A syntactically-based query reformulation technique for information retrieval

C. Lioma *, I. Ounis

University of Glasgow, Department of Computing Science, Lilybank Gardens, Glasgow G12 8QQ, UK

Abstract

Whereas in language words of high frequency are generally associated with low content [Bookstein, A., & Swanson, D. (1974). Probabilistic models for automatic indexing. *Journal of the American Society of Information Science, 25*(5), 312–318; Damerau, F. J. (1965). An experiment in automatic indexing. *American Documentation, 16*, 283–289; Harter, S. P. (1974). A probabilistic approach to automatic keyword indexing. PhD thesis, University of Chicago; Sparck-Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation, 28*, 11–21; Yu, C., & Salton, G. (1976). Precision weighting – an effective automatic indexing method. *Journal of the Association for Computer Machinery (ACM), 23*(1), 76–88], shallow syntactic fragments of high frequency generally correspond to lexical fragments of high content [Lioma, C., & Ounis, I. (2006). Examining the content load of part of speech blocks for information retrieval. In *Proceedings of the international committee on computational linguistics and the association for computational linguistics (COLING/ACL 2006), Sydney, Australia*]. We implement this finding to Information Retrieval, as follows. We present a novel automatic query reformulation technique, which is based on shallow syntactic evidence induced from various language samples, and used to enhance the performance of an Information Retrieval system. Firstly, we draw shallow syntactic evidence from language samples of varying size, and compare the effect of language sample size upon retrieval performance, when using our syntactically-based query reformulation (SQR) technique. Secondly, we compare SQR to a state-of-the-art probabilistic pseudo-relevance feedback technique. Additionally, we combine both techniques and evaluate their compatibility. We evaluate our proposed technique across two standard Text REtrieval Conference (TREC) English test collections, and three statistically different weighting models. Experimental results suggest that SQR markedly enhances retrieval performance, and is at least comparable to pseudo-relevance feedback. Notably, the combination of SQR and pseudo-relevance feedback further enhances retrieval performance considerably. These collective experimental results confirm the tenet that high frequency shallow syntactic fragments correspond to content-bearing lexical fragments.
© 2007 Published by Elsevier Ltd.

---

* Corresponding author.
  E-mail addresses: xristina@dcs.gla.ac.uk (C. Lioma), ounis@dcs.gla.ac.uk (I. Ounis).

1. Introduction

The task of an information retrieval (IR) system is to retrieve documents from a collection, in response to a user need, which is expressed in the form of a query. Retrieval consists in matching the query against the documents, and in returning the ones that appear closest, in ranked lists of assumed relevance (van Rijsbergen, 1979). Usually, the words found in the queries and documents are associated with individual weights, which capture the importance of these words to the content of the corresponding queries and documents. Such weights, commonly referred to as term weights, can be computed using various term weighting models. On top of these term weighting models, other performance-boosting techniques are often used to increase retrieval performance, especially for ad hoc retrieval. Ad hoc retrieval is denned as the prototypical document retrieval task, where the document set is a static collection of text documents, and where a subset of the documents are to be retrieved in response to a user's query. The ad hoc retrieval task is similar to how a researcher might use a library – the collection is known, but the questions likely to be asked are not known (Voorhees & Harman, 2005). One of the performance-boosting techniques that are often used to increase retrieval performance is pseudo-relevance feedback (PRF) (Salton & Buckley, 1990; Salton, Fox, & Voorhees, 1985). In PRF, the query is usually enriched with more relevant terms, and thus reformulated, in order to facilitate the retrieval of documents relating to the user need. PRF is often realised on the basis of the lexical features of the queries, such as word frequency counts and co-occurrence statistics (Amati & van Rijsbergen, 2002; Efthimiadis & Biron, 1993; Fuhr & Robertson, 1992; Rocchio, 1971; Xu & Croft, 1996). By lexical, we denote the information that relates to the canonical form of words, such as *apple* for example. By shallow syntactic, we in fact denote the surface-syntactic information that relates to the part of speech of words, such as *noun* for example. Table 1 illustrates this correspondence between lexicon and syntax in language, with a few examples.

Generally in natural languages, words are lexical manifestations of meaning, while sentences are shallow syntactic arrangements of parts of speech. Our assumption is that, in textual IR, apart from lexical information, shallow syntactic information, namely part of speech classification information, can also be used to reformulate the queries, and increase their likelihood of fetching more relevant documents. We propose syntactically-based query reformulation (SQR) as a technique that aims to automatically enhance the performance of an IR system, on the basis of natural language shallow syntactic knowledge.

A lot of research in the area of IR has focussed on the automatic processing of words, most probably motivated by the fact that words are the explicit carriers of the information intended in any language communication. However, syntax also plays an implicit role in communicating meaning, namely by regulating the relations that bind words together into coherent sentences. We believe that shallow syntactic information may model the structure of language, by showing which shallow syntactic structures are more likely to occur

Table 1
Sample correspondences between the lexicon and syntax of natural language

Natural language	
Lexicon	Syntax
Canonical word form	Part of speech
apple	*noun*
write	*verb*
the	*article*
well	*adverb*
tall	*adjective*
and	*conjunction*
in	*preposition*
his	*pronoun*
are	*auxiliary verb*
three	*cardinal number*
done	*participle*
i.e.	*particle*
's	*possessive ending*
@	*symbol*

and/or co-occur. In order for this shallow syntactic information to be considered as shallow syntactic evidence, it needs to be extracted from a relatively large language sample, which represents language use in general, and not domain-specific language use.

The shallow syntactic structures that we use in SQR consist of shallow syntactic blocks, that may be arbitrarily set to any length. shallow syntactic blocks are sentential fragments, where lexical tokens have been replaced by their corresponding shallow syntactic category, namely part of speech. We call these shallow syntactic structures, part of speech blocks (POS blocks). The following example illustrates this point, with a sentential lexical fragment and its corresponding POS block. The length of the POS block used in this example is set at four tokens.

[lexical fragment]

genetic or environmental factors

[**POS block**]

adjective conjunction adjective noun

Such fixed-length POS blocks are extracted from large language samples, and smoothed on the basis of their frequency and the overall size of the language sample used. Both the length of the POS block, and the types of shallow syntactic (part of speech) categories that are contained within POS blocks, may be tailored to fit specific and distinct research needs. For example, longer POS blocks may be preferred for a shallow syntactic modeling of the argumentative structure of documents; similarly, the shallow syntactic category of proper nouns may be specified as a compulsory presence within all POS blocks, when seeking named entities.

Our underlying assumption, which is laid out and justified in Section 3.1, is that POS blocks that occur more often in language may be more content-bearing, than other less frequently occurring POS blocks. This follows from the fact that arrangements of parts of speech have different properties than lexical items, with regard to frequency of occurrence (as is discussed in Section 3.1). Following from this, our aim is to reduce the original queries to their content-rich fragments only, using POS blocks. Thus, we adopt the following steps, which are further detailed in Section 4.

- Firstly, we extract frequently occurring POS blocks from a large representative language sample.
- Secondly, we select the lexical parts of the queries that correspond to these frequently occurring POS blocks.
- Thirdly, we reduce the query to the lexical form of the selected frequently occurring POS blocks.

In this respect, our proposed technique reformulates queries by reducing the amount of assumed content-poor and hence noisy fragments found in them. The resulting query becomes our syntactically-based reformulated query.

The main objective of this work is to further test the assumption that POS blocks that occur more often in language may be more content-bearing, than other less frequently occurring POS blocks, through an application, namely IR, for which we have some quantitative measure of performance, namely *mean average precision* (MAP). SQR applies that assumption (Lioma & Ounis, 2006). Other applications testing this claim may also be done in the fields of automatic summarisation, automatic indexing, and so on.

The remainder of this paper is organised as follows. Section 2 discusses studies relating to this work. Section 3 introduces the motivating intuition behind SQR, and the research questions addressed in this paper. Section 4 presents our methodology. Section 5 details the experimental settings used to evaluate our technique, and the corresponding evaluation outcomes. Section 6 discusses further applications of SQR in IR. Section 7 provides our concluding remarks and some future research directions.

2. Related studies

The use of natural language syntax to enhance retrieval performance is not a new idea. In the context of IR, efforts have been made to use shallow syntactic information to enhance retrieval, ranging from using pseudo-syntactic rules (Bruandet, 1987), to developing conceptual frames (Croft & Lewis, 1987), and

focussing on specific surface syntactic groups (Smeaton & van Rijsbergen, 1988). However, numerous as such studies may have been (DeJaco & Garbolino, 1986; Jacobs, 1992; Jacobs & Rau, 1988; Karlgren, 1993; Mauldin, Carbonell, & Thomason, 1987; Salton, 1991; Smeaton, 1986, 1999; Sparck-Jones & Tait, 1984; Strzalkowski, 1992; Walker, Karlgren, & Kay, 1977; Xu & Croft, 1996; Zukerman & Raskutti, 2002), they have not made use of the type of shallow syntactic structures, namely POS blocks, which we present in this work.

