Word sense discrimination in information retrieval: a spectral clustering-based approach

Adrian-Gabriel Chifu, Florentina Hristea, Josiane Mothe, Marius Popescu

Word sense discrimination in information retrieval: A spectral clustering-based approach

Adrian-Gabriel Chifu [a], Florentina Hristea [b], Josiane Mothe [c], Marius Popescu [b]

[a] IRIT UMR5505, CNRS, Université de Toulouse, Université Paul Sabatier, 118 Route de Narbonne, F-31062 TOULOUSE CEDEX 9, France
[b] University of Bucharest, Faculty of Mathematics and Computer Science, Department of Computer Science, Academiei 14, RO-010014 Bucharest, Romania
[c] IRIT UMR5505, CNRS, Université de Toulouse, Ecole Supérieure du Professorat et de l'Education, 118 Route de Narbonne, F-31062 TOULOUSE CEDEX 9, France

Keywords:
Information retrieval
Word sense disambiguation
Word sense discrimination
Spectral clustering
High precision

A B S T R A C T

Word sense ambiguity has been identified as a cause of poor precision in information retrieval (IR) systems. Word sense disambiguation and discrimination methods have been defined to help systems choose which documents should be retrieved in relation to an ambiguous query. However, the only approaches that show a genuine benefit for word sense discrimination or disambiguation in IR are generally supervised ones. In this paper we propose a new unsupervised method that uses word sense discrimination in IR. The method we develop is based on spectral clustering and reorders an initially retrieved document list by boosting documents that are semantically similar to the target query. For several TREC ad hoc collections we show that our method is useful in the case of queries which contain ambiguous terms. We are interested in improving the level of precision after 5, 10 and 30 retrieved documents (P@5, P@10, P@30) respectively. We show that precision can be improved by 8% above current state-of-the-art baselines. We also focus on poor performing queries.

1. Introduction

According to Lin (1997), "given a word, its context and its possible meanings, the problem of word sense disambiguation (WSD) is to determine the meaning of the word in that context".[1] Although WSD is generally easy for humans, it represents an issue for computers. The problem becomes even more difficult to solve when an ambiguous word occurs in short chunks of texts, such as a query in an information retrieval (IR) system.

Applying WSD to improve IR results is a well studied problem, but with controversial results as evidenced in the literature. Several authors have concluded that WSD in IR does not lead to significant retrieval performance improvement (Guyot, Falquet, Radhouani, & Benzineb, 2008; Sanderson, 1994). Various studies (Krovetz & Croft, 1992; Uzuner, Katz, & Yuret, 1999; Voorhees, 1993) have argued that the main problem in improving retrieval performance when using WSD is the inefficiency of the existing disambiguation algorithms, a problem which increases in the case of short queries.

In more recent years the issue remained "as to whether less than 90% accurate automated WSD can lead to improvements in retrieval effectiveness" (Stokoe, Oakes, & Tait, 2003). This remark refers primarily to the traditional task of WSD which

[1] For a complete discussion of state-of-the-art WSD see the monograph (Agirre & Edmonds, 2006).

http://dx.doi.org/10.1016/j.ipm.2014.10.007

identifies the meaning of the ambiguous word in context. This type of WSD is generally based on external sources, such as dictionaries or WordNet(WN)-like knowledge bases for labeling senses (Carpineto & Romano, 2012; Guyot et al., 2008) and is therefore knowledge-based.

Attempts to use knowledge-based WSD in IR have been numerous. In (Gonzalo, Verdejo, Chugur, & Cigarran, 1998) as well as in (Mihalcea & Moldovan, 2000) positive results were reported. These studies made use of semantic indexing based on WN synsets. However, they were all conducted on small data sets. As commented in (Ng, 2011), the evaluation is scaled up to a large test collection in (Stokoe et al., 2003) but the reported improvements are from a weak baseline. Positive results are also reported in (Kim, Seo, & Rim, 2004), although the quantum of improvements is small.

Zhong and Ng (2012) are among the few authors who more recently have expressed a growing belief in the benefits brought by WSD to IR – when using a supervised WSD technique. They constructed their supervised WSD system directly from parallel corpora. Experimental results on standard TREC collections show that, using the word senses tagged by this supervised WSD system, significant improvements over a state-of-the-art IR system can be obtained (Zhong & Ng, 2012). However, it is well known that supervised WSD cannot be used on a large scale in practice due to the absence of the necessary annotated/parallel corpora.

In contrast to all these authors, we are suggesting and investigating the usage of an unsupervised WSD technique. In this paper, we present an approach that aims at identifying clusters from similar contexts, where each cluster shows a polysemous word being used for a particular meaning. It is our belief that IR is an application for which this type of analysis is useful. Our approach is therefore not concerned with performing a straightforward WSD, but rather with differentiating among the meanings of an ambiguous word. Considering word sense discrimination rather than straightforward WSD avoids the use of external sources such as dictionaries or WN type synsets which are commonly used (Carpineto & Romano, 2012).

In this paper, we propose a new word sense discrimination method for IR based on spectral clustering. This state of the art clustering technique is now a hot topic; for example, Takacs and Demiris (2009) studied the use of spectral clustering in multi-agent systems while Borjigin and Guo (2012) recently discussed the cluster number determination in spectral clustering. Spectral clustering has been used in WSD for the first time by Popescu and Hristea (2011) who point out the importance of the clustering method used in unsupervised WSD.

We hereby show that WS discrimination based on spectral clustering outperforms the baseline when no WS discrimination is applied and also when using another unsupervised method (Naïve Bayes).

The present paper is organized as follows: in Section 2 we present the related works on WSD in IR; the focus is on unsupervised methods. Section 3 presents word sense discrimination based on spectral clustering. Section 4 presents the two step IR process using the proposed WS discrimination model. The evaluation is presented in Section 5. A more thorough analysis of the obtained results is performed in Section 6. Section 7 lays out the impact of automatically generated context on our proposed method. Section 8 concludes this paper.

2. Related work

Word sense ambiguity is a central concern in natural language processing (NLP). SENSEVAL defined the first evaluation framework for word sense disambiguation (WSD) in NLP (Kilgarriff, 1997). According to Kilgarriff and Rosenzweig (2000), SENSEVAL participants defined systems that can be classified into two categories: supervised systems, which use training instances of sense-tagged words and non-supervised systems. According to (Navigli, 2009), supervised systems are typically employed when a restricted number of words have to be disambiguated, while this type of system encounters more difficulties when all open-class words from a text have to be disambiguated. In addition to general WSD, many recent papers consider disambiguation of individuals (Artiles, Gonzalo, & Sekine, 2007; D'Angelo, Giuffrida, & Abramo, 2011; Piskorski, Wieloch, & Sydow, 2009) and disambiguation of place names (Leidner, 2007). Indeed, WSD has many applications, such as text processing, machine translation and information retrieval (IR), for which this type of disambiguation – proper names – can be useful (although not sufficient).

Krovetz and Croft (1992) were among the first to conduct a thorough analysis of ambiguity in IR. They used the CACM and TIME test collections and compared query word sense with word senses in retrieved documents. They found that sense mismatch occurs more often when the document is non-relevant to the query and when there are few common words bridging the query and the retrieved document. Another large scale study of word sense disambiguation in IR was conducted by Voorhees (1993). The automatic indexing process she developed used the "is-a" relations from WN and constructed vectors of senses to represent documents and queries. This approach was compared to a stem-based approach for 5 small collections (CACM, CISI, CRAN, MED, TIME). The results showed that the stem-based approach was superior overall, although the sense-based approach improved the results for some queries (Voorhees, 1993). Sanderson (1994) used the Reuters collection in his experiments and showed that disambiguation accuracy should be of at least 90% in order for it to be of practical use. He used pseudo-words in his experiments.

Schütze introduced word sense discrimination in IR (Schütze & Pedersen, 1995; Schütze, 1998). Moreover, Schütze considers that, in some cases, WSD can be defined as a two-stage process: first sense discrimination, then sense labeling. Sense discrimination aims at classifying the occurrences of a word into categories that share the same word sense. This type of approach is quite distinct from the traditional task of WSD, which, as already mentioned, classifies words relative to existing senses. Schütze and Pedersen (1995), Schütze (1998) created a lexical co-occurrence based thesaurus. They associated each

ambiguous term with a word vector where coordinates correspond to co-occurring term frequencies. Words with the same meaning were assumed to have similar vectors. Word vectors were clustered together to determine the word uses. Similarity was based on the cosine measure. The application in IR consisted of modifying the standard word-based vector-space model. The words from the "bag of words" text representation were replaced by word senses. Evaluation of TREC 1 showed that average precision is improved when using sense-based retrieval rather than word-based retrieval. Combining word and sense-based retrieval improves precision as well. They were the first to demonstrate that disambiguation, even if imperfect, can indeed improve text retrieval performance (Schütze & Pedersen, 1995).

Schütze (1998) context group discrimination uses a form of average link clustering known as McQuitty's Similarity Analysis. Schütze adapts LSI/LSA so that it represents entire contexts rather than single word types using second-order co-occurrences of lexical features. The created clusters are made up of contexts that represent a similar or related sense. In Schütze (1998) it is again shown that unlabeled clusters of occurrences of a word representing the same sense result in improved IR.

Unlike that described in Schütze and Pedersen (1995), Schütze (1998), the method we propose in the present paper is based on re-ranking and not on modifying document representation.

