UNIBA: Combining Distributional Semantic Models and Sense Distribution for Multilingual All-Words Sense Disambiguation and Entity Linking

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

Abstract

This paper describes the participation of the UNIBA team in the Task 13 of SemEval-2015 about Multilingual All-Words Sense Disambiguation and Entity Linking. We propose an algorithm able to disambiguate both word senses and named entities by combining the simple Lesk approach with information coming from both a distributional semantic model and usage frequency of meanings. The results for both English and Italian show satisfactory performance.

1 Introduction

SemEval-2015 Task 13 (Moro and Navigli, 2015) aims to evaluate systems that provide a comprehensive representation of text through linking of both words and entities with concepts in a knowledge base. Besides the traditional difficulties of word sense disambiguation, this task requires specific methods able to tackle the challenges posed by the named entity recognition, disambiguation and linking steps.

This paper proposes a unified strategy for word sense and named entity disambiguation which leverages BabelNet, a multilingual resource that encompasses both encyclopedic and lexicographic knowledge (Navigli and Ponzetto, 2012). Our approach relies on the Distributional Lesk (DL-WSD) algorithm (Basile et al., 2014), which is able to disambiguate a word occurrence by computing the similarity between word context and the glosses associated with all possible word meanings. Such a similarity is computed through a Distributional Semantic Model (DSM) (Sahlgren, 2006).

In this work we describe an extension of the DL-WSD algorithm that exploits a specific module for entity discovery given a list of possible surface forms. In particular, we build an index in which each surface form (i.e. candidate entity) is paired to the list of all its possible meanings in a semantic network. This index of surface forms is exploited to look up all candidate entities in a text.

The rest of this paper is structured as follows: Section 2 provides details about the adopted strategy, and describes the two main steps: 1) Entity Recognition and 2) Disambiguation. An experimental evaluation, along with details about results, is presented in Section 3, while conclusions close the paper.

2 Methodology

Our methodology is a two-step algorithm consisting in an initial identification of all possible entities mentioned in a text followed by the disambiguation of both words and named entities through the DL-WSD algorithm. The semantic network is exploited twice in order to 1) extract all the possible surface forms related to entities, and 2) retrieve glosses used in the disambiguation process.

2.1 Entity Recognition

In order to speed up the entity recognition step we build an index in which for each surface form (entity) the set of all its possible meanings in the semantic network is reported. Lucene[1] is exploited to

[1] http://lucene.apache.org/

Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 360–364,
Denver, Colorado, June 4-5, 2015.

build the index, specifically for each surface form (lexeme) occurring in BabelNet, a document composed of two fields is created. The first field stores the surface form, while the second one contains the list of all possible BabelSynsets that refer to the surface form in the first field. The index is built separately for each language, Italian and English. The entity recognition module exploits this index in order to find entities in a text. Given a text fragment, the module performs the following steps:

- Building all n-grams up to five words;
- Querying the index and retrieving the list of the top t matching surface forms for each n-gram. It is possible to enable a multi-match strategy; for example the 3-gram "European Union Commission" can match two entities: "European Union" and "European Union Commission". The multi-match strategy provides disambiguation for all the possible entities, otherwise the longest surface form is selected;
- Scoring each surface form by exploiting two different approaches:
  - **EXACT_MATCH** computes the linear combination between the score provided by the search engine and a string similarity function based on the Levenshtein Distance between the n-gram and the candidate surface form in the index;
  - **PARTIAL_MATCH** computes the linear combination between the two scores provided by the EXACT_MATCH and the Jaccard Index in terms of common words between the n-gram and the candidate surface form;
- Filtering the candidate entities recognized in the previous steps; entities are removed if the score computed in the previous step is below a given threshold and/or the sequence of PoS-tags related to the n-gram does not match a set of defined patterns;
- Assigning to each candidate entity two additional scores according to the percentage of: 1) stop words, and 2) words that do not contain at least one upper-case character. A threshold can be fixed for each score to filter out some entities.

Moreover, for each entity we build a set of alternatives. For example, given the candidate entity "European Union" we create the set of alternative surface forms {*European, Union, EU, E.U.*}. Then, we add all the BabelSynsets of "European Union" to the list of possible meanings of those words that follow the candidate entity and belong to the set of alternative forms.

The output of the entity recognition module is a list of candidate entities in which a set of possible meanings (BabelSynset) is assigned to each surface form in the list. The set of named entities extracted by this module and the list of all the words in the text are the input to the DL-WSD algorithm.

2.2 DL-WSD

We exploit the distributional Lesk algorithm proposed by Basile et al. (2014) for disambiguating words and named entities. The algorithm replaces the concept of word overlap initially introduced by (Lesk, 1986) with the broader concept of semantic similarity computed in a distributional semantic space. Let $w_1, w_2, ...w_n$ be a sequence of words/entities, the algorithm disambiguates each target word/entity w_i by computing the semantic similarity between the glosses of senses associated with the target word/entity and its context. This similarity is computed by representing in a DSM both the gloss and the context as the sum of 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. We exploit the word2vec tool[2](Mikolov et al., 2013) in order to build a DSM, by analyzing all the pages in the last English/Italian Wikipedia Dump. The correct sense for a word is the one whose gloss maximizes the semantic similarity with the word/entity context. The sense description can still be too short for a meaningful comparison with the word/entity context. Following this observation, we adopted an approach inspired by the adapted Lesk (Banerjee and Pedersen, 2002), and

[2]https://code.google.com/p/word2vec/

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. The algorithm consists of the following steps.