POS blocks should not be confused with textual chunks, which are defined as non-recursive and typically non-overlapping cores of intra-clausal constituents (Abney, 1991, 1996; Ramshaw & Marcus, 1995). Chunks can be (and most of the times are) non-overlapping, while POS blocks are *always* overlapping. Chunks aim to model word groups that are grammatically related, noun phrases for example. POS blocks do not aim to model words groups that are grammatically related, they only aim to model tags (and hence their corresponding words) that occur next to one another. More simply, even through a POS block may include a noun phrase, or any other constituent, it does not necessarily have to, and more importantly, it does not aim to. Hence, chunking can be viewed as a search process, with applications like named entity identification, while extracting POS blocks *cannot* be viewed as a search process. Chunks can vary in length, while POS blocks are always *fixed-length*.

The extraction of POS blocks for SQR is similar as a process to the class-based *n*-gram model (Brown, Pietra, deSouza, Lai, & Mercer, 1992). The class-based *n*-gram model uses *n*-grams of shallow syntactic classes to determine the probability of occurrence of a shallow syntactic class, given its preceding shallow syntactic class, and the probability of occurrence of a particular word, given its own shallow syntactic class. This model differs from our here-proposed approach in that it addresses the problem of shallow syntactic tagging, and has never been applied to Information Retrieval. More importantly, in SQR we treat POS blocks as tokens, and we do not examine the relations between their members. Moreover, a plethora of other studies have examined the distribution of character or word *n*-grams, as in the case of language modeling for information retrieval (Croft & Lafferty, 2002; Hiemstra, 2001; Kraaij, 2004), for example. Specifically, both in language modeling and in SQR, the length of the patterns or *n*-grams to be extracted is fixed, while the extraction of the patterns or *n*-grams is realised on a recurrent and overlapping basis (see Section 4, step 1, for a detailed presentation of the way in which POS blocks are extracted in SQR). Nevertheless, in language modeling for Information Retrieval, *n*-grams tend to consist of 'lexical' units, such as character or word groups or subgroups, and not of shallow syntactic categories, such as parts of speech. The POS blocks used in SQR do not replace the terms of the documents and queries, but are used on top of them, as an extra layer of processing. On the contrary, IR language modeling is a type of weighting and retrieval model, which can be used on its own to weight and match *n*-grams, and therefore to retrieve documents that are relevant to a query. SQR is a query reformulation technique, which cannot be used on its own, but only on top of any existing weighting model.

SQR differs from conventional pseudo-relevance feedback (PRF) techniques in two distinct ways. Firstly, conventional PRF mainly consists in processing lexical items, namely words. On the contrary SQR focuses mainly on arrangements of parts of speech, namely POS blocks. Secondly, PRF is often a query expansion mechanism, which adds assumed relevant words to the queries, in order to boost the presence of content-bearing items in them, thus increasing query size, and reweights the resulting query terms. To the contrary, SQR, reduces query size by keeping in the query only sentential fragments that are assumed to be content-bearing, on the basis of their syntax. SQR does not reweight the terms of the reformulated query in itself, but relies on the term weighting model used to do so.

3. Syntactically-based query reformulation (SQR)

3.1. Motivation

One of the core functions of natural language is to communicate information. The role of natural language syntax is to regulate the arrangement of words into phrases, so that these phrases are well-formed and communicate the information intended. There is more than one way of saying the same thing. Hence, there is more than one syntactic arrangement that communicates the same piece of information. We capture syntactic

arrangement in language, using recurrent arrangements of parts of speech, namely POS blocks. We assume that the most frequently occurring POS blocks in language must be the ones that capture the most information content possible in the least effort-consuming way, in line with the principle of least effort (Zipf, 1949). The validity of this statement is well-known in the field of linguistics, where complicated and/or difficult syntactic structures and features are used less frequently with time, until they become extinct from language as a whole (Kiparsky, 1976). It is this exact relation between syntax and information content that we wish to model and utilise in order to enhance retrieval performance, in the framework of an IR system.

Our intuition might appear at first glance to contradict one of the main tenets of textual IR, which holds that the relation between the frequency of occurrence of words and the amount of content that they convey is approximately inversely proportional (Bookstein & Swanson, 1974; Damerau, 1965; Harter, 1974; Sparck-Jones, 1972; Yu & Salton, 1976). Nevertheless, our intuition does not relate to single words, namely lexical items, but to arrangements of parts of speech, namely POS blocks. Specifically, we assume that the relation between the frequency of occurrence of POS blocks in a representative language sample and the amount of content that they bear is approximately directly proportional. Note that the evaluation of this assumption is not within the scope of this paper. Even so, experimental results suggest that this assumption holds (Lioma & Ounis, 2006). We further confirm these findings with the application of SQR to ad hoc retrieval, described in this study.

We model language use statistically, using language samples of considerable size. We consider the most frequent POS blocks that are contained within these samples to be representative of the overall 'easiest' way of communicating as much content as possible. Such POS blocks form the shallow syntactic evidence, which we employ to reformulate the queries as part of our SQR technique. SQR reformulation consists in reducing the query to only those sentential fragments that correspond to POS blocks assumed to be content-rich, on the basis of their high frequency.

We refer to our technique as syntactically-based, rather than syntactic, query reformulation, because it does not make any attempt to process syntax, such as exploring the relations that bind the members of a sentence for example. SQR uses surface-syntactic blocks, namely POS blocks, induced from text, and treats these POS blocks as individual units, i.e. tokens. We assume that for these 'surface-syntactic tokens', high frequency of occurrence indicates the presence of equivalent content-bearing lexical items (Lioma & Ounis, 2006). It is this exact theoretical claim that we implement as part of the retrieval process, namely in SQR, for which we have some quantitative measure of performance, such as MAP for example.

3.2. Our research questions

Within the above-described framework of syntactically-based query reformulation, we purport to address the following research questions:

1. Does the relative size of the language samples from which we draw shallow syntactic evidence for SQR affect retrieval performance?
2. Is SQR an effective and robust query reformulation technique for IR systems?
3. Can SQR be effectively combined with pseudo-relevance feedback (PRF)?

We investigate the aforementioned research questions as follows. It has been shown that collection size has an effect on retrieval performance (Hawking & Robertson, 2003; Macdonald et al., 2005), and when using external collections for query expansion (Diaz & Metzler, 2006; Macdonald et al., 2005). We ask if collection size also has an effect on SQR. To answer this first question, we use shallow syntactic evidence induced from various language samples of different sizes. To answer the second question, we compare SQR to a strong pseudo-relevance feedback technique. To answer the third question, we apply SQR and PRF together. We evaluate our approach on two standard Text REtrieval Conference (TREC) English test collections, using three statistically different term weighting models. We consider short queries as the least noise-bearing queries possible, and long queries as the most noise-bearing queries. Thus, we compare the effect of SQR on long queries, using short queries as our baseline, and observe the amount, if any, of noise reduction, as reflected upon retrieval performance.

4. SQR methodology

This section presents the steps realised in order to apply SQR within the context of an IR application. These steps, which are presented in details below, are:

Step 1. Draw shallow syntactic evidence from language samples.
Step 2. Represent the queries syntactically.
Step 3. Reformulate the queries using shallow syntactic evidence.

4.1. Step 1. Drawing shallow syntactic evidence from language samples

We set off with various language samples of different sizes, and induce shallow syntactic evidence from them as follows. We feed each language sample separately into an automatic shallow syntactic tagger, for example the Tree Tagger (Schmidt, 1997). The shallow syntactic analysis is realised on a part of speech level. Thus, we obtain syntactically tagged versions of these samples, from which we extract POS blocks. These blocks are of fixed length, which is set empirically, depending on the application (see Section 1). The blocks are overlapping, and are extracted as indicated in the following example. The collective number of POS blocks of length n that can be extracted from a sentence that contains l terms is: $l - (n - 1)$. Thus, no more blocks are extracted after there remain less than n terms in a sentence. In the following example, a sample sentence is presented both in its lexical and shallow syntactic form, while the length of the POS blocks extracted is set at four tokens. Note that, in this example, even though *farm* is a noun, it is tagged as an adjective, because it occupies a pronominal position, and thus assumes the syntactic role of an adjective.

<p style="text-align:center">[sample sentence – lexical form]</p>

The mechanisation of farm work has reduced the need for manual labour.