Much more recently, Chifu and Ionescu (2012) also show that the combination of word-based ranking and sense-based ranking is beneficial for improving IR performance, but on the lowest precision queries only (Chifu & Ionescu, 2012). They used a classical clustering technique based on the Naïve Bayes model (for which a WN-based feature selection is performed). However, in the case of their best obtained result, the highest difference between this result and the baseline under consideration was 0.1091.

The present paper will investigate a similar type of technique for IR that uses spectral clustering. Our aim is not only to restate the benefits of unsupervised WSD in IR, but also to point out the importance of the clustering technique involved in this task. While Chifu and Ionescu (2012), in spite of performing WN-based feature selection, were not able to move beyond the baseline when considering all queries, and therefore only targeted the lowest precision ones, we hereby show that, when using spectral clustering (that performs its own feature weighting) the same baseline is, in most cases, surpassed.

Analysis of the results provided by the newly proposed method will be carried out (see Section 5.6) against these two major approaches existing in the literature (Chifu & Ionescu, 2012; Schütze & Pedersen, 1995). The obtained results will be shown as promising in sustaining the concept of sense discrimination being beneficial for IR applications, especially when used from a re-ranking perspective.

3. Spectral clustering-based word sense discrimination

Word sense discrimination can be considered as a clustering problem since a way to solve it is to group the contexts of an ambiguous word into a number of groups and to discriminate between these groups without labeling them. As is well known, linguistic data is structurally highly complex, thus turning clustering into a difficult task. Recently, a variety of clustering algorithms have been proposed in order to deal with situations where the data is not linearly separable and the clusters are non-convex. In particular, two related families of methods, kernel and spectral methods, have proven to be very effective in solving different tasks.

In computational linguistics, spectral clustering has been used for machine translation (Gangadharaiah, Brown, & Carbonell, 2006; Zhao, Xing, & Waibel, 2005), name disambiguation for author citation identification (Han, Zha, & Giles, 2005), and in unsupervised WSD (Popescu & Hristea, 2011).

Spectral clustering has been used in WSD for the first time by Popescu and Hristea (2011) who point out the importance of the clustering method used in unsupervised WSD. Spectral clustering has been shown (Popescu & Hristea, 2011) as strong enough to make up for the lack of external knowledge of all types, solving many problems on its own, including that of feature selection for WSD. Disambiguation results, after using an unsupervised algorithm based on spectral clustering (that uses its own feature weighting) were superior to those obtained using a classical unsupervised algorithm (with an underlying Naïve Bayes model, for which feature selection was performed) for all parts of speech (Popescu & Hristea, 2011).

The disambiguation accuracy obtained when using spectral clustering in unsupervised WSD, relative to all parts of speech, encouraged us to adopt this clustering technique for sense discrimination in the context of IR.

3.1. Spectral clustering method

The method of spectral clustering is briefly presented here. For more details and justification of the method the reader is referred to von Luxburg (2007) and Hastie, Tibshirani, and Friedman (2008).

Given a set of observations $x_1, \ldots, x_n$ and some notion of similarity $s_{ij} \geqslant 0$ between all pairs of observations x_i and x_j, the intuitive goal of clustering is to divide the observations into several groups such that observations in the same group are similar and observations in different groups are dissimilar to each other. One possible way to represent the pairwise similarities between observations is via an undirected *similarity graph* $G = (V, E)$. The vertices of the graph represent the observations (the vertex v_i represents the observation x_i). Two vertices are connected if the similarity s_{ij} between the corresponding observations x_i and x_j is positive (or exceeds some threshold). The edges are weighted by the s_{ij} values. The problem of clustering can then be reformulated as a graph-partition problem, where we identify connected components

with clusters. Our intention is to find a partition of the graph such that the edges between different groups have very low weights (which means that observations in different clusters are dissimilar to each other) and the edges within a group have high weights (which means that observations within the same cluster are similar to each other).

An important element in spectral clustering is to construct similarity graphs that reflect the local neighborhood relationships between observations. Starting from a similarity matrix, there are many ways (Maier, Hein, & von Luxburg, 2009; von Luxburg, 2007) to define a similarity graph that reflects local behavior: ε-neighborhood graph, k-nearest neighbor graphs, fully connected graph. One of the most popular graphs and the one that we will use for unsupervised WSD, is the mutual k-nearest-neighbor graph. The vertex v_i is connected to the vertex v_j if, according to the similarity matrix s_{ij}, the observation x_i is among the k-nearest neighbors of the observation x_j or the observation x_j is among the k-nearest neighbors of the observation x_i. The weight of the edge $v_i v_j$ will be $w_{ij} = s_{ij}$ in this case.

In order to formally present the method of spectral clustering we introduce the following notations.

Let $G = (V, E)$ be an undirected graph with vertex set $V = v_1, \ldots, v_n$. In the following we assume that the graph G is weighted, that each edge between two vertices v_i and v_j carries a non-negative weight $w_{ij} \geqslant 0$. The weighted adjacency matrix of the graph is the matrix $W = (w_{ij})_{i,j=1,\ldots,n}$. $w_{ij} = 0$ means that the vertices v_i and v_j are not connected by an edge. As G is undirected, we require that $w_{ij} = w_{ji}$. The degree of a vertex $v_i \in V$ is defined as $d_i = \sum_{j=1}^{n} w_{ij}$. The degree matrix D will be the diagonal matrix with the degrees $d_1, \ldots, d_n$ on the diagonal.

Given a subset of vertices $A \subseteq V$, we denote its complement $V \setminus A$ by $\overline{A}$ and its cardinal by $|A|$. For two not necessarily disjoint sets $A, B \subseteq V$ we define

$$W(A, B) = \sum_{v_i \in A, v_j \in B} w_{ij}. \tag{1}$$

We can now formulate the graph-partition problem in relation to spectral clustering. For a given number k of subsets (clusters) there is a partition of V $A_1, \ldots, A_k$ which minimizes[2]:

$$\text{RatioCut}(A_1, \ldots, A_k) = \frac{1}{2} \sum_{i=1}^{k} \frac{W(A_i, \overline{A_i})}{|A_i|} \tag{2}$$

Unfortunately, the above optimization problem is NP hard (von Luxburg, 2007). Spectral clustering solves a relaxed version of this problem. Relaxing "RatioCut" leads to unnormalized spectral clustering.

The unnormalized graph Laplacian matrix of a similarity graph G is defined as:

$$L = D - W \tag{3}$$

Spectral clustering finds the m eigenvectors $U_{n \times m}$ that correspond to the m smallest eigenvalues of L (ignoring the trivial constant eigenvector corresponding to the eigenvalue 0). Using a standard method like K-means, the rows of U are clustered, giving a clustering of the original observations.

The unnormalized spectral clustering algorithm is summarized in Algorithm 1.

Algorithm 1. Unnormalized spectral clustering algorithm

Input: Similarity matrix $S \in \mathbb{R}^{n \times n}$, number k of clusters to construct.

- Construct a similarity graph in one of the standard ways, for example by using the mutual k-nearest-neighbor graph. Let W be its weighted adjacency matrix.
- Compute the unnormalized Laplacian L.
- Compute the first $k - 1$ eigenvectors $u_1, \ldots, u_{k-1}$ of L corresponding to the $k - 1$ smallest eigenvalues of L (ignoring the trivial constant eigenvector corresponding to the eigenvalue 0).
- Let $U \in \mathbb{R}^{n \times (k-1)}$ be the matrix containing the vectors $u_1, \ldots, u_{k-1}$ as columns.
- For $i = 1, \ldots, n$, let $y_i \in \mathbb{R}^{k-1}$ be the vector corresponding to the i-th row of U.
- Cluster the points $(y_i)_{i=1,\ldots,n}$ in $\mathbb{R}^{k-1}$ with the k-means algorithm into clusters $C_1, \ldots, C_k$.

Output: Clusters $A_1, \ldots, A_k$ with $A_i = \{j | y_j \in C_i\}$.

3.2. Using spectral clustering for unsupervised WSD

There are a number of issues that must be dealt with when applying spectral clustering in practice. One must choose how to compute the similarity between observations and how to transform these similarities into a similarity graph. In the case of the mutual k-nearest-neighbor graph, the parameter k, representing the number of nearest neighbors, must be set. In the light of all these issues we follow the approach adopted by Popescu and Hristea (2011) where spectral clustering was used in WSD for the first time.

[2] The "RatioCut" is not the only objective function optimized in spectral clustering. See (von Luxburg, 2007) for other variants such as "Ncut".

In unsupervised WSD, the observations are represented by contexts of the ambiguous word. The contextual features are given by the actual "neighboring" content words of the target (ambiguous) word. They occur in a fixed position near the target, in a window of fixed length, centered or not centered on the target. A window of size n denotes the consideration of n content words to the left and n content words to the right of the target, whenever possible. The total number of words considered for disambiguation is therefore $2n + 1$. When not enough features are available, the entire sentence in which the target word occurs represents the context window. The classical window size, generally used in WSD, and also in the present paper, is 25 ($n = 25$). Within this representation, the value of a feature is given by the number of occurrences of the corresponding word in the given context window. Thus, a context is represented as a feature vector and the similarity between two contexts is given by the value of the dot product of the corresponding feature vectors. The dot product was chosen as the measure of similarity between feature vectors because of the success of the linear kernel in supervised WSD (Màrquez, Escudero, Martínez, & Rigau, 2006).

