Using Knowledge-Based Relatedness for Information Retrieval

Arantxa Otegi, Xabier Arregi, Eneko Agirre
IXA NLP Group, University of the Basque Country, UPV/EHU
{arantza.otegi,xabier.arregi,e.agirre}@ehu.es

Abstract

Traditional information retrieval (IR) systems use keywords to index and retrieve documents. The limitations of keywords were recognized since the early days, specially when different but closely-related words are used in the query and the relevant document. Query expansion techniques like pseudo-relevance feedback (PRF) and document clustering techniques rely on the target document set in order to bridge the gap between those words. This paper explores for the first time the use of WordNet-based semantic relatedness techniques to overcome the vocabulary mismatch between the query and documents, both on Information Retrieval and Passage Retrieval. We performed both query expansion and document expansion, with positive effects over a language modeling baseline on three datasets (Robust, Yahoo! and ResPubliQA), and over PRF on two of these datasets (Yahoo! and ResPubliQA). Our analysis shows that our models and PRF are complementary, in that PRF is better for easy queries and our models are stronger for difficult queries. We also show that our models generalize better to other collections, and are more robust to parameter adjustments. Our methods can be easily applied to other relevant knowledge sources like medical ontologies or linked-data repositories.

Keywords: *Knowledge-Based Systems, Semantic Similarity, Semantic Relatedness, Information Retrieval, Query and Document Expansion.*

1 Introduction

The potential pitfalls of keyword retrieval have been noted since the earliest days of information retrieval (IR). Keyword retrieval proves ineffective when different but closely-related words are used in the query and the relevant document. The use of different words creates a lexical gap between the query and the document. Lexical gaps are also a problem for question answering, as well as related passage-retrieval and answer-finding systems (Moldovan and Surdeanu, 2003; Riezler et al.,

Q: How *fast* does a *tractor* *go*?

D: This Directive shall apply only to *tractors* defined in paragraph 1 which are fitted with pneumatic tyres and which have two axles and a maximum design *speed* between 6 and 25 *kilometres per hour*.

(a)

Q: How do you *cook* an *apple pie*?

D: There are many good *recipes* for *apple pies* but there are also some important things to remember that are usually not in the recipe. That is you should make sure the bottom of the crust will *bake* as well and not remain soggy. To do this, coat the inside of the crust with butter before adding the filling and place the baking dish on a dark metal pan so the bottom will get more heat.

(b)

Figure 1: Examples of lexical gap from ResPubliQA and Yahoo! datasets

2007). Those systems face the task of recovering short pieces of information that satisfy the users' needs, passages or exact answers respectively, instead of whole documents. They also differ in the nature of the queries: while IR queries are usually composed by a few keywords, the queries on question answering scenarios are formulated as natural language questions. Furthermore, as noted in (Berger et al., 2000), users who submit questions to a question answering system can not be expected to anticipate the lexical content of an optimal response, and there is often little overlap between the terms in the question and those appearing in its answer. The datasets we have used in this work have been selected to test the effects of the lexical gap problem in ad-hoc IR, passage-retrieval and answer-finding tasks.

Figure 1 shows two examples from two of the datasets used in this article exemplifying lexical gaps. In each example there is a query (Q) and a relevant document (D) which answers the question using different but related words. For example, the question in Figure 1a contains *fast*, *tractor* and *go*. Only one of these words appears in the document (*tractor*), but other words related to the query are also present, like *speed* and *kilometres per hour*. Something similar happens on Figure 1b, where the query keyword *cook* does not occur on the document, which does contain related words like *recipes* or *bake*.

	tractor	speed	kmh	recipe	apple pie	bake
fast	**1.34**	**8.09**	1.11	**1.24**	0.78	1.17
cook	1.68	1.90	0.87	**5.93**	**2.57**	**4.31**

Table 1: Relatedness between *fast* and *cook* in each of the queries in Fig. 1a and 1b, respectively, with highlighted words in the respective documents. The numbers are produced by a WordNet-based relatedness software (Agirre et al., 2009). Numbers are scaled by 10^3. Three highest relatedness numbers in each row in bold.

	tractor	speed	kmh	recipe	apple pie	bake
How fast does a tractor go?	**408.59**	**7.06**	**-0.02**	-0.04	-0.03	-0.18
How do you cook an apple pie?	-0.03	-0.11	-0.03	**0.48**	**13.28**	**14.14**

Table 2: Scores for selected words in documents from Fig. 1, returned by random walks initialized with the queries (Agirre et al., 2009), where higher scores indicate higher relatedness to the query words. Numbers are scaled by 10^3. Three highest scores in each row in bold.

In order to bridge the gap, IR has resorted to distributional models. Most research concentrated on Query Expansion (QE) methods, which typically analyze term co-occurrence statistics in the corpus and/or in the highest scoring documents in order to select terms for expanding the query (Manning et al., 2009). Pseudo-relevance feedback (PRF) is one of the most notorious techniques in this area. Document expansion (DE) is a natural alternative to QE. Several researchers have used distributional methods from similar documents in the collection in order to expand the documents with related terms that do not actually occur in the document (Liu et al., 2004b; Kurland and Lee, 2004; Tao et al., 2006; Mei et al., 2008; Huang et al., 2009). The work presented here is complementary to those works, in that we explore QE and DE, but use relatedness-based methods (on WordNet) instead of distributional ones.

As an alternative to distributional methods, WordNet has been used with great success in psycholinguistic datasets of word similarity and relatedness, where it surpasses distributional methods based on keyword matches (Agirre et al., 2009; Agirre et al., 2010b). Table 1 shows the relatedness between some words in the queries and documents in Figure 1, as returned by the WordNet relatedness software proposed in (Agirre et al., 2009). As the table shows, the word which is most related to *fast* among the highlighted words in the documents is *speed* and the word most related to *cook* is *recipe*. Given these relatedness scores, each query could be paired with the corresponding document automatically. This example shows the motivation of our approach, where we want to use WordNet-based relatedness to bridge the lexical gap.

WordNet has been applied to IR before. Some authors extended the query with synonyms from WordNet (Voorhees, 1994; Liu et al., 2005), while others have explicitly represented and indexed word senses after performing word sense disambiguation (WSD) (Gonzalo et al., 1998; Stokoe et al., 2003; Kim et al., 2004). More recently, a CLEF task was organized[1] where queries and documents were semantically disambiguated. Some high-scoring participants reported significant improvements when using WordNet information.

The work reported here is novel in that we use WordNet-based relatedness beyond synonymy for query and document expansion. As computing and using word-by-word relatedness as in Table 1 is a costly process, we compute the related words for whole queries or documents instead. Given a query (or full document), a relatedness algorithm using random walks over the WordNet graph (Agirre et al., 2009) returns the concepts which are closely related to the words in the query (or document). This is in contrast to previous WordNet-based works which focused on WSD to replace or supplement words with their senses. Our method discovers important concepts, even if they are not explicitly mentioned in the query or document. Table 2 shows that using this technique for the two queries in Fig. 1, the words in the respective documents get the highest scores.

In this work we adopt a language modeling framework to implement the query likelihood and pseudo-relevance feedback baselines, as well as our relatedness-based query expansion and document expansion methods. In order to test the performance of our method we selected several datasets with different domains, topic typologies and document lengths, including ad-hoc IR, passage retrieval and answer-finding. Given the relevance among the community using WordNet-related methods, we selected the Robust-WSD dataset from CLEF (Agirre et al., 2010c), which is a typical ad-hoc dataset on news.

We think that our method is specially relevant for question answering tasks, but instead of evaluating our method as a component of a complex question-answering system, we preferred to evaluate directly on an answer-finding and a passage retrieval dataset. The first is the Yahoo! Answers dataset, which contains questions and answers as phrased by real users on diverse topics (Surdeanu et al., 2008). The second is ResPubliQA, a passage retrieval task on European Union laws organized at CLEF (Peñas et al., 2009).

The results show that our methods provide improvements in all three datasets when compared to the query likelihood baseline, and that they compare favorably to PRF in two datasets. The analysis suggests that our models and PRF are complementary, in that PRF improves results for easy queries and our models are stronger

[1] http://ixa2.si.ehu.es/clirwsd/

for difficult queries. We also show that our models are more robust in face of sub-optimal parameters.

The work presented in this article follows (Agirre et al., 2010a), which used the same WordNet-based relatedness algorithm for document expansion but in a probabilistic setting, and (Otegi et al., 2011), where we explored query expansion. In the present work we subsume both works providing an implementation on a language modeling framework for IR, and provide additional analysis, including the factors that affect the performance of the algorithm.

The article is structured as follows. We first introduce the random walk model and the relatedness-based models for query and document expansion. Section 3 presents the experimental setup. Section 4 shows our main results, followed by a discussion and analysis section. Section 6 reviews related work. Finally, the conclusions and future work are mentioned.

2 Relatedness-based Expansion Models

In this section we describe the relatedness-based method to expand queries and documents, followed by the expansion models we propose for information retrieval.

2.1 Obtaining Expansion Terms

The key insight of our model is to expand the query or the document with related words according to the background information in WordNet (Fellbaum, 1998), which provides generic information about general vocabulary terms. WordNet groups nouns, verbs, adjectives and adverbs into sets of synonyms (synsets), each expressing a distinct concept. Synsets are interlinked with conceptual-semantic and lexical relations, including hypernymy, meronymy, causality, etc.