<p style="text-align:center">[sample sentence – shallow syntactic form]</p>

```
article noun preposition adjective noun aux_verb participle article noun
                    preposition adjective noun
```

[POS blocks generated from sample sentence]

$$\underbrace{article}_{The} \quad \underbrace{noun}_{mechanisation} \quad \underbrace{preposition}_{of} \quad \underbrace{adjective}_{farm} \tag{1}$$

$$\underbrace{noun}_{mechanisation} \quad \underbrace{preposition}_{of} \quad \underbrace{adjective}_{farm} \quad \underbrace{noun}_{work} \tag{2}$$

$$\underbrace{Preposition}_{of} \quad \underbrace{adjective}_{farm} \quad \underbrace{noun}_{work} \quad \underbrace{aux\_verb}_{has} \tag{3}$$

$$\underbrace{adjective}_{farm} \quad \underbrace{noun}_{work} \quad \underbrace{aux\_verb}_{has} \quad \underbrace{participle}_{reduced} \tag{4}$$

$$\underbrace{noun}_{work} \quad \underbrace{aux\_verb}_{has} \quad \underbrace{participle}_{reduced} \quad \underbrace{article}_{the} \tag{5}$$

$$\underbrace{aux\_verb}_{has} \quad \underbrace{participle}_{reduced} \quad \underbrace{article}_{the} \quad \underbrace{noun}_{need} \tag{6}$$

$$\underbrace{participle}_{reduced} \quad \underbrace{article}_{the} \quad \underbrace{noun}_{need} \quad \underbrace{preposition}_{for} \tag{7}$$

$$\underbrace{article}_{the} \quad \underbrace{noun}_{need} \quad \underbrace{preposition}_{for} \quad \underbrace{adjective}_{manual} \tag{8}$$

$$\underbrace{noun}_{need} \quad \underbrace{preposition}_{for} \quad \underbrace{adjective}_{manual} \quad \underbrace{noun}_{labour} \tag{9}$$

[most frequent POS blocks(occurring twice)]

(1 & 8) `article noun preposition adjective`

(2 & 9) `noun preposition adjective noun`

This step is realised during indexing time. The relevant computational cost is low, as both shallow syntactic tagging and block extraction are speedy processes, the aggregation of which does not require much disk space.

4.2. Step 2. Representing the queries syntactically

We syntactically tag the queries using the same automatic shallow syntactic tagger as in Step 1. We extract from the shallow syntactic representations of the queries the top k most probable POS blocks drawn from the language samples. The decision regarding the number k of the most probable POS blocks of the language samples we use is made empirically. The following example illustrates Step 2. In this example, we select POS blocks of frequency higher than 1. Note that, in this example, even though *Chevrolet* is a proper noun, it is tagged as an adjective, because it occupies a pronominal position, and thus assumes the syntactic role of an adjective.

[sample query – lexical form]

Find documents that address the types of Chevrolet trucks available

[sample query – shallow syntactic from]

`verb noun wh-determiner verb article noun preposition adjective noun adjective.`

We map the two most frequent POS blocks from Step 1 to the tagged query. Fragments that have not been mapped appear in reduced font size. The selected mapped fragments appear in brackets. Overlapping fragments are highlighted in bold:

> *Find documents that address*
>
> `verb noun relative_pronoun verb`
>
> ($_1$ *the* ($_2$ `types of Chevrolet`)$_1$ *trucks*)$_2$
>
> ($_1$ *article* ($_2$ **noun preposition adjective**)$_1$ *noun*)$_2$
>
> *available*
>
> `adjective`

In the preceding example, brackets numbered 1 and 2 contain query fragments that correspond to each of the two POS blocks employed, respectively. Thus, bracket numbered 1 contains the fragment *the types of Chevrolet*, and bracket numbered 2 contains the fragment *types of Chevrolet trucks*.

4.3. Step 3. Reformulating the queries using shallow syntactic evidence

We use the original lexical query text that corresponds to the selected POS blocks of the queries as the sole content of the reformulated query. If there exists an overlap between terms that are shared by two or more POS blocks, we keep one occurrence of the overlapping terms, so as to avoid word repetition and thus lexical misrepresentation in the query. The output of this process becomes our new query. The following example illustrates Step 3.

Lexical fragments corresponding to the most frequent POS blocks extracted from the language sample (Step 1) and mapped to the query (Step 2):

> *the types of Chevrolet*
>
> *types of Chevrolet trucks*

Final reformulated query without overlap:

    the types of Chevrolet trucks

The computational overhead of steps 2 and 3 for small values of k, such as the ones used in this paper, is negligible, compared to the cost associated with core retrieval processing.

5. Evaluation

This section presents the experiments conducted in order to investigate the research questions formulated in Section 3.2, across a selection of retrieval settings. Section 5.1 introduces the experimental settings we employ. Section 5.2 presents our evaluation results and discusses our findings.

5.1. Experimental settings

We induce shallow syntactic evidence from five language samples of significantly different sizes, in order to investigate the effect of language sample size upon retrieval performance when using SQR. Specifically, the first language sample we use is the English component of the second release of the Europarl parallel corpus, namely Europarl_En (75MB)[1], to be referred to as LS1. This corpus contains parliamentary proceedings of the European Parliament, crawled from the proceedings dating between 1996 and 2003. Our second language sample, to be referred to as LS2, contains journalistic articles collected from the Los Angeles Times archives in 1994 (425MB)[2]. Our third language sample, to be referred to as LS3, is the TREC[3] AP collection (742MB), which contains newswire stories collected between 1988 and 1990. The fourth and fifth language samples we use, to be referred to as LS4 and LS5 correspondingly, are two standard TREC Web test collections, namely WT2G (2GB), from TREC-8, and WT10G (10GB), from TREC-9 and TREC-10, respectively. Both LS4 and LS5 were crawled from the Web in 1997. We syntactically tag these document collections using the TreeTagger (Schmidt, 1997), which is a probabilistic part of speech tagger.

We estimate the probability of occurrence of the shallow syntactic blocks induced from the above language samples, using Good-Turing statistical smoothing (Manning & Schutze, 1999). Good-Turing smoothing estimates the probability of occurrence P_{GT} of a POS block as

$$P_{GT} = \frac{r^*}{N}$$

where N denotes the number of POS block tokens extracted from the language sample, and r^* is an adjusted frequency, given by:

$$r^* = \frac{(r+1)E(Nr+1)}{E(Nr)}$$

where r is the frequency of a given unique type of POS block, E is the expectation of a random variable, and Nr is the frequency of r (also known as count-count) of a given unique type of POS block.

We empirically vary the k most probable POS blocks found in each of the language samples that we use for query reformulation, to the following values: 100, 50, 30, 10, and 5. Additionally, we empirically set the size of POS blocks to four tokens. We have also experimented with varying the n size of POS blocks to 3, 5, and 6. These experiments have produced retrieval performances that are consistent with the scores reported in this paper. The choice of four tokens as the size of POS blocks follows from the intuition that the size of the POS block should be large enough for the block to contain shallow syntactic constraints, yet no so large as to suppress constraints that make use of strict adjacency.

Table 2 displays the five most frequent POS blocks extracted from each of the five language samples employed. POS blocks common to two or more language samples in the top five most frequent places appear

---

Table 2
Top 5 most frequent POS blocks per language sample

POS block	Language sample
noun preposition article noun (5/5)	
preposition article noun preposition (2/5)	
article noun preposition article	LSI
preposition article adjective noun (4/5)	(75MB)
article noun preposition noun	
noun noun noun noun (4/5)	
noun preposition article noun (5/5)	
noun preposition noun noun (2/5)	LS2
preposition article noun noun	(425MB)
preposition article adjective noun (4/5)	
noun noun noun noun (4/5)	
noun preposition article noun (5/5)	
preposition article noun noun (2/5)	LS3
noun article noun noun	(742MB)
preposition article adjective noun (4/5)	
noun noun noun noun (4/5)	
noun preposition article noun (5/5)	
noun preposition noun noun (3/5)	LS4
preposition article noun preposition (2/5)	(2GB)
preposition article adjective noun (4/5)	
noun noun noun noun (4/5)	
noun preposition article noun (5/5)	
noun preposition noun noun (3/5)	LS5
noun noun preposition noun	(10GB)
preposition article noun noun (2/5)	

POS blocks common to two or more language samples in the top 5 frequency rankings appear in shaded boldface. The ratio of language samples sharing a given *POS block* in the top 5 appears in brackets.

in boldface, followed by a bracketed ratio of the language samples they occur in. We can see that LS1 shares the fewest most frequent common POS blocks with the other language samples, namely three, whereas LS2, LS3, and LS5 share four, and LS4 shares five. This is symptomatic of the fact that LS1 is representative of elaborate spoken language, as it is the transcription of the proceedings of the European Parliament, while the remaining language samples are in fact collections of written language, with a focus on factual, rather than elaborate language.

For our retrieval experiments, we use ad hoc queries from the TREC Web track. Specifically, we use the ad hoc queries from the TREC-8 Web track, numbered 401-450, when retrieving documents from the WT2G collection, and the ad hoc queries from the TREC-9 Web track, numbered 451-500, when retrieving documents from the WT10G collection. For both sets of queries, respective relevance assessments are provided. Each query contains three fields, namely a *title*, a *description*, and a *narrative* field. The *title* contains a few keywords, which are essential to the core content of the query. In the vast majority of cases, the *title* contains no noise, either in the form of stopwords, or in the form of vague/generic/irrelevant terms. The *description* of a query usually consists of a single sentence, which expands on the query content conveyed in the keywords in the *title*. Very often, the description sentence is a straight-forward question. In the vast majority of cases, the type of noise that is contained in the *description* sentence consists of simple stopwords. These stopwords are removed during the stage of stopword removal, which takes place prior to retrieval. The *narrative* of a query usually consists of more than one sentence, elaborating on what might be of relevance, and what might not be of relevance, to the topic in question. The *narrative* differs from the *title* and *description* fields of the query, not simply because it is longer, but mainly, because it contains much more noise, and more importantly, because it contains a different type of noise, namely noise that cannot be entirely removed only by dropping stopwords. Specifically, this type of noise consists of periphrastic structures, i.e. phrases introducing a point, rather than the factual point itself. These periphrastic structures contain both stopwords and other words that are neverthess vague/generic/irrelevant to the topic. We address exactly this type of noise with

Table 3
Values of the term weighting parameters used, per document collection and for all query field combinations (T, TD, and TDN)

Collection	TF · IDF		BM25			PL2	
	k_1	b	k_1	b	k_3	c	c
	(T, TD, TDN)	(T, TD, TDN)	(T, TD, TDN)	(T, TD, TDN)	(T, TD, TDN)	(T)	(TD, TDN)
WT2G	1.2	0.75	1.2	0.75	1000	10.99	4.80
WT10G	1.2	0.75	1.2	0.75	1000	13.13	5.58

SQR. As was mentioned in Section 1, SQR is an application of an automatic information processing model, which identifies and categorises content, on the basis of shallow syntactic recurrence statistics, and absolutely not on the basis of the kind of lexical and morphological evidence that defines stopwords. Thus, SQR is not analogous to stopword removal, as a noise reduction process. Hence, we apply SQR to the *narrative* field of the queries, not only because this field contains longer sentences, but also, and in fact mainly, because the *narrative* field contains vague/general words and phrases, which are not always relevant to the topic. We use all three query fields to match query terms to documents.