As a method for building the similarity graph from the similarity matrix we use the mutual k-nearest-neighbor graph method. This involves the choice of the parameter k, the number of neighbors. As in Popescu and Hristea (2011), we use a value of 30 for the number of neighbors.[3]

4. Word sense discrimination in IR

The main contribution of this paper is the proposal of a new spectral clustering-based method for performing word sense discrimination for IR. This method aims to increase the top level precision for queries which contain ambiguous words. Our suggested approach reorders an initially retrieved document list by pushing to the fore documents that are semantically similar to the target query.

To start with, for each query we retrieve a set of documents by means of a state of the art search engine; documents are ordered according to their scores. Our objective is then to pinpoint the documents which are more relevant to the information need because they share the term sense with the query; and to enable these documents to improve their position at the top of the document list.

The first phase in the method is based on clustering the retrieved document set with respect to the senses of each ambiguous word and to decide which documents share the query term sense. Spectral clustering is used in this phase. In the second phase, we reorder the initially retrieved document list by boosting documents belonging to the selected cluster.

4.1. Query WSD

Before we can apply the WSD technique described in Section 3.2 to the queries, we need to process the data within a preprocessing step. The preprocessing step identifies the polysemous words of a query (as a result of their occurrence in multiple WN synsets).

Term discrimination uses a sub-set of documents. More precisely, for each query, we consider the first n documents retrieved by the IR system. For each document we thus know the score which indicates how similar that document is to a specific query. The query is added to this set of retrieved documents as if it was another document. The feature set for each document is then calculated by creating an incidence matrix with rows representing the documents and columns representing the features. Each element of this matrix is either 1 or 0, depending whether or not the feature indicated by the column index is present in the document indicated by the row index. The number of WN senses and the incidence matrix obtained after data preprocessing is used as input for the WSD algorithm. Thus, the WSD process is performed on $n + 1$ documents (n initially retrieved documents and the query itself).

Let us now describe in more details the entire WSD process in relation to a single polysemous target word.

The first step is to build the corresponding feature set. The first processing step is to eliminate the stopwords. The remaining words are stemmed using the Porter stemmer algorithm (Porter, 1980). The stem corresponding to the target word is not retained, while the remaining stems, alphabetically ordered, represent the final set of features that are used in the WSD process.

The second step is to build the incidence matrix that indicates what features occur in each document. We determine the position of the target word within each of the documents. In our experiments we used a context window of size 25, as suggested in Hristea, Popescu, and Dumitrescu (2008) in order to obtain the best possible disambiguation accuracy. The features that occur in the context window are stored in the row of the matrix that corresponds to the analyzed document. If a certain document contains the target several times, we only consider its first occurrence. This is done in accordance with the "one-sense-per-discourse" heuristic (Gale, Church, & Yarowsky, 1992) which is largely used in WSD and which states the tendency of a word to preserve its meaning across all its occurrences in a given discourse.

For each query and for each ambiguous term occurring in that query, we cluster the retrieved documents into a number of clusters equal to the number of senses the ambiguous term has, according to a lexical database (WN), which will be used as sense inventory. The final task of the WSD process for a given polysemous word is to determine the document clusters relative to that word. Each obtained cluster corresponds to a specific sense of the polysemous target word, with one of

[3] See (Popescu & Hristea, 2011) for a justification concerning the choice of this number of neighbors.

the clusters containing the query itself. Note that two documents are similar (and thus belong to the same cluster), from the WSD point of view, if the polysemous word has the same sense in both documents. Therefore, disambiguating a polysemous term results in retaining only those documents occurring in the same cluster as the query.

A query can contain several ambiguous terms. In this case, as many clusters of documents as the number of ambiguous words in the query are retained. In order to form a unique list of documents, we fuse these sets of documents; we consider the initial values obtained by the search engine to be the document scores. Various fusion functions that can be used for this purpose have been defined in the literature (Shaw & Fox, 1995). In order to obtain a unique cluster per query, we apply the fusion function CombMNZ (Shaw & Fox, 1995).

The CombMNZ function computes the final document scores as follows:

$$S_f^i = c_i \sum_{j=0}^{c_i} \left(S_j^i \right), \tag{4}$$

where S_f^i represents the final score for each document d_i, S_j^i represents the score of the document d_i from the cluster j (if the document d_i does not exist in one particular cluster j, then $S_j^i = 0$), and c_i represents the number of nonzero scores for each document i ($c_i = k$ if the document d_i occurs in k clusters).

We should point out that our unsupervised WSD method performs word sense discrimination and therefore does not give the actual word sense (since we do not know which cluster refers to a specific sense). However, it is not necessary to pair clusters with senses, as document clusters are sufficient for explicit automatic disambiguation in IR.

4.2. Document Re-ranking

Our approach aims to improve the top retrieved document list. The first phase in the discussed method leads to a set of documents extracted by the search engine, corresponding to each query, and in the clusters of documents obtained as described in Section 4.1. Our main purpose for using a WSD technique in IR is to find the most probable relevant documents and to assign them a higher rank in the initial document list. The second phase of the method thus corresponds to a re-ranking method (Meister, Kurland, & Kalmanovich, 2011).

In our approach, the way to reach this goal is to modify the order of the retrieved documents by pushing those documents to the fore that are semantically similar to the query, as defined by the WSD results. To do this, we fuse the initial set of documents with those obtained as a result of clustering. According to the method discussed above, the documents obtained by the search engine and the set of documents obtained after clustering have different levels of importance in the final results. We therefore use a parameter to assign a weight to the fusion function.

This function has the following structure:

$$\begin{aligned} &S_f^i = S_1^i + \alpha S_2^i \text{ with :} \\ &S_1^i = score(d_i) \\ &S_2^i = \begin{cases} score(d_i), & \text{if } d_i \text{ exists in } Clust \\ 0, & \text{otherwise} \end{cases}, \end{aligned} \tag{5}$$

where S_f^i represents the final score of a document d_i, $score(d_i)$ represents the score of that document d_i when considered in the initially retrieved document set, *Clust* is the document cluster containing the query itself and $\alpha \in [0, 1]$ represents the weight of the clustering method for the final results.

As reported in the evaluation section (Section 5), we started with $\alpha = 0$ and then increased this parameter by 0.01 at each trial.

Finally, in Section 6, the method is used on subgroups of queries with the purpose of analyzing its behavior with regard to different types of queries. The criterion for creating these subgroups of queries is their performance after being sent to the search engine.

5. Evaluation framework

5.1. Data collection features

To evaluate the described method we have used data collections from the TREC competition. TREC (Text REtrieval Conference) is an annual workshop hosted by the US government's National Institute of Standards and Technology which provides the necessary infrastructure for the large-scale evaluation of text retrieval methods.[4] The TREC ad hoc tasks allow us to investigate the performance of systems that search a static set of documents using new information needs (called topics). We opted for three collections for use in the ad hoc task: TREC7, TREC8 and WT10G. For TREC7 and TREC8, the competition provided approximately 2 gigabytes worth of documents and a set of 50 natural language topic statements (per collection).

[4] http://mitpress.mit.edu/catalog/item/default.asp?ttype=2&tid=10667.

Table 1
The number of ambiguous queries for the data collections.

Collection	Number of queries with X ambiguous terms					Total number of ambiguous queries
	$X = 0$	$X = 1$	$X = 2$	$X = 3$	$X = 4$	
TREC7	15	22	10	3	0	35
TREC8	13	22	6	9	0	37
WT10G	18	15	11	5	1	32

The documents were articles from newspapers like the Financial Times, the Federal Register, the Foreign Broadcast Information Service and the LA Times. The WT10G collection provided approximately 10 gigabytes worth of Web/Blog page documents.

TREC distinguishes between a statement of information need (the *topic*) and the text that is actually processed by a retrieval system (the *query*). The TREC test collections provide topics. What is now considered the "standard" format of a TREC topic statement comprises a topic *ID*, a *title*, a *description* and a *narrative*. The title contains two or three words that represent the key words a user could have used to send a query to a search engine. The description contains one or two sentences that describe the topic area. The narrative part gives a concise description of what makes a document relevant (Voorhees & Harman, 1998). Both the descriptive and the narrative parts can offer clues about the word senses used in the title part.

5.2. Ambiguous queries and ambiguous terms

In our approach, the ambiguity of a query was evaluated with reference to the title part of the topic. The ambiguous terms were detected using the WN knowledge database. If the term occurred in multiple WN synsets, then it was considered as an ambiguous term. A query is defined as ambiguous if it contains at least one ambiguous term. The queries from the three collections contained from zero to four ambiguous words, as presented in Table 1, with most of them being nouns.

5.3. Evaluation measures

In TREC, ad hoc tasks are evaluated using the `trec_eval` package. This package provides various performance measures, including some single valued summary measures that are derived from the two basic measures in IR: recall and precision. The precision is the fraction of the retrieved documents that are relevant, while the recall represents the fraction of the documents relevant to the query that are successfully retrieved. The average precision is defined as:

$$AP(q) = \frac{\sum_{r=1}^{R}[p(r)rel(r)]}{relev(q)} \tag{6}$$

where $relev(q)$ represents the number of documents relevant to the query q, R is the number of retrieved documents, r is the rank, $p(r)$ is the precision of the top r retrieved documents and $rel(r)$ equals 1 if the rth document is relevant and 0 otherwise. The MAP (Mean Average Precision) stands for the mean of the average precision scores for each query. The `trec_eval` package also implements the precision at certain cut-off levels. A cut-off level is a rank that defines the retrieved set. For example, a cut-off level of ten defines the retrieved set as the top ten documents in the ranked list (P@10).

