Exploring Query Categorisation for Query Expansion: A Study

Dipasree Pal *, 1, Mandar Mitra †, 1, and Samar Bhattacharya ‡, 2

1Indian Statistical Institute, 203 B.T. Road, Kolkata-700108, India
2Jadavpur University, 188, Raja SCM Road, Kolkata-700032, India

Abstract

The vocabulary mismatch problem is one of the important challenges facing traditional keyword-based Information Retrieval Systems. The aim of query expansion (QE) is to reduce this query-document mismatch by adding related or synonymous words or phrases to the query.

Several existing query expansion algorithms have proved their merit, but they are not uniformly beneficial for all kinds of queries. Our long-term goal is to formulate methods for applying QE techniques tailored to individual queries, rather than applying the same general QE method to all queries. As an initial step, we have proposed a taxonomy of query classes (from a QE perspective) in this report. We have discussed the properties of each query class with examples. We have also discussed some QE strategies that might be effective for each query category.

In future work, we intend to test the proposed techniques using standard datasets, and to explore automatic query categorisation methods.

1 Introduction

The use of Search Engines (SEs) has become an inseparable part of the activities of most computer users. People use SEs in various forms to find information in a wide variety of contexts: from Web search through desktop search and email search to searching through document archives belonging to specific domains such as the medical and legal domains. Depending on the information need, finding the desired information can be a more or less difficult task.

The well-known vocabulary mismatch problem is one significant factor that makes searching difficult. A user’s query Q and a useful document D in a document collection may use different vocabulary to refer to the same concept. Retrieval systems that rely on keyword-matching may not detect a match between Q and D. A good retrieval system must bridge the potential vocabulary gap that exists between useful documents and the user’s query. Query Expansion (QE), the addition of related terms to a user’s query, is one important technique that attempts to solve this problem by increasing the likelihood of a match between the query and relevant documents.

Most lay users prefer to keep their interaction with a retrieval system simple. Thus, most QE methods are completely automatic and involve little or no additional effort on the part of a user. Of course, a completely automatic QE method may end up adding unrelated terms to a user’s query, thus changing the query’s focus. This is known as query drift. In such cases, QE causes performance to deteriorate rather than improve.

* dipasree.pal.gmail.com; Corresponding author. Fax.: +91 33 2577 3035; Tel.: +91 33 2575 2858.
† mandar.mitra@gmail.com; Fax.: +91 33 2577 3035; Tel.: +91 33 2575 2858.
‡ samar_bhattacharya@ee.jdvu.ac.in; Fax.: +91 33 2577 3035; Tel.: +91 33 2414 6129
Over the years, many different QE techniques have been proposed. A recent survey of such
techniques can be found in [Carpineto and Romano, 2012]. While a number of QE techniques
have been shown to be effective on average (i.e. when their overall impact across a large set
of queries is measured), the effect of different QE techniques for individual queries can vary
greatly. Figure 1 makes this point graphically. The points on the X-axis represent individual
queries; the Y-axis denotes the relative improvement in performance obtained for each query
by using query expansion. The lines labelled QE1 and QE2 correspond to two different QE
techniques. Points on QE1 (resp. QE2) that lie above the X-axis correspond to queries for
which this expansion method improves performance, while a point below the X-axis corresponds
to a query for which the method hurts retrieval effectiveness.

Table 1 shows the Mean Average Precision (MAP) scores for three retrieval methods: a baseline
strategy that uses the original, unexpanded queries, QE1 and QE2. QE1 and QE2 are clearly
superior to the baseline on average. This reinforces the claim above that QE techniques often
improve overall performance. However, it is clear from Figure 1 that the impact of QE1 or QE2
varies greatly across queries. Specifically, QE1 and QE2 result in decreased performance for a
number of queries. Also, while the overall performance figures for QE1 and QE2 are comparable,
each of these methods outperforms the other on about half the queries used in this experiment.
The hypothetical performance that would be obtained if one could predict in advance the most
effective technique for a query — no expansion vs. QE1 vs. QE2 — is shown in the last column
(MAX) of Table 1. Notice that such a capability would lead to nearly 35% improvement in
retrieval effectiveness.

|              | Baseline | QE1      | QE2      | MAX       |
|--------------|----------|----------|----------|-----------|
| MAP          | 0.1842   | 0.2191 (+18.95%) | 0.2183 (+18.51%) | 0.2473 (+34.26%) |

Table 1: Potential improvement obtainable in principle by judiciously choosing QE techniques

In their overview of the NRRC Reliable Information Access (RIA) Workshop [Harman and Buckley, 2009] make a similar point: “it may be more important for research to discover what current techniques should be applied to which topics, rather than to come up with new techniques”.

