# Robust and Efficient Page Rank for Word Sense Disambiguation

**Diego De Cao, Roberto Basili, Matteo Luciani, Francesco Mesiano, Riccardo Rossi**
Dept. of Computer Science,
University of Roma Tor Vergata, Roma, Italy
{decao,basili}@info.uniroma2.it
{matteo.lcn,fra.mesiano,ricc.rossi}@gmail.com

## Abstract

Graph-based methods that are *en vogue* in the social network analysis area, such as centrality models, have been recently applied to linguistic knowledge bases, including unsupervised Word Sense Disambiguation. Although the achievable accuracy is rather high, the main drawback of these methods is the high computational demanding whenever applied to the large scale sense repositories. In this paper an adaptation of the PageRank algorithm recently proposed for Word Sense Disambiguation is presented that preserves the reachable accuracy while significantly reducing the requested processing time. Experimental analysis over well-known benchmarks will be presented in the paper and the results confirm our hypothesis.

## 1 Introduction

Lexical ambiguity is a fundamental aspect of natural language. Word Sense Disambiguation (WSD) investigates methods to automatically determine the intended sense of a word in a given context according to a predefined set of sense definitions, provided by a semantic lexicon. Intuitively, WSD can be usefully exploited in a variety of NLP (e.g. Machine Translation (Chan et al., 2007; Carpuat and Wu, 2007)) and Information Retrieval tasks such as *ad hoc retrieval* (Krovetz, 1997; Kim et al., 2004) or Question Answering (Beale et al., 2004). However controversial results have been often obtained, as for example the study on text classification reported in (Moschitti and Basili, 2004). The impact of WSD on IR tasks is still an open issue and large scale assessment is needed. For this reason, unsupervised approaches to inductive WSD are appealing. In contrast with supervised methods that strongly rely on manually labeled data sets, those methods do not require annotated examples for all words and can thus support realistic (large scale) benchmarks, as needed in IR research.

In recent years different approaches to Word Sense Disambiguation task have been evaluated through comparative campaigns, such as the earlier Senseval evaluation exercises. (Palmer et al., 2001; Snyder and Palmer, 2004) or the most recent (Pradhan et al., 2007).

The best accuracy is reached by WSD based on supervised methods that exploit large amounts of hand-tagged data to train discriminative or generative disambiguation models. The common alternative to supervised systems are knowledge-based WSD systems that try to exploit information made available by large Lexical Knowledge Bases (LKB). They enable the definition of several metrics to estimate semantic similarity (e.g. (Lesk, 1986) or (Agirre and Rigau, 1996), (Basili et al., 2004) methods) and then use it to rank the alternative senses according to the incoming context. Moreover they make available large relationship sets between pairs of lexical meaning units, such as synonymy, hyponymy or meronymy. The resulting networks represent at various grains and degrees of approximation models of the mental lexicons. It is not by chance that early research on WSD based on semantic dictionaries were applying models of network activation processes (in particular simulated annealing as in (Cowie et al., 1992)) for precise and fast disambiguation.

It has been more recently that graph-based methods for knowledge-based WSD have gained much attention in the NLP community ((Sinha and Mihalcea, 2007), (Navigli and Lapata, 2007), (Agirre and Soroa, 2008), (Agirre and Soroa, 2009)). In these methods a graph representation for senses (nodes) and relation (edges) is first built. Then graph-based techniques that are sensible to the structural properties of the graph are used to find the best senses for words in the incoming contexts. The relation employed by the different methods are of several types such as synonymy, antonymy but also co-occurrence based lexical similarity computed externally over a corpus. These give rise to real-valued weights that determine large weighted directed graphs. Usu-



*Proceedings of the 2010 Workshop on Graph-based Methods for Natural Language Processing, ACL 2010*, pages 24–32,
Uppsala, Sweden, 16 July 2010. 

ally, the employed disambiguation is carried out by ranking the graph nodes. Thus the concepts with highest ranks are assigned to the corresponding words. In (Agirre and Soroa, 2009), a comparative analysis of different graph-based models over two well known WSD benchmarks is reported. In the paper two variants of the random surfer model as defined by PageRank model (Brin and Page, 1998) are analyzed. A special emphasis for the resulting computational efficiency is also posed there. In particular, a variant called *Personalized PageRank* ($PPR$) is proposed (Agirre and Soroa, 2009) that tries to trade-off between the amount of the employed lexical information and the overall efficiency. In synthesis, along the ideas of the Topic sensitive PageRank (Haveliwala, 2002), $PPR$ suggests that a proper initialization of the teleporting vector $\vec{p}$ suitably captures the context information useful to drive the random surfer PageRank model over the graph to converge towards the proper senses in fewer steps. The basic idea behind the adoption of $PPR$ is to impose a personalized vector that expresses the contexts of all words targeted by the disambiguation. This method improves on the complexity of the previously presented methods (e.g. (Agirre and Soroa, 2008)) as it allows to contextualize the behaviors of PageRank over a sentence, without asking for a different graph: in this way the WordNet graph is always adopted, in a word or sentence oriented fashion. Moreover, it is possible to avoid to rebuild a graph for each target word, as the entire sentence can be coded into the personalization vector. In (Agirre and Soroa, 2009), a possible, and more accurate alternative, is also presented called PPR word2word ($PPRw2w$) where a different personalization vector is used for each word in a sentence. Although clearly less efficient in terms of time complexity, this approach guarantees the best accuracy, so that it can be considered the state-of-the art in unsupervised WSD.