The last two language samples from which we draw POS blocks, namely LS4 and LS5, are also the test collections from which we retrieve relevant documents, namely WT2G and WT10G. Even though these are Web collections, as mentioned in Section 1, we are not experimenting with Web retrieval, but with the ad hoc task of the relevant Web track. We select these two Web collections, instead of a non-Web TREC collection, such as the AP collection, for example, because we have more queries available for WT2G and WT10G. Thus, these two collections provide a better testbed for our experiments.

We do not expect using the same language sample from which we extract shallow syntactic evidence as a test collection for retrieval to affect retrieval performance, as this shallow syntactic evidence is used to reformulate the queries, and does not affect the test collections used for retrieval in any way. This point is experimentally confirmed and discussed in Section 5.2.

During indexing, we remove stopwords, and stem the collections and the queries, using Porter's stemming algorithm[4]. We use the Terrier (Ounis et al., 2005, 2006) IR platform, and apply three different term weighting schemes to match query terms to document descriptors. By doing so, we aim to test the SQR technique on top of three statistically different term weighting approaches. Specifically, we use the classical TF · IDF weighting scheme (Robertson, 1995; Sparck-Jones, 1972), the well-established probabilistic BM25 weighting scheme (Robertson, 1995), and the more recent PL2 weighting scheme from the Divergence From Randomness (DFR) framework (Amati, 2003). In DFR, term relevance is measured in terms of the divergence of the actual term distribution from that obtained under a random process. For all three weighting schemes, we calculate the query term weight as follows: $qtw = \frac{qtf}{qtf_{max}}$, where qtf is the query term frequency, and qtf_{max} is the maximum qtf among all query terms.

We use the default values of all parameters of the above weighting schemes (Amati, 2003; He & Ounis, 2003; Robertson, 1995), which are shown in Table 3. We use default values, instead of tuning the term weighting parameters, because our focus lies in evaluating our query reformulation technique, and not in optimising overall retrieval performance. If these term weighting parameters are optimised, retrieval performance may be further improved. We measure retrieval performance using the Mean Average Precision (MAP) measure.

After we have examined the retrieval performance associated with SQR across the aforementioned different settings, we compare SQR against the recent Bo1 pseudo-relevance feedback (PRF) model from the DFR framework, which is based on the Bose-Einstein statistics. Bo1 measures the importance of a term by the divergence of its distribution in a pseudo-relevance document set from a random distribution (Amati, 2003). The Bo1 PRF model has been shown to be a very effective and robust query expansion mechanism (Amati, 2003; Amati & van Rijsbergen, 2002; Macdonald et al., 2005; Plachouras, He, & Ounis, 2004). Bo1 estimates the weight $w(t)$ of term t in the expanded query as follows:

---

[4] http://snowball.tartarus.org/

Table 4

Values of the PRF parameters used, per document collection, term weighting model, and for all query field combinations (T, TD, and TDN)

$\frac{\text{Top relevant terms}}{\text{Top retrieved documents}}$ ratio used in PRF (BO$_1$)

Collection	TF · IDF	BM25	PL2	
	(T, TD, TDN)	(T, TD, TDN)	(T)	(TD, TDN)
WT2G	$\frac{20}{5}$	$\frac{20}{5}$	$\frac{30}{5}$	$\frac{30}{5}$
WT10G	$\frac{10}{5}$	$\frac{20}{5}$	$\frac{5}{5}$	$\frac{5}{5}$

$$w(t) = tf_x \cdot \log_2 \frac{1 + P_n}{P_n} + \log_2(1 + P_n)$$

where *tfx* is the frequency of the query term in the *x* top-ranked documents, P_n is given by $\frac{F}{N}$, *F* is the frequency of the term in the collection, and *N* is the number of documents in the collection. The Bo1 PRF mechanism expands the query by merging the extracted terms with the original query terms. The query term weight *qtw* in the expanded query is given by the following formula:

$$qtw = \frac{qtf}{qtf_{\max}} + \frac{w(t)}{\lim_{F \to tf_x} w(t)} = F_{\max}\log_2\frac{1 + P_{n,\max}}{P_{n,\max}} + \log_2(1 + P_{n,\max})$$

where $\lim_{F \to tf_x}$ is the theoretical upper bound of $w(t)$, $P_{n,\max}$ is given by $F_{\max}/N$, and $F_{\max}$ is the frequency *F* of the term with the maximum $w(t)$ in the top-ranked documents. If an original query term does not appear in the most informative terms extracted from the top-ranked documents, its query term weight remains equal to the original one.

Bo1 has two parameters, namely the relevant terms added, and the top retrieved documents from which those relevant terms are selected. These two parameters are given as a ratio of relevant terms / top retrieved documents. We tune this ratio empirically, on the basis of the corresponding relevance assessments available for the queries and collections employed, so as to maximise retrieval performance. This provides us with a strong PRF baseline, against which we can compare our proposed SQR technique. The selected most relevant terms from the top retrieved documents are displayed in Table 4.

In order to undergo a fair comparison between SQR and PRF, we empirically select the settings associated with the highest retrieval performance for SQR, on the basis of the language sample size and top *k* shallow syntactic blocks.

To recapitulate, we organise our evaluation in three parts, which are listed below. It should be noted that each part uses a different baseline.

- *Part 1* (Section 5.2.1): we use different term weighting models at default settings as a baseline. We apply SQR on top of these weighting models, and investigate the effect of varying two parameters of SQR, namely (i) the size of the language sample from which shallow syntactic evidence is induced, and (ii) the number *k* of top most frequent POS blocks used to reformulate the queries. We vary language sample size by using five differently sized language samples. We also test the effect of employing different values of the top *k* most frequent POS blocks upon SQR, by testing five different *k* values. This part answers our first research question, which was presented in Section 3.2 (page 7).
- *Part 2* (Section 5.2.2): we use the Bo1 PRF model, set at empirically optimised settings, as our new baseline. Against this new baseline, we compare the SQR runs that are associated with the highest retrieval performance, as per Part 1. This part answers our second research question, which was presented in Section 3.2 (page 7).
- *Part 3* (Section 5.2.3): we select the single best between PRF and SQR as our new baseline. Against this new baseline, we now compare a technique that combines SQR + PRF. For the SQR + PRF combination, we use the previously reported optimal settings of SQR and PRF (Part 2). This part answers our third research question, which was presented in Section 3.2 (page 7).

5.2. Evaluation results

5.2.1. Evaluation of SQR with respect to (i) language sample size, and (ii) the number k of top most frequent POS blocks

In our evaluation experiments, we first set out to investigate the effect of the size of the language sample used to draw shallow syntactic evidence for SQR, upon retrieval performance. To that end, we employ a baseline which consists of the three term weighting models, with each of the two test collections, and the original queries. We apply SQR using shallow syntactic evidence drawn from five language samples of different sizes. Additionally, we test the effect of varying the number k of top most frequent POS blocks used for SQR upon retrieval performance. Tables 5 and 6 contain the corresponding MAP scores, when retrieving from the WT2G and WT10G test collections, respectively.

From Tables 5 and 6 we see that, overall, SQR can markedly improve retrieval performance, compared to the baseline consisting solely of the weighting models. The size of the language sample from which we draw shallow syntactic evidence does not seem to have a great effect on retrieval performance, as can be seen from the variance of the MAP scores across the language samples and as per weighting model used (column σ^2 in Tables 5 and 6). Specifically, there is little variation in retrieval performance as language sample size changes, suggesting that the language samples used are representative of general language use, to the extent that their size does not affect their representativeness. Tables 5 and 6 also indicate that when we vary the amount of shallow syntactic evidence employed, in terms of the k most probable POS blocks, retrieval performance remains relatively unaffected, as can be seen from the variance of the MAP scores across the range of k values tested (rows $\sigma^2_{\text{TF-IDF}}$, σ^2_{BM25}, and σ^2_{PL2} in Tables 5 and 6).