In contrast with previous work using WordNet, we select those concepts that are most closely related to the text as a whole. As we will see in the following sections, this text could be a query or a document. For that, we use a technique based on random walks over the graph representation of WordNet concepts and relations (Hughes and Ramage, 2007), which has been successfully used in word similarity (Agirre et al., 2009) and word sense disambiguation (Agirre and Soroa, 2009), and made publicly available by the authors[2].

We represent WordNet as a graph as follows: graph nodes represent WordNet concepts (synsets) and dictionary words; relations among synsets are represented by undirected edges; and dictionary words are linked to the synsets associated to them by directed edges. We use version 3.0, with all relations provided, including the gloss relations. This was the setting obtaining the best results in a word

[2] http://ixa2.si.ehu.es/ukb/

similarity dataset as reported by Agirre et al. (2009).

Given a text and the graph-based representation of WordNet, we obtain a ranked list of WordNet concepts as follows: (1) We first pre-process the text to obtain the lemmas and parts of speech of the open category words. (2) We then assign a uniform probability distribution to the terms found in the text. The rest of nodes are initialized to zero. (3) We compute Personalized PageRank (Haveliwala, 2002) over the graph, using the previous distribution as the reset distribution, and producing a probability distribution over WordNet concepts. The higher the probability for a concept, the more related it is to the given text. (4) Given the topology of the graph, some concepts from very dense areas receive high probabilities, regardless of the words used to initialize the random walk. In order to avoid this effect, we run PageRank over the whole graph, which produces a probability independent of the specific target words, and subtracted the resulting probability from each concept. That is, the score of each concept is obtained subtracting the PageRank from the probability returned by Personalized PageRank. Table 2 shows the scores attained by Personalized PageRank when initialized with each of the two queries. The positive scores show that the Personalized PageRank value is higher than that of the PageRank value, indicating high relevance to the query, while negative scores show the contrary.

Basically, Personalized PageRank is computed by modifying the random jump distribution vector in the traditional PageRank equation. In our case, we concentrate all probability mass in the concepts corresponding to the words in the text. Let G be a graph with N vertices $v_1, \dots, v_N$ and d_i be the outdegree of node i; let M be a $N \times N$ transition probability matrix, where $M_{ji} = \frac{1}{d_i}$ if a link from i to j exists, and zero otherwise. Then, the calculation of the *PageRank vector* $\mathbf{Pr}$ over G is equivalent to resolving Equation (1).

$$\mathbf{Pr} = cM\mathbf{Pr} + (1-c)\mathbf{v} \tag{1}$$

In the equation, $\mathbf{v}$ is a $N \times 1$ vector and c is the so called *damping factor*, a scalar value between 0 and 1. The first term of the sum on the equation models the voting scheme described in the beginning of the section. The second term represents, loosely speaking, the probability of a surfer randomly jumping to any node, e.g. without following any paths on the graph. The damping factor, usually set in the $[0.85..0.95]$ range, models the way in which these two terms are combined at each step.

The second term on Eq. (1) can also be seen as a smoothing factor that makes any graph fulfill the property of being aperiodic and irreducible, and thus guarantees that PageRank calculation converges to a unique stationary distribution.

In the traditional PageRank formulation the vector $\mathbf{v}$ is a stochastic normalized vector whose element values are all $\frac{1}{N}$, thus assigning equal probabilities to all nodes in the graph in case of random jumps. In the case of Personalized PageRank as used here, $\mathbf{v}$ is initialized with uniform probabilities for the terms in the document, and 0 for the rest of terms.

PageRank is actually calculated by applying an iterative algorithm which computes Eq. (1) successively until a fixed number of iterations are executed. In our case, we used a publicly available implementation[3] with the default values provided by the software, i.e. a damping value of 0.85, and 30 iterations.

In order to select the expansion terms, we choose the top N highest scoring concepts, and get all the words that lexicalize the given concept. When expanding the documents (see Section 2.2) we follow the work in (Agirre et al., 2010a), and fix N to 100. When expanding the queries (cf. Section 2.3) we explore several values of N, and tune it in order to get the optimum value, as discussed in Section 3.

Figure 2 shows the expansion process for a document. After applying the graph algorithm to the document in 2a, we obtain the concepts with the *synset* numbers, as partially shown in 2b, sorted by relatedness to the document in decreasing order. The words that lexicalize these concepts are shown in Figure 2c. The words that are in the original document are in **bold**, their synonyms are in *italic* and other related words are highlighted. In addition to synonyms, words that are not in the document but are related to related concepts are suggested for expansion, as for instance, *phone company* and *computer*.

Similarly, Figure 3 illustrates query expansion. After applying the graph algorithm to the query in 3a, we obtain the concepts with the *synset* numbers, as partially shown in 3b, sorted by relatedness to the query in decreasing order. The words that lexicalize these concepts are shown in Figure 3c. We can see that words like *vehicle* and *distance*, which are not in the query but are related to it, are suggested for expansion.

2.2 Relatedness-based Document Expansion (RDE)

The relatedness-based document expansion approach requires the document collection to be pre-processed to obtain a list of most related terms for each document, following the method explained in Section 2.1. These related terms are indexed separately. Documents are ranked by their probability of generating the query (Ponte and Croft, 1998), where this probability is estimated as a weighted combination of query likelihoods from the different document representations:

$$P_{\scriptscriptstyle RDE}(Q \mid \Theta_{\scriptscriptstyle RDE}) = P(Q \mid \Theta_D)^w P(Q \mid \Theta_E)^{1-w} \tag{2}$$

where Θ_D and Θ_E are the language models estimated from the original document representation and the expanded document representation, respectively, and w is

[3] http://ixa2.si.ehu.es/ukb/

You should only need to turn off **virus** and **anti**-spy not uninstall. And that's done within each of the **softwares** themselves. Then turn them back on later after **installing** any **DSL softwares**.

(a)

06566077-n	→ *computer software, package*, **software**, *software package, software program, software system*
03196990-n	→ *digital subscriber line*, **dsl**
09796809-n	→ **anti**
01569566-v	→ *instal*, **install**, *put in, set up*
01328702-n	→ **virus**
04402057-n	→ line, phone line, subscriber line, telephone circuit, telephone line
08186221-n	→ phone company, phone service, telco, telephone company, telephone service
03082979-n	→ computer, computing device, computing machine, data processor, electronic computer

(b) (c)

Figure 2: Example of the expansion for document 1005121303076 of the Yahoo! dataset: (a) original document; (b) *synset* numbers for some of the concepts to be expanded; (c) words obtained from the expansion.

the weight given to the original document language model set in the [0..1] range. Query likelihood is estimated following the multinomial distribution (we show the document model, but the expansion model is analogous):

$$P(Q \mid \Theta_D) = \prod_{i=1}^{|Q|} P(q_i \mid \Theta_D)^{\frac{1}{|Q|}} \tag{3}$$

where q_i is a query term of query Q and $|Q|$ is the length of Q. And following the Dirichlet smoothing (Zhai and Lafferty, 2001) we have

$$P(q_i \mid \Theta_D) = \frac{tf_{q_i D} + \mu \frac{tf_{q_i C}}{|C|}}{|D| + \mu} \tag{4}$$

where $tf_{q_i D}$ and $tf_{q_i C}$ are the frequency of the query term q_i in the document D and the entire collection, respectively, and μ is the smoothing free parameter.

2.3 Relatedness-based Query Expansion (RQE)

In this approach, we expand each query with the terms obtained following the expansion technique described in Section 2.1. Thus, we retrieve documents based

What is the **lowest speed** in **miles per hour** which can be **shown** on a **speedometer**?

(a)

(b)	(c)
04273796-n	→ **speedometer**, *speed indicator*
15280346-n	→ **miles per hour**, *mph*
03791235-n	→ motor vehicle, automotive vehicle
00393149-r	→ **low**
00922867-v	→ *read*, *register*, **show**, *record*
04524313-n	→ vehicle
15282696-n	→ **speed**, *velocity*
06879521-n	→ *display*, **show**
05129565-n	→ distance, length

Figure 3: Example of the expansion for query 91 of the ResPubliQA dataset: (a) original query; (b) *synset* numbers for some of the concepts to be expanded; (c) words obtained from the expansion.

on the expanded query, which contains the original terms of the query and the expansion terms. Documents are ranked by their probability of generating the whole expanded query (Q_{RQE}), which is given by:

$$P_{RQE}(Q_{RQE} \mid \Theta_D) = P(Q \mid \Theta_D)^w P(Q' \mid \Theta_D)^{1-w} \tag{5}$$

where w is the weight given to the original query and Q' is the expansion of query Q. The query likelihood probability $P(Q \mid \Theta_D)$ is again calculated following a multinomial distribution and Dirichlet smoothing, as specified in Equation 3 and Equation 4. The probability of generating the expansion terms is defined as

$$P(Q' \mid \Theta_D) = \prod_{q'_i}^{|Q'|} P(q'_i \mid \Theta_D)^{\frac{w_i}{W}} \tag{6}$$

where q'_i is a expansion term, $W = \sum_{i=1}^{|Q'|} w_i$ and w_i is the weight we give to a expansion term, which we can see as the relatedness between the original query Q and the expansion term, and is computed as

$$w_i = P(q' \mid Q) = \sum_{j=1}^{N} P(q' \mid c_j) P(c_j \mid Q) \tag{7}$$

where c is a concept returned by the expansion algorithm (see Section 2.1), N is the number of concepts we chose for the expansion, $P(q' \mid c_j)$ is estimated using the sense probabilities estimated from Semcor (i.e. how often the query term q' occurs with sense c_j), and $P(c_j \mid Q)$ is the similarity weight that the mentioned expansion algorithm assigned to c_j concept.