In this paper a different approach to personalization of the PageRank is presented, aiming at preserving the suitable efficiency of the sentence oriented *PPR* algorithm for WSD but achieving an accuracy at least as high as the $PPRw2w$ one. We propose to use distributional evidence that can be automatically acquired from a corpus to define the topical information encoded by the personalization vector, in order to amplify the bias on the resulting $PPR$ and improve the performance of the sentence oriented version. The intuition is that distributional evidence is able to cover the gap between word oriented usages of the $PPR$ as for the $PPRw2w$ defined in (Agirre and Soroa, 2009), and its sentence oriented counterpart. In this way we can preserve higher accuracy levels while limiting the number of PageRank runs, i.e. increasing efficiency.

The paper is structured as follows. We first give a more detailed overview of the *PageRank* and *Personalized PageRank* algorithms in Section 2. In Section, 3 a description of our distributional approach to the personalized PageRank is provided. A comparative evaluation with respect to previous works is then reported in Section 4 while section 5 is left for conclusions.

## 2 Graph-based methods for Word Sense Disambiguation

Word sense disambiguation algorithms in the class of graph-based method are unsupervised approaches to WSD that rely almost exclusively on the lexical KB graph structure for inferring the relevance of word senses for a given context. Much current work in WSD assume that meaning distinctions are provided by a reference lexicon (the LKB), which encodes a discrete set of senses for each individual word. Although the largely adopted reference resource is WordNet (Miller et al., 1990), the graph-based algorithms are not limited to this particular lexicon. In these methods, nodes are derived from the sense units, i.e. the synsets, and edges are derived from semantic relations established between synsets. We will hereafter use WordNet to discuss the details of the different steps. Every algorithm can be decomposed in a set of general steps:

**Building the graph**. The first step proceeds to the definition of the graph structure. As introduced before, WordNet is mapped into a graph whose nodes are concepts (represented by *synsets* (i.e., synonym sets)) and whose edges are semantic relations between concepts (e.g., *hyperonymy*, *meronymy*). For each sentence, a graph $G = (V, E)$ is built, which is derived from the entire graph of the reference lexicon. More formally, given a sentence $\sigma = w_1, w_2, \ldots, w_n$, where $w_i$ is a word, the following steps are executed to build $G$: (1) the sense vocabulary $V_\sigma$ is derived as $V_\sigma := \bigcup_{i=1}^{n} Senses(w_i)$, where $Senses(w_i)$ is the set of senses of any of the $w_i$ of the sen-

tence. (2) For each node $v \in V_\sigma$, a visit of the WordNet graph is performed: every time a node $v' \in V_\sigma (v' \neq v)$ is encountered along a path $v \rightarrow v_1 \rightarrow \ldots \rightarrow v_k \rightarrow v'$ all intermediate nodes and edges on the path from $v$ to $v'$ are added to the graph: $V := V \bigcup \{v_1, \ldots, v_k\}$ and $E := E \bigcup \{(v, v_1), \ldots, (v_k, v')\}$. The constructed graph is the subgraph covering the nodes and relations of all the relevant vocabulary in the sentence.

**Sense Ranking**. The derived graph is then used with different ranking models to find the correct senses of words into the sentence $\sigma$. A suitable interpretation of the source sentence can be in fact obtained by ranking each vertex in the graph $G$ according its centrality. In (Navigli and Lapata, 2007) different ranking models are described. The specific algorithm presented in (Agirre and Soroa, 2008) is the major inspiration of the present paper, and makes use of PageRank (Brin and Page, 1998) to rank edges in the graph $G$. PageRank tries to separate these nodes from the other candidate synsets of words in $\sigma$, which are expected to activate less relations on average and remain isolated. Let the vector $\vec{Rank}$ express the probability to reach any of the vertices $V_\sigma$, and let $M$ represent the edge information. The expected rank between senses satisfies:

$$\vec{Rank} = (1 - \alpha)M \times \vec{Rank} + \alpha\vec{p} \qquad (1)$$

whereas $0 \leq \alpha \leq 1$. $\alpha$ is called the *damping factor*. It models the amount of likelihood that a generic Web surfer, standing at a vertex, randomly follows a link from this vertex toward any other vertex in the graph: the uniform probability $p_i = \frac{1}{N} \quad \forall i$, is assigned to each one of the $N$ vertices in $G$. While it guarantees the convergence of the algorithm, it expresses the trade-off between the probability of following links provided by the Web graph and the freedom to violate them. An interesting aspect of the ranking process is the initial state. Many algorithms (as well as the one proposed by (Agirre and Soroa, 2009)) initialize the ranks of the vertex at a uniform value (usually $1/N$ for a graph with $N$ vertices). Then Equation 1 is iterated until convergence is achieved or a maximum fix number of iterations has been reached.

**Disambiguation**. Finally, the disambiguation step is performed by assigning to each word $w_i$ in the source sentence $\sigma$, the associated $j$-th concept $sense_{ij}$ (i.e. the $j$-th valid interpretation for $w_i$) associated to the maximum resulting rank. In case of ties all the concepts with maximum rank are assigned to $w_i \in \sigma$.

