelatedMap", with the exception of "antonym" senses. Each word in g^*_{ij} is weighted by a function inversely proportional to the distance d between s_{ij} and the related glosses where the word occurs. Moreover, in order to emphasize more discriminative words among the different senses, we introduce in the weight a variation of the inverse document frequency (*idf*) for retrieval that we named inverse gloss frequency (*igf*). The *igf* for a word w_k occurring gf^*_k times in the set of extended glosses for all the senses in S_i, the sense inventory of w_i, is computed as $IGF_k = 1 + log_2 \frac{|S_i|}{gf^*_k}$. The final weight for the word w_k appearing h times in the extended gloss g^*_{ij} is given by:

$$weight(w_k, g^*_{ij}) = h \times IGF_k \times \frac{1}{1+d} \quad (1)$$

Building the context. The context C for the word w_i is represented by all the words that occur in the text.

Building the vector representations. The context C and each extended gloss g^*_{ij} are represented as vectors in the *SemanticSpace* built through the DSM described in Subsection 2.1.

Sense ranking. The algorithm computes the cosine similarity between the vector representation of each extended gloss g^*_{ij} and that of the context C. Then, the cosine similarity is linearly combined with a function which takes into account the usage of the meaning in the language. In this paper we investigate the two functions described in Subsection 2.2. The output of this step is a ranked list of synsets. The sense with the highest similarity is selected.

2.1 Distributional Semantics

Distributional Semantic Models are a means for representing concepts through vectors in *Semantic* (or *Word*) *Spaces*. Building the *SemanticSpace* only requires the analysis of big amounts of text data in order to collect evidence about word usage in the language in a complete unsupervised method. These methods rely on the construction of a word-to-word matrix M, which reflects the paradigmatic relations between words that share the same contexts, e.g. between words that can be used interchangeably. In this space, the vector proximity expresses the semantic similarity between words, traditionally computed as the cosine of the angle between the two word-vectors. Moreover, the concept of *semantic similarity* can be extended to whole sentences via the vector addition ($+$) operator. A sentence can always be represented as the sum of the word vectors it is composed of. Then, vector addition can be exploited to represent both the extended gloss and the target word context in order to asses their similarity.

2.2 Sense Distribution

We analyse two functions to compute the probability assigned to each synset. The first one has already been proposed in the original version of the distributional Lesk algorithm (Basile et al., 2014), the second one is based on synset frequency. It is important to point out that many synsets in BabelNet refer to named entities that do not occur in WordNet. In order to compute the probability of these synsets using a synset-tagged corpus we try to map them to WordNet and select the WordNet synset with the maximum probability. If no WordNet synset is provided, we assign a uniform probability to the synset.

Distribution based on conditional probability (SDprob). We define the probability $p(s_{ij}|w_i)$ that takes into account the sense distribution of s_{ij} given the word w_i. The sense distribution is computed as the number of times the word w_i is tagged with the sense s_{ij} in a sense-tagged corpus. Zero probabilities are avoided by introducing an additive (Laplace) smoothing. The probability is computed as follows:

$$p(s_{ij}|w_i) = \frac{t(w_i, s_{ij}) + 1}{\#w_i + |S_i|} \quad (2)$$

where $t(w_i, s_{ij})$ is the number of times the word w_i is tagged with the sense s_{ij}.

Distribution based on frequency (SDfreq). We compute the probability $p(s_{ij})$ of a meaning s_{ij} in a tagged corpus. The frequency is computed by taking into account all the occurrences of the whole set of meanings assigned to the word w_i. Given S_i, the set of the k possible meanings of w_i, the frequency of each s_{ij} in S_i is computed as:

$$p(s_{ij}) = \frac{t(s_{ij}) + 1}{\sum_{k=1}^{l}(t(s_{ik})) + |S_i|} \quad (3)$$

where $t(s_{ij})$ are the occurrences of s_{ij} in the tagged corpus.

3 Evaluation

The evaluation is performed using the dataset provided by the organizers of the Multilingual WSD (Task-12) of SemEval-2013 (Navigli et al., 2013), a traditional WSD all-words experiment where BabelNet is used as sense inventory. Our evaluation aims at: 1) analysing the algorithm performance changes in function of both the two synset distribution functions and the corpus used to built the DSM; 2) comparing our system with respect to the other task participants for the Italian language.