3 Experimental setup

In order to test the performance of our method we selected several datasets from different domains, topics, typologies and document lengths, including ad-hoc IR, passage retrieval and answer finding. Table 3 shows some statistics for the three selected datasets. Note that all three datasets have a separate subset for developing and training the systems.

The first is the English dataset of the **Robust-WSD** task at CLEF 2009 (Agirre et al., 2010c), a typical ad-hoc dataset on news. This dataset has been widely used among the community interested on WSD and WordNet-related methods for the following reasons. Note that we need to reuse existing relevance judgments (customary on standard datasets), which were pooled among participants of the task, and thus systems that are based on different expansion strategies (e.g. WSD or WordNet) might return relevant documents which were not available in the pool that was manually judged at competition time. For this reason, the organizers of the Robust-WSD dataset used relevance judgments obtained pooling both monolingual and multilingual runs. The organizers of the exercise hoped that the inclusion of multilingual runs, with a larger variability due to translation strategies, would include relevance judgments for query-document pairs where different wording had been used (Agirre et al., 2010c).

The documents in the Robust-WSD comprise news collections from LA Times 94 and Glasgow Herald 95. The topics are statements representing information needs, consisting of three parts: a brief title statement; a one-sentence description; a more complex narrative describing the relevance assessment criteria. Following the rules of the Robust-WSD task, we use the title and the description parts of the topics in our experiments.

As we think that our method is specially relevant for question answering, we also evaluated our methods on an answer-finding dataset, Yahoo! Answers, which contains questions and answers as phrased by real users on diverse topics (Surdeanu et al., 2008), and a paragraph retrieval task, ResPubliQA, which related to European Union laws, organized at CLEF (Peñas et al., 2009).

The **Yahoo! Answers** corpus is a subset of a dump of the Yahoo! Answers web site, where people post questions and answers, all of which are public to any web user willing to browse them[4] (Surdeanu et al., 2008). The task is to find the

[4] Yahoo! Webscope dataset "ydata-yanswers-manner-questions-v1_0" http://webscope.sandbox.yahoo.com/

	documents		training queries		test queries	
dataset	#	length	#	length	#	length
Robust	166,754	532	150	8.37	160	8.64
Yahoo!	89,610	104	1,000	11.32	30,000	11.25
ResPubliQA	1,379,011	20	100	10.22	500	10.71

Table 3: Statistics for each of the dataset: number of documents, average document length, number of queries and average query length for train and test.

	QL	PRF				RDE		RQE		
dataset	μ	μ	d	t	w	μ	w	μ	N	w
Robust	1000	1000	10	50	0.3	1200	0.8	2000	100	0.5
Yahoo!	200	200	2	20	0.8	200	0.8	200	50	0.7
ResPubliQA	100	100	10	30	0.8	100	0.7	100	125	0.7

Table 4: Optimal values in each dataset for free parameters.

document which contains the answer. Before releasing the dataset, the Yahoo team filtered the dataset as follows: (1) It comprised a subset of the questions, selected for their linguistic properties (for example they all start with ”how {to — do — did — does — can — would — could — should}”). (2) Questions and answers of obvious low quality were removed. (3) The document set was created with the best answer of each question (only one for each question). We use the dataset as released by its authors.

The other collection is the English dataset of **ResPubliQA** exercise at the Multilingual Question Answering Track at CLEF 2009 (Peñas et al., 2009). The exercise is aimed at retrieving paragraphs that contain answers to a set of 500 natural language questions. The document collection is a subset of the JRC-Acquis Multilingual Parallel Corpus, and consists of 21,426 documents for English which are aligned to a similar number of documents in other languages[5]. For evaluation, we used the gold standard released by the organizers, which contains a single correct passage for each query.

Documents and queries have been lemmatized and tagged with parts of speech. The Robust dataset is already tagged with WordNet senses, as well as with lemmas and parts of speech. We have used OpenNLP[6] to tag the other two datasets.

Our experiments were performed using the Indri search engine (Strohman et al., 2005), which is a part of the open-source Lemur toolkit[7]. To determine whether the

[5] Note that Table 3 shows the number of paragraphs, which conform the units we indexed.

[6] http://incubator.apache.org/opennlp/

[7] http://www.lemurproject.org

two expansion models we developed are useful to improve retrieval performance, we set up a number of experiments in which we compared our expansion models with other retrieval approaches. We used two baseline retrieval approaches for comparison purposes. One of the baselines is the default query likelihood (**QL**) language modeling method implemented in the Indri search engine. The other one is pseudo-relevance feedback (**PRF**) using a modified version of Lavrenko's relevance model (Lavrenko and Croft, 2001), where the final query is a weighted combination of the original and expanded queries, analogous to Eq. 5. As in our own model presented in the previous sections, we chose the Dirichlet smoothing method for the baselines. We consider **QL** and **PRF** to be strong, reasonable baselines.

All the methods have several free parameters. The PRF model has three parameters: number of documents (d) and terms (t), and w (cf. Eq. 5). The RDE model also has w (cf. Eq. 2). The RQE model has two parameters: w (cf. Eq.. 5) and N the number of concepts for the expansion (Eq. 7). In addition, all methods use Dirichlet smoothing, which has a smoothing parameter μ. We used the train part of each dataset to tune all these parameters via a simple grid-search. The μ parameter was tested on the [100,1200] range for ResPubliQA and Yahoo! and [100,2000] for Robust, with increments of 100. The w parameter ranged over [0,1] with 0.1 increments. The d parameter ranged over [2,50] and the t and N in the range [1,200] (we tested 10 different values in the respective ranges). The parameter settings that maximized mean average precision for each model and each collection are shown in Table 4.

4 Results

In this section we present the results for the baseline query likelihood model (QL), the pseudo relevance feedback model (PRF) and our relatedness-based expansion models: query expansion (RQE) and document expansion (RDE).

The main evaluation measure for Robust is Mean Average Precision (MAP), as customary. In two of the datasets (Yahoo! and ResPubliQA), there is a single correct answer per topic, and therefore we use Mean Reciprocal Rank (MRR). Note that in this setting MAP is identical to MRR. We also report Mean Precision at ranks 5 and 10 (P@5 and P@10). GMAP is also included, we will introduce and mention it afterwards. Statistical significance was computed using Paired Randomization Test (Smucker et al., 2007). In the tables throughout the paper, we use * to indicate statistical significance for 90% confidence level, ** for 95% and *** for 99%.

		QL	PRF		RDE		RQE	
dataset	measure	result	result	Δ QL	result	Δ QL	result	Δ QL
Robust	MAP	0.3322	**0.3669** ***	10.44%	**0.3387** **	1.95%	**0.3367**	1.36%
	GMAP	0.1321	**0.1438** ***	8.90%	**0.1351**	2.26%	**0.1434** **	8.59%
	P@5	0.4250	**0.4363**	2.65%	**0.4300**	1.18%	0.4225	-0.59%
	P@10	0.3531	**0.3738** ***	5.84%	**0.3556**	0.71%	**0.3581**	1.42%
Yahoo!	MRR	0.2636	**0.2640**	0.15%	**0.2752** ***	4.42%	**0.2722** ***	3.26%
	P@5	0.0667	0.0663 **	-0.56%	**0.0691** ***	3.64%	**0.0688** ***	3.21%
	P@10	0.0395	**0.0396**	0.25%	**0.0412** ***	4.29%	**0.0410** ***	3.91%
ResPubl.	MRR	0.4877	0.4633 ***	-5.00%	**0.4926**	1.02%	**0.4978**	2.07%
	P@5	0.1244	0.1200 *	-3.54%	0.1236	-0.64%	**0.1268**	1.93%
	P@10	0.0680	0.0678	-0.29%	**0.0694**	2.06%	0.0678	-0.29%

Table 5: Results of all methods. Δ columns show relative improvement with respect to QL. Bold means better than QL.

Comparison with Respect to QL

Our main results are shown in Table 5. The first three columns of results in Table 5 shows the results for QL and PRF, and the performance difference between them. The results for PRF are mixed. In Yahoo! the improvement is small in MRR and P@10, without statistical significance, but P@5 is lower. In ResPubliQA the results are bad, with statistical significant degradation in MRR. In contrast, it is very effective in the Robust dataset, with dramatic improvements, specially in MAP. This finding is common for relevance feedback algorithms, which is a recall-enhancing technique at the cost of precision (Manning et al., 2009; Ruthven and Lalmas, 2003). The results that we obtained in Robust are partly consistent with this statement, as apart from improving recall (5.81%), we also have improved precision at early rank (in a less degree, but still significant for P@10). Note that all differences are statistically significant, except for P@5. As MAP encapsulates both precision and recall aspects, is the one with largest improvement. Note that there is nothing to say about recall in the other datasets as there is only one relevant document for each query.

Continuing rightwards with Table 5, the last columns show the results for RDE and RQE, together with their difference with respect to QL. RDE and RQE improve QL in nearly all datasets and measures. The strongest improvements are in Yahoo!. For Robust, the improvements in precision are not so substantial, but the recall improvements are significant, 1.36% for RDE and 4.67% for RQE.