1.1 Problem statement

In this study, we consider the important problem of predicting the most effective QE technique
for a given query (including the possibility that not expanding certain queries may be most
effective). We explore one possible approach to this question. We examine a number of different
criteria that can be used to classify queries. For each query category, we discuss what QE
techniques (or more generally, what query processing techniques) might be most effective.

Our eventual goal is to find methods that can automatically (or semi-automatically, i.e.,
with some assistance from a user) classify a given query into one (or sometimes more) of several

1Details about the dataset and the techniques used to generate these plots can be found in the Appendix.
pre-defined categories. We will then apply the QE method that is most appropriate for this category. Our hypothesis, supported by Table 1 and Harman and Buckley [2009], is that overall performance should improve if we apply QE techniques specifically tailored to a given query, rather than applying the same general QE method to all queries.

1.2 Outline

The rest of this report is organised as follows. Section 2 discusses related work and its relationship to this study. Section 3 presents a taxonomy of query categories. For each category, we provide examples of queries belonging to that class. We also discuss QE techniques that are likely to be most effective for that category. Details about the data presented in this Section are given in Appendix A. We conclude in Section 4 by presenting a summary of the work done along with a roadmap for future work.

2 Related Work

Related work can be broadly classified into three categories: research related to query categorisation, prior work on query expansion, and research on query performance prediction.

2.1 Query classification

Automatic query categorisation (QC) is a well-known problem that has been studied for many years in both the Information Retrieval and Machine Learning communities. QC is usually treated as a multi-class categorisation problem. It is quite different from normal text categorisation, since queries are not as long as text documents.

Different types of query classification approaches have been defined according to the purpose that the classification is intended to serve. A well-known classification of Web queries [Broder, 2002] uses three categories: informational, navigational, and transactional. Navigational queries are entered by users looking for a specific website, whereas informational queries cover a broad topic, for which there are typically many relevant documents. Transactional queries have commercial / transactional purposes. Transactional queries or queries with commercial intent are further classified in [Ashkan and Clarke, 2009] depending on whether the user has “on-line commercial intent” (i.e. intention to purchase a product or utilise a commercial service). Naturally, these categories are not applicable to general-purpose queries that have no commercial intent. On a somewhat related note, Baeza-Yates et al. [2006] classify Web queries according to whether they are informational, non-informational or ambiguous.

Another traditional approach classifies queries according to the domain or subject area targeted by the query. For example, the KDDCUP competition 2005 [Shen et al., 2006] focused on a Web query classification task. This task defined 67 query categories organised into a hierarchical taxonomy, for example Computers / Security, Computers / Software, Entertainment / Celebrities, Sports / Tennis. A single query could belong to multiple categories. For example, relevant documents for the query “Beijing 2008”, may belong to the following domains: Sports / Olympic Games, Information / Local & Regional, Living / Travel and Vacation and Information / Law and Politics. Thus, this query belongs to multiple categories. Competitors were required to classify 800,000 real user queries into the 67 categories. Out of these queries, only 800 queries (randomly chosen) were labeled manually, among which 682 queries belonged to multiple categories [Cao et al., 2009b]. Beitzel et al. [2004] reported 16 categories of Web queries. These query classes are also based on the subject domain of relevant documents, like music, games, entertainment, computer, health,
US-sites. The authors analysed Web traffic on an hourly basis using these query types. They showed that music related queries cover 50% of the total queries, while queries targeted at US-sites cover 35% queries, and queries related to entertainment comprise 5% of the overall query set. Cao et al. [2009a] also classify Web queries into 17 groups with the aim of improving personalised search, but details about these query groups are not available.