Selecting the top 5 and top 10 most probable POS blocks works best, when retrieving relevant documents both from WT2G and WT10G (see shaded cells in Tables 5 and 6). This may be explained by the fact that there is a noted division in the way POS blocks are distributed, in terms of their frequency and frequency rank.

Table 5
Mean average precision (MAP) scores of the SQR runs, for WT2G, for each of the five language samples (LS1-5) used; base is the baseline; Δ% is the % difference in MAP from the baseline; *marks statistical significance ($p < 0.05$) between SQR runs and the respective baseline runs, as per the Wilcoxon matched-pair signed rank test; k is the number of most frequent POS blocks used; σ^2 in the right-most column is the variance of the MAP scores across the five language samples as per weighting model; σ^2 in the last three rows is the variance of the MAP scores across the five k values tested, for each language sample and weighting model; best scores across all language samples per k are in italics; best scores across all k values per language sample are in square brackets (italics); best overall score is in bolditalic

WT2G collection													
Model	Base	LS1 75MB	Δ%	LS2 425MB	Δ%	LS3 742MB	Δ%	LS4 2GB	Δ%	LS5 10GB	Δ%	k	σ^2
TF · IDF	0.276	0.274	−0.7	0.285	+3.3	0.286	+3.6	0.276	none	0.275	−0.4	100	2.7^{-05}
BM25	0.280	0.278	−0.7	0.288	+2.8	0.289	+3.2	0.280	none	0.280	none		2.1^{-05}
PL2	0.268	0.279	+4.1	*0.292**	+8.9	0.290*	+8.2	0.280	+4.5	0.283	+5.6		2.8^{-05}
TF · IDF	0.276	0.282	+2.2	0.284	+2.9	0.285	+3.3	0.285	+3.3	0.283	+2.5	50	1.4^{-06}
BM25	0.280	0.282	+0.7	0.285	+1.8	0.288	+2.8	0.287	+2.5	0.286	+2.1		4.2^{-06}
PL2	0.268	0.280	+4.5	0.290*	+8.2	0.292*	+8.9	0.291*	+8.6	*0.293**	+9.3		2.2^{-05}
TF · IDF	0.276	0.290	+5.1	0.287	+4.0	0.287	+4.0	0.291	+5.4	0.288	+4.3	30	2.6^{-06}
BM25	0.280	0.289	+3.2	0.289	+3.2	0.287	+2.5	0.293	+4.6	0.291	+3.9		4.2^{-06}
PL2	0.268	0.291*	+8.6	0.297*	+10.8	0.297*	+10.8	*0.299**	*+11.6*	0.297*	+10.8		7.4^{-06}
TF · IDF	0.276	0.291	+5.4	0.293	+6.1	0.293	+6.1	0.292	+5.8	0.291	+5.4	10	8.0^{-07}
BM25	0.280	0.291	+3.9	0.293	+4.6	0.293	+4.6	0.292	+4.3	0.292	+4.3		5.6^{-07}
PL2	0.268	0.296*	+10.4	[0.312]*	[+16.4]	[0.311]*	[+16.0]	0.307*	+14.5	0.308*	+14.9		3.2^{-05}
TF · IDF	0.276	0.290	+5.1	0.292	+5.8	0.292	+5.8	0.291	+5.4	0.298	+8.0	5	7.8^{-06}
BM25	0.280	0.289	+3.2	0.293	+4.6	0.293	+4.6	0.292	+4.3	0.298	+6.4		8.4^{-06}
PL2	0.268	[0.301]*	[+12.3]	0.309*	+15.3	0.309*	+15.3	[0.308]*	[+14.9]	[*0.314*]*	[*+17.2*]		1.7^{-05}
$\sigma^2_{TF·IDF}$	–	4.3^{-05}	–	1.3^{-05}	–	$1.1^{-0.5}$	–	3.6^{-05}	–	6.0^{-05}	–		
σ^2_{BM25}	–	2.4^{-05}	–	9.4^{-06}	–	6.4^{-06}	–	2.4^{-05}	–	3.7^{-05}	–		
σ^2_{PL2}	–	7.5^{-05}	–	8.0^{-05}	–	7.5^{-05}	–	1.1^{-04}	–	1.2^{-04}	–		

Table 6

Mean average precision (MAP) scores of the SQR runs, for WT10G, for each of the five language samples (LS1-5) used; base is the baseline; Δ% is the % difference in MAP from the baseline; *marks statistical significance ($p < 0.05$) between SQR runs and the respective baseline runs, as per the Wilcoxon matched-pair signed rank test; k is the number of most frequent POS *blocks* used; σ^2 in the right-most column is the variance of the MAP scores across the five language samples as per weighting model; σ^2 in the last three rows is the variance of the MAP scores across the five k values tested, for each language sample and weighting model; best scores across all language samples per k are in italics; best scores across all k values per language sample are in square brackets (italics); best overall score is in bolditalic

WT10G collection													
Model	Base	LSI 75MB	Δ%	LS2 425MB	Δ%	LS3 742MB	Δ%	LS4 2GB	Δ%	LS5 10GB	Δ%	k	σ^2
TF · IDF	0.231	0.240	+3.9	0.243	+5.2	0.245	+6.1	0.247	+6.9	0.245	+6.1		5.6^{-06}
BM25	0.234	0.242	+3.4	0.244	+4.3	0.247	+5.5	0.247	+5.5	0.246	+5.1	100	3.8^{-06}
PL2	0.237	0.249	+5.1	0.252	+6.3	0.254	+7.2	*0.255*	*+7.6*	*0.255*	*+7.6*		5.2^{-06}
TF · IDF	0.231	0.237	+2.6	0.240	+3.9	0.243	+5.2	0.242	+4.8	0.240	+3.9		4.2^{-06}
BM25	0.234	0.237	+1.3	0.240	+2.6	0.243	+3.8	0.243	+3.8	0.241	+3.0	50	5.0^{-06}
PL2	0.237	0.250	+5.5	0.253	+6.7	*0.257*	*+8.4*	*0.257*	*+8.4*	0.256	+8.0		7.4^{-06}
TF · IDF	0.231	0.238	+3.0	0.238	+3.0	0.238	+3.0	0.240	+3.9	0.238	+3.0		6.4^{-07}
BM25	0.234	0.239	+2.1	0.238	+1.7	0.239	+2.1	0.240	+2.6	0.239	+2.1	30	4.0^{-07}
PL2	0.237	0.253	+6.7	0.255	+7.6	0.252	+6.3	*0.257*	*+8.4*	0.256	+8.0		3.4^{-06}
TF · IDF	0.231	0.230	−0.4	0.241	+4.3	0.243	+5.2	0.244	+5.6	0.244	+5.6		2.8^{-05}
BM25	0.234	0.233	−0.4	0.242	+3.4	0.244	+4.3	0.245	+4.7	0.244	+4.3	10	1.9^{-05}
PL2	0.237	0.250	+5.5	0.258	+8.9	*0.261*	*+10.1*	0.256	+8.0	*0.261*	*+10.1*		1.6^{-05}
TF-IDF	0.231	0.233	+0.9	0.243	+5.2	0.245	+6.1	0.248	+7.3	0.243	+5.2		2.5^{-05}
BM25	0.234	0.236	+0.8	0.243	+3.8	0.245	+4.7	0.248	+6.0	0.244	+4.3	5	1.6^{-05}
PL2	0.237	[0.256]	[+8.0]	[0.260]	[+9.7]	[0.262]	[+10.5]	[**0.264**]*	[**+11.4**]	[0.263]	[+11.0]		8.0^{-06}
$\sigma^2 TF · IDF$	–	1.3^{-05}	–	3.6^{-06}	–	6.6^{-06}	–	9.0^{-06}	–	6.8^{-06}	–		
σ^2_{BM25}	–	9.0^{-06}	–	4.6^{-05}	–	7.0^{-06}	–	8.2^{-06}	–	6.2^{-06}	–		
σ^2_{PL2}	–	6.6^{-06}	–	9.0^{-06}	–	1.5^{-05}	–	1.0^{-05}	–	1.0^{-05}	–		

As Fig. 1 graphically displays, there are a few very highly probable POS blocks, after which the remaining POS blocks decrease in probability of occurrence sharply. The top 5 and top 10 POS blocks that seem to work best for SQR correspond to these few highly probable POS blocks.

Additionally, we observe that all three term weighting models perform relatively consistently with SQR, throughout the variations in language sample size and number k of most frequent POS blocks used. TF · IDF and BM25 are shown to benefit less from SQR than does PL2. Notably, PL2 with SQR marks the overall highest MAP score, when retrieving relevant documents both from WT2G and WT10G.

Overall, SQR is associated with an improvement in retrieval performance over the baseline of using solely weighting models, which is sometimes statistically significant, with one exception. This exception consists of the extreme case of using the largest of the used k values of POS blocks, namely 100, from the smallest language sample, namely LS1. In this case, SQR results in a slight deterioration in retrieval performace for TF · IDF and BM25 only, and solely with the WT2G test collection (Table 5). This case is described as extreme for the following two reasons:

(i) the language sample used represents an atypical domain of language, namely the spoken language used in the European Parliament (Section 5.1). Thus, the POS blocks induced from it are likely to be less representative of the average written language use, than the POS blocks extracted from the remaining language samples;

(ii) by inducing the highest of the used values k, namely 100, POS blocks from such an atypical (language-wise) collection, we effectively introduce atypical, and thus 'syntactically noisy' blocks to the list of POS blocks assumed to be content-bearing by SQR.