The above process has several sources of complexity, but the major burden is related to the *Sense ranking* step. While complex methods have been proposed (as discussed in (Navigli and Lapata, 2007)), sentence oriented algorithms, that build the graph $G$ once per each sentence $\sigma$, whatever the number of $w_i \in \sigma$ is, are much more efficient. The problem is twofold:

- How different sentences can be targeted without major changes in the graph $G$? How the matrix $M$ can be made as much reusable as possible?
- How to encode in Eq. 1 the incoming context in order to properly address the different words in the sentence $\sigma$?

In order to address the above problems, in line with the notion of topic-sensitive PageRank, a personalized PageRank approach has been recently devised (Agirre and Soroa, 2009) as discussed in the next section.

### 2.1 Personalizing PageRank for WSD

In (Agirre and Soroa, 2009), a novel use of PageRank for word sense disambiguation is presented. It aims to present an optimized version of the algorithm previously discussed in (Agirre and Soroa, 2008). The main difference concerns the method used to initialize and use the graph $G$ for disambiguating a sentence with respect to the overall graph (hereafter $GKB$) that represents the complete lexicon.

Previous methods (such as (Agirre and Soroa, 2008)) derive $G$ as the subgraph of $GKB$ whose vertices and edges are particularly relevant for the given input sentence $\sigma$. Such a subgraph is often called the *disambiguation subgraph* $\sigma$, $GD(\sigma)$. $GD$ is a subgraph of the original $GKB$, obtained by computing the shortest paths between the concepts of the words co-occurring in the context. These are expected to capture most of the information relevant to the disambiguation (i.e. sense ranking) step.

The alternative proposed in (Agirre and Soroa, 2009) allows a more static use of the full LKB. Context words are newly introduced into the graph $G$ as nodes, and linked with directed edges (i.e. the lexical relations) to their respective concepts (i.e. synsets). Topic-sensitive PageRank over the graph $G$ (Haveliwala, 2002) is then applied: the initial probability mass is concentrated uniformly

over the newly introduced word nodes through the setting of the personalization vector $\vec{p}$ in Eq. 1 (Haveliwala, 2002). Words are linked to the concepts by directed edges that act as sources to propagate probability into the $GKB$ concepts they are associated with. A personalized PageRank vector is finally produced that defines a measure of the (topological) relevance of the GKB nodes (concepts) activated by the input context. The overall time complexity is limited by the above sketched *Personalized PageRank* approach ($PPR$) as a single initialization of the graph $GKB$ is requested for an entire target sentence. This *sentence oriented* method reuses the $GKB$ of the entire lexicon, while the second step runs the sense ranking once for all the words. This method reduces the number of invocations of PageRank thus lowering the average disambiguation time.

A *word oriented* version of the algorithm is also proposed in (Agirre and Soroa, 2009). It defines different initializations for the different words $w_i \in \sigma$: these are obtained by setting the initial probability mass in $\vec{p}$ to 0 for all the senses $Sense(w_i)$ of the targeted $w_i$. In this way, only the context words and not the target are used for the personalization step[1]. This approach to the personalized PageRank is termed word-by-word or $PPRw2w$ version in (Agirre and Soroa, 2009). $PPRw2w$ is run on the same graph but with $n$ different initializations where $n$ is the number of words in $\sigma$. Although less efficient, $PPRw2w$ is shown to outperform the sentence oriented $PPR$ model.

## 3 A distributional extension of PageRank

The key idea in (Agirre and Soroa, 2009) is to adapt the matrix initialization step in order to exploit the available contextual evidence. Notice that personalization in Word Sense Disambiguation is inspired by the topic-sensitive PageRank approach, proposed in (Haveliwala, 2002), for Web search tasks. It exploits a context dependent definition of the vector $\vec{p}$ in Eq. 1 to influence the link-based sense ranking achievable over a sentence. Context is used as only words of the sentence (or words co-occurring with the target $w_i$ in the $w2w$ method) are given non zero probability mass in $\vec{p}$: this provides a $topical$ bias to PageRank. A variety of models of topical information have been proposed in IR (e.g. (Landauer and Dumais, 1997)) to generalize documents or shorter texts (e.g. query). They can be acquired through large scale corpus analysis in the so called distributional approaches to language modeling. While *context*s can be defined in different ways (e.g as the set of words surrounding a target word), their analysis over large corpora has been shown to effectively capture topical and paradigmatic relations (Sahlgren, 2006). We propose to use the topical information about a sentence $\sigma$, acquired through Latent Semantic Analysis (Landauer and Dumais, 1997), as a source information for the initialization of the vector $\vec{p}$ in the $PPR$ (or $PPRw2w$) disambiguation methods.

SVD usually improves the word similarity computation for three different reasons. First, SVD tends to remove the random noise present in the source matrix. Second, it allows to discover the latent meanings of a target word through the corpus, and to compute second-order relations among targets, thus improving the similarity computation. Third, similarities are computed in a lower dimensional space, thus speeding up the computation. For the above reasons by mapping a word, or a sentence, in the corresponding Latent Semantic Space, we can estimate the set of its similar words according to implicit semantic relations acquired in an unsupervised fashion. This can be profitably used as a personalization model for $PPR$.