Comparison with Respect to PRF

Results of PRF, RDE and RQE are repeated in Table 6 to better compare results with respect to PRF. Note that figures in bold mean better performance than PRF.

		PRF	RDE		RQE	
dataset	measure	result	result	Δ PRF	result	Δ PRF
Robust	MAP	0.3669	0.3387 ***	-7.69%	0.3367 ***	-8.22%
	GMAP	0.1438	0.1351 **	-6.10%	0.1434	-0.29%
	P@5	0.4363	0.4300	-1.43%	0.4225	-3.15%
	P@10	0.3738	0.3556 ***	-4.85%	0.3581 *	-4.18%
Yahoo!	MRR	0.2640	**0.2752** ***	4.26%	**0.2722** ***	3.11%
	P@5	0.0663	**0.0691** ***	4.22%	**0.0688** ***	3.79%
	P@10	0.0396	**0.0412** ***	4.03%	**0.0410** ***	3.65%
ResPubliQA	MRR	0.4633	**0.4926** ***	6.33%	**0.4978** ***	7.44%
	P@5	0.1200	**0.1236**	3.00%	**0.1268** ***	5.67%
	P@10	0.0678	**0.0694**	2.36%	0.0678	0.00%

Table 6: Results of PRF, RDE and RQE. Δ columns show relative improvement with respect to PRF. Bold means better than PRF.

We can see that the best results vary across datasets, with PRF yielding the best results for Robust, RDE for Yahoo! and RQE for ResPubliQA. Both RDE and RQE improve over PRF in Yahoo! and ResPubliQA, with mostly statistically significant differences.

PRF is known to perform well for some topics and datasets but not for others (Ruthven and Lalmas, 2003). We have included results for the GMAP in the Robust dataset (it is not relevant in the other datasets). GMAP tries to promote systems which are able to perform well for all topics, in contrast to systems that perform better in some but worse in others (Robertson, 2006). The figures show that RQE gets worse results for MAP but approximates the performance of PRF for GMAP. In that way, we will analyze the results of each of the queries one by one and we will see that RDE and RQE perform better for some queries, concretely, for difficult queries.

5 Analysis and discussion

In order to understand the behavior of our method, we performed some detailed analysis. First of all, we will analyze the performance of each technique on a query by query basis. Next, we will analyze the intercollection generalization of each technique, followed by their sensitivity to model parameters. We will also check the associated computational costs. The relation between our techniques and word sense disambiguation will be examined next. Finally, we will summarize the main points.

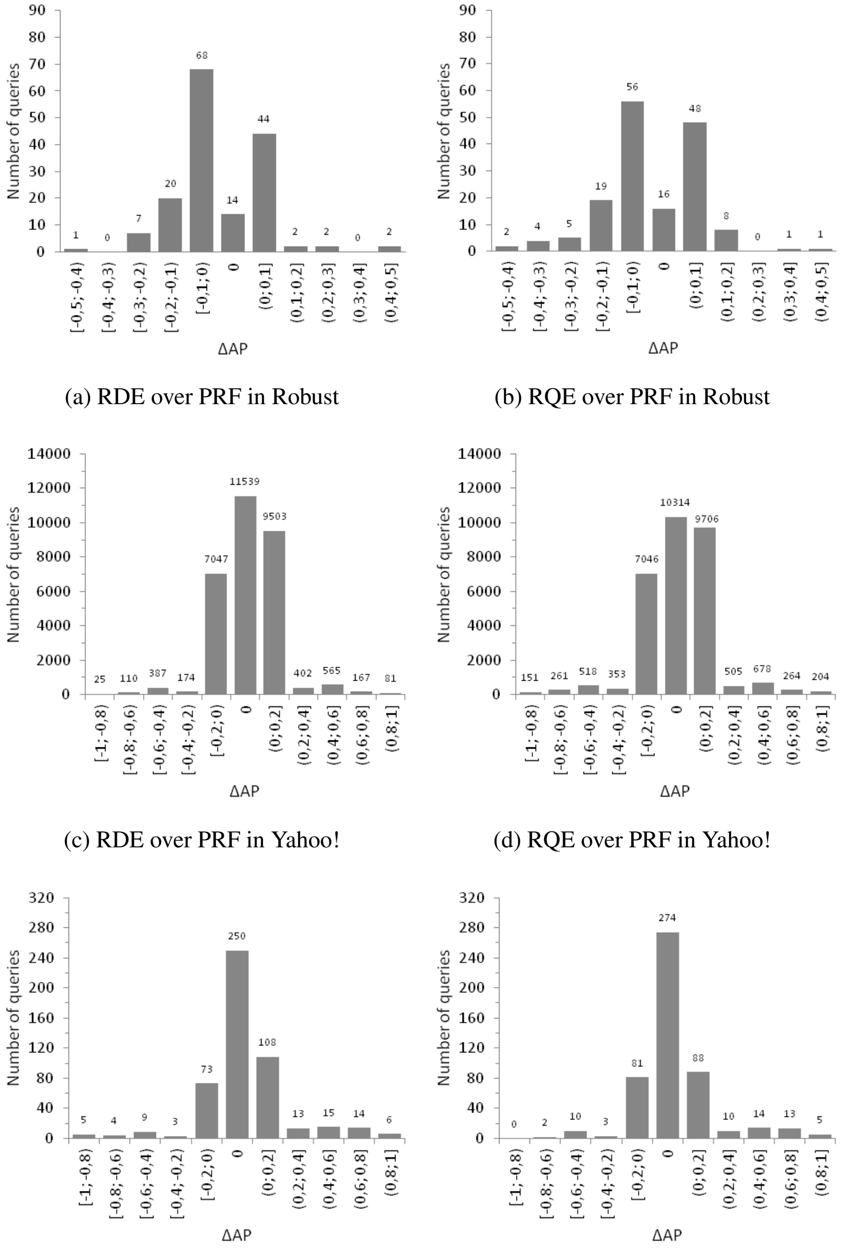
5.1 Analyzing queries

We first compared the performance of RDE and RQE with respect to PRF, calculating, for each query, the difference with respect to PRF in terms of average precision (ΔAP). We sorted the queries by decreasing ΔAP, grouped the queries according to ΔAP ranges, and plotted the number of queries falling into each bucket, as shown in Figure 4. A positive difference indicates an improvement over PRF for those queries.

The plot for Robust confirms that PRF performs better than RDE and RQE in this dataset, with more queries with negative ΔAP, but note that our expansion models outperform PRF for some of the queries. The situation is reversed for Yahoo! and ResPubliQA, with more queries getting worse results with PRF. In addition, the plots show that for ResPubliQA the majority of queries get the same performance with either method (ΔAP equals 0). In Yahoo! the trend is similar, but less steep. Surprisingly, in the Robust dataset the number of queries getting the same performance is very low, showing that PRF and our methods are complementary. The plots of our methods versus the QL baseline show the same trends. We have omitted them for the sake of brevity.

In order to study the behavior of our expansion models with respect to easy and hard topics, Figure 5 shows the performance of each query according to MAP (MRR for Yahoo! and ResPubliQA) obtained by our expansion methods (vertical axis) and PRF (horizontal axis). Hard queries are those which get low performance, and are located close to the origin, on the bottom-left quadrant. The best fitting line for the Robust plots show that PRF does better than RDE and RQE on easy queries (i.e. those with high performance, on the right), but the performance on difficult queries is better for RQE and specially RDE. The best fitting lines for Yahoo! and ResPubliQA also show that the RDE and RQE are performing better on difficult queries. It thus seems that PRF and our expansion techniques are complementary, with one doing better on easy queries and the others doing better on hard queries. The plots with respect to the QL baseline (not shown for the sake of brevity) are very similar, with RQE and RDE doing specially better for queries with low performance.

Figure 6a shows an example of a difficult query from ResPubliQA. Both QL and PRF obtain a low MRR for this query (0.33), while question expansion (RQE) gets a perfect score of 1. Figure 6b shows some of the words proposed by RQE for query expansion. The expansion words include *vehicle*, *distance* and *mph*, which are contained in the relevant document (cf. Figure 6c).



