# UNIBA @ EVALITA 2009
# Lexical Substitution Task

Pierpaolo Basile and Giovanni Semeraro

Department of Computer Science
University of Bari
Via E. Orabona, 4 - 70125 Bari - ITALY
{basilepp, semeraro}@di.uniba.it

**Abstract.** This paper presents the participation of the University of Bari (UNIBA) at the EVALITA 2009 Lexical Substitution Task. The goal of the task is to substitute a word in a particular context providing the best synonyms which fit in that context. This task is a different way to evaluate Word Sense Disambiguation (WSD) algorithms. Indeed, understanding the meaning of the target word is necessary to find the best substitution. An important aspect of this kind of task is the possibility of finding synonyms without using a particular sense inventory thus allowing the participation of unsupervised approaches. UNIBA proposes two systems: the former is based on a knowledge-based WSD algorithm which exploits ItalWordNet as knowledge-base, the latter is based on an unsupervised approach which relies on a large corpus in order to find the different contexts in which words are used.

**Keywords:** Lexical Substitution, Word Sense Disambiguation, NLP, Evaluation.

## 1 Introduction

The goal of the Lexical Substitution Task consists in selecting an alternative word for a given one in a particular context preserving the meaning. This task is not easy one since lists of candidate words are not provided by the organizers. Therefore, the system must identify a set of candidate words and then select only those words which fit the context. The organizers propose this kind of task to promote unsupervised systems. An example of the task follows. Consider the sentence:

È morta tra le braccia della sua tata che amava così *tanto*.

In the previous sentence the target word is "*tanto*". Taking into account the meaning of the word "*tanto*", in this particular context the best synonyms are: "*intensamente*" and "*fortemente*".

We propose two systems to solve the problem of lexical substitution: the former is based on a knowledge-based WSD algorithm which exploits ItalWordNet

as knowledge-base, the latter is based on an unsupervised approach which relies on a large corpus in order to find the different contexts in which words are used. Moreover in the second approach we adopt two different lexical resources to select the list of possible synonyms for a given word. In particular, we use ItalWordNet as thesaurus and "Il dizionario dei sinonimi e contrari, De Mauro Paravia"[1] as dictionary.

The paper is organized as follows: Section 2 describes our WSD strategy adopted for the lexical substitution task, while Section 3 presents the method which exploits a large corpus to find the best substitutions. An experimental session was carried out in order to evaluate the proposed approaches and results are presented in Section 4. Conclusions are discussed in Section 5.

## 2 JIGSAW: a knowledge-based WSD algorithm

The goal of a WSD algorithm consists in assigning a word $w_i$ occurring in a document $d$ with its appropriate meaning or sense $s$, by exploiting the *context* $C$ in which $w_i$ is found. The sense $s$ is selected from a predefined set of possibilities, usually known as *sense inventory*. In the proposed algorithm, the sense inventory is obtained from ItalWordNet. JIGSAW is a WSD algorithm based on the idea of combining three different strategies to disambiguate nouns, verbs, adjectives and adverbs. The main motivation behind our approach is that the effectiveness of a WSD algorithm is strongly influenced by the Part-of-Speech (PoS) of the target word. An adaptation of Lesk dictionary-based WSD algorithm has been used to disambiguate adjectives and adverbs [1], an adaptation of the Resnik algorithm has been used to disambiguate nouns [2], while the algorithm we developed for disambiguating verbs exploits the nouns in the *context* of the verb as well as the nouns both in the glosses and in the phrases exploited by ItalWordNet to describe the usage of a verb. The algorithm is based on three different procedures for nouns, verbs, adverbs and adjectives, called $JIGSAW_{nouns}$, $JIGSAW_{verbs}$, $JIGSAW_{others}$, respectively. A short description of the first two procedures follows, whereas the detailed description for all procedure can be found in [3].