For the WSD task, our aim is to exploit an externally acquired semantic space to expand the incoming sentence $\sigma$ into a set of *novel* terms, different but *semantically related* with the words in $\sigma$. In analogy with topic-driven PageRank, the use of these words as a seed for the iterative algorithm is expected to amplify the effect of local information (i.e. $\sigma$) onto the recursive propagation across the lexical network: the interplay of the global information provided by the whole lexical network with the local information characterizing the initialization lexicon is expected to maximize their independent effect.

More formally, let the matrix $W_k := U_k S_k$ be the matrix that represents the lexicon in the $k$-dimensional LSA space. Given an input sentence $\sigma$, a vector representation $\overrightarrow{w_i}$ for each term $w_i$ in $\sigma$ is made available. The corresponding representation of the sentence can be thus computed as the

[1] This seems to let the algorithm to avoid strong biases toward pairs of senses of a given word that may appear in some semantic relations (thus connected in the graph), that would be wrongly emphasized by the PPR method.

linear combination through the original $tf \cdot idf$ scores of the corresponding $\overrightarrow{w_i}$: this provides always an unique representation $\overrightarrow{\sigma}$ for the sentence. $\overrightarrow{\sigma}$ locates the sentence in the LSA space and the set of terms that are *semantically related* to the sentence $\sigma$ can be easily found in the neighborhood. A lower bound can be imposed on the cosine similarity scores over the vocabulary to compute the lexical expansion of $\sigma$, i.e. the set of terms that are enough similar to $\overrightarrow{\sigma}$ in the $k$ dimensional space. Let $D$ be the vocabulary of all terms, we define as the lexical expansion $T(\sigma) \subset D$ of $\overrightarrow{\sigma}$ as follows:

$$T(\sigma) = \{w_j \in D : sim(\overrightarrow{w_j}, \overrightarrow{\sigma}) > \tau\} \quad (2)$$

where $\tau$ represents a real-valued threshold in the set $[0, 1)$. In order to improve precision it is also possible to impose a limit on the cardinality of $T(\sigma)$ and discard terms characterized by lower similarity factors.

Let the $t = |T(\sigma)|$ be the number of terms in the expansion, we extend the original set $\sigma$ of terms in the sentence, so that the new seed vocabulary is $\sigma \cup T(\sigma) = \{w_1, w_2, \ldots, w_n, w_{n+1}, \ldots, w_{n+t}\}$. The nodes in the graph $G$ will be thus computed as $Vext_\sigma := \bigcup_{i=1}^{n+t} Senses(w_i)$ and a new personalization vector $\vec{p_{ext}}$ will then replace $\vec{p}$ in Eq. 1: it will assign a probability mass to the words $w_1, ..., w_{n+t}$ proportional to their similarity to $\overrightarrow{\sigma}$, i.e.

$$p_{k_i} = \frac{sim(\overrightarrow{w_i}, \overrightarrow{\sigma})}{\sum_{j=1}^{n+t} sim(\overrightarrow{w_j}, \overrightarrow{\sigma})} \quad \forall i = 1, ..., n+t \quad (3)$$

whereas $k_i$ is the index of the node corresponding to the word $w_i$ in the graph. Finally, the later steps of the PPR methods remain unchanged, and the PageRank works over the corresponding graph $G$ are carried out as described in Section 2.

## 4 Empirical Evaluation

The evaluation of the proposed model was focused on two main aspects. First we want to measure the impact of the topical expansion at sentence level on the accuracy reachable by the personalized PageRank PPR. This will be done also comparatively with the state of the art of unsupervised systems over a consolidated benchmark, i.e. Semeval 2007. In Table 1 a comparison between the official Semeval 2007 results for unsupervised methods is reported. Table 1 shows also the results of the standard PPR methods over the Semeval 2007 dataset. Second, we want to analyze the efficiency of the algorithm and its impact in a sentence (i.e. $PPR$) or word oriented (i.e. $w2w$) perspective. This will allow to asses its applicability to realistic tasks, such as query processing or document indexing.

**Experimental Set-up** In order to measure accuracy, the Senseval 2007 coarse WSD dataset[2] (Navigli et al., 2007) has been employed. It includes 245 sentences for a total number of 2,269 ambiguous words. In line with the results reported in (Agirre and Soroa, 2009), experiments against two different WordNet versions, 1.7 and 3.0, have been carried out. Notice that the best results in (Agirre and Soroa, 2009) were obtained over the enriched version of the LKB, i.e. the combination of WordNet and extra information supplied by *extended WordNet* (Harabagiu and Moldovan, 1999).

The adopted vector space has been acquired over a significant subset of the BNC 2.0 corpus, made of 923k sentences. The most frequent 200k words (i.e. the contextual features) were acquired through LSA. The corpus has been processed with the LTH parser (Johansson and Nugues, 2007) to obtain POS tags for every token. Moreover, a dimensionality reduction factor of $k = 100$ was applied.

In subsection 4.1, a comparative analysis of the accuracy achieved in the disambiguation task is discussed. Subsection 4.2 presents a corresponding study of the execution times aiming to compare the relative efficiency of the methods and their application into a document semantic tagging task.