This study uses three cut-off levels: P@5, P@10 and P@30, which are high precision measures.

5.4. Baselines

The present study is based on runs constructed by Terrier. Terrier is an open source search engine that implements state-of-the-art indexing and retrieval functionalities. Terrier was developed at the School of Computing Science, University of Glasgow.[5] The sets of documents retrieved by Terrier are the first 1000 ranked documents returned by the search engine. We tried several configurations of the Terrier parameters. For each collection, we chose as our baselines the settings with the highest MAP. The MAP values for our baselines are consistent with the literature (He & Ounis, 2005; Zhong & Ng, 2012). Runs (associated with the baselines), which determine the set of documents to be used by our WSD method, were constructed as additional baselines.

The best configuration for TREC7 was the following: a two step indexation (both direct and inverted index), with active indexation by block and the use of the BB2 (parameter $c = 1$) as a weighting model. As a query expansion model our choice was the parameter-free KL model (KLbfree). The configuration required 3 documents to be used for the query expansion. A term has to occur in two documents in order to be considered relevant. Finally, the number of terms to be added to the query for the process of query expansion was set at 10. The queries use all the three topic parts (title, descriptive and narrative). However, for TREC8 and WT10G, the parameter configuration with the best results in terms of MAP stays in place, except for two differences: the weighting model which is changed with the DFRee model (no parameters) and the narrative part of the topic which is not taken into account when the query is constructed. The values of the MAP corresponding to the best initial

[5] http://www.terrier.org.

Table 2
Topic and document features from the data collections.

Collection	No. of topics	Topic number	No. of documents	Baseline MAP
TREC7	50	351–400	528,155	0.2851
TREC8	50	401–451	528,155	0.2577
WT10G	50	451–500	1,692,096	0.1733

results for the data collections we use are briefly introduced in Table 2, in addition to some collection features, such as the number of topics and the number of documents.

It is worth mentioning that, since we re-ordered an initial retrieved document list, we were unable to retrieve documents that would not have initially been retrieved (no recall improvement). We target high precision improvements.

5.5. WSD settings

For each query, the set of the top 1,000 documents retrieved by the best settings for Terrier was considered. These 1,000 documents are the documents considered as the most similar to the information need, sorted by their obtained score, in descending order. The method we promote aims to filter and reorder those documents before retrieving them for the user.

The target terms for the described WSD process are taken from the title part of the TREC topic only. However, our approach needs to have a context for the term. Since topic titles are generally too short, in order to form the context window for the ambiguous terms in the title, all three parts of the topic were used (title, narrative and descriptive). The stopwords from the resulting text were removed (Terrier's stopwords list is used).

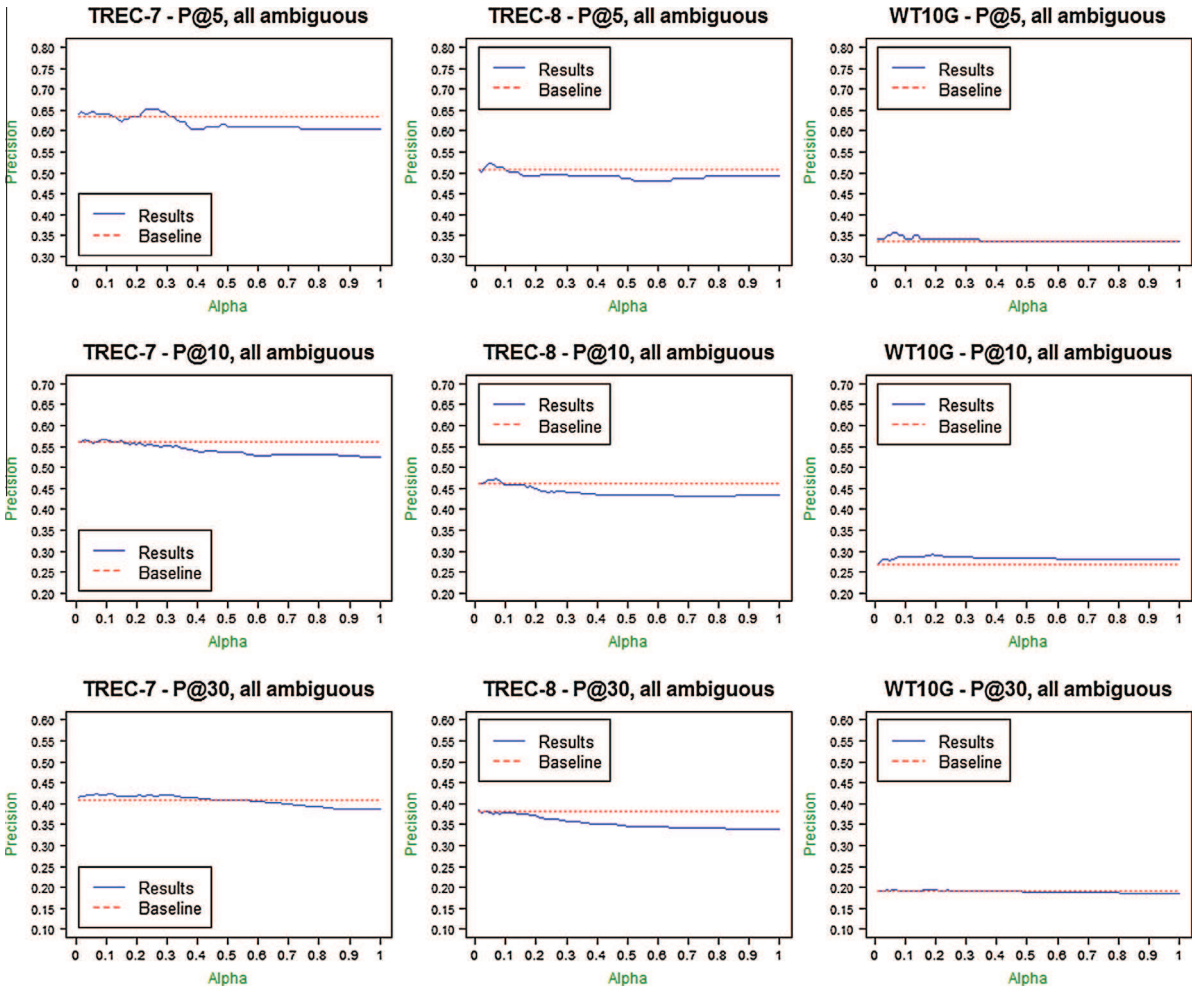

Fig. 1. The results for the three test collections by the top levels of precision.

Table 3
Comparison with the co-occurrence-based methods and with the Naïve Bayes-based method, by the top precisions, for the TREC7 collection.

Prec.	Best	Naïve Bayes		Spectral clustering		Schütze & Pedersen			
	Run	Peak res.	Average res.	Peak res.	Average res.	Sense-based	CombRank	CMNZ-WB-SB	CMNZ-WB-SB-0.1
P@5	0.6343	0.6286	0.5736	**0.6514****	0.6193	0.3829	0.4400	0.4286	0.4914
P@10	0.5600	0.5629	0.5152	**0.5657****	0.5409	0.3400	0.3686	0.3743	0.4257
P@30	0.4095	0.4171**	0.3999	**0.4248****	0.4068	0.2438	0.2629	0.2752	0.3210

Bold characters are used in order to highlight the column with the highest peak results, by the top precisions.
** Represents p-value $< 10^{-6}$.

5.6. Results

The results as compared to the baselines are presented in Fig. 1. The graphs illustrate the manner in which the top levels of precision evolve with respect to the alpha parameter. Alpha is the parameter which gives a greater or smaller level of importance, in the final score, to the scores of the documents in the cluster, as presented in Section 4.2 (Eq. (5)). On the first row, the results for P@5 are shown, in comparison with the three collections. The next two rows present the results for P@10 and P@30 respectively. For each cut-off level, the vertical axis is recalibrated in order to obtain a clearer view.

Fig. 1 shows that the best results were obtained when the value of the alpha parameter was between 0.02 and 0.20. This observation holds for all the top levels of precision (P@5, P@10 and P@30) and for all the three collections involved in the study. p-values smaller than 10^{-6} of T-Tests have confirmed the statistical significance of our results. T-Tests have used the two following populations: the baseline value (fixed across the alpha parameter) and our results per alpha, respectively. It is also noticeable that for an alpha parameter which is greater than 0.2, the results usually fall below the baselines. Since alpha is the parameter that assigns a weight to the clustering method for the final results, one can notice that it is best for the cluster documents to participate with not more than $1/5$ in the final document score. The alpha parameter is similar to lambda in the case of RM3, since the optimal lambda values for the short queries vary between 0.01 and 0.4 (Zhai & Lafferty, 2004). The baselines were surpassed by the obtained results for each collection and for each top level of precision. While for P@5 and P@10 the difference between the results and the baselines is clear, for P@30 the curve representing the results remains closer to the baseline. The best improvement occurred for the WT10G collection in the case of P@10. A precision value of 0.2937 was obtained (the baseline was 0.2688), which represents an improvement of 8.48%, and is statistically significant with a p-value $< 10^{-6}$ (T-Test).