Building the glosses. We retrieve the set $S_i = \{s_{i1}, s_{i2}, ..., s_{ik}\}$ of senses associated to the word/entity w_i. For named entities such a set is provided by the entity recognition module, while for words the set is obtained by firstly looking up to the WordNet portion of BabelNet, then if no sense is found we seek for senses from Wikipedia. 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 meanings 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 discriminative words among the different senses, in the weight we introduce 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.

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. We analyse a function that computes the probability assigned to each synset given a word/named entity as follows:

Word. We exploit a synset-tagged corpus and we attempt to map each word occurrence to WordNet (Miller, 1995). Then, we select the WordNet sysnet with the maximum probability.

Named Entity. We retrieve from BabelNet the Wikipedia title pages related to the BabelSynset and count the number of times a Wikipedia page is linked from another page. In this way we use Wikipedia as a synset-tagged corpus.

We define the probability $p(s_{ij}|w_i)$ that takes into account the sense distribution of s_{ij} given the word/entity w_i. The sense distribution is computed as the number of times the word/entity w_i is tagged with the sense. 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/entity w_i is tagged with the sense s_{ij}.

3 Evaluation

The evaluation aims at comparing the system result against a gold standard manually annotated using synsets from BabelNet 2.5.1. Test data consists of four documents that belong to three different domains: biomedical, maths and computer science, and general. The idea is to evaluate the algorithm performance both in general and specific domains. We submitted three runs with different parameter settings that mainly affected the entity recognition module. System settings are reported in Table 1.

Run	Match	PoS-Tag	Threshold
Run1	EXACT	YES	1.0
Run2	PARTIAL	YES	0.75
Run3	PARTIAL	NO	0.75

Table 1: System settings.

The Match column indicates the type of matching used during the entity recognition step, PoS-Tag reports the usage of the filter based on PoS-Tag patterns, and finally the table reports the Threshold used by the matching filter. Moreover, we set the number

	EN							IT						
Run	all	NE	WSD	n	v	r	a	all	NE	WSD	n	v	r	a
best	65.8	88.9	64.6	70.3	57.7	79.0	79.5	59.9	54.9	61.3	56.6	62.7	62.5	69.6
Run1	58.4	84.4	56.5	63.3	57.1	79.0	-	50.8	48.5	51.0	53.7	61.1	60.0	-
Run2	58.3	82.9	56.5	63.2	57.1	79.0	-	50.9	48.5	51.0	53.8	61.1	60.0	-
Run3	58.3	82.9	56.5	63.2	57.1	79.0	-	50.9	50.0	51.0	53.7	61.1	60.0	-

Table 2: Official task results.

	EN				IT			
Run	all	NE	WSD	a	all	NE	WSD	a
Run1	**61.3**	88.1	59.5	48.2	**59.5**	51.0	59.9	77.7
Run2	61.0	85.2	59.3	47.6	59.6	51.0	60.0	77.7
Run3	60.8	84.4	59.2	47.6	**59.5**	51.0	59.9	77.7

Table 3: Task results after the adjective fix.

of entities retrieved by the search engine to 25, and the thresholds for stop-word and lower-case filters to 0.3.

Table 2 reports the official results released by the task organizers. Our best system ranks 4th among 17 submissions for English, and 4th among 8 for Italian. As reported in Table 2, our system is not scored for adjective. This issue is due to a problem with PoS-tag: in trial data adjectives are tagged with 'A', while in the test data with 'J'. Inadvertently, we did not report this modification in our system during the testing. After the release of the gold standard, we fixed that issue in our system and performed a new experiment whose results are reported in Table 3. Since results for noun, verbs and adverbs are not affected by the fix, they are not reported again in the table. Considering the new results reported in Table 3, our system is able to rank 3rd for English, and 2nd for Italian.

Another goal of the task is to evaluate system performance on different domains. In particular three domains were provided: biomedical (**bio**), maths and computer science (**math**), and general domain (**gnr**). Results for each domain and language are reported in Table 4. Our performance on each domain shows a trend very similar to the best system for each language: the math/computer science domain is the hardest to disambiguate, while the biomedical one seems to be the easiest. A deep analysis of domain results shows that our system is the best to disambiguate named entities for Italian biomedical and math/computer science domains, while it provides the lowest performance in the general domain for both Italian and English. It is important to note that the system settings seem not to affect the overall performance, while a deep analysis focused on the only named entities reveals slight differences between settings. This behaviour is due to the different methods used to recognize named entities. The task description paper reports more details about results (Moro and Navigli, 2015).

	EN			IT		
Run	bio	math	gnr	bio	math	gnr
best	71.2	54.1	67.2	65.5	52.1	61.0
Run1	66.6	50.8	62.0	64.4	51.2	58.4
Run2	66.4	50.8	60.7	64.4	51.2	58.7
Run3	66.4	50.8	60.2	64.4	51.2	58.4

Table 4: System performance for each domain.

4 Conclusions

We presented a unified approach to entity linking and word sense disambiguation which relies on a distributional extension of the simple Lesk disambiguation algorithm. This algorithm has been extended with an entity recognition module able to recognize candidate named entities. We evaluated three different configurations of such recognition module within the Task 13 of SemEval-2015. Experimental evaluation showed competitive results, with our best run ranked among the top systems.

References

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.

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.

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.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

George A Miller. 1995. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41.

Andrea Moro and Roberto Navigli. 2015. SemEval-2015 Task 13: Multilingual All-Words Sense Disambiguation and Entity Linking. In *Proceedings of SemEval-2015*.

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.

Magnus Sahlgren. 2006. *The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces*. Ph.D. thesis.