### 4.1 Comparative evaluation: accuracy on the Semeval '07 data

The approaches proposed in Semeval 2007 can be partitioned into two major types. The supervised or semi-supervised approaches and the unsupervised ones that rely usually on WordNet. As the basic *Page Rank* as well as our LSA extension makes no use of sense labeled data, we will mainly focus on the comparative evaluation among unsupervised WSD systems. In order to compare the quality of the proposed approach, the results of the personalized PageRank proposed in (Agirre and Soroa, 2009) over the same dataset are reported in Table 1 (The * systems, denoted by UKB). As also suggested in (Agirre and Soroa, 2009) the best per-

[2]The dataset is publicly available from http://nlp.cs.swarthmore.edu/semeval/tasks/task07/data.shtml

| System | P | R | F1 |
|---|---|---|---|
| *LSA_UKB_1.7x* | 71.66 | 71.53 | **71.59** |
| UKB_1.7x * | 71.38 | 71.13 | 71.26 |
| TKB-UO | 70.21 | 70.21 | 70.21 |
| UKB_3.0g * | 68.47 | 68.05 | 68.26 |
| *LSA_UKB_3.0g* | 67.02 | 66.73 | 66.87 |
| *LSA_UKB_1.7* | 66.96 | 65.66 | 66.31 |
| *LSA_UKB_3.0* | 66.60 | 65.31 | 65.95 |
| RACAI-SYNWSD | 65.71 | 65.71 | 65.71 |
| UKB_3.0 * | 63.29 | 61.92 | 62.60 |
| SUSSX-FR | 71.73 | 52.23 | 60.44 |
| UKB_1.7 * | 59.30 | 57.99 | 58.64 |
| UOFL | 52.59 | 48.74 | 50.60 |
| SUSSX-C-WD | 54.54 | 39.71 | 45.96 |
| SUSSX-CR | 54.30 | 39.53 | 45.75 |

Table 1: Official Results over the Semeval'07 dataset. The * systems was presented in (Agirre and Soroa, 2009). The *LSA_UKB_1.7* and *LSA_UKB_3.0* show the rank of the model proposed in this paper.

formances are obtained according to the *PPRw2w* word oriented approach.

For sake of comparison we applied the LSA-based expansion to the Personalized Page Rank in a sentence oriented fashion (i.e., only one PageRank is run for all the target words of a sentence, $PPR$). Notice that $PPR$ models the context of the sentence with a single iterative run of PageRank, while $PPRw2w$ disambiguates each word with a dedicated PageRank. In line with (Agirre and Soroa, 2009), different types of WordNet graphs are employed in our experiments:

**WN17** all hyponymy links between synsets of the WN1.7 dictionary are considered;

**WN17x** all hyponymy links as well as the extended 1.7 version of WordNet, whereas the syntactically parsed glosses, are semantically disambiguated and connected to the corresponding synsets;

**WN3.0** all hyponymy links between synsets of the WN3.0 dictionary are considered;

**WN30g** all hyponymy links as well as the extended 3.0 version of WordNet, whereas the syntactically parsed glosses, are semantically disambiguated and connected to the corresponding synsets;

The impact of the LSA sentence expansion technique proposed in this paper on the different involved resources, i.e. WN1.7 to WN30g, has been measured. The 1.7 configuration provides

| Model | Iter. | PPR | | | w2w | | |
|---|---|---|---|---|---|---|---|
| | | Prec | Rec | F1 | Prec | Rec | F1 |
| 17_LSA100 | 5 | 65.8 | 64.5 | **65.2** | 65.7 | 64.4 | **65.1** |
| | 15 | 65.6 | 64.3 | **65.0** | 66.3 | 65.0 | **65.7** |
| 17_UKB | 5 | 60.9 | 59.7 | 60.3 | 65.3 | 63.8 | 64.5 |
| | 15 | 61.3 | 60.1 | 60.7 | 61.6 | 60.2 | 60.9 |
| 17x_LSA100 | 5 | 71.5 | 71.4 | **71.5** | 71.1 | 71.0 | **71.1** |
| | 15 | 71.5 | 71.4 | **71.4** | 71.6 | 71.5 | **71.5** |
| 17x_UKB | 5 | 67.4 | 67.3 | 67.4 | 70.9 | 70.6 | 70.7 |
| | 15 | 67.5 | 67.4 | 67.5 | 71.3 | 71.1 | 71.2 |
| 30_LSA100 | 5 | 66.5 | 65.2 | **65.8** | 65.7 | 64.4 | **65.1** |
| | 15 | 66.9 | 65.6 | **66.2** | 66.6 | 65.3 | **65.9** |
| 30_UKB | 5 | 61.7 | 60.5 | 61.1 | 64.7 | 63.3 | 64.0 |
| | 15 | 63.5 | 62.2 | 62.8 | 63.2 | 61.9 | 62.6 |
| 30g_LSA100 | 5 | 66.6 | 66.3 | **66.4** | 66.6 | 66.3 | 66.5 |
| | 15 | 66.7 | 66.4 | **66.5** | 67.0 | 66.7 | 66.8 |
| 30g_UKB | 5 | 60.8 | 60.5 | 60.6 | 68.1 | 67.7 | **67.9** |
| | 15 | 60.7 | 60.5 | 60.6 | 68.4 | 68.0 | **68.2** |