(a) RDE over PRF in Robust

(b) RQE over PRF in Robust

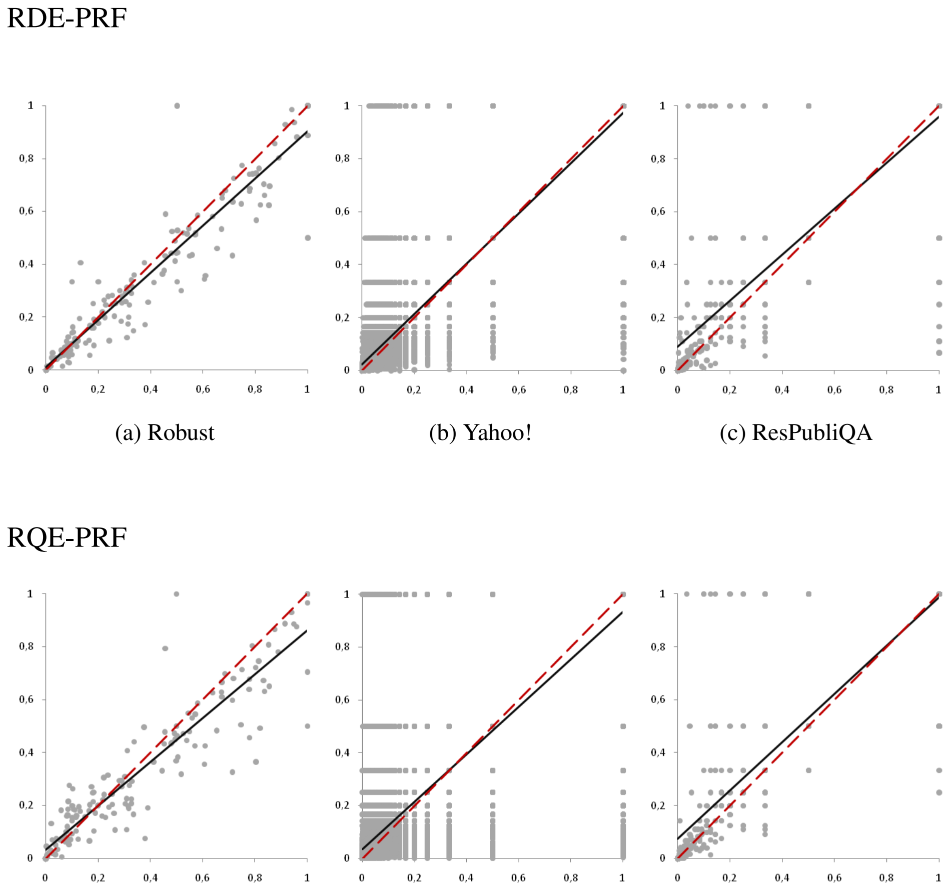
(c) RDE over PRF in Yahoo!

(d) RQE over PRF in Yahoo!

(e) RDE over PRF in ResPubliQA

(f) RQE over PRF in ResPubliQA

Figure 4: Queries grouped by differences in improvement over PRF for all datasets.



(a) Robust (b) Yahoo! (c) ResPubliQA

(d) Robust (e) Yahoo! (f) ResPubliQA

Figure 5: MAP plots (MRR for Yahoo! and ResPubliQA) of all queries, comparing RDE and RQE (y axis) to PRF (x axis). RDE plots on the top row, RQE on the bottom. Best fitting linear trend (solid) and equality ($y = x$, dashed) lines are also shown.

What is the lowest speed in miles per hour which can be shown on a speedometer?

(a) The English query (number 91).

speedometer speed_indicator miles_per_hour **mph** motor_vehicle automotive_vehicle low read register show record **vehicle** speed velocity display show **distance** length

(b) Some of the words obtained by query expansion.

where a **vehicle** is intended for sale in a Member State where imperial **distances** are used, the speedometer must also be graduated in **mph** (miles per hour), with subdivisions of 1, 2, 5 or 10 **mph**. Marked numerical speed value intervals must not exceed 20 **mph** and must begin at either 10 **mph** or 20 **mph**;

(c) A relevant document for the query (jrc32000L0007-en/92).

Figure 6: A difficult query from ResPubliQA which has been correctly answered with query expansion, including expansion terms proposed by relatedness and the relevant document.

5.2 Intercollection generalization

In Table 4 we showed the optimum parameters for each technique and dataset, developed according to cross-validation results on the training subset of each dataset. In most practical situations, though, there is no training data to adjust the parameters, and parameters estimated on other scenarios are used, with some performance loss.

In this section we analyze the behavior of the methods when parameters adjusted on other datasets are used. This analysis was named *intercollections generalization* in (Metzler, 2006). Metzler proposed to measure generalization properties of a model by computing the effectiveness ratio, which is the ratio of the observed effectiveness of a target model with respect to the optimal effectiveness (when optimal values in train are used). Thus, an effectiveness ratio of 100% represents a model that generalizes optimally. We take a simpler approach, and apply the idea directly to MAP (or MRR) values, obtaining a MAP (or MRR) ratio for each combination of training/testing datasets, and macro-averaging across all possible combinations (cf. Table 7). Note that, in order to keep the analysis simpler, we kept μ fixed at the optimal values. The smoothing parameter μ has a direct relation with document length, and can be thus adjusted according to past experiences easily.

For instance, the *Rob* column for PRF shows a negative ratio of -6.3 when Yahoo! is used to estimate the parameters and the system is tested on Robust, meaning that the performance is 6.3% less than when using parameters estimated on the

	PRF			RDE			RQE		
	Rob	Yah	Res	Rob	Yah	Res	Rob	Yah	Res
Robust	—	-9.7	-18.8	—	0.0	1.0	—	-4.3	-7.6
Yahoo!	-6.3	—	1.7	0.0	—	1.0	0.5	—	-0.4
ResPubliQA	-7.3	-0.9	—	-0.7	-0.7	—	0.9	1.3	—
average	-6.9%			0.11%			-1.60%		

Table 7: Effectiveness ratios for *intercollections generalization* (based on MAP or MRR). The first column specifies the training dataset for the respective row and the columns the test dataset. Empty slots correspond to the reference (0.0%). The average row shows the macro-average of all differences above it.

training subset of Robust. The figures in the table show that RDE is the least sensitive to optimization (it actually improves performance), with RQE losing some performance and PRF with the largest losses, -6.9%.

5.3 Sensitivity to model parameters

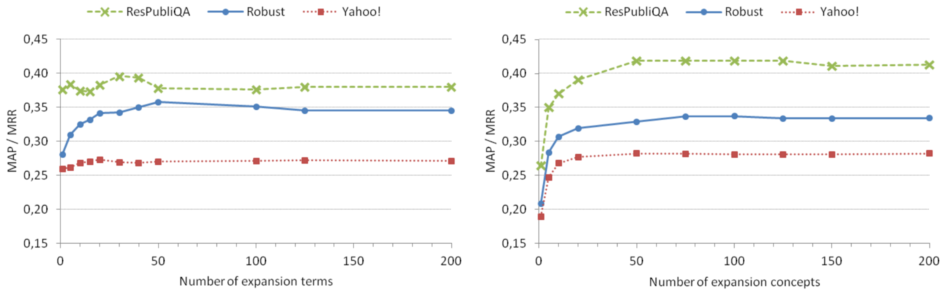
We will now explore the sensitivity of the results to changes in the parameters. For that purpose, we will display the effects of the results for different models varying one parameter each time, maintaining the other parameters in their optimal values. This analysis has been performed on the training subsets of the dataset, and thus the figures reported here are not directly comparable to those obtained on the test datasets.

The number of terms

The main parameter when expanding queries is the number of terms that are added to the query. Figure 7 shows the behaviour of PRF and RQE with respect of the number of query terms, when keeping the other parameters fixed. Figure 7a shows that PRF behaves differently on each datasets, with maximum performances at different points. Figure 7b shows that, for RQE, all datasets respond similarly to each newly added term, growing steadily until they plateau at around 20-75 concepts.

The weight of the original query or document

Figure 8 displays the effect of varying the weight of the original query or document. For PRF we observe that the best value is very different for each dataset, with approximately 0.3 for Robust and 0.8 for the other two. RDE obtains the best result for similar values, around 0.7 or 0.8. For RQE the best results range from 0.5 for Robust to 0.7 for the other two. RDE shows the most consistent behavior



(a) PRF

(b) RQE

Figure 7: Results for varying the number of expansion terms for each of the models.

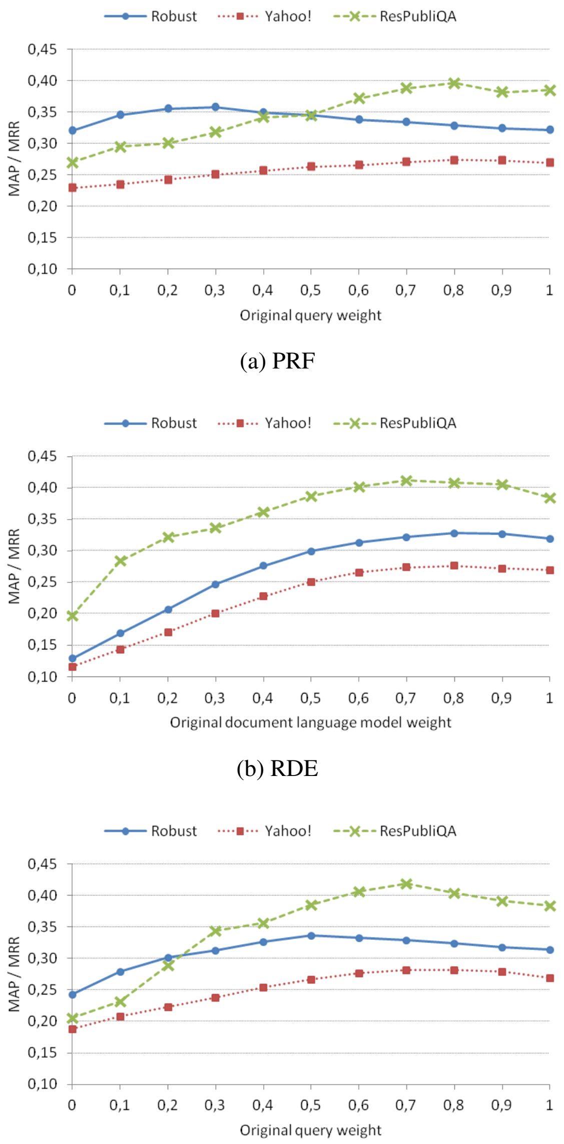
from all three methods, with PRF behaving worst.