Additionally, from Tables 5 and 6, we can see clearly that inducing shallow syntactic evidence from the test collection used for retrieval does not affect SQR performance. This observation provides an answer to our first
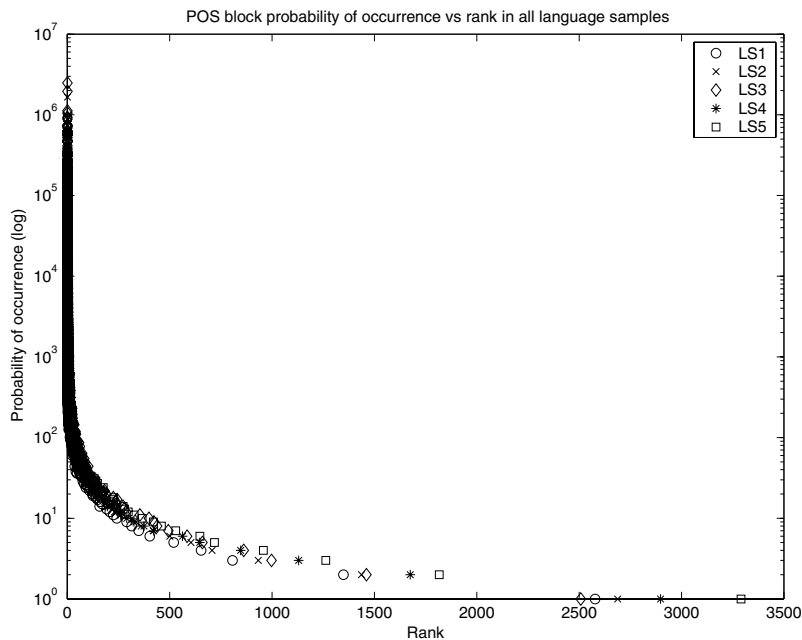
Fig. 1. POS block rank (x-axis) plotted against the POS block probability of occurrence (y-axis) in all language samples.

research question, which was posed in Section 3.2. The best MAP score when retrieving documents from the WT2G test collection is associated with POS blocks induced from LS5, which is the WT10G collection; similarly, the best MAP score when retrieving documents from the WT10G test collection is associated with POS blocks drawn from LS4, which is the WT2G collection. This may be explained by the fact that, generally, the lexical properties of test collections might differ from one test collection to another, as some collections might include unique terms that other collections may not include. On the contrary, the shallow syntactic evidence used by SQR is not unique to specific collections, but actually present in all of them. This follows from the fact that POS blocks are very highly recurrent in language. The specific type of POS blocks used in SQR consists of combinations of 14 unique types of part of speech, which are contained in Table 1. Put more simply, it is as if a whole document collection contained solely 14 unique terms. SQR extracts POS blocks of those 14 unique terms (which are in fact the 14 parts of speech). Since the POS blocks used in SQR are present in all language samples, the only difference between the type of POS blocks induced from different language samples is their frequency ranking. Drawing POS blocks from the same collection that is used to retrieve documents does not affect retrieval performance, because it does not add any unique information to the query that the other language samples do not. This reasoning is based on the assumption that shallow syntactic evidence is induced from representative language samples.

The above observations collectively indicate that the proposed SQR technique is a robust automatic query reformulation technique for IR.

5.2.2. Evaluation of SQR with respect to pseudo-relevance feedback (PRF)

Having thus 'roadtested' our SQR technique, we now wish to compare it against a strong, state-of-the-art PRF technique. To this end, and in order to have a fair comparison between the two, we use the tuning settings that are associated with the best retrieval performance, for both PRF and SQR, as mentioned in Section 5.1. We optimise Bo1, so as to have a strong baseline against which to compare our empirically-tuned SQR technique. Also, we empirically select the best reported settings for SQR (as per Tables 5 and 6), namely:

(i) $k = 5$, and language sample size = 10GB (LS5), when retrieving from WT2G, and
(ii) $k = 5$, and language sample size = 2GB (LS4), when retrieving from WT10G.

This empirical selection is the choice of language sample size and k settings, associated with the best retrieval performance, as displayed in Tables 5 and 6. Table 7 displays the MAP scores of the optimised PRF and SQR runs, separately for WT2G and WT10G.

From Table 7 we can conclude that our proposed SQR technique is at least comparable to the Bo1 PRF mechanism. For the WT2G collection, SQR is almost negligibly outperformed by PRF with TF-IDF and BM25, but outperforms PRF with PL2, and achieves the best overall performance. With the WT10G collection, SQR outperforms PRF at all times, with a statistically significant best overall score for PL2. The general trend emerging from these runs is that SQR benefits PL2 much more that it benefits TF-IDF and BM25. Overall, our proposed SQR technique is shown to perform satisfactorily, robustly, and comparably to a strong PRF baseline, as it outperforms PRF, for the majority of runs, while also producing the best overall MAP score, for both test collections.

5.2.3. Evaluation of SQR and PRF combined

Having tested the robustness and effectiveness of our proposed SQR as a query reformulation technique for IR, we assess its compatibility with conventional PRF. By doing so, we wish to evaluate the theoretical assumption of our SQR model, which holds the following: just as in natural language, communication is achieved by the combination of lexical and shallow syntactic features, similarly in IR, the automatic processing of lexical information may be successfully assisted, in a compatible way, by the automatic processing of shallow syntactic information. We test this by combining SQR with PRF, and comparing them to the best corresponding run achieved either by SQR or PRF alone. By 'combining', we denote the process of:

Table 7

Mean average precision (MAP) scores of the optimised PRF and SQR runs; $\Delta\%$ is the % difference in MAP between them; *indicates statistical significance ($p < 0.05$) between corresponding SQR and PRF runs, as per the Wilcoxon matched-pair signed rank test; best overall scores appear in boldface

Model	PRF	SQR	$\Delta\%$
WT2G collection			
TF-IDF	0.299	0.298	−0.3
BM25	0.302	0.298	−1.3
PL2	0.285	**0.314**	+10.2
WT10G collection			
TF-IDF	0.234	0.248	+6.0
BM25	0.240	0.248	+3.3
PL2	0.237	**0.264***	+11.4

Table 8

Mean average precision (MAP) scores of the best among SQR and PRF (in column *best single*), against the merged SQR + PRF; $\Delta\%$ is the % difference in MAP between the corresponding runs; * marks statistical significance ($p \leqslant 0.05$) between the corresponding best single and combined runs, as per the Wilcoxon matched-pair signed rank test; best overall scores appear in boldface

Model	Best single	SQR + PRF (best)	$\Delta\%$
WT2G Collection			
TF · IDF	0.299 (PRF)	0.331 (**0.332**)	+10.7 (+11.0)
BM25	0.302(PRF)	0.326 (0.331)	+7.9 (+9.6)
PL2	0.314 (SQR)	0.324 (0.327)	+3.2 (+5.1)
WT10G collection			
TF · IDF	0.248 (SQR)	**0.270*** (–)	+8.9 (–)
BM25	0.248 (SQR)	0.264* (–)	+6.4 (–)
PL2	0.264 (SQR)	0.265 (–)	+0.4 (–)

The bracketed scores in the third column relate to SQR + PRF runs, for which the language sample size and top k POS blocks used are retuned as follows: (i) $k = 10$, and language sample size = 10 GB for the WT2G collection, and (ii) $k = 5$, and language sample size = 2 GB for the WT10G collection; – indicates identical figures.

(i) dropping query terms using SQR, followed by;
(ii) expanding the remaining query terms using PRF.

Table 8 includes the relevant MAP scores. For the combined SQR + PRF runs, we use the same weighting model and PRF parameters that we used for the corresponding single SQR and PRF runs separately (Tables 3 and 4, p. 16).

Table 8 reveals that SQR can be combined with PRF successfully, indicating that the two techniques complement one another. The MAP scores of the merged SQR and PRF runs achieved lead to a pronounced improvement in retrieval performance. More importantly, we see that lexical relevance feedback and shallow syntactic query reformulation can work together successfully, in an equally robust way.

Table 9 displays the difference of: (1) the best SQR, (2) the best PRF,and (3) the best PRF + SQR runs in MAP over a baseline that uses only the weighting model. The best SQR and PRF scores are displayed in Table 7, while the best PRF + SQR scores are displayed in Table 8. The baseline scores that use only the weighting: model are displayed in Tables 5 and 6, for WT2G and WT10G, respectively. We observe that:

(i) PL2 benefits from SQR more than do TF · IDF and BM25, for both collections, and
(ii) PL2 benefits from PRF less than do TF · IDF and BM25, for both collections. This difference seems to be bridged with PRF + SQR, where all three weighting schemes appear to improve in MAP approximately similarly (as is indicated by the more or less uniform Δ% figures under column Δ% SQR + PRF). The only divergence from this appears to be the difference in MAP from the baseline marked by TF · IDF for WT10G when using PRF + SQR, which is not very pronounced anyway. Overall, we may conclude that any pronounced differences that exist between how much PRF helps retrieval and how much SQR helps retrieval separately, become less pronounced when PRF and SQR are combined. Hence, it appears that PRF + SQR have an additive, rather than dampening effect on retrieval performance.