$JIGSAW_{nouns}$: The procedure is obtained by making some variations to the algorithm designed by Resnik for disambiguating noun groups. Given a set of nouns $W = \{w_1, w_2, \ldots, w_n\}$, obtained from document $d$, with each $w_i$ having an associated sense inventory $S_i = \{s_{i1}, s_{i2}, \ldots, s_{ik}\}$ of possible senses, the goal is assigning each $w_i$ with the most appropriate sense $s_{ih} \in S_i$, according to the *similarity* of $w_i$ with the other words in $W$ (the context for $w_i$). The idea is to define a function $\varphi(w_i, s_{ij})$, $w_i \in W$, $s_{ij} \in S_i$, that computes a value in $[0, 1]$ representing the confidence with which word $w_i$ can be assigned with sense $s_{ij}$. $JIGSAW_{nouns}$ differs from the original algorithm by Resnik in several ways. First, in order to measure the relatedness

[1] Italian dictionary of synonyms and antonyms, available on line: http://www.demauroparavia.it/

of two words we adopted a modified version of the Leacock-Chodorow measure [4], which computes the length of the path between two concepts in a hierarchy by passing through their *Most Specific Subsumer* (MSS). Moreover, in the similarity computation, we introduced both a Gaussian factor $G(pos(w_i), pos(w_j))$, which takes into account the distance between the position of the words in the text to be disambiguated, and a factor $R(k)$, which assigns $s_{ik}$ with a numerical value, according to the word meaning frequency. In particular, the $R(k)$ function takes into account the real distribution of word meanings. The idea is to compute the sense rank frequency using MultiSemCor [5] corpus and then infer a statistical distribution of word meanings for each PoS. As Kilgarriff describes in [6], the ZIPF distribution approximates well the natural distribution of meanings. The ZIPF formula is:

$$f(k; N; s) = \frac{1/k^s}{\sum_{n=1}^{N} 1/n^s} \tag{1}$$

where:

- $N$ is the number of word meanings;
- $k$ is the word meaning rank. We adopt the ItalWordNet synset rank;
- $s$ is the value of the exponent characterizing the distribution.

In order to set the parameter $s$, we have computed the frequency of the word meaning rank for each PoS exploiting MultiSemCor. We approximated the only unknown parameter of the distribution that is $s$ using the *Pearson's chi-square* $\chi^2$ test method described in [7]. Finally, we adopted the ZIPF distribution as $R(k)$ function using a different value of $s$ based on the PoS.

$JIGSAW_{verbs}$: We define the *description* of a synset as the string obtained by concatenating the gloss and the sentences exploited by ItalWordNet to explain the usage of a synset. First, $JIGSAW_{verbs}$ includes, in the context $C$ for the target verb $w_i$, all the nouns in the window of $2n$ words surrounding $w_i$. For each candidate synset $s_{ik}$ of $w_i$, the algorithm computes $nouns(i, k)$, that is the set of nouns in the description for $s_{ik}$. Then, for each $w_j$ in $C$ and each synset $s_{ik}$, the following value is computed:

$$max_{jk} = max_{w_l \in nouns(i,k)} \{sim(w_j, w_l, depth)\} \tag{2}$$

where $sim(w_j, w_l, depth)$ is the same similarity measure adopted by $JIGSAWnouns$. In other words, $max_{jk}$ is the highest similarity value for $w_j$ wrt the nouns related to the $k$-th sense for $w_i$. Finally, an overall similarity score among $s_{ik}$ and the whole context $C$ is computed:

$$\varphi(i, k) = R(k) \cdot \frac{\sum_{w_j \in C} G(pos(w_i), pos(w_j)) \cdot max_{jk}}{\sum_h G(pos(w_i), pos(w_h))} \tag{3}$$

where both $R(k)$ and $G(pos(w_i), pos(w_j))$, that gives a higher weight to words closer to the target word, are defined as in $JIGSAWnouns$. The synset assigned to $w_i$ is the one with the highest $\varphi$ value.

## 3 Lexical Substitution exploiting a large Corpus

The second strategy which we proposed relies on a large corpus of documents. We adopted ItWaC-Italian Web Corpus[2] [8] which is a large corpus of about 1,900,000 documents and it is built automatically from the Web. The idea is to index ItWaC and then try to find phrases in which synonyms of the target word occur in the same context. We use Apache LUCENE API[3] to index and search ItWaC. Moreover, we need a lexical resource which provides a list of candidate synonyms for the target word. We exploited two resources:

- ItalWordNet: is the Italian version of WordNet. The basic unit of ItalWordNet is the synset that is a set of words which refer to the same meaning (a set of synonyms).
- "Il dizionario dei sinonimi e contrari, De Mauro Paravia": is an Italian dictionary of synonyms and antonyms.