In recent times, query classification has become a particularly important problem since most Web search engines earn their revenue via targeted advertisements provided alongside search results. Gabrilovich et al. [2009] classify queries onto a fine-grained commercial taxonomy with approximately 6000 nodes, arranged in a hierarchy with median depth 5 and maximum depth 9. The key idea in this approach is to determine the class of a query by classifying the search results retrieved for that query.

Apart from the targeted domain, queries may also be classified based on certain features, for example, (i) ambiguous queries, (ii) short queries, and (iii) hard or difficult queries. Xu et al. [2009] classify queries into three categories, based on their relationship with Wikipedia topics. These categories are: (i) queries about specific entities; (ii) ambiguous queries; and (iii) all other queries. Hard queries were studied in the Robust Track at TREC [Voorhees, 2003b]. Substantial work has also been done on queries that have multiple aspects [Harman, 1988, Buckley, 2009].

In the next section (Section 3), we discuss various criteria for query classification, including some of the criteria mentioned above. While some of these query types have been defined by other researchers in earlier work, we specifically investigate the relationship between query categories and appropriate QE strategies.

### 2.2 Query performance prediction

Query performance prediction may be regarded as a special case of the query categorisation problem, in which the objective is to classify a query as being either hard or easy for a given retrieval system. Cronen-Townsend et al. [2002] were among the earliest to study the Query Performance Prediction (QPP) problem. They defined the Clarity Score as the relative entropy between a query language model and the corresponding collection language model. This score is intended to measure the ambiguity of a query with respect to a document collection. The authors showed that the Clarity Score is positively correlated with average precision (a standard evaluation measure) on a variety of benchmark datasets.

QPP methods may be classified into two broad categories.

1. **Pre-retrieval** methods. These methods (e.g., the method proposed in He and Ounis [2006]) use only the initial query, and term statistics from the target document corpus collected at indexing time. In particular, no preliminary retrieval results are needed. Hauff et al. [2008] present a survey of pre-retrieval QPP methods.

2. **Post-retrieval** methods. These methods additionally make use of the results retrieved in response to the initial query, usually by analysing the similarity scores for the retrieved documents. The prediction method proposed by Shtok et al. [2009, 2012] may be regarded as a representative post-retrieval method. It is based on the hypothesis that “high standard deviation of retrieval scores in the result list correlates with reduced query-drift, and consequently, with improved effectiveness.”

A good introduction to work in this area can be found in a monograph by Carmel and Yom-Tov [2010]. The monograph provides the background and motivation for the QPP problem. It covers pre-retrieval and post-retrieval methods, as well as methods that combine these two approaches.
Finally, it also discusses applications of query difficulty estimation. A more up-to-date overview is provided in a tutorial by Carmel and Kurland [2012].

Recently, Kurland et al. [2012] have proposed a probabilistic framework for QPP that unifies various earlier, apparently diverse approaches [Cronen-Townsend et al., 2002, Vinay et al., 2006, Yom-Tov et al., 2003, Zhou and Croft, 2006, 2007]. Sondak et al. [2013] generalise this framework by modelling the user’s actual information need (as represented by the query). Their framework makes it possible to integrate pre-retrieval, post-retrieval, and query-representativeness based predictors.

2.3 Query expansion

A great deal of work has been done on QE. Carpineto and Romano [2012] provide a comprehensive and up-to-date survey of various automatic QE techniques. In earlier work on QE, we find that expansion terms (i.e. the terms that are added to the original query) are generally selected from 3 types of sources. On the basis of the source of expansion terms, QE strategies can be divided into the following groups.

- **Local**: “Local” QE techniques select candidate expansion terms from a set of documents retrieved in response to the original (unexpanded) query. Ideally, expansion terms should be drawn from some initially retrieved relevant documents. Since these documents are relevant, terms present in these documents are expected to be related to the query, and should help to retrieve other similar documents which are also likely to be relevant. If the user does not provide any feedback about which of the initially retrieved documents are relevant, certain simplifying assumptions may be made. Usually, in the absence of user feedback, a few top-ranked documents are assumed to be relevant. This is called pseudo relevance feedback (PRF). This method has an obvious drawback: if several of the documents assumed to be relevant are in fact non-relevant, then the words added to the query (drawn mostly from these documents) are unlikely to be useful expansion terms, and the quality of the documents retrieved using the expanded query is likely to be poor. Mitra et al. [1998] propose a local expansion method that tries to prevent query drift by ensuring that the query is expanded in a balanced way. Xu and Croft [1996, 2000] present a method called local context analysis that also obtains candidate expansion terms from a few top-ranked documents. These terms are scored on the basis of their co-occurrence patterns with all of the query terms. The highest scoring terms are added to the query. Recently, Colace et al. [2015] have demonstrated the effectiveness of a new expansion method that extracts weighted word pairs from relevant or pseudo-relevant documents. Researchers have also applied learning to rank methods to select useful terms from a set of candidate expansion terms within a PRF framework [Xu et al., 2015].