Analysis of the results has been carried out against the two major approaches existing in the literature. We compare our results with those obtained when implementing the disambiguation method proposed in Schütze and Pedersen (1995), as well as with those reported in Chifu and Ionescu (2012). The latter authors also test over the TREC7 benchmark, while employing a Naïve Bayes clustering technique. The baselines are therefore identical and represent the performance of the best runs (see Section 5.4), for all the 35 ambiguous queries considered in our study.

In the case of the terms co-occurrence-based method (Schütze & Pedersen, 1995) we have reimplemented this method and have organized the same testing setup as the one originally used by Schütze and Pedersen (1995). The stop words have been removed (Terrier stop-words list) and the target term-centered context window was set to size 40 (20 terms before the target term, 20 terms after the target term). We have identified 1,466,983 unique terms that induced the 1,466,983 $\times$ 1,466,983 sparse term co-occurrence matrix. We mention that, due to the high number of vocabulary terms, the co-occurrence matrix is difficult to handle from a computational point of view. The SVD for the co-occurrence matrix was set to 100 dimensions and the reduced matrix was computed using the `irlba` package of `R`[6] (with 100 iterations). To classify the context vectors we used the Buckshot algorithm implemented using `R` (packages `hc` and `kmeans` of `R`), with 10% sampling for the initial hierarchical clustering step. The query terms that occurred less than 100 times in the corpus were not considered ambiguous since, according to its authors (Schütze & Pedersen, 1995), the method uses $f/50$ as the number of senses, with f denoting the occurrence number of the target term in the corpus. Creating context vectors for each target word occurrence, as well as reindexing after replacing words with their senses (for each set of queries), also represent time and resource consuming operations.

In Table 3 we present the results of our comparison in terms of high precision. **Best Run** represents the best run obtained with Terrier, treated as baseline (see Section 5.4) and also treated as the word-based retrieval for the terms co-occurrence-based method. **Naïve Bayes** represents the method from Schütze and Pedersen (1995). **Spectral Clustering** is the method proposed in this article. **Sense-based** represents the terms co-occurrence-based method of Schütze and Pedersen (1995) and **CombRank** represents the modified co-occurrence method, also presented in Schütze and Pedersen (1995), which considers the sum of ranks from the word-based retrieval and from the sense-based retrieval as the final rank for a retrieved document. It was reported (Schütze & Pedersen, 1995) as better than using the sense-based method alone. **CMNZ-WB-SB** represents the combined document list resulting from the word-based and sense-based retrievals, using the CombMNZ function (Shaw & Fox, 1995). **CMNZ-WB-SB-*alpha*** represents the **CMNZ-WB-SB** results with the sense-based retrieved list weighted by an *alpha* parameter. We tested various *alpha* values. The best turned out to be 0.1.

[6] http://www.r-project.org/.

Table 4
The number of improved queries, by the top precisions, for **CMNZ-WB-SB-0.1** and **Spectral Clustering-0.1**.

Prec.	Number of improved queries	
	CMNZ-WB-SB-0.1	**Spectral Clustering-0.1**
P@5	3	18
P@10	3	19
P@30	5	17

The various combinations help to improve the initial performance of Sense-based results, although the baseline results (**Best Run**) are not surpassed. The number of the improved queries with respect to **Best Run** was also computed for the **CMNZ-**0.1 (which is the best alpha for co-occurrence-based results) and for the **Spectral Clustering** results, with the same *alpha* value of 0.1. The results are presented in Table 4. Only very few queries are improved by the terms co-occurrence-based method.

As opposed to the **Sense-based** model, the peak results of **Spectral Clustering** outperform the **Best Run** baseline, for all the levels of high precision (from 1.01% to 3.73%). The average results do not overcome the baseline due to the performance decrease after a certain value of alpha (see Fig. 1).

The **Naïve Bayes** peak results outperform the **Best Run** only for P@10 and P@30 (0.51% and 1.85%, respectively). In addition, our method outperforms the Naïve Bayes method both on average and on peak results. The average is computed across all alpha parameter values, considering all the ambiguous queries. This again suggests the importance of the clustering technique used in unsupervised WSD for IR. We hereby conclude that spectral clustering is an appropriate clustering method for the purpose of sense discrimination in IR.

The present method was also tested for 5000 document runs, but the results were not improved. We think the reason for this is that, once more documents per run are taken into account, a significant amount of noise is also introduced and the re-ranking method cannot reach efficiency at the top level of precision (P@5, P@10 and P@30). We also considered a two cluster model in which documents could either be clustered in the query cluster if similar enough to the query, or in the non-query cluster. Results were better when as many clusters as WN senses were considered.

6. Further analysis of the results

This section aims to deepen the analysis of the results obtained when considering two types of query clusters: those based on the query performance and those taking into account the number of ambiguous terms per query.

6.1. Improvements in baseline precision intervals

Following previous research showing that results can differ according to query difficulty (Bigot, Chrisment, Dkaki, Hubert, & Mothe, 2011) and with the purpose of observing where the proposed method behaves most accurately, (independently for each collection), all the results from all the three test collections were gathered into a single data set. All the runs corresponding to the 104 ambiguous queries were divided into 5 groups, according to the baseline precision (0.0–0.2, 0.2–0.4, . . ., 0.8–1.0) and for each of the top levels of precision being investigated (P@5, P@10 and P@30, respectively).

The P@5 results for all the intervals are depicted in Fig. 2. For the queries with low or very low performance (intervals 0.0–0.2 and 0.2–0.4) we obtained significant improvements. This fact suggests that the reordering of the poorly performing list of documents retrieved by the search engine puts more relevant documents at the top of the list. On the other hand, for the queries with a good or very good performance (0.6–0.8 and 0.8–1.0), our re-ranking method could not bring more relevant documents to the fore because the search engine's results were either already as good as possible (0.8 for the interval 0.6–0.8, or 1.0 for the interval 0.8–1.0), or very close to this. The results can hence overcome the baseline only by chance. All the comparisons are statistically significant, with the p-values $< 10^{-6}$ (T-Test for columns **Baseline** vs. **Peak Result** and **Baseline** vs. **Average Result**, respectively). The conclusions for P@5 are also consistent with P@10. The good results for P@10 can be explained by the fact that there is a higher chance of obtaining new relevant documents in a list of 10 documents than in a list of 5. However, for P@30, the improvements were not as significant as for the other top levels of precision. In order to improve the performance in a list of 30 retrieved documents it would be necessary to bring more than 1 or 2 new relevant documents from the re-ranking process. (For P@5, 1 new relevant document represents a 20% improvement, while for P@30, 1 new relevant document represents only a 3.33% improvement).

In Table 5 we present the peak (the best results) and average improvements for each baseline precision interval in the case of P@5 and P@10 respectively.

6.2. Detailed results after taking into account the number of ambiguous terms

The queries from the data set utilized in this study contain from 0 to 4 ambiguous terms (see Table 1). A high number of ambiguous terms also suggests an increased level of query difficulty caused by multiple possible combinations of senses