Finally, we summarise our results and present them alongside those related to shorter queries, namely T and TD. For these runs, we employ the same default values of all weighting model and PRF parameters displayed in Tables 3 and 4, respectively (p. 16). We compare the retrieval performance associated with T and TD queries, without and with PRF, to the best retrieval performance associated with long queries (TDN), when using either SQR alone, or SQR + PRF. Table 10 displays the relevant scores associated with these runs.

The retrieval performance scores displayed in Table 10 indicate the following points:

(i) TDN queries assisted by SQR perform better than the best-performing between T and TD queries that use only the weighting model, with one exception (PL2 when retrieving TD queries from the WT2G collection), where TD slightly outperforms TDN with SQR. Notably, when retrieving from the WT10G collection, TDN queries assisted by SQR outperform, all T and TD queries, without and with PRF.

Table 9
Difference in mean average precision (MAP) scores of the baseline versus SQR and PRF, both separately and combined: $\Delta\%_{BASE\,vs\,SQR}$ is the % difference in MAP between the baseline versus the best SQR: $\Delta\%_{BASE\,vs\,PRF}$ is the % difference in MAP between the baseline versus the best PRF: $\Delta\%_{BASE\,vs\,SQR\,+\,PRE}$ is the % difference in MAP between the baseline versus the best SQR + PRF: biggest Δ% appears in boldface

Model	Δ% SQR	Δ% PRF	Δ% SQR + PRF
WT2G collection			
TF · IDF	+7.8	+8.3	+20.29
BM25	+6.4	+7.9	+18.21
PL2	+17.2	+6.3	**+22.01**
WT10G collection			
TF · IDF	+7.4	+1.3	**+16.9**
BM25	+6.0	+2.6	+12.8
PL2	+11.4	–	+11.8

Table 10
Mean average precision (MAP) scores of the best scores marked for T, TD and TDN queries; *w.m.* denotes the use of the weighting model alone; best overall scores appear in boldface

Model	w.m.			SQR	PRF			PRF + SQR
	T	TD	TDN	TDN	T	TD	TDN	TDN
WT2G collection								
TF · IDF	0.270	0.295	0.276	0.298	0.313	0.327	0.299	**0.332**
BM25	0.276	0.293	0.280	0.298	0.314	**0.331**	0.302	**0.331**
PL2	0.227	0.319	0.268	0.314	0.234	0.325	0.285	**0.327**
WT10G collection								
TF · IDF	0.188	0.231	0.231	0.248	0.192	0.253	0.234	**0.270**
BM25	0.189	0.233	0.234	0.248	0.192	0.240	0.240	**0.264**
PL2	0.210	0.255	0.237	0.264	0.231	0.255	0.237	**0.265**

(ii) TDN queries assisted by SQR + PRF perform markedly better than the best-performing between T and TD queries that use the weighting model and PRF, apart from one run (BM25 when retrieving from WT2G), for which retrieval performance is the same.

Note that the MAP scores displayed in the SQR + PRF column of Table 10 compare favourably to the high-scoring equivalent TREC runs, namely, 0.324 (and 0.383 when using Web evidence) (Robertson & Walker, 2000), for TREC-8, and 0.269 (Fujita, 2001), for TREC-9. On these grounds, we may conclude that the underlying assumption implemented in SQR, namely the fact that content fragments may be approximately identified on the basis of shallow-syntactic recurrence statistics, seems to be valid. This is justified by the experimental evidence presented in this section, which shows that long queries, when assisted by SQR, tend to outperform shorter queries, when in fact, shorter queries are proven to be most efficient.

We have thus seen that SQR is a robust query reformulation technique, which is not greatly affected by the size of the language sample from which shallow syntactic evidence is drawn, provided that this sample is representative of language use in general. This point means that SQR is a flexible automatic query reformulation technique, portable to language samples of different size. Additionally, SQR works similarly robustly across a varied number of top most probable POS blocks used as shallow syntactic evidence. The advantage of this feature is that SQR can lead to improved retrieval performance, using few POS blocks, which keeps the processing time and general computational cost associated at low levels. We have also shown that the assistance of SQR to retrieval performance is due to the fact that it reduces the amount of noise in the queries successfully, since it outperforms even shorter queries, that are generally considered more effective. Finally, the evaluation results presented in this work indicate that SQR is at least comparable to, if not significantly better than, the effective Bo1 PRF technique, while the combination of the two proves to be overwhelmingly effective in terms of retrieval performance.

6. Applications

In this paper, we have applied SQR to long queries, for two main reasons. Firstly, long queries are more likely to contain full sentences, which we can POS tag and hence extract POS blocks from, than shorter queries. Secondly, long queries are more likely to contain noise, which we can attempt to reduce using POS blocks, than shorter queries. The type of long queries used in our experiments are typical of the TREC ad hoc style, which was created by TREC as part of a controlled experimental environment, with the aim to allow for a measurable and analytical evaluation of IR as a process (Voorhees & Harman, 2005). It is exactly under the same light that we consider long queries, as the best setting which would allow us to measure and analyse our motivation, namely that there seems to exist an approximately proportional relation between the frequency and content associated to POS blocks (see Section 3.1). Hence, in this paper we report on the validation of our assumption, within a controlled experimental setting of long queries. We consider this to be a first step towards applying this assumption in different ways, and towards more realistic retrieval applications, such

as a filter that cleans document collections from noise, or as a criterion of index compression, for example. Indeed, we successfully applied the latter in our participation in the Terabyte ad hoc track of TREC 2006 (Lioma et al., 2006). SQR could be also applied as a filter for first-pass retrieval for successfull when document summaries have been used (Sakai & Sparck Jones, 2001). Such application, namely index filtering, indexing compression, summary-assisted PRF, have been reported to be very beneficial for early-precision, a feature which is most attractive to real-user retrieval tasks, since most users do not look further than the top ten-twenty retrieved documents. Lastly, more general applications of our assumption could be found in fields that aim to automatically identify and process content, such as document summarisation, for example.

7. Conclusion

We described a syntactically-based query reformulation (SQR) technique, applied to enhance the performance of an IR system. We formulated three research questions, namely: (i) is the size of the language sample from which shallow syntactic evidence is drawn important? (ii) is SQR an effective query reformulation technique for IR? (iii) can SQR be successfully combined with pseudo-relevance feedback? We investigated these questions using five language samples of different sizes, across two standard test collections, and three statistically different term weighting models. We compared SQR to a state-of-the-art pseudo-relevance feedback (PRF) technique, firstly by comparing the two techniques to one another, and secondly by merging them into one relevance feedback combination. We foun that SQR is generally not affected by the size of the language sample from which shallow syntactic evidence is drawn, nor by any linguistic similarity between the language sample and the test collection used for retrieval. Moreover, experimental findings indicate that SQR is an overall robust and effective query reformulation technique. Additionally, SQR is shown to be at least comparable, if not significantly better, to a good PRF technique, and even to complement PRF, as the best overall retrieval performance is marked when both techniques are merged.

In the future, we wish to investigate combining POS blocks of different lengths for SQR, as well as specifying shallow syntactic categories within the POS blocks used in SQR.

References

Abney, S. (1991). Parsing by chunks. In R. Bernwick, S. Abney, & C. Tenny (Eds.), *Principle-based parsing*. Kluwer Academic Publishers.
Abney, S. (1996). Chunk stylebook. Internal report. http://www.vinartus.net/spa/96i.pdf.
Amati, G. (2003). Probabilistic models for information retrieval based on divergence from randomness. PhD thesis, University of Glasgow.
Amati, G., & van Rijsbergen, C. J. (2002). Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Transactions on Information Systems (TOIS), 20*(4), 357–389.
Bookstein, A., & Swanson, D. (1974). Probabilistic models for automatic indexing. *Journal of the American Society of Information Science, 25*(5), 312–318.
Brown, P. F., Pietra, V. J. D., deSouza, P. V., Lai, J. C., & Mercer, R. L. (1992). Class-based *n*-gram models of natural language. *Computational Linguistics, 18*(4), 467–479.
Bruandet, M. F. (1987). Outline of a knowledge base model for an intelligent information retrieval system. In C. T. Yu & C. J. van Rijsbergen (Eds.), *Proceedings of the tenth annual international ACM/SIGIR conference on research and development in information retrieval (SIGIR 1987)* (pp. 33–43). New Orleans, LA: ACM Press.
Croft, W. B., & Lafferty, J. (2002). *Language modeling for information retrieval*. New York: Springer.
Croft, W. B., & Lewis, D. D. (1987). An approach to natural language processing for document retrieval. In C. T. Yu & C. J. van Rijsbergen (Eds.), *Proceedings of the tenth annual international ACM/SIGIR conference on research and development in information retrieval (SIGIR 1987)* (pp. 26–32). New Orleans, LA: ACM Press.
Damerau, F. J. (1965). An experiment in automatic indexing. *American Documentation, 16*, 283–289.
DeJaco, D., & Garbolino, D. (1986). An information retrieval system based on artificial intelligence techniques. In F. Rabitti (Ed.), *Proceedings of the ninth annual international ACM/SIGIR conference on research and development in information retrieval (SIGIR 1986)* (pp. 214–220). Pisa, Italy: ACM Press.
Diaz, F., & Metzler, D. (2006). Improving the estimation of relevance models using large external Corpora. In E. N. Efthimiadis, S. T. Dumais, D. Hawking, & K. jarvelin (Eds.), *Proceedings of the twenty-ninth annual international ACM/SIGIR conference on research and development in information retrieval (SIGIR 2006)* (pp. 151–161). Seattle, Washington: ACM Press.
Efthimiadis, E. N., & Biron, P. V. (1993). UCLA-Okapi at TREC-2: Query expansion experiments. In *Proceedings of the second text retrieval conference (TREC 1993)* (pp. 278–290). Gaithersburg, MD: NIST.