System Setup. Our algorithm[1] is developed in JAVA and exploits the BabelNet API 1.1.1[2]. We adopt the standard Lucene analyzer to tokenize both glosses and the context. The *SemanticSpace*s for the two corpora are built using proprietary code derived from (Widdows and Ferraro, 2008) which relies on two Lucene indexes, denoted as ItWaC and Wiki, containing documents form ItWaC Corpus (Baroni et al., 2009) and the Wikipedia dump for Italian, respectively. For each corpus, the co-occurrence matrix M contains information about the top 100,000 most frequent words. Co-occurrences are computed by taking into account a window of 5 words. M is built by using Random Indexing and by setting a reduced dimension equal to 400 and the seed to 10. Sense distribution functions are computed over MultiSemCor (Bentivogli and Pianta, 2005), a parallel (English/Italian) sense labelled corpus of SemCor. Since BabelNet Italian glosses are taken from MultiWordNet, which does not contain glosses for all the synsets, we replaced each missing gloss with the other synonym words that belong to the

[1] Available on line: `https://github.com/pippokill/lesk-wsd-dsm`

[2] Available on line: `http://lcl.uniroma1.it/babelnet/download.jsp`

Table 1: Comparison between DSMs with different Sense Distribution functions.

Run	*DSM*	*SenseDistr.*	*P*	*R*	*F*	*A*
MFS	-	-	0.572	0.572	0.572	-
ItWaC	ItWaC	-	0.614	0.613	0.613	99.73%
Wiki	Wiki	-	0.596	0.594	0.595	99.73%
ItWaCprob	ItWaC	SDprob	0.732	0.730	0.731	99.73%
ItWaCfreq	ItWaC	SDfreq	0.718	0.716	0.717	99.73%
Wikiprob	Wiki	SDprob	0.703	0.700	0.701	99.73%
Wikifreq	Wiki	SDfreq	0.700	0.698	0.699	99.73%

synset. The gloss term scoring function is always applied, since it provides better results. The synset distance d used to expand the gloss is fixed to 1 (the experiments with a distance d set to 2 did not result in any improvement). The sense distribution is linearly combined with the cosine similarity score through a coefficient set to 0.5. Using only sense distribution to select a sense is somehow similar to the most frequent sense (MFS) technique, i.e. the algorithm always assigns the most probable meaning. The MFS reported in Table 1 and Table 2 is the one computed by the task organizers in order to make results comparable. Evaluation is performed in terms of F measure.

Results of the Evaluation. Table 1 shows the results obtained by the distributional Lesk algorithm on the Italian language by exploiting different corpora and sense distribution functions. It is well known that the MFS approach obtains very good performance and it is hard to be outperformed, especially by unsupervised approaches. However, all the proposed systems are able to outperform the MFS, even those configurations that do not make use of sense distribution (ItWaC and Wiki). With respect to DSM, ItWaC corpus consistently provides better results (ItWaC vs. Wiki, ItWaCprob vs. Wikiprob, and ItWaCfreq vs. Wikifreq). By analysing the sense distribution functions, the best overall result is obtained when the SDprob function is exploited (ItWaCprob vs. ItWaCfreq), while there are no differences between SDprob and SDfreq in the DSM built on Wikipedia (Wikiprob vs. Wikifreq).

Table 2 compares the two systems built on the ItWaC corpus, with and without the sense distribution (SDprob), to the other task participants (UMCCDLSI2, DAEBAK!, GETALPBN) (Navigli et al., 2013). Moreover, we report the results of Babelfy (Moro et al., 2014) and UKB (Agirre et al., 2010), which hitherto have given the best performance on this dataset. While the system without sense distribution (ItWaC) is over the baseline but still below many task participants, the run which exploits the sense distribution (ItWaCprob) always outperforms the other systems.

Table 2: Comparison with other systems.