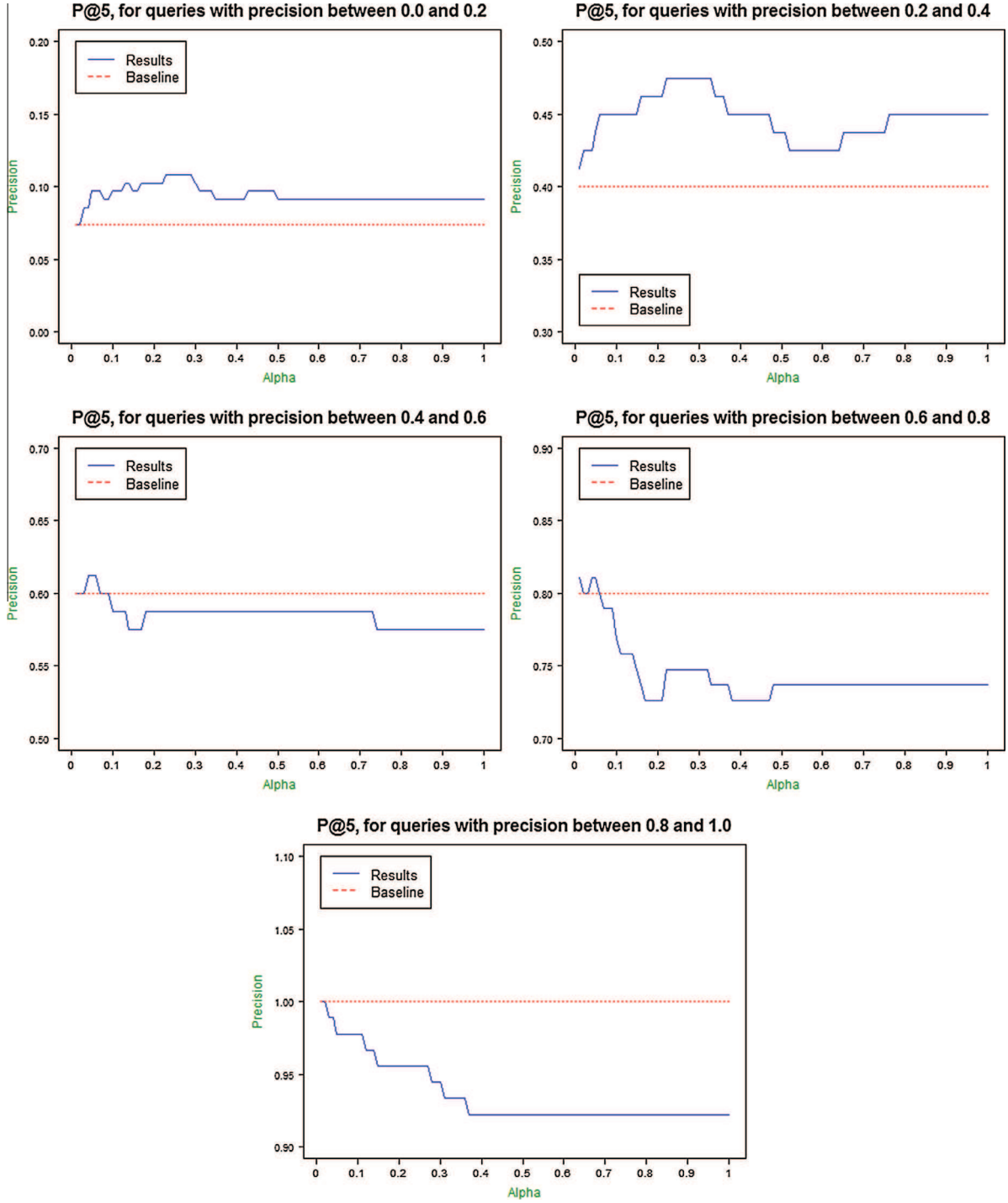

Fig. 2. P@5 by the intervals of performance corresponding to the TREC7, TREC8 and WT10G collections.

between terms. Keeping this aspect in mind, we investigated the behavior of our method over clusters of queries classified by the number of ambiguous terms. We proceeded as in Section 6.1 (independently for each collection) by grouping all three data sets into a single one. All of the 104 ambiguous queries were divided into three classes: queries that contain 1 ambiguous term, 2 ambiguous terms and 3 ambiguous terms respectively. Our method was not applied to the queries that contained no ambiguous terms (see Section 5.2). The population for the cluster corresponding to queries with 4 ambiguous terms was very weak (only one query) and therefore was also not taken into account.

Table 5
The peak and average improvements for each baseline precision interval.

Precision	Interval	Baseline	Peak result	Peak improvement (%)	Average result	Average improvement (%)
P@5	0.0–0.2	0.0742	**0.1085**	**46.15****	**0.0942**	**26.95****
P@5	0.2–0.4	0.4000	**0.4750**	**18.75****	**0.4477**	**11.92****
P@5	0.4–0.6	0.6000	0.6125	2.08**	0.5851	−2.48**
P@5	0.6–0.8	0.8000	0.8105	1.31	0.7433	−7.08**
P@5	0.8–1.0	1.0000	1.0000	0.00	0.9360	−6.40**
P@10	0.0–0.2	0.0696	**0.1121**	**60.87****	**0.1048**	**50.57****
P@10	0.2–0.4	0.3560	**0.3920**	**10.11****	**0.3731**	**4.80****
P@10	0.4–0.6	0.5312	0.5437	2.35**	0.4916	−7.45**
P@10	0.6–0.8	0.7533	0.7600	0.88	0.6546	−13.10**
P@10	0.8–1.0	0.9533	0.9533	0.00	0.9227	−3.20**

Bold values underline the best improvements, both on avearage and peak, with respect to baseline precision intervals.
** Represents p-value < 10^{-6}.

The peak results and the percentages of improvements for each cluster, by the top levels of precision, are presented in Table 6.

The highest values were obtained for the clusters of queries containing 3 ambiguous terms, which suggests that our method best improves the most ambiguous queries. For P@30 the improvement was almost 8%. The results are statistically significant with p-values $< 10^{-6}$ (T-Test for columns **Baseline** vs. **Peak Res.**). It is also worth mentioning that constant improvements were also obtained for the other two clusters being investigated.

7. The spectral clustering method using automatically generated context

The method we propose in this paper uses all the three parts of the TREC topics, title, description and narrative (TDN), as disambiguation context. TDN implies the assumption that a context exists for the query, which is not the case in real world applications. For this reason, in this section we automatically build a context in order to validate our approach. This automatic context is not optimal (weaker performance than for TDN) and it is not optimized since our point was only to validate that our method still works with automatic context. We present the automatic contextualization method and we discuss the obtained results.

7.1. Automatic contextualization using pseudo relevance feedback

We chose a straightforward pseudo relevance feedback (PRF) approach in order to obtain the context (Attar & Fraenkel, 1977; Buckley, Salton, Allan, & Singhal, 1994). The option for this type of contextualization is motivated by the assumption that the first retrieved documents have high chances to be relevant and thus they presumably contain the target words with the correct sense.

First of all, we run retrieval on the initial query (title part of the TREC topic) over the TREC document collections and we retain the first three retrieved documents, as it was done in the baseline (see Section 5.4). This parameter value is used in query expansion models (He & Ounis, 2009) based on the assumption that, when taking into account more than five documents, the probability of treating irrelevant documents increases. Having these top documents, we concatenate the texts, we remove the stopwords and we search for the presence of at least two query terms in a moving context window of 50 words. If this presence occurs, we keep the text in the context window and add it to our context. The search for at least two query terms together is motivated by the assumption that two ambiguous words tend to disambiguate each other when found together, for example "java" and "island" (Andrews, Pane, & Zaihrayeu, 2011).

External sources such as Wikipedia were avoided when building the context due to differences in terms of actuality. Moreover, the relevance judgments were constructed considering the information in the description and narrative parts of the topic, suggesting some kind of a closed circuit. For instance, supposing that we have obtained a context with senses for target words different than the senses suggested for pooling, this would lead the evaluation of the disambiguation process to complete failure.

Table 6
The peak improvements, by the number of ambiguous terms, for each top precision.

Precision	1 Ambiguous Term			2 Ambiguous Terms			3 Ambiguous Terms		
	Baseline	Peak res.	Peak improv. (%)	Baseline	Peak res.	Peak improv. (%)	Baseline	Peak res.	Peak improv. (%)
P@5	0.5767	0.5936	2.9307**	0.4000	0.4148	3.6952**	0.3882	0.4118	**6.0575****
P@10	0.5238	0.5404	3.1712**	0.3307	0.3445	4.1724**	0.3176	0.3353	**5.5688****
P@30	0.3911	0.4001	2.3072**	0.2323	0.2434	4.7841**	0.2727	0.2941	**7.9076****

** Represents p-value < 10^{-6}.

Table 7
TREC7, P@X for the considered runs.

Run evaluation	P@5	P@10	P@30
TD(N)	0.6429	0.5679	0.4226
Spectral-TD(N)	0.6571	0.5714	0.4381
Title	0.5310	0.4586	0.3425
Title + Context	0.5448	0.4103	0.2770
Spectral-Title	0.5379	0.4586	0.3402
Spectral-Title + Context	**0.5655(6.5%)****	**0.4724(3.0%)****	**0.3460(1.0%)****

Table 8
TREC8, P@X for the considered runs.

Run evaluation	P@5	P@10	P@30
TD(N)	0.5081	0.4622	0.3811
Spectral-TD(N)	0.5243	0.4730	0.3829
Title	0.5371	0.4800	0.3848
Title + Context	0.3829	0.2543	0.1524
Spectral-Title	0.5486	**0.5114**	0.3886
Spectral-Title + Context	**0.5543(3.2%)****	0.5029(4.8%)**	**0.4019(4.4%)****

Table 9
WT10G, P@X for the considered runs.

Run evaluation	P@5	P@10	P@30
TD(N)	0.3538	0.2885	0.1923
Spectral-TD(N)	0.3692	0.3115	0.1949
Title	0.3500	0.2714	0.1917
Title + Context	0.2071	0.1357	0.0679
Spectral-Title	0.3357	0.2679	0.1857
Spectral-Title + Context	**0.3571(2.0%)****	**0.2786(2.7%)****	**0.1929(0.6%)****

The usage of our PRF-based context and insights regarding the performance are presented in the following subsection.

7.2. Experiments and results

In order to prove the effectiveness of our method in the case of automatically generated context we created four TREC runs, as follows:

- **Title**: retrieved documents when the query represents only the title part of the topic;
- **Title + Context**: retrieved documents when the query represents the title part of the topic, together with the automatically built context;
- **Spectral-Title**: the re-ranked documents after applying the spectral clustering method, when the query is represented only by the title part of the topic;
- **Spectral-Title + Context**: the re-ranked documents after using the automatic context as WSD context for the spectral clustering method.

For few queries in each collection, our method was not able to provide any context either due to a title part of the TREC topic formed only by one term, or due to the complete nonexistence of co-occurrences of at least two terms in the context window, in the retained text. Hence, we considered only the queries containing ambiguous terms and for which the automatic method was able to provide a context, as follows: 29 out of 35 ambiguous queries in TREC7, 35 out of 35 ambiguous queries in TREC8 and 28 out of 32 ambiguous queries in WT10G, respectively.

Tables 7–9 provide precision values at 5, 10 and 30 retrieved documents after evaluating the above mentioned runs, for each collection. For comparison we recall the results obtained using the reformulated TD(N) runs (from Section 5). We mention that the queries without automatic context were also removed from the TD(N) evaluations, in order to maintain the same comparison basis. The best values per collection are written in bold. Statistical significance of results (*T*-Test between the basic **Title** run and **Spectral-Title + Context**) is also marked with asterisks in the tables (p-value $< 10^{-6}$).