The method description follows. Given the sentence:

Solo oggi, con lo spoglio *completo* dei tabulati, se ne potrà sapere di più.

where "*completo*" is the target word, the first step is to retrieve the list of possible synonyms $CS$ of the target word from ItalWordNet or De Mauro dictionary, and then find the best synonyms, since we need to define a function to rank the candidate synonyms. The idea is to search the number of phrases into the corpus in which the synonym occurs in the same context. The context is built using the $n$-gram strategy, for example given the synonym $s_i \in CS$ and $n = 3$ we build the following phrase queries: *"lo spoglio* $s_i$*"*, *"spoglio* $s_i$ *dei"* and *"*$s_i$ *dei tabulati"*. For each $s_i$ a score is computed according to Equation 4:

$$score(s_i) = ndoc * (1/slop) * boost_{s_i} \tag{4}$$

where *ndoc* is the number of documents in which the phrase occurs, *slop* is the distance between words and $boost_{s_i}$ is a boost factor. The *slop* factor allows to find words which are a within a specific distance away, in this way an exact match between the context and the phrase query is not required. The factor $(1/slop)$ gives more weight to synonyms which occur into the context with less *slop*. The searching step starts with $slot = 1$ and if no results are retrieved the *slop* is incremented by one until *slop* is equal to a specified value $slop_{max}$. The $boost_{s_i}$ is used to give more weight to some synonyms, this strategy is adopted when we use a thesaurus to extract the list of candidate synonyms. More details about $boost_{s_i}$ and $slop_{max}$ are provided in Section 4. The previous strategy is applied to each candidate synonym. Finally, we obtain a list of candidate synonyms sorted by Equation 4.

[2] Online: http://wackybook.sslmit.unibo.it/
[3] Lucene API: http://lucene.apache.org

## 4 Evaluation

The goal of the evaluation is to measure the system ability to find correct synonyms for a given word. The dataset provided by the organizer contains 2,011 instances in XML format. All the features useful to run the algorithms are extracted by a NLP tool called META-MultilanguagE Text Analyzer [9].

Regarding the system setup, we adopt LUCENE to index ItWaC. The output of the indexing process is an index of 8,6 G-bytes, 1,867,618 documents and 2,256,895 terms. To run the evaluation, we need to set some parameters such as $slop_{max}$ and *boost* factors for candidate synonyms. After a training step, using the data provided by the task organizers, we set $slop_{max} = 30$ and modify the *boost* adopting the following strategies:

- candidate synonyms provided by the dictionary not have a boost factor
- candidate synonyms provided by ItalWordNet have a boost factor equal to 1, while words in hypernym synsets of the candidate synonyms have a boost factor equal to 0.5. We exploit hypernyms to produce a rich list of candidates, this is useful when a synset contains few words. For example, in ItalWordNet, the second synset of the word "casa" contains only the word "casa", to overcome this problem we enrich the list of candidate synonyms using words in hypernym synsets.

The systems are evaluated using two scoring types: **best** scores the best guessed synonym and out-of-ten (**oot**) scores the best 10 guessed synonyms. For each scoring type precision (P) and recall (R) are computed. Mode precision (P-mode) and mode recall (R-mode) calculate precision and recall, respectively, against the synonym chosen by the majority of the annotators (if there is a majority). The details on the evaluation and scoring types are provided in the task guidelines [10]. Results of the evaluation are reported in Table 1. Our systems are tagged as follow: **uniba1** is the algorithm based on $JIGSAW$, **uniba2** is the method based on ItWaC corpus and De Mauro dictionary, and **uniba3** is the method based on ItWaC and ItalWordNet. C and B are other two participants.