Fuhr, N., & Robertson, S. E. (1992). Machine learning and relevance feedback. In *Proceedings of the first text retrieval conference (TREC 1992)* (pp. 369–370). Gaithersburg, MD: NIST.

Fujita, S. (2001). Reflections on "Aboutness": TREC-9 evaluation experiments at justsystem. In *Proceedings of the ninth text retrieval conference (TREC 2001)* (pp. 281–288). Gaithersburg, MD: NIST.

Harter, S. P. (1974). A probabilistic approach to automatic keyword indexing. PhD thesis, University of Chicago.

Hawking, D., & Robertson, S. (2003). On collection size and retrieval effectiveness. *Information Retrieval, 6*(1), 99–105.

He, B., & Ounis, I. (2003). A study of parameter tuning for term frequency normalization. In *Proceedings of the twelfth international conference on information and knowledge management (CIKM 2003)* (pp. 10–16). New Orleans, USA: ACM Press.

Hiemstra, D. (2001). Using language models for information retrieval. PhD thesis, University of Twente.

Jacobs, P. S. (1992). Joining statistics with NLP for text categorization. In M. Bates & O. Stock (Eds.), *Proceedings of the third conference on applied natural language processing (ANLP 1992)* (pp. 178–185). Trento, Italy: Association for Computational Linguistics.

Jacobs, P. S., & Rau, L. F. (1988). Natural language techniques for intelligent information retrieval. In Y. Chiaramella (Ed.), *Proceedings of the eleventh annual international ACM/SIGIR conference on research and development in information retrieval (SIGIR 1988)* (pp. 85–99). Grenoble, France: ACM Press.

Karlgren, J. (1993). Syntax in information retrieval. In *Proceedings of the first nordic doctoral symposium on computational linguistics*. Copenhagen, Denmark: NORFA.

Kiparsky, P. (1976). Historical linguistics and the origin of language. In R. S. Hamad, H. D. Steklis, & J. Lancaster (Eds.). *Origins and evolution of language and speech* (vol. 280, pp. 97–103). Annals of the New York Academy of Sciences.

Kraaij, W. (2004). Variations on language modeling for information retrieval. PhD thesis, University of Twente.

Lioma, C., & Ounis, I. (2006). Examining the content load of part of speech blocks for information retrieval. In *Proceedings of the international committee on computational linguistics and the association for computational linguistics (COLING/ACL 2006), Sydney, Australia.*

Lioma, C., Macdonald, C., Plachouras, V., Peng, J., He, B., & Ounis, I. (2006). University of Glasgow at TREC 2006: Experiments in terabyte and enterprise tracks with terrier. In *Proceedings of the fifteenth text retrieval conference (TREC 2006)*. Gaithersburg, MD: NIST.

Macdonald, C., He, B., Plachouras, V., & Ounis, I. (2005). University of Glasgow at TREC 2005: Experiments in terabyte and enterprise tracks with terrier. In *Proceedings of the fourteenth text retrieval conference (TREC 2005)*. Gaithersburg, MD: NIST.

Manning, C. D., & Schutze, H. (1999). *Foundations of statistical language processing*. London: The MIT Press.

Mauldin, M., Carbonell, J., & Thomason, R. (1987). Beyond the keyword barrier: knowledge-based information retrieval. In *Proceedings of the twenty-ninth annual conference of the national federation of abstracting and information services*. Elsevier Press.

Ounis, I., Amati, G., Plachouras, V., He, B., Macdonald, C., & Johnson, D. (2005). Terrier information retrieval platform. In D. E. Losada, & J. M. Fernandez-Luna (Eds.), *Proceedings of the twenty-seventh european conference on information retrieval research (ECIR 2005), Santiago de Compostella, Spain.*

Ounis, I., Amati, G., Plachouras, V., He, B., Macdonald, C., & Lioma, C. (2006). Terrier: A high performance and scalable information retrieval platform. In *Proceedings of the twenty-ninth annual international ACM/SIGIR conference on research and development in information retrieval workshop on open source information retrieval (OSIR 2006)*. Seattle, Washington: ACM Press.

Plachouras, V., He, B., & Ounis, I. (2004). University of Glasgow at TREC-2004: Experiments in web, robust and terabyte tracks with terrier. In *Proceedings of the thirteenth text retrieval conference (TREC 2004)*. Gaithersburg, MD: NIST.

Ramshaw, L., & Marcus, M. (1995). Text chunking using transformation-based learning. In D. Yarovsky & K. Church (Eds.), *Proceedings of the third ACL workshop on very large corpora (WVLC 1995)* (pp. 82–94). Massachussetts, Cambridge: MIT.

Robertson, S. E. (1995). Okapi at TREC-3. In D. K. Harman (Ed.), *Overview of the third text retrieval conference (TREC-3)*. Gaithersburg, MD: NIST.

Robertson, S. E., & Walker, S. (2000). Okapi/Keenbow at TREC-8. In D. K. Harman (Ed.), *Overview of the eighth text retrieval conference (TREC-8)*. Gaithersburg, MD: NIST.

Rocchio, J. J. (1971). Relevance feedback in information retrieval. In *The SMART Retrieval System – Experiments in automatic document processing*. New Jersey: Prentice Hall.

Sakai, T., & Sparck Jones, K. (2001). Generic summaries for indexing in information retrieval. In *Proceedings of the twenty-fourth annual international ACM/SIGIR conference on research and development in information retrieval (SIGIR 2001)* (pp. 190–198). New Orleans, USA: ACM Press.

Salton, G. (1991). Syntactic approaches to book indexing. In *Proceedings of the twenty-ninth annual meeting of the association for computational linguistics (ACL 1991), Berkeley, CA.*

Salton, G., & Buckley, C. (1990). Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science, 41*(4), 288–297.

Salton, G., Fox, E. A., & Voorhees, E. (1985). Advanced feedback methods in information retrieval. *Journal of the American Society for Information Science, 36*(3), 200–210.

Schmidt, H. (1997). Probabilistic Part-of-Speech tagging using decision trees. In D. Jones & H. Somers (Eds.), *New methods in language processing studies. Computational linguistics*. UCL Press.

Smeaton, A. F. (1986). Incorporating syntactic information into a document retrieval strategy: An investigation. In F. Rabitti (Ed.), *Proceedings of the ninth annual international ACM/SIGIR conference on research and development in information retrieval (SIGIR 1986)* (pp. 103–113). Pisa, Italy: ACM Press.

Smeaton, A. F. (1999). *Using NLP or NLP resources for information retrieval tasks: Natural language information retrieval*. Dordrecht, NL: Kluwer Academic Publishers.

Smeaton, A. F., & van Rijsbergen, C. J. (1988). Experiments on incorporating syntactic processing of user queries into a document retrieval strategy. In Y. Chiaramella (Ed.), *Proceedings of the eleventh annual international ACM/SIGIR conference on research and development in information retrieval (SIGIR 1988), Grenoble, France*, pp. 31–51.

Sparck-Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation, 28*, 11–21.

Sparck-Jones, K., & Tait, J. I. (1984). Automatic search term variant generation. *Journal of Documentation, 10*(1), 50–66.

Strzalkowski, T. (1992). TTP: A fast and robust parser for natural language. In *Proceedings of the international conference on computational linguistics (COLING'92), Nantes, France*, pp. 198–204.

van Rijsbergen, C. J. (1979). *Information retrieval*. London: Butterworth.

Voorhees, E. M., & Harman, D. K. (2005). *TREC: Experiment and evaluation in information retrieval*. London: The MIT Press.

Walker, D. E., Karlgren, H., & Kay, M. (1977). *Natural language in information science*. Stockholm: Scriptor.

Xu, J., & Croft, W. B. (1996). Query expansion using local and global document analysis. In *Proceedings of the nineteenth annual international ACM/SIGIR conference on research and development in information retrieval (SIGIR 1996), Zurich, Switzerland*, pp. 4–11.

Yu, C., & Salton, G. (1976). Precision weighting – an effective automatic indexing method. *Journal of the Association for Computer Machinery (ACM), 23*(1), 76–88.

Zipf, G. K. (1949). *Human behaviour and the principle of least effort*. Cambridge, MA: Addison-Wesley Press.

Zukerman, I., & Raskutti, B. (2002). Lexical query paraphrasing for document retrieval. In *Proceedings of the nineteenth international conference on computational linguistics (COLING 2002), Taipei, Taiwan*.