Note that when the weight for the original query or the original document language model is 0 the results show the performance of using PRF or expansions alone. PRF terms seem to yield good results on their own, with RQE terms performing slightly worse. Regarding RDE, when the weight is 0 the query is processed using the terms that expand the document alone, and the results are very low.

5.4 Computational Cost

Improved performance comes at a computational cost. A query of the Robust test set (160 queries) takes 0.14 seconds on average for the QL baseline on a server with two Intel QuadCore Xeon X5460 processors at 3160MHz with 32 GB of memory. PRF takes 2.75 seconds, RDE 0.37 seconds, and RQE 8.53 seconds per query on average. The larger cost for PRF and RQE at query time comes from the added complexity of examining additional terms in the expanded query. Given that RQE is using more terms than PRF, the cost is higher. The added cost for RDE is the overhead of searching in two indexes and merging the results from both indexes.

In addition, running the random walk on one query or document takes approximately 6 seconds. In the case of RDE, the process can be easily parallelized and done in batch in advance. In the case of RQE, query time computations could be sped up using less iterations in the random walk algorithm, or we could have precomputed the random walks for each word in advance. In the later case, at query time one would just need to do a linear combination of the probability vectors of the words in the query. For the future, we would like to check whether there is any performance loss involved in these computational improvements.



(a) PRF

(b) RDE

(c) RQE

Figure 8: Plots of the results when varying the weight of the original query for each of the models.

Title: Computer Mouse RSI
Desc: Find documents that report on computer mouse repetitive strain injuries (RSI).
Narr: Relevant documents report injuries that are caused by the continuous use of a computer mouse. Documents proposing ways to avoid repetitive strain injuries (RSI) when using the computer are also relevant.

(a) Topic 10.2452/064-AH from Robust dataset.

computer mouse rsi repetitive strain injuries

(b) The formulated query using the *title* and *desc* fields of the topic.

Figure 9: A query example from the Robust dataset to illustrate polysemy.

5.5 Relation to WSD

mouse-1: any of numerous small rodents typically resembling diminutive rats having pointed snouts and small ears on elongated bodies with slender usually hairless tails.
mouse-4, computer mouse: a hand-operated electronic device that controls the coordinates of a cursor on your computer screen as you move it around on a pad; on the bottom of the device is a ball that rolls on the surface of the pad.

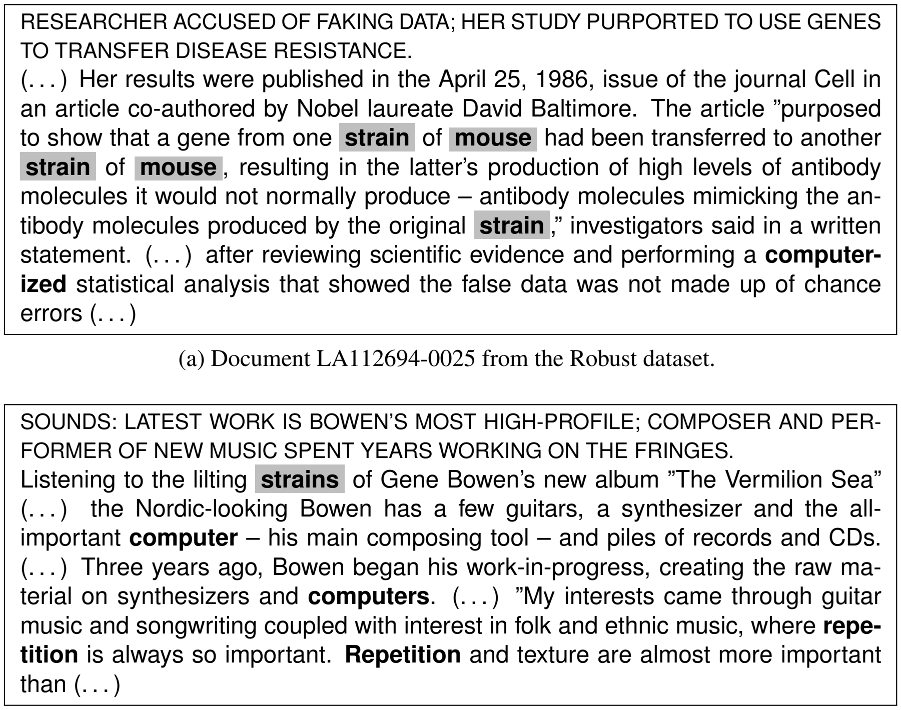
(a)

strain-3, tune, melody, air, melodic line, line, melodic phrase: a succession of notes forming a distinctive sequence; *she was humming an air from Beethoven*.
strain-5, breed, stock: a special variety of domesticated animals within a species; *he experimented on a particular breed of white rats*.
strain-7: injury to a muscle (often caused by overuse), results in swelling and pain.

(b)

Figure 10: Some nominal senses for the words *mouse* and *strain*, as given by WordNet.

As mentioned in the introduction, most of the techniques based on WordNet have focused on performing explicit word sense disambiguation before doing expansion. Our method initializes the random walk using a set of words. If the words are polysemous, the random walk follows all senses of the words, but the probabilities of the senses which are close to other senses in the input set raise, and the probabilities of unrelated senses decrease. When selecting the concepts for expansion, those concepts which are close to the intended senses of the input words will get higher



RESEARCHER ACCUSED OF FAKING DATA; HER STUDY PURPORTED TO USE GENES TO TRANSFER DISEASE RESISTANCE.
(. . .) Her results were published in the April 25, 1986, issue of the journal Cell in an article co-authored by Nobel laureate David Baltimore. The article ”purposed to show that a gene from one **strain** of **mouse** had been transferred to another **strain** of **mouse**, resulting in the latter’s production of high levels of antibody molecules it would not normally produce – antibody molecules mimicking the antibody molecules produced by the original **strain**,” investigators said in a written statement. (. . .) after reviewing scientific evidence and performing a **computerized** statistical analysis that showed the false data was not made up of chance errors (. . .)

(a) Document LA112694-0025 from the Robust dataset.

SOUNDS: LATEST WORK IS BOWEN’S MOST HIGH-PROFILE; COMPOSER AND PERFORMER OF NEW MUSIC SPENT YEARS WORKING ON THE FRINGES.
Listening to the lilting **strains** of Gene Bowen’s new album ”The Vermilion Sea” (. . .) the Nordic-looking Bowen has a few guitars, a synthesizer and the all-important **computer** – his main composing tool – and piles of records and CDs. (. . .) Three years ago, Bowen began his work-in-progress, creating the raw material on synthesizers and **computers**. (. . .) ”My interests came through guitar music and songwriting coupled with interest in folk and ethnic music, where **repetition** is always so important. **Repetition** and texture are almost more important than (. . .)

(b) Document LA063094-0099 from the Robust dataset.

Figure 11: Some non-relevant documents retrieved for the given query in the previous example, due to polysemy.

scores than the rest. Table 2 shows this effect, with *tractor*, *speed* and *kmh* getting high scores for the first query (“*How fast does a tractor go*”) and *recipe*, *apple pie* and *bake* getting higher scores for the second query (“*How do you cook an apple pie*”). We thus interpret that our algorithm does implicit word sense disambiguation. In fact, an algorithm based on random walks has been successfully used to perform word sense disambiguation (Agirre and Soroa, 2009).

For instance, Figure 9 shows an information need and respective query from the Robust dataset. Some of the terms in this query are polysemous. Figure 10 shows a subset of the senses of the words *mouse* and *strain* in WordNet[8]. Given this query, the IR baseline system retrieves, among others, the documents displayed in Figure 11, which are not relevant to the given query. These documents are considered to be relevant by the system because they contain two of the query words (highlighted), but used with a different meaning. For instance, the word *mouse* in the query is

[8] http://wordnetweb.princeton.edu/perl/webwn

used in the computer mouse sense, whereas in the document it refers to a kind of animal. In the case of the word *strain*, the query refers to an injury in the muscle, while the document in Figure 11a refers to a variety of an animal, and the document in Figure 11b refers to a tune or melody of music.

For the future, we would like to check whether a state-of-the-art dedicated system doing WSD prior to running the random walks would improve the performance of our expansion methods.

5.6 Summary

We have shown that our two methods provide improvements in all three datasets when compared to the query likelihood baseline. PRF is beneficial in two datasets, but degrades performance in ResPubliQA. RDE and RQE compare favorably to PRF in two datasets, but perform worse in Robust. Our analysis shows that our models and PRF are complementary, in that PRF is better for easy queries and our models are stronger for difficult queries. Note also that, in the robust dataset, there are very few queries where PRF and our models perform equally, underscoring the possibilities for future combinations. RQE, and specially RDE, generalize very well across collections, with PRF suffering 7% on average. The analysis of each individual parameter also show that RQE and RDE behave nicely regarding the number of terms and the weight of the original query or document. The analysis of performance shows that, at query time, RDE is the most efficient. Finally, we have shown that our method is implicitly doing WSD, and that it could possibly be improved using other WSD methods.