- **Global**: “Global” QE techniques select expansion terms from the entire database of documents. Candidate terms are usually identified by mining term-term relationships from the target corpus. Qiu and Frei [1993] propose a global QE technique that makes use of a similarity thesaurus. A similarity thesaurus is a matrix containing term-term similarity scores as its entries. These similarity scores are computed based on how word-pairs co-occur in the documents contained in a corpus. Expansion terms are selected on the basis of a probabilistic measure of a term’s relationship to the query concept. Jing and Croft [1994] also propose a global technique, called phrasefinder, that is based on term co-occurrence information in the corpus. Each term $T$ corresponds to a vector $V_T$ of associated (or co-occurring) terms. A term $T$ is assigned a similarity score based on the similarity between the original query and $V_T$. The terms that are most similar to the query are selected as expansion terms.
Gauch et al. [1999] define two words as similar if they occur in similar contexts, where a word’s context is defined in terms of its neighbouring words in a corpus. Words that are similar to the query words are selected for inclusion in the expanded query.

Carpineto et al. [2001] use a combination of local and global approaches. Their hypothesis is that a useful term will occur more frequently in relevant documents than in non-relevant documents or in the whole corpus. Vechtomova et al. [2003] also combine local and global information in the form of long-span collocates — words that significantly co-occur with query terms. Collocates of query terms are extracted from both the entire corpus, as well as from a subset of retrieved documents. The significance of association between collocates is estimated using modified Mutual Information and Z score.

- **External:** “External” QE techniques comprise methods that obtain expansion terms from other resources besides the target corpus. These resources may include other document corpora (including the Web), linguistic resources like Wordnet, and user-query logs. Li et al. [2007] use Wikipedia as a source of expansion terms. Given an initial query, Wikipedia pages are retrieved and reranked on the basis of Wikipedia category information. The “best” wiki pages provide terms for inclusion in the expanded query. Xu et al. [2009] also used Wikipedia as a source of expansion terms. For each query word, the related Wikipedia page (if any) is found; terms from this page are ranked, and top-ranked terms are added to the query. This approach needs few parameter settings, since for each term, only one document is selected.

Voorhees [1994] used Wordnet synsets to find terms related to query words. She showed that only the addition of synonyms of query words does not consistently improve performance. More recently, Fang [2008] showed that Wordnet-based query expansion can yield good results if the definitions (or glosses) of words provided by Wordnet are used instead of simply relying on the semantic relations defined within Wordnet. A comprehensive survey of the uses of ontologies in query expansion can be found in [Bhogal et al., 2007].

### 2.4 Selective query expansion

As mentioned in the Introduction, many of the above QE techniques have been shown to be effective on the whole over large query sets, even though they may cause retrieval effectiveness for individual queries to suffer. Our eventual goal is to formulate a method by which the type of a given query is first determined, and an appropriate expansion strategy is then used based on the query category. In other words, we hope to be able to apply QE techniques tailored to individual queries, rather than applying any particular QE technique uniformly to all queries.

As a special case of this problem, researchers have looked at selective query expansion, i.e., the question of whether to expand a query at all. Amati et al. [2004] define an information theoretic measure that indicates, for a given query, whether it is likely to benefit from expansion. This measure is used to selectively apply QE to only some queries. The authors show that their approach works better than applying QE uniformly across all topics in a test collection. Similarly, Cronen-Townsend et al. [2004] show that a comparison between language models constructed on the basis of the results retrieved by the unexpanded and a given expanded query can be used to predict whether expansion has resulted in altering the sense of the original query. In such cases, QE should be avoided. This idea was shown to be effective in improving the robustness of expansion strategies.
3 Query types

As discussed in Section 2.1, queries may be classified into a wide variety of query types. Thus far, customising online advertising and search result presentation has been the main motivation behind query classification: search engines may tailor the format of the results page or the advertisements displayed in response to a query according to its category. Our goal in this study is to focus on query types from a QE perspective. In other words, we are interested in classification criteria that are likely to have some relation to query expansion. The types we consider are not mutually exclusive. Our intention is that the retrieval system (or the user) will decide the (possibly multiple) categories that a particular query belongs to, and then select the appropriate QE method for these categories.