**Table 1.** Evaluation results

| System | best | | | | oot | | | |
|---|---|---|---|---|---|---|---|---|
| | P | R | P-mode | R-Mode | P | R | P-mode | R-Mode |
| **uniba2** | **8.16** | 7.18 | **10.58** | 10.58 | **41.46** | 36.50 | **47.23** | 47.23 |
| B1 | 6.26 | 6.01 | 11.28 | 10.84 | 16.65 | 16.00 | 16.00 | 24.97 |
| **uniba1** | 6.80 | 5.53 | 8.90 | 8.90 | 37.74 | 30.69 | 34.84 | 34.84 |
| **uniba3** | 6.28 | 5.46 | 8.13 | 8.13 | 28.54 | 24.79 | 34.58 | 34.58 |
| C3 | 3.95 | 3.21 | 6.58 | 6.58 | 23.48 | 19.11 | 26.58 | 26.58 |
| C2 | 3.9 | 3.17 | 6.71 | 6.71 | 23.00 | 18.72 | 26.32 | 26.32 |
| C1 | 3.16 | 3.16 | 6.97 | 6.97 | 20.09 | 20.09 | 27.74 | 27.74 |
| C4 | 3.52 | 2.80 | 5.03 | 5.03 | 18.62 | 14.78 | 20.52 | 20.52 |

The results show that the method which combines ItWaC corpus and dictionary obtains the best performance. Moreover $JIGSAW$, a knowledge-based WSD algorithm, achieves very encouraging results which prove the effectiveness of WSD in this kind of task. Finally, the results obtained by uniba3 show that De Mauro dictionary is the best synonyms resource with respect to ItalWordNet. We can conclude that in this kind of task dictionary works better than thesaurus. Finally, our systems achieve the best results wrt the other participants.

## 5 Conclusions

We described our participation in EVALITA Lexical Substitution Task proposing two systems: a knowledge-based WSD algorithm and a method based on a large corpus. Moreover, we adopt two resources to retrieve the list of candidate substitutions: ItalWordNet and an Italian Dictionary. The results prove that the method based on a large corpus is more effective than the method based on WSD, but the results obtained by the WSD are very encouraging in spite of the past beliefs. Moreover, the Italian Dictionary combined with the method based on a large corpus provides the best task result.

## References


1. Banerjee, S., Pedersen, T.: An adapted lesk algorithm for word sense disambiguation using wordnet. In: Proceedings of 3rd International Conference on Computational Linguistics and Intelligent Text Processing (CICLing'02), pp. 136–145. Springer-Verlag (2002)
2. Resnik, P.: Disambiguating noun groupings with respect to WordNet senses. In: Proceedings of the Third Workshop on Very Large Corpora, pp. 54–68. Association for Computational Linguistics (1995)
3. Basile, P., de Gemmis, M., Gentile, A., Lops, P., Semeraro, G.: Jigsaw algorithm for word sense disambiguation. In: Proceedings of SemEval-2007: 4th International Workshop on Semantic Evaluations, pp. 398–401. ACL press (2007)
4. Leacock, C., Chodorow, M.: Combining local context and wordnet similarity for word sense identification. In: C. Fellbaum (ed.) WordNet: An Electronic Lexical Database, pp. 305–332. MIT Press (1998)
5. Bentivogli, L., Pianta, E.: Exploiting parallel texts in the creation of multilingual semantically annotated resources: the MultiSemCor Corpus. Natural Language Engineering, vol. 11, issue 3, pp. 247–261 (2005)
6. Kilgarriff, A.: How Dominant Is the Commonest Sense of a Word? In: P. Sojka, I.K., Pala, K. (eds.) TSD 2004, Text, Speech and Dialogue 7th International Conference, vol. 2448, pp. 1–9. Springer-Verlag Berlin Heidelberg (2004)
7. Chernoff, H., Lehmann, E.L.: The use of maximum likelihood estimates in $\chi^2$ tests for goodness-of-fit. The Annals of Mathematical Statistics, vol. 25, pp. 579–586 (1954)
8. Baroni, M., Kilgarriff, A.: Large linguistically-processed Web corpora for multiple languages. In: Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics, pp. 87–90 (2004)

9. Basile, P., de Gemmis, M., Gentile, A., Iaquinta, L., Lops, P., Semeraro, G.: META-MultilanguagE Text Analyzer. In: Proceedings of the Language and Speech Technnology Conference-LangTech, pp. 137–140 (2008)
10. Toral, A.: EVALITA 2009 Lexical Substitution Task - Guidelines for Participants, http://evalita.fbk.eu/lexical.html