6 Related Work

Our work stems from the use of random walks over the WordNet graph to compute the similarity and relatedness between pairs of words (Hughes and Ramage, 2007). In this approach, WordNet is represented as a graph, with word senses and concepts as vertices, and relations between concepts as edges (cf. Section 2.1 for more details). The method first computes a random walk over the graph for a single word, obtaining the probability distribution over all WordNet concepts. The probability distribution represents the meaning of the word in the concept space. To judge the degree of similarity between any two words, it suffices to compute the similarity of the probability distributions of each word. In later work different configurations of the graph were tested (Agirre et al., 2009; Agirre et al., 2010b), obtaining the best results on a word similarity benchmark among WordNet-based systems to date. Note that the results were comparable to the results of a distributional similarity method which used a crawl of the entire web (Agirre et al., 2009). The same method also ranks highest among WordNet-based methods for relatedness (Agirre et al., 2009), where the task is to judge the degree to which the words are related to each other. The random-walk software is open-source[9], and it is the same as we use in this work.

As mentioned in the introduction, Information Retrieval relies heavily on keyword match. As an alternative to bridge lexical mismatches between query and documents, query expansion and document expansion methods have been proposed.

Query expansion (QE) methods analyze user query terms and incorporate related terms automatically (Voorhees, 1994), and are usually divided into local and global methods. Local methods adjust a query relative to the documents that initially appear to match the query (Manning et al., 2009). Pseudo-relevance feedback (PRF) is one of the most widely used expansion methods (Rocchio, 1971; Xu and Croft, 1996). This method assumes that the top-ranked documents returned by the original query are relevant (and in some cases, that low-ranked documents are irrelevant), and selects additional query terms from the top-ranked documents. Since Rocchio presented an algorithm for relevance feedback (Rocchio, 1971), lots of variations have been developed. The TREC 2008 Relevance Feedback Track results confirmed that relevance feedback consistently improves different kinds of retrieval models, but the amount of relevance information needed to improve results and the use or not of non-relevant information varied among systems (Buckley and Sanderson, 2008).

Global methods are techniques for expanding query terms without checking the results returned by the query. These methods analyze term co-occurrence statistics in the entire corpus or use external knowledge sources to select terms for expansion (Manning et al., 2009). As an example of the former, (Bai et al., 2005) proposed a language modeling approach that integrates term relationships mined from documents in a query expansion model. They considered two specific types of term relationship: co-occurrence relationships and inferential relationships extracted from documents. As examples of the later, several researchers have expanded queries with synonyms from WordNet after performing word sense disambiguation (WSD) with some success (Voorhees, 1994; Liu et al., 2004a; Liu et al., 2005; Cao et al., 2005; Fang, 2008; Zhong and Ng, 2012). For instance, Zhong and Ng (2012) use a combination of PRF, WSD and query expansion using WordNet relations. They use the top documents returned by the query to provide a context for disambiguating the queries, in a way reminiscent of pseudo-relevance feedback. The senses and the synonyms of the senses are then used to smooth term probabilities in a language modeling approach to IR. They show very strong results, with significant improvements and state-of-the-art results, but their expansion system might suffer

[9] http://ixa2.si.ehu.es/ukb

on datasets where pseudo-relevance feedback is not effective.

The query expansion method proposed here is also a global expansion technique based on WordNet, but in contrast to the references just cited, it does not require explicit WSD, and uses related words beyond synonyms for expansion. As mentioned above, the use of explicit WSD could further improve our technique to suggest expansion techniques.

An alternative to QE is to perform the expansion in the document. Document Expansion (DE) was first proposed in the speech retrieval community (Singhal and Pereira, 1999), where the task is to retrieve speech transcriptions which are quite noisy. Singhal and Pereira proposed to enhance the representation of a noisy document by adding to the document vector a linearly weighted mixture of related documents. In order to determine related documents, the original document is used as a query into the collection, and the ten most relevant documents are selected. Two related papers (Liu et al., 2004b; Kurland and Lee, 2004) followed a similar approach on the TREC ad-hoc document retrieval task. They use document clustering to determine similar documents, and document expansion is carried out with respect to these. Both papers report significant improvements over non-expanded baselines. Instead of clustering, more recent work (Tao et al., 2006; Mei et al., 2008; Huang et al., 2009) use language models and graph representations of the similarity between documents in the collection to smooth language models with some success.

The document expansion method presented here is complementary to those methods, in that we also explore DE, but use WordNet instead of distributional methods. The comparison with respect to other DE techniques and the exploration of potential combinations will be the focus of future research.

Another strand of WordNet-based IR work has explicitly represented and indexed word senses after performing WSD, without performing any expansion proper (Gonzalo et al., 1998; Stokoe et al., 2003; Kim et al., 2004). Word senses conform a different space for document representation, but contrary to us, these works incorporate concepts for all words in the documents, and are not able to incorporate concepts that are not explicitly mentioned in the document. Stokoe et al. (2003) performed WSD on Wordnet senses for both documents and queries, and achieved significant improvements over a vector-space model baseline. Unfortunately the baseline was very weak, making it difficult to judge whether the word senses would be helpful in a stronger IR system. Kim et al. (2004) tagged nouns with 25 semantic tags from Wordnet, and adjusted term weights in the baseline IR system according to the sense matches between query and document, improving over a strong system. More recently, a CLEF task was organized (Agirre et al., 2010c) where terms were semantically disambiguated to see the improvement that this would have on retrieval. Several teams participated, exploring different ways to index word senses. The conclusions were mixed, with some participants slightly improving results over baselines with information from WordNet.

Our method to find related concepts both for queries and documents is complementary to those methods, in that we could have used an index of concepts and word senses in addition to the additional index in RDE. We would like to explore this possibilities in the future.

As an alternative to Wordnet, other authors have used Wikipedia as the word sense or concept repository. For instance, Egozi *et al.* (2011) use a method to augment text with concepts from Wikipedia, based on Explicit Semantic Analysis (Gabrilovich and Markovitch, 2007). In order to improve over the baseline they need to use feature selection methods to prune the concept representation, and combine concept and bag-of-words retrieval.

In previous work (Agirre et al., 2010a), we used the same WordNet-based relatedness method in order to expand documents, following the BM25 probabilistic method for IR, obtaining some improvements, specially when parameters had not been optimized. Subsequently we moved to a language modeling approach, experimenting with query expansion and comparing the performance with PRF (Otegi et al., 2011). The work presented here extends (Otegi et al., 2011) with an implementation of RDE in a language modeling framework, and provides more extensive analysis and experimentation.

Finally, we would like to mention the performance of other systems on the same datasets. The systems which performed best in the Robust evaluation campaign (Agirre et al., 2010c) report 0.4509 MAP, but note that they deployed a complex system combining probabilistic and monolingual translation-based models. In ResPubliQA (Peñas et al., 2009), the official evaluation included manual assessment, and we cannot therefore reproduce those results exactly. As an alternative, the organizers released all runs, but only the first ranked document for each query was included, so we could only compute P@1. The P@1 of the best run was 0.40, which is not so far from our best P@1 result, as we obtain 0.3940 P@1 for RDE. Regarding Yahoo!, (Surdeanu et al., 2008) report an MRR of around 0.68. This number in an overestimation of the real performance, as they evaluate only in the questions where the correct answer is retrieved by their document retrieval engine in the top 50 answers, and it is thus not directly comparable to our setting.

7 Conclusions

In this paper we explore a generic method to improve IR results using structured knowledge for both query and document expansion. Our work has been motivated by the success of knowledge-based methods in word similarity and relatedness tasks (Agirre et al., 2009). Note that distributional similarity is closely related to query expansion and clustering techniques for IR. In the first case, techniques such as pseudo-relevance feedback (PRF) expand the query with terms which are deemed to be related to the query according to the retrieved documents (Xu and Croft, 1996). In the second case, documents are clustered, and terms from related documents are used to re-estimate counts and to expand the documents with new terms (Singhal and Pereira, 1999).

Our expansion method is based on random walks over a graph-representation of a knowledge base. The random walks return sets of concepts which are related to the input query (or document), even if those concepts are not explicitly mentioned in the texts. The query (or document) is then expanded using the terms lexicalizing the related concepts. In this work we focused on WordNet, but any other knowledge structure could be used.

We adopted a language modeling framework to implement the query likelihood and pseudo-relevance feedback baselines, as well as our relatedness-based query expansion (RQE) and document expansion (RDE) methods, where the expansion terms for documents are indexed separately. We wanted to check the performance on a diverse range of document collections, ranging from ad-hoc information retrieval, answer-finding and passage retrieval: Robust-WSD dataset from CLEF (ad-hoc dataset on news which got the attention of the WSD community), Yahoo! Answers (answer-finding collection, with questions and answers as phrased by real users on diverse topics) and ResPubliQA (a passage retrieval task on European Union laws in the context of question answering exercises).

Our two methods provide improvements in all three datasets, when compared to the query likelihood baseline. PRF is beneficial in two datasets, but degrades performance in ResPubliQA. RDE and RQE compare favorably to PRF in two datasets, but perform worse in Robust. Our analysis shows that our models and PRF are complementary, in that PRF is better for easy queries and our models are stronger for difficult queries. We also show that our models generalize better to other collections, and are more robust to parameter adjustments.

Given the very positive results obtained with WordNet, we are currently exploring other knowledge bases and resources. For instance, we used random walks over Wikipedia for query expansion on the ad-hoc and semantic enrichment task held at CHiC 2012 (Cultural Heritage in CLEF), obtaining promising results, specially in the semantic enrichment task (Agirre et al., 2012). On the other hand, we performed random walks on the UMLS metathesaurus (Humphreys et al., 1998) in order to expand the queries for the TREC 2012 Medical Track[10]. In both tasks we obtained positive results.