Table 2: Accuracy of the LSA-based sentence expansion PageRank model, as compared with the sentence ($PPR$) and word oriented ($w2w$) versions of the personalized PageRank over the Semeval 2007 datasets. 17x and 30g refer to the extended resources of WordNet 1.7 and 3.0, respectively.

the most efficient one as it runs the original PPR against a graph built around the only hyponymy relations among synsets. We used the Senseval'02 and Senseval'03 datasets to fine tune parameters of our LSA model, that are: (1) the dimensionality cut $k$ to derive the LSA space; (2) the threshold $\tau$ to determine the expansion dictionary in the LSA space for every POS tag (e.g. noun or adjectives), that may require different values; (3) the damping factor $\alpha$ and (4) the number of iteration over the graph. In (Agirre and Soroa, 2009) the suggested parameters are $\alpha = 0.85$ as the damping factor and 30 as the upper limit to the PageRank iterations. We always adopted this setting to estimate the performances of the standard $PPR$ and $PPRw2w$ algorithms referred through $UKB$. Due the novel configuration of the graph that in our model also includes many other similar terms, the damping factor and the number of iterations have been re-estimated. $k$ has been set to 100 as different values did not seem to influence accuracy. We adopted fixed limits for sentence expansion where values from 20 up to 150 terms have been tested. The good scores obtained on the development set suggested that a number of iterations lower than 30 is in general enough to get good accuracy levels: 15 iterations, instead of 30, have been judged adequate. Finally, on average, the total number of lexical items in the expanded sentence $T(\sigma)$ includes about 40% of nouns, 30% of verbs, 20% of adjectives and 10% of adverbs.

Finally, a damping factor $\alpha = 0.98$ has been used.

Table 2 reports Precision, Recall and F1 scores of the different models as obtained over the test SemEval '07 data. Every row pair compares the LSA model with the original corresponding UKB version over a given graph (from WN1.7 to WN30g). For each model the accuracy corresponding to two iterations (5 and 15) is reported to analyze also the overall trend during PageRank. The best F1 scores between any pair are emphasized in bold, to comparatively asses the results. As a confirmation of the outcome in (Agirre and Soroa, 2009), different lexical resources achieve different results. In general by adopting the graph derived from WN3.0 (i.e. WN30 and WN30g) lower performance can be achieved. Moreover, the word-by-word model (last three columns for the w2w side of the Table) is evidently superior. Interestingly, almost on every type of graph and for every approach (sentence or word oriented) the LSA-based method outperforms the original UKB PPR. This confirms that the impact of the topical information provided by the LSA expansion of the sentence is beneficial for a better use of the lexical graph. An even more interesting outcome is that the improvement implied by the proposed LSA method on the sentence oriented model (i.e. the standard PPR method of (Agirre and Soroa, 2009)) is higher, so that the difference between the performances of the $PPRw2w$ model are no longer strikingly better than the $PPR$ one. For example, on the simple WN1.7 hyponymy network the $PPR - LSA100$ [3] method abolishes the gap of about 4% previously observed for the PPR-UKB model. When LSA is used, it seems that the word-by-word approach is no longer required. On the contrary, in the WN17x case the best figure after 5 iterations is obtained by the PPR-LSA100 method instead of the w2w-LSA100 one (71.5% vs. 71.1%). The good accuracy reachable by the sentence oriented strategy (i.e. LSA100 and w2w) is also very interesting as for the higher efficiency of the PPR approach with respect to the word-by-word $PPRw2w$ one.

### 4.2 Time Efficiency

In the attempt to validate the hypothesis that LSA is helpful to improve time complexity of the WSD, we analyzed the processing times of the different data sets, in order to cross compare methods and resources[4]. The aim of the evaluation is to study the contribution of the sentence expansion using Latent Semantic Analysis and the Page Rank algorithm. Tests were performed comparing different parameter values (e.g. cardinality $t$ of the sentence expansion, different values for the acceptability threshold) as well as several settings of the damping factor for the personalized PageRank algorithm (Eq 1) and the number of iterations over the KB Graph. In figure 1, the processing speed, measured as seconds per sentence, has been plot for different graphs and configurations. Notice that one sentence is equivalent on average to 9,6 target words. As clearly shown in the figure, the processing times for the word-by-word method over the extended WN 1.7 (i.e. WN17x) are not acceptable for IR tasks such as query processing, or document indexing. For an entire document of about 20 sentences the overall amount of processing required by the w2w_17x_UKB method is about 45 minutes. Word-by-word methods are just slightly more efficient whenever applied to graphs with lower connectivity (e.g. WN17 vs. WN17x as in Fig. 1 left plot). The same tasks with PPR methods are solved in a quite faster way, with a general ratio of 1:14 with the extended versions and 1:6 with the hyponymy graphs. The processing time of the proposed LSA method is thus at least 6 times faster than the UKB method with the comparable accuracy level. Moreover, as accuracy between PPR and w2w is comparable when LSA is adopted, this efficiency can be guaranteed at no loss in accuracy. By integrating the evidence of Figure 1 with the ones of Table 1, we observe that accuracy reachable by LSA-UKB is independent by the standard or word-by-word configuration so that the overall process can be made about 10 times faster. Notice that the representation in the LSA space that is projected for a target sentence can be easily obtained also for longer text fragments. Moreover, as for the *one sense per discourse* hypothesis it is possible that every word can be processed once in an entire text. This suggests that a document oriented usage of the personalized PageRank based on LSA can be designed achieving the maximal efficiency. In order to evaluate the corresponding impact on accuracy a dedicated dataset has been defined and more tests have been run, as discussed hereafter.