Table 2 lists the query categories that we are interested in, along with very brief descriptions. Some of these categories can be determined automatically, while for some, a user’s inputs may be required (these categories are marked M). In some cases, it may be difficult to categorise queries before an initial retrieval (and evaluation). For example, to know if a query is hard or not, we need to examine the initial retrieval results. Generally, we need to expand the queries only if we are not satisfied by the initial retrieval. The following sections discuss these categories in more detail.

| No. | Name                              | Characteristics                              |
|-----|-----------------------------------|----------------------------------------------|
| 1   | Short query                       | Few query terms                              |
| 2   | Hard query                        | Low average precision                        |
| 3   | Ambiguous query                   | Meaning of query not clear                   |
| 4   | Query containing negative terms   | Presence of negation                         |
| 5   | Query involving named entities    | Named entities in query                      |
| 6   | Multi-aspect query                | Query contains multiple sub-topics           |
| 7   | High-level query                  | Query uses abstract terms                    |
| 8   | Recall-oriented query (M)         | User requires all/many relevant documents    |
| 9   | Context implicit in query (M)     | Meaning of query determined by context       |
| 10  | Domain specific query (M)         | Related to a particular domain               |
| 11  | Query needing short answer (M)    | Specific answer needed                       |
| 12  | Query needing special processing  | Special indexing techniques may be required   |
|     | Multilingual query (M)            | Query uses more than one language            |
|     | Noisy query (M)                   | Query contains some textual error            |

Table 2: Query categories

3.1 Short / long queries

A query may be classified as short or long based on the number of terms or keywords that it contains. In order to make this notion concrete, we adopt the following definitions.

- **short** queries: queries containing fewer than **four** words
- **long** queries: queries containing more than **ten** terms

These definitions may be regarded as rather arbitrary; however, they are only intended to be indicative. If a query consists of a single named entity that is four words long, it should really be regarded as a short query.

It is generally believed that casual users tend to formulate short queries, while more experienced or professional searchers formulate longer queries that better represent their information need.
Queries provided by various test collections (e.g., TREC “topics”) usually have both a short and a long version. They typically consist of a title, a description and a narrative. The title fields of these queries are short, since they are mostly intended to model queries created by casual users; the descriptions are longer. The Narrative section is only intended to provide a detailed specification of what the user deems relevant; it should generally not be used as a source of keywords. Table 3 shows the maximum and minimum lengths (in words) of different parts of TREC queries.

| Number | Query field | Maximum length | Minimum length |
|--------|-------------|----------------|----------------|
| 1      | Title       | 21             | 1              |
| 2      | Desc        | 46             | 5              |
| 3      | Narr        | 129            | 14             |

(400 queries have narr)

Table 3: TREC queries 1-450: query length in words

Table 4 shows the distribution of the length of the title field of TREC queries 1–450 (queries 201–250 are omitted from this table since they do not contain a title field). We can see from the table that more than half the queries contain no more than 3 words. Only occasionally are they any longer, for example, when the title contains some well known phrase or a long proper name.

| Query length | Number of Queries |
|--------------|------------------|
| 1            | 18               |
| 2            | 101              |
| 3            | 113              |
| 4            | 50               |
| 5            | 32               |
| 6            | 30               |
| >6           | 56               |

Table 4: Distribution of length of titles of TREC queries 1-200 and 251-450

**Benefits of expanding short queries.** We now turn to the relationship between the length of a query and how it may be affected by QE. Given their brevity, it is reasonably likely that a short query is an incomplete representation of the user’s information need. Expanding a short query is likely to yield a more complete representation of the user’s information need. Thus, the benefits of QE are expected to be substantial in the case of short queries. On the other hand, a long query is usually a more comprehensive statement of the searcher’s information need. A higher level of retrieval effectiveness can generally be obtained using long queries, and there is less opportunity for QE techniques to yield dramatic improvements for such queries.