In terms of top level precision, the Spectral-Title + context run is better than the **Title** run, which is better than the **Title + Context** run. This suggests that the generated context is harmful for the retrieval process itself but beneficial for the spectral clustering method (**Spectral-Title + Context** run is generally better than **Spectral-Title**). We believe that this

is due to the amount of "noisy" terms in the context. Unlike the feature selection process using a Naïve Bayes technique (Chifu & Ionescu, 2012), spectral clustering automatically selects its useful features, hence "noise" filters out and it remains less of a problem than for the retrieval process.

We notice that in 89% of cases the **Spectral-Title + Context** run has the greatest performance. Even if the relative improvement (0.6–6.5%) is not very high, this improvement allows us to state that our method remains effective even with automatic contextualization. In addition, the results we obtained in Section 5 show that using a better context would improve the results even more.

The least improved results are to be noticed in the case of WT10G (Table 9). Here the initial retrieval (Title only) has the lowest performance among all three considered collections, therefore the context quality decreases, since the P@3 is relatively low (TREC7: 0.5977, TREC8: 0.5619, WT10G: 0.3810). Having a poor context implies a less performing WSD process.

8. Conclusions

This paper presents a re-ranking method for IR. It shows a remarkable improvement in high rank precision for ambiguous queries. We believe that this represents a very important aspect, considering the fact that IR systems are predisposed to failure in the case of this particular type of queries (Stokoe et al., 2003; Mothe & Tanguy, 2007).

Several previous studies (Sanderson, 1994; Guyot et al., 2008) have failed to prove the usefulness of WSD in IR. On the contrary, we show that unsupervised WSD, namely WS discrimination, can improve IR results. We are of the opinion that WS discrimination is sufficient in IR and that WS disambiguation is not compulsory, as opposed to text translation, for example. Analysis of the obtained results has been carried out by us with respect to the two major approaches existing in the literature (Chifu & Ionescu, 2012; Schütze & Pedersen, 1995), as detailed in Section 5. In the only existing related approach, when using a Naïve Bayes-based clustering technique, Chifu and Ionescu (2012) also demonstrated that WS discrimination can improve IR performance. However, in their work they only recorded very small improvement and only on some subcases, hence the importance of the clustering technique used for WS discrimination in IR, another point which we have made here.

Our method exploits TREC topic definitions which have a level of detail in their description, a level which is not normally available in an IRS; the topic definition is used to provide a context to the query. Other works from the literature use these structural elements and most often the Descriptive part of the query helps in improving the results (He & Ounis, 2004). However, using the complete statement of the topic could lead to valid criticism of experiments such as ours because we exploit this detail. In this paper, we were interested in proving that, if this level of detail is available, then it can be used in a beneficial way to improve retrieval effectiveness, and we have achieved this goal. However, even if our goal was not to develop mechanisms which can capture in an optimal way the needed level of detail, we do propose a method to capture the context of the query and show that our own method for WS discrimination in IR remains useful. Such a method of contextualization, namely the usage of PRF (Buckley et al., 1994), has been employed by us in Section 7 for validating our conclusions in the presence of automatically generated context. Indeed, contextualizing short texts, such as tweet contextualization (SanJuan, Bellot, Moriceau, & Tannier, 2011) and query expansion (Ogilvie, Voorhees, & Callan, 2009) is an active research domain and we think it will be worth considering new contextualization techniques in our WS discrimination method as a future goal.

Our future work will also concentrate on query difficulty prediction, which is already an active research area (Carmel & Yom-Tov, 2010; Mothe & Tanguy, 2005; Pehcevski, Thom, Vercoustre, & Naumovski, 2010). The fact that our method rather improves poor performing queries (Section 6.1), especially those with multiple ambiguous terms (Section 6.2), should drive in-depth research along this path.

Acknowledgements

The authors would like express their gratitude to Taoufiq Dkaki from the University of Toulouse and Radu Ionescu from University of Bucharest for their useful comments and discussions, as well as to the ANR agency who partially funded this work.

References

Agirre, E., & Edmonds, P. (Eds.). (2006). *Word sense disambiguation: Algorithms and applications*. Berlin: Springer-Verlag.

Andrews, P., Pane, J., & Zaihrayeu, I. (2011). Semantic disambiguation in folksonomy: A case study. In *Proceedings of the 2009 International Conference on advanced language technologies for digital libraries* (pp. 114–134). Berlin, Heidelberg: Springer-Verlag. <http://dl.acm.org/citation.cfm?id=2039901.2039909>.

Artiles, J., Gonzalo, J., & Sekine, S. (2007). The semeval-2007 weps evaluation: Establishing a benchmark for the web people search task. In *Proceedings of the fourth international workshop on semantic evaluations (semeval-2007)*. ACL. <http://www.aclweb.org/anthology/W/W07/W07-2012>.

Attar & Fraenkel (1977). Local feedback in full-text retrieval systems. *JACM: Journal of the ACM*, 24.

Bigot, A., Chrisment, C., Dkaki, T., Hubert, G., & Mothe, J. (2011). Fusing different information retrieval systems according to query-topics: A study based on correlation in information retrieval systems and trec topics. *Information Retrieval, 14*(6), 617–648. doi: 10.1007/s10791-011-9169-5, <http://dx.doi.org/10.1007/s10791-011-9169-5>.

Borjigin, S., & Guo, C. (2012). Non-unique cluster numbers determination methods based on stability in spectral clustering. *Knowledge and Information Systems*. doi: 10.1007/s10115-012-0547-0, <http://www.springerlink.com/index/10.1007/s10115-012-0547-0>.

Buckley, C., Salton, G., Allan, J., & Singhal, A. (1994). Automatic query expansion using SMART: TREC 3. In Trec (p.0).

Carmel, D., & Yom-Tov, E. (2010). Estimating the query difficulty for information retrieval. In *Proceedings of the 33rd international ACM SIGIR conference on research and development in information retrieval proceedings of the 33rd international acm sigir conference on research and development in information retrieval* (pp. 911). New York, NY, USA: ACM. doi: 10.1145/1835449.1835683, <http://doi.acm.org/10.1145/1835449.1835683>.

Carpineto, C., & Romano, G. (2012). A survey of automatic query expansion in information retrieval. *ACM Computing Surveys, 44*(1), 1. http://doi.acm.org/10.1145/2071389.2071390.

Chifu, A., & Ionescu, R. T. (2012). Word sense disambiguation to improve precision for ambiguous queries. *Central European Journal of Computer Science, 2*(4), 398–411.

D'Angelo, C. A., Giuffrida, C., & Abramo, G. (2011). A heuristic approach to author name disambiguation in bibliometrics databases for large-scale research assessments. *JASIST, 62*(2), 257–269. http://dx.doi.org/10.1002/asi.21460.

Gale, W. A., Church, K. W., & Yarowsky, D. (1992). One sense per discourse. In *Proceedings of the workshop on speech and natural language* (pp. 233–237). Stroudsburg, PA, USA: Association for Computational Linguistics. doi:10.3115/1075527.1075579, http://dx.doi.org/10.3115/1075527.1075579.

Gangadharaiah, R., Brown, R. D., & Carbonell, J. G. (2006). Spectral clustering for example based machine translation. In *HLT-NAACL'06 Hlt-naacl'06* (pp. -1–1).

Gonzalo, J., Verdejo, F., Chugur, I., & Cigarran, J. (1998). Indexing with WordNet synsets can improve Text Retrieval. <http://arxiv.org/abs/cmp-lg/9808002>.

Guyot, J., Falquet, G., Radhouani, S., & Benzineb, K. (2008). Analysis of word sense disambiguation-based information retrieval. In C. Peters (Ed.). *Clef* (Vol. 5706, pp. 146–154). Springer. http://dx.doi.org/10.1007/978-3-642-04447-2.

Han, H., Zha, H., & Giles, C. L. (2005). Name disambiguation in author citations using a K-way spectral clustering method. In *Proceedings of the Fifth ACM/IEEE-CS joint conference on digital libraries. Proceedings of the fifth acm/ieee-cs joint conference on digital libraries*.

Hastie, T., Tibshirani, R., & Friedman, J. (2008). *The elements of statistical learning: Data mining, inference and prediction* (2nd ed.). Springer.

He, B., & Ounis, I. (2004). Inferring query performance using pre-retrieval predictors. In *11th international conference, spire 2004, proceedings* (pp. 43–54). Padova, Berlin, Heidelberg: Springer. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.107.102 doi:10.1007/978-3-540-30213-1_5.

He, B., & Ounis, I. (2005). Term frequency normalisation tuning for BM25 and DFR models. In D. E. Losada & J. M. Fernández-Luna (Eds.). *ECIR – Advances in information retrieval, 27th european conference on IR research, ECIR 2005, Santiago de Compostela, Spain, March 21–23, 2005, Proceedings* (Vol. 3408, pp. 200–214). Springer. http://dx.doi.org/10.1007/978-3-540-31865-1_15.

He, B., & Ounis, I. (2009). Finding good feedback documents. In D. W.-L. Cheung, I.-Y. Song, W. W. Chu, X. Hu, & J. J. Lin (Eds.), *Proceedings of the 18th ACM conference on information and knowledge management, CIKM 2009, Hong Kong, China, November 2–6, 2009* (pp. 2011–2014). ACM. http://doi.acm.org/10.1145/1645953.1646289.