[3] 100 refers to the dimension $k$ of the LSA space

[4] Tests were carried out on a 32-bit machine with a 3.2 Ghz CPU and 2 Gbyte Memory. Gnu/Linux operative system is installed on it, with the kernel 2.6.28-16-generic.

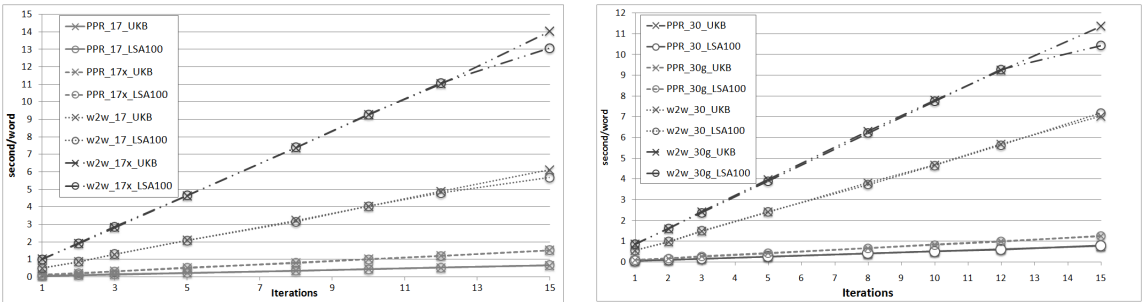


Figure 1: Processing Times for the $PPR$, $w2w$ and LSA methods as applied on the WN 1.7 (left plot) and WN 3.0 (right plot) resources, respectively: 17x and 30g refer to test over the extended resources.

### 4.3 Document oriented PPR

While the LSA model has been actually applied to determine an expansion for the entire target sentence, nothing prevents to apply it to larger text units, in order to bias the PageRank for all words in a document. In order to verify if such a process disambiguation could preserve the same accuracy, we measured the accuracy reachable over the same Semeval'07 data organized in documents. The sentences have been grouped in 5 documents, made of about about 250 sentences: during the tagging process, the system generates a lexical expansion for an entire document, about 450 target words on average. Then PageRank is carried out and the resulting ranking is projected to the senses of all the targeted words in the document. Due to the much wider locality managed in this process, a larger cardinality for the expansion is used and the most similar 400 words are collected as a bias for the PageRank. The accuracy reachable is reported in Table 4.3. As expected, the same trends as for the sentence based approach are observed: the best resource is still the WN17x for which the best results is obtained. However, the crucial result here is that no drop in performance is also observed. This implies that the much more efficient document oriented strategy can be always applied through LSA without major changes in accuracy. Also results related to the processing time follow the trends of the sentence based method. Accordingly 28 seconds required to process a document in the worst case is an impressive achievement because the same accuracy was obtained, without LSA, in 2 orders of magnitude more time.

| Model | Iter. | Prec | Rec | F1 |
|---|---|---|---|---|
| PPR_17_LSA400 | 5 | 0.6670 | 0.6540 | **0.6604** |
| | 15 | 0.6800 | 0.6668 | **0.6733** |
| PPR_17_UKB | 5 | 0.6440 | 0.6316 | 0.6377 |
| | 15 | 0.6360 | 0.6236 | 0.6297 |
| PPR_17x_LSA400 | 5 | 0.7130 | 0.7118 | **0.7124** |
| | 15 | 0.7152 | 0.7140 | **0.7146** |
| PPR_17x_UKB | 5 | 0.7108 | 0.7096 | 0.7102 |
| | 15 | 0.7073 | 0.7060 | 0.7067 |
| PPR_30_LSA400 | 5 | 0.6593 | 0.6465 | **0.6529** |
| | 15 | 0.6688 | 0.6558 | 0.6622 |
| PPR_30_UKB | 5 | 0.6445 | 0.6320 | 0.6382 |
| | 15 | 0.6724 | 0.6593 | **0.6658** |
| PPR_30g_LSA400 | 5 | 0.6636 | 0.6606 | **0.6621** |
| | 15 | 0.6653 | 0.6624 | **0.6639** |
| PPR_30g_UKB | 5 | 0.6543 | 0.6514 | 0.6528 |
| | 15 | 0.6565 | 0.6536 | 0.6550 |

Table 3: Accuracy of the LSA-based $PPR$ model when applied in a document oriented fashion on the Semeval '07 dataset. LSA400 stands for the size $t$ of the applied sentence expansion $T(\sigma)$.