System	*F*
ItWaCprob	0.731
UKB	0.673
Babelfy	0.666
UMCC-DLSI-2	0.658
ItWaC	0.613
DAEBAK	0.613
MFS	*0.572*
GETALP-BN	0.528

4 Conclusions and Future Work

This paper proposed an analysis for the Italian language of an enhanced version of Lesk algorithm, which replaces the word overlap with distributional similarity. We analysed two DSM built over the ItWaC and Wikipedia corpus along with two sense distribution functions (SDprob and SDfreq). The sense distribution functions were computed over MultiSemCor, in order to avoid missing references between Italian and English synsets. The combination of the ItWaC-based DSM with the SDprob function resulted in the best overall result for the Italian portion of SemEval Task-12 dataset.

Acknowledgements

This work fulfils the research objectives of the project "VINCENTE - A Virtual collective INtelligenCe ENvironment to develop sustainable Technology Entrepreneurship ecosystems" (PON 02 00563 3470993) funded by the Italian Ministry of University and Research (MIUR).

References

Eneko Agirre, Aitor Soroa, and Mark Stevenson. 2010. Graph-based Word Sense Disambiguation of Biomedical Documents. *Bioinformatics*, 26(22):2889–2896, November.

Satanjeev Banerjee and Ted Pedersen. 2002. An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, volume 2276 of *Lecture Notes in Computer Science*, pages 136–145. Springer Berlin Heidelberg.

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.

Pierpaolo Basile, Annalina Caputo, and Giovanni Semeraro. 2014. An Enhanced Lesk Word Sense Disambiguation Algorithm through a Distributional Semantic Model. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1591–1600, Dublin, Ireland, August. Dublin City University and Association for Computational Linguistics.

Luisa Bentivogli and Emanuele Pianta. 2005. Exploiting Parallel Texts in the Creation of Multilingual Semantically Annotated Resources: The MultiSemCor Corpus. *Natural Language Engineering*, 11(3):247–261.

Adam Kilgarriff and Joseph Rosenzweig. 2000. Framework and Results for English SENSEVAL. *Computers and the Humanities*, 34(1-2):15–48.

Michael Lesk. 1986. Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone. In *Proceedings of the 5th Annual International Conference on Systems Documentation*, SIGDOC '86, pages 24–26, New York, NY, USA. ACM.

Tristan Miller, Chris Biemann, Torsten Zesch, and Iryna Gurevych. 2012. Using Distributional Similarity for Lexical Expansion in Knowledge-based Word Sense Disambiguation. In *Proceedings of COLING 2012*, pages 1781–1796, Mumbai, India, December. The COLING 2012 Organizing Committee.

Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity Linking meets Word Sense Disambiguation: a Unified Approach. *TACL*, 2:231–244.

Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The Automatic Construction, Evaluation and Application of a Wide-Coverage Multilingual Semantic Network. *Artificial Intelligence*, 193:217–250.

Roberto Navigli, David Jurgens, and Daniele Vannella. 2013. SemEval-2013 Task 12: Multilingual Word Sense Disambiguation. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 222–231, Atlanta, Georgia, USA, June. Association for Computational Linguistics.

Siddharth Patwardhan and Ted Pedersen. 2006. Using WordNet-based Context Vectors to Estimate the Semantic Relatedness of Concepts. In *Proceedings of the EACL 2006 Workshop on Making Sense of Sense: Bringing Computational Linguistics and Psycholinguistics Together*, pages 1–8, Trento, Italy, April.

Siddharth Patwardhan, Satanjeev Banerjee, and Ted Pedersen. 2003. Using Measures of Semantic Relatedness for Word Sense Disambiguation. In *Proceedings of the 4th International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing'03, pages 241–257, Berlin, Heidelberg. Springer-Verlag.

Florentina Vasilescu, Philippe Langlais, and Guy Lapalme. 2004. Evaluating Variants of the Lesk Approach for Disambiguating Words. In *Proceedings of the Conference on Language Resources and Evaluation (LREC)*, pages 633–636.

Dominic Widdows and Kathleen Ferraro. 2008. Semantic Vectors: a Scalable Open Source Package and Online Technology Management Application. In *Proceedings of the Conference on Language Resources and Evaluation (LREC)*.