In the future, we would like to evaluate separately the concepts obtained from the

[10] http://trec.nist.gov/pubs/call2012.html

random walks, in order to study which are the words that have good expansions that contribute to improve performance. We also would like to combine our relatedness method with other WSD-based techniques. We also plan to explore the ability of RQE and RDE to perform well on difficult queries, perhaps combining them with PRF and document clustering techniques.

A limitation of our method is that it would suffer in the case of noisier datasets like blogs or tweets, where informal language abound. Recent work on normalization (Han and Baldwin, 2011) would be very helpful, as the text could be normalized prior to checking the lexical resources, making our method amenable to the task.

References

E. Agirre and A. Soroa. 2009. Personalizing PageRank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 33–41, Athens, Greece. Association for Computational Linguistics.

E. Agirre, A. Soroa, E. Alfonseca, K. Hall, J. Kravalova, and M. Paşca. 2009. A study on similarity and relatedness using distributional and WordNet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 19–27, Stroudsburg, PA, USA. Association for Computational Linguistics.

E. Agirre, X. Arregi, and A. Otegi. 2010a. Document expansion based on WordNet for robust IR. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 9–17, Stroudsburg, PA, USA. Association for Computational Linguistics.

E. Agirre, M. Cuadros, G. Rigau, and A. Soroa. 2010b. Exploring knowledge bases for similarity. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

E. Agirre, G. M. Di Nunzio, T. Mandl, and A. Otegi. 2010c. CLEF 2009 Ad Hoc Track Overview: Robust-WSD Task. In *Multilingual Information Access Evaluation I. Text Retrieval Experiments, CLEF 2009*, volume 6241 of *Lecture Notes in Computer Science*, pages 36–49. Springer.

E. Agirre, P. Clough, S. Fernando, M. Hall, A. Otegi, and M. Stevenson. 2012. The Sheffield and Basque Country Universities Entry to CHiC: Using Random Walks and Similarity to Access Cultural Heritage. In P. Forner, J. Karlgren, and C. Womser-Hacker, editors, *CLEF (Online Working Notes/Labs/Workshop)*.

J. Bai, D. Song, P. Bruza, J. Y. Nie, and G. Cao. 2005. Query expansion using term relationships in language models for information retrieval. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, CIKM '05, pages 688–695, New York, NY, USA. ACM.

A. Berger, R. Caruana, D. Cohn, D. Freitag, and V. Mittal. 2000. Bridging the lexical chasm: statistical approaches to answer-finding. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 192–199. ACM.

C. Buckley and M. Sanderson. 2008. Relevance Feedback Track Overview: TREC 2008. In *Proceedings of The Seventeenth Text REtrieval Conference, TREC 2008*, volume Special Publication 500-277. National Institute of Standards and Technology (NIST).

G. Cao, J.Y. Nie, and J. Bai. 2005. Integrating word relationships into language models. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '05, pages 298–305, New York, NY, USA. ACM.

O. Egozi, S. Markovitch, and E. Gabrilovich. 2011. Concept-based information retrieval using explicit semantic analysis. *ACM Transactions on Information Systems*, 29(2):8:1–8:34.

H. Fang. 2008. A re-examination of query expansion using lexical resources. In *Proceedings of ACL-08: HLT*, pages 139–147, Columbus, Ohio. Association for Computational Linguistics.

C. Fellbaum. 1998. *WordNet: an electronic lexical database and some of its applications*. MIT Press, Cambridge, Mass.

E. Gabrilovich and S. Markovitch. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In M. M. Veloso, editor, *IJCAI*, pages 1606–1611.

J. Gonzalo, F. Verdejo, I. Chugur, and J. Cigarran. 1998. Indexing with WordNet synsets can improve text retrieval. In *Proceedings of the COLING/ACL Workshop on Usage of WordNet in Natural Language Processing Systems*, pages 38–44.

B. Han and T. Baldwin. 2011. Lexical normalisation of short text messages: Makn sens a #twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 368–378, Portland, Oregon, USA. Association for Computational Linguistics.

T. H. Haveliwala. 2002. Topic-sensitive PageRank. In *Proceedings of WWW '02*, pages 517–526.

Y. Huang, L. Sun, and J. Nie. 2009. Smoothing document language model with local word graph. In *Proceeding of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 1943–1946. ACM.

T. Hughes and D. Ramage. 2007. Lexical semantic relatedness with random graph walks. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 581–589.

L. Humphreys, D. Lindberg, H. Schoolman, and G. Barnett. 1998. The Unified Medical Language System: an informatics research collaboration. *Journal of the American Medical Informatics Association*, 1(5):1–11.

S. Kim, H. Seo, and H. Rim. 2004. Information retrieval using word senses: root sense tagging approach. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '04, pages 258–265, New York, NY, USA. ACM.

O. Kurland and L. Lee. 2004. Corpus structure, language models, and ad hoc information retrieval. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '04, pages 194–201, New York, NY, USA. ACM.

V. Lavrenko and W. B. Croft. 2001. Relevance based language models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '01, pages 120–127, New York, NY, USA. ACM.

S. Liu, F. Liu, C. Yu, and W. Meng. 2004a. An effective approach to document retrieval via utilizing wordnet and recognizing phrases. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 266–272. ACM.

X. Liu, W. B. Croft, and W. Bruce. 2004b. Cluster-based retrieval using language models. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '04, pages 186–193, New York, NY, USA. ACM.

S. Liu, C. Yu, and W. Meng. 2005. Word sense disambiguation in queries. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, CIKM '05, pages 525–532, New York, NY, USA. ACM.

C. D. Manning, P. Raghavan, and H. Schütze. 2009. *An introduction to information retrieval*. Cambridge University Press, UK.

Q. Mei, D. Zhang, and C. Zhai. 2008. A general optimization framework for smoothing language models on graph structures. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 611–618. ACM.

D. Metzler. 2006. Estimation, sensitivity, and generalization in parameterized retrieval models. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, CIKM '06, pages 812–813, New York, NY, USA. ACM.

D. Moldovan and M. Surdeanu. 2003. On the role of information retrieval and information extraction in question answering systems. *Information Extraction in the Web Era*, pages 129–147.

A. Otegi, X. Arregi, and E. Agirre. 2011. Query expansion for ir using knowledge-based relatedness. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1467–1471, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

A. Peñas, P. Forner, R. Sutcliffe, A. Rodrigo, C. Forăscu, I. Alegria, D. Giampiccolo, N. Moreau, and P. Osenova. 2009. Overview of ResPubliQA 2009: question answering evaluation over European legislation. In *Proceedings of the 10th cross-language evaluation forum conference on Multilingual information access evaluation: text retrieval experiments*, CLEF'09, pages 174–196, Berlin, Heidelberg. Springer-Verlag.

J. M. Ponte and W. B. Croft. 1998. A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '98, pages 275–281, New York, NY, USA. ACM.

S. Riezler, A. Vasserman, I. Tsochantaridis, V. Mittal, and Y. Liu. 2007. Statistical machine translation for query expansion in answer retrieval. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 464–471, Prague, Czech Republic. Association for Computational Linguistics.

S. Robertson. 2006. On GMAP: and other transformations. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, CIKM '06, pages 78–83, New York, NY, USA. ACM.

J. J. Rocchio. 1971. Relevance feedback in information retrieval. In *The Smart retrieval system - experiments in automatic document processing*, pages 313–323. Englewood Cliffs, NJ: Prentice-Hall.

I. Ruthven and M. Lalmas. 2003. A survey on the use of relevance feedback for information access systems. *Knowl. Eng. Rev.*, 18(2):95–145.

A. Singhal and F. Pereira. 1999. Document expansion for speech retrieval. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99, pages 34–41, New York, NY, USA. ACM.

M. D. Smucker, J. Allan, and B. Carterette. 2007. A comparison of statistical significance tests for information retrieval evaluation. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, CIKM '07, pages 623–632, New York, NY, USA. ACM.

C. Stokoe, M. P. Oakes, and J. Tait. 2003. Word sense disambiguation in information retrieval revisited. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, SIGIR '03, pages 159–166. ACM.

T. Strohman, D. Metzler, H. Turtle, and W. B. Croft. 2005. Indri: a language-model based search engine for complex queries. Technical report, Proceedings of the International Conference on Intelligent Analysis.

M. Surdeanu, M. Ciaramita, and H. Zaragoza. 2008. Learning to rank answers on large online QA collections. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, pages 719–727. The Association for Computer Linguistics.

T. Tao, X. Wang, Q. Mei, and C. Zhai. 2006. Language model information retrieval with document expansion. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, HLT-NAACL '06, pages 407–414. Association for Computational Linguistics.

E. M. Voorhees. 1994. Query expansion using lexical-semantic relations. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '94, pages 61–69, New York, NY, USA. Springer-Verlag New York, Inc.

J. Xu and W. B. Croft. 1996. Query expansion using local and global document analysis. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '96, pages 4–11, New York, NY, USA. ACM.

C. Zhai and J. Lafferty. 2001. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th annual international ACM*

SIGIR conference on Research and development in information retrieval, SIGIR '01, pages 334–342, New York, NY, USA. ACM.

Z. Zhong and H. T. Ng. 2012. Word sense disambiguation improves information retrieval. In *ACL (1)*, pages 273–282. The Association for Computer Linguistics.