Table 5 illustrates these points. It shows the number of queries for which a standard QE technique results in better / worse performance. QE improves effectiveness for 98 out of 150 short, title-only queries (T). The maximum improvement in MAP over all queries is as much as 0.6016. In contrast, QE yields improvements for fewer medium (TD) or long (TDN) queries; further, the maximum improvement obtained is also substantially smaller for long queries (about 0.45).

**Risks related to expanding long / short queries.** Short queries usually contain only important keywords. Users generally do not include stop-words (words such as articles, con-
Table 5: Improvements due to QE for short / long queries (Query set: TREC678 (queries 301–450), IR system: TERRIER, term-weighting method: IFB2c1.0, QE method: Bo1-based pseudo relevance feedback (40 terms from top ranked 10 documents))

| Query field(s) | MAP (no expansion) | MAP (after QE) | # queries improved (best difference) | # queries hurt (worst difference) |
|----------------|--------------------|----------------|-------------------------------------|----------------------------------|
| T              | 0.2181             | 0.2630         | 98 (0.6016)                         | 50 (0.3404)                     |
| TD             | 0.2560             | 0.2693         | 80 (0.5824)                         | 70 (0.3827)                     |
| TDN            | 0.2567             | 0.2749         | 79 (0.4537)                         | 70 (0.3320)                     |

junctions, prepositions that have a primarily grammatical function) in short queries. Thus, short queries are often not grammatically well-formed sentences or phrases, but this feature is generally an advantage for many QE techniques: all query terms can be assumed to be informative, and every query term is likely to be important during expansion. In contrast, long queries may contain “weak” (relatively less useful / informative) terms in addition to the important keywords. Two examples from the TREC topic set illustrate the important differences between long and short queries.

- **Oil Spill (number-154)**
  Long: A relevant document will note the location of the spill, amount of oil spilled, and the responsible corporation, if known. This will include shipborne accidents, offshore drilling and holding tank spills, but should not include intentional spills such as Iraq/Kuwait or leakage from broken pipes. References to legislation brought about by a spill, litigation and clean up efforts associated with a spill are not relevant unless specifics of the spill are included.

- **Black Monday (number-105)**
  Long: Document will state reasons why U.S. stock markets crashed on 19 October 1987 (“Black Monday”), or report on attempts to guard against another such crash.

The short version of these search topics (“Oil Spill”, “Black Monday”) contain only keywords, but they do not properly describe the user’s information need. In contrast, the long queries contain a clear and detailed specification of the user’s requirement in natural language. However, they contain a number of unimportant or general terms (e.g., relevant, document, note, include, etc.) that would be inappropriate in a keyword-only version of these queries. At the time of expansion, therefore, special care is needed in order to identify the strong terms and to avoid adding words related to weak terms, since this may result in query drift.

On the other hand, because a short query contains few words, it has a greater chance of being ambiguous. Compare, for example, the single term query “SVM” with the longer queries “SVM pattern recognition” (in which SVM refers to Support Vector Machines) and “SVM admission criteria” (in which SVM expands to School of Veterinary Medicine). Expanding such a single-term query by adding words related to the “wrong” sense will also result in query drift. Further, short queries lie outside the scope of QE techniques that use some form of language analysis.

**Special processing for verbose queries.** Most Web search queries are also short, being generally 2 or 3 words long [Beitzel et al., 2005]. However, over the last ten years or so, long, verbose queries are becoming more frequent. In 2006, Yahoo claimed that 17% of its queries contained 5 words or more [Gupta and Bendersky, 2015]. Users create long queries for a variety of reasons. A number of techniques for processing verbose queries have been proposed over the years. Many of these focus on automatic methods for assigning weights to the original query terms that distinguish between useful terms and weak terms [Bendersky and Croft, 2008, Lease...
| Category                                                                 | Examples                                                                 | r  | R  | Remarks                                                                 |
|-------------------------------------------------------------------------|--------------------------------------------------------------------------|----|----|-------------------------------------------------------------------------|
| Queries for which there are very few relevant documents                 | Q303, Q320, Q344                                                         | 10 | 10 | Expanding such queries to target the few “needles in the haystack” is unlikely to be beneficial in any real sense. |
| Queries with several relevant documents, for which recall is reasonably high, but ranking is poor | Q374, Q399, Q435                                                       | 203| 204| Since the relevant documents are retrieved at poor ranks, global expansion techniques may work better. |
| Queries with several relevant documents, but for which recall is poor   | Q307, Q336, Q389                                                       | 25 | 215| Automatic expansion techniques are likely to be inappropriate for such queries. Manual, interactive expansion may work well. |