Hristea, F., Popescu, M., & Dumitrescu, M. (2008). Performing word sense disambiguation at the border between unsupervised and knowledge-based techniques. *Artificial Intelligence Review, 30*(1–4), 67–86. http://dx.doi.org/10.1007/s10462-009-9117-6.

Kilgarriff, A. (1997). What is word sense disambiguation good for? CoRR, cmp-lg/9712008. <http://arxiv.org/abs/cmp-lg/9712008>.

Kilgarriff, A., & Rosenzweig, J. (2000). Framework and results for english SENSEVAL. *Computers and the Humanities, 34*(1-2), 15–48. http://dx.doi.org/10.1023/A:1002693207386.

Kim, S.-B., Seo, H.-C., & Rim, H.-C. (2004). Information retrieval using word senses. In *Proceedings of the 27th annual international conference on research and development in information retrieval – SIGIR '04* (pp. 258). New York, New York, USA: ACM Press. doi: 10.1145/1008992.1009038, <http://dl.acm.org/citation.cfm?id=1008992.1009038>.

Krovetz, R., & Croft, B. (1992). Lexical ambiguity and information retrieval lexical ambiguity and information retrieval. *ACM Transactions on Information Systems, 10*(2), 115–141.

Leidner, J. L. (2007). Toponym resolution in text: Annotation, evaluation and applications of spatial grounding. *SIGIR Forum, 41*(2), 124–126. http://doi.acm.org/10.1145/1328964.1328989.

Lin, D. (1997). Using syntactic dependency as local context to resolve word sense ambiguity. In *Meeting of the association for computation linguistics* (pp. 64–71). <citeseer.nj.nec.com/lin97using.html>.

Maier, M., Hein, M., & von Luxburg, U. (2009). Optimal construction of k-nearest-neighbor graphs for identifying noisy clusters. *Theoretical Computer Science, 410*(19), 1749–1764.

Màrquez, L., Escudero, G., Martínez, D., & Rigau, G. (2006). Supervised corpus-based methods for WSD. In E. Agirre & P. Edmonds (Eds.). *Word sense disambiguation algorithms and applications* (Vol. 33, pp. 167–216). Dordrecht: Springer Netherlands. <http://www.springerlink.com/content/n426u44w1q274218/> doi: 10.1007/978-1-4020-4809-8.

Meister, L., Kurland, O., & Kalmanovich, I. G. (2011). Re-ranking search results using an additional retrieved list. *Information Retrieval, 14*(4), 413–437. http://dx.doi.org/10.1007/s10791-010-9150-8.

Mihalcea, R., & Moldovan, D. (2000). Semantic indexing using WordNet senses. *Proceedings of the acl-2000 workshop on recent advances in natural language processing and information retrieval held in conjunction with the 38th annual meeting of the association for computational linguistics* (Vol. 11, pp. 35). Morristown, NJ, USA: Association for Computational Linguistics. doi: 10.3115/1117755.1117760, <http://dl.acm.org/citation.cfm?id=1117755.1117760>.

Mothe, J. & Tanguy, L. (2005). Linguistic features to predict query difficulty – a case study on previous TREC campaigns. In *ACM conference on research and development in information retrieval, SIGIR, Predicting query difficulty – methods and applications workshop, Salvador de Bahia, Brésil*, 15/08/05-19/08/05 (pp. 7–10). ACM. <http://www.irit.fr/recherches/IRI/SIG/personnes/mothe/pub/SIGIR05b.pdf>.

Mothe, J. & Tanguy, L. (2007). Linguistic analysis of users' queries: towards an adaptive information retrieval system. In *International conference on SIGNAL-IMAGE TECHNOLOGY & INTERNET–BASED SYSTEMS (SITIS), Shanghai, China*, 16/12/07-19/12/07 (pp. 77–84). http://www.seerc.info: South-East European Research Center (SEERC). <ftp://ftp.irit.fr/IRIT/SIG/2007_SITIS_MT.pdf>.

Navigli, R. (2009). Word sense disambiguation: A survey. *ACM Computing Surveys, 41*(2). <http://doi.acm.org/10.1145/1459352.1459355>.

Ng, H. T. (2011). Does word sense disambiguation improve information retrieval? In *Proceedings of the fourth workshop on exploiting semantic annotations in information retrieval – esair '11* (pp. 17). New York, USA: ACM Press. <http://dl.acm.org/citation.cfm?id=2064713.2064724>, doi: 10.1145/2064713.2064724.

Ogilvie, P., Voorhees, E. M., & Callan, J. (2009). On the number of terms used in automatic query expansion. *Information Retrieval, 12*(6), 666–679. http://dx.doi.org/10.1007/s10791-009-9104-1.

Pehcevski, J., Thom, J., Vercoustre, A.-M., & Naumovski, V. (2010). Entity ranking in wikipedia: Utilising categories, links and topic difficulty prediction. *Information Retrieval, 13*, 568–600. http://dx.doi.org/10.1007/s10791-009-9125-9 (10.1007/s10791-009-9125-9).

Piskorski, J., Wieloch, K., & Sydow, M. (2009). On knowledge-poor methods for person name matching and lemmatization for highly inflectional languages. *Information Retrieval, 12*(3), 275–299. http://dx.doi.org/10.1007/s10791-008-9085-5.

Popescu, M., & Hristea, F. (2011). State of the art versus classical clustering for unsupervised word sense disambiguation. *Artificial Intelligence Review, 35*(3), 241–264. http://dx.doi.org/10.1007/s10462-010-9193-7.

Porter, M. F. (1980). An algorithm for suffix striping. *Program, 14*(3), 130–137.

Sanderson, M. (1994). Word sense disambiguation and information retrieval. In *Proceedings of SIGIR-94, 17th ACM international conference on research and development in information retrieval proceedings of sigir-94, 17th acm international conference on research and development in information retrieval* (pp. 49–57). Dublin, IE.

SanJuan, E., Bellot, P., Moriceau, V., & Tannier, X. (2011). Overview of the inex 2010 question answering track (qa@inex). In *Proceedings of the 9th international conference on initiative for the evaluation of xml retrieval: Comparative evaluation of focused retrieval* (pp. 269–281). Berlin, Heidelberg: Springer-Verlag. <http://dl.acm.org/citation.cfm?id=2040369.2040399>.

Schütze, H. (1998). Automatic word sense discrimination. *Computational Linguistics, 24*(1), 97–123.

Schütze, H. & Pedersen, J. (1995). Information retrieval based on word senses. In *Proceedings of the 4th annual symposium on document analysis and information retrieval* (pp. 161–175). Las Vegas, USA.

Shaw, J.A. & Fox, E.A. (1995). Combination of multiple searches. In *Overview of the Third Text REtrieval Conference (TREC-3) Overview of the third text retrieval conference (trec-3)* (pp. 105–108). NIST Gaithersburg.

Stokoe, C., Oakes, M. P., & Tait, J. (2003). Word sense disambiguation in information retrieval revisited. In *SIGIR Sigir* (pp. 159–166). ACM. http://doi.acm.org/10.1145/860435.860466.

Takacs, B., & Demiris, Y. (2009). Spectral clustering in multi-agent systems. *Knowledge and Information Systems, 25*(3), 607–622. <http://www.springerlink.com/index/10.1007/s10115-009-0272-5>, doi: 10.1007/s10115-009-0272-5.

Uzuner, O., Katz, B., & Yuret, D. (1999). Word sense disambiguation for information retrieval. In *Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference innovative applications of artificial intelligence* (pp. 985). Menlo Park, CA, USA: American Association for Artificial Intelligence. <http://dl.acm.org/citation.cfm?id=315149.315639>.

von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing, 17*(4), 395–416. http://dx.doi.org/10.1007/s11222-007-9033-z.

Voorhees, E. (1993). Using wordnet to disambiguate word senses for text retrieval. In *Proceedings of the 16th annual international acm sigir conference on research and development in information retrieval* (pp. 171–180). New York, NY, USA: ACM. doi: 10.1145/160688.160715, http://doi.acm.org/10.1145/160688.160715.

Voorhees, E. & Harman, D. (1998). Overview of the Seventh Text REtrieval Conference (TREC-7). In *Text retrieval conference (trec) trec-7 proceedings* (pp. 1–23). Department of Commerce, National Institute of Standards and Technology. Retrieved form papers/overview_7.ps.gz; papers/overview_7.pdf.gz (NIST Special Publication 500-242: The Seventh Text REtrieval Conference (TREC 7)).

Zhai, C., & Lafferty, J. (2004). A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems, 22*(2), 179–214. doi: 10.1145/984321.984322, http://doi.acm.org/10.1145/984321.984322.

Zhao, B., Xing, E. P., & Waibel, A. (2005). Bilingual word spectral clustering for statistical machine translation. In *Proceedings of the acl workshop on building and using parallel texts* (pp. 25–32). Ann Arbor, Michigan: Association for Computational Linguistics. <http://www.aclweb.org/anthology/W/W05/W05-0804>.

Zhong, Z. & Ng, H. T. (2012). Word sense disambiguation improves information retrieval. In Acl (1) (pp. 273–282). The Association for Computer Linguistics. <http://www.aclweb.org/anthology/P12-1029>.