## 5 Conclusions

In this paper an extension of a PageRank-based algorithm for Word Sense Disambiguation has been presented. It suggests a kind of personalization based on sentence expansion, obtained as a side effect of Latent Semantic Analysis. The major results achieved are in terms of improved efficiency that allows to use smaller resources or less iterations with similar accuracy results. The resulting speed-up can be also improved when the disambiguation is run in a document oriented fashion, and the PageRank is run once per each document. The overall results can achieve a speed-up of two order of magnitude at no cost in accuracy. Moreover the presented approach constitutes the state-of-the-art among the unsupervised WSD algorithms over the Semeval'07 datasets, while improving the efficiency of the PPR method by a factor 10 in the worst case. This work opens perspectives towards more sophisticated distributional models (such as syntax-driven ones) as well as cross-linguistic applications supported by multilingual lexical sense repositories.

## References


E. Agirre and G. Rigau. 1996. Word sense disambiguation using conceptual density. In *Proceedings of COLING-96*, Copenhagen, Denmark.

Eneko Agirre and Aitor Soroa. 2008. Using the multilingual central repository for graph-based word sense disambiguation. In *Proceedings of the LREC'08*, Marrakech, Morocco, May.

E. Agirre and A. Soroa. 2009. Personalizing pagerank for word sense disambiguation. In *Proceedings of the 12th conference of EACL '09*, Athens, Greece, March 30 - April 3.

R. Basili, M. Cammisa, and F.M. Zanzotto. 2004. A semantic similarity measure for unsupervised semantic disambiguation. In *Proceedings of LREC-04*, Lisbon, Portugal.

Stephen Beale, Benoit Lavoie, Marjorie McShane, Sergei Nirenburg, and Tanya Korelsky. 2004. Question answering using ontological semantics. In *TextMean '04: Proceedings of the 2nd Workshop on Text Meaning and Interpretation*, pages 41–48, Morristown, NJ, USA. Association for Computational Linguistics.

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117.

M. Carpuat and D. Wu. 2007. Improving statistical machine translation using word sense disambiguation. In *Proceedings of the Joint Conference EMNLP-CoNLL '09*, Prague, Czech Republic.

Y. Chan, H. Ng, and D. Chiang. 2007. Word sense disambiguation improves statistical machine translation. In *Proceedings of the ACL '09*, Prague, Czech Republic.

Jim Cowie, Louise Guthrie, and Joe Guthrie. 1992. Lexical disambiguation using simulated annealing. In *Proc. of 14th Int. Conf. COLING '92*, pages 359–365, Nantes, France.

Sanda M. Harabagiu and Dan I. Moldovan. 1999. Enriching the wordnet taxonomy with contextual knowledge acquired from text. In *in Iwanska, L.M., and Shapiro, S.C. eds 2000. Natural Language Processing and Knowledge Representation: Language*, pages 301–334. AAAI/MIT Press.

T. H. Haveliwala. 2002. Topic-sensitive pagerank. In *Proc. of 11th Int. Conf. on World Wide Web*, page 517526, New York, USA. ACM.

Richard Johansson and Pierre Nugues. 2007. Semantic structure extraction using nonprojective dependency trees. In *Proceedings of SemEval-2007*, Prague, Czech Republic, June 23-24.

S. B. Kim, H. Seo, and H. Rim. 2004. Information retrieval using word senses: root sense tagging approach. In *Proceedings of the International ACM-SIGIR Conference '09*, Sheffield, UK, July.

H. Krovetz. 1997. Homonymy and polysemy in information retrieval. In *Proceedings of the 35th ACL '09*.

Tom Landauer and Sue Dumais. 1997. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 104:211–240.

M. Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *SIGDOC '86: Proceedings of the 5th annual international conference on Systems documentation*, New York, NY, USA.

G. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. 1990. An on-line lexical database. *International Journal of Lexicography*, 13(4):235–312.

Alessandro Moschitti and Roberto Basili. 2004. Complex linguistic features for text classification: A comprehensive study. In *Proc. of the European Conf. on IR, ECIR*, pages 181–196, New York, USA.

Roberto Navigli and Mirella Lapata. 2007. Graph connectivity measures for unsupervised word sense disambiguation. In *Proceedings of IJCAI'07*, pages 1683–1688, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Roberto Navigli, Kenneth C. Litkowski, and Orin Hargraves. 2007. Semeval-2007 task 07: coarse-grained english all-words task. In *SemEval '07*, pages 30–35, Morristown, NJ, USA. Association for Computational Linguistics.

M. Palmer, C. Fellbaum, S. Cotton, L. Delfs, and H.T. Dang. 2001. English tasks: All-words and verb lexical sample. In *Proceedings of SENSEVAL-2*, Tolouse, France, July.

S. Pradhan, E. Loper, D. Dligach, and M. Palmer. 2007. Semeval-2007 task-17: English lexical sample srl and all words. In *Proceedings of SemEval-2007*, Prague, Czech Republic, June.

Magnus Sahlgren. 2006. *The Word-Space Model*. Department of Linguistics, Stockholm University.

Ravi Sinha and Rada Mihalcea. 2007. Unsupervised graph-based word sense disambiguation using measures of word semantic similarity. In *IEEE ICSC 2007*.

B. Snyder and M. Palmer. 2004. The english all-words task. In *Proceeding of ACL 2004 Senseval-3 Workshop*, Barcelona, Spain, July.