Table 6: Types of hard queries with examples from the TREC query collection. $R$ denotes the total number of relevant documents for a query, and $r$ denotes the number of relevant documents retrieved for that query within the top 1000 ranks. CHECK: WHAT SYSTEM?

3.2 Hard queries

We characterise a query as hard if no automatic retrieval method yields good performance (as measured by Average Precision (AP), or by the number of relevant documents initially retrieved, for example) for the query.

A number of tracks at TREC have focused on hard queries. The goal of the Robust Track [1] (2003–2005) was to study queries for which performance is generally poor. In 2003, the topic set for this task consisted of a total of 100 queries. The minimum and maximum number of relevant documents for any topic was 4 and 115 respectively. The following year (2004), fifty new topics (651–700) were created for the Robust Track. Later, in its final year, the Million Query Track (2007 – 2009) [2] defined hard queries based on the Average Average Precision (AAP) score for a query, which is the average of AP estimates for a single query over all submitted runs. Difficulty levels were automatically assigned to queries by partitioning the AAP score range into three intervals: [0,0.06) (hard queries), [0.06,0.17) (medium queries), and [0.17,1.0] (easy queries). These intervals were chosen so that queries would be roughly evenly distributed. Of the three, the hard category comprises 38% of all queries, and includes all queries for which no relevant documents were found.

Types of hard queries. Hard queries may be grouped into the sub-categories shown in Table 6 based on their properties.

By definition, the initial retrieval results are poor for hard queries. In other words, the top retrieved set contains more irrelevant documents than relevant ones. PRF-based expansion, which assumes that the top-ranked documents are relevant, is unlikely to work well for such queries, and may result in severe performance degradation due to query drift. For example, if we search for information about the TERRIER IR system using only the term “terrier”, most /

\[2009, \text{Bendersky et al., 2011, Paik and Oard, 2014}. \] For a comprehensive overview of these and other approaches to handling verbose queries, please see [Gupta and Bendersky, 2015a].
all top retrieved documents may be related to the breed of dog. Instead of using PRF, adding the terms “IR” and “system” to the query manually is likely to yield definite improvements.

On the other hand, for an easy query (e.g., TREC queries 365, 403 and 423), the original query terms are generally good enough for retrieving relevant documents. Thus, most of the desired documents are retrieved early in the first round, resulting in high AP (AP values for the above queries are 0.8213, 0.8891, 0.7402 resp.). As the user is likely to be satisfied with the results of the initial retrieval, query expansion should be done in a fairly conservative way, if at all, i.e., only a small number of terms (possibly zero) that are strongly related to the original query terms need be added to the query.

For such queries, since the baseline AP is high, AQE techniques (which may be modelled as having an element of stochastic error) are more likely to lead to performance degradation.

Of course, while these categories can be defined easily for a TREC-like test collection, earlier work discussed in Section 2.2 suggests that automatically differentiating between these query types is non-trivial in a real-life setting. The easiest option may be to have the user look at the initially retrieved set and decide whether a given query is hard or easy, and accordingly determine whether expansion is needed or not.

3.3 Ambiguous queries

According to WordNet [Miller, 1995], the term ambiguous means “open to two or more interpretations” or “of uncertain nature or significance” or “(often) intended to mislead”. Extending this definition, we can define an ambiguous query as one whose meaning is not clear, or one which admits of multiple valid interpretations. We categorise ambiguous queries into two groups (analogous to the grouping in Santos et al. [2015]), which are discussed in the rest of this section.

3.3.1 Queries containing polysemous words

We first consider queries that are ambiguous because they contain one or more polysemous words, i.e., words that have multiple meanings. For such queries, a match with a document on an ambiguous term is only weakly suggestive of relevance, since the term may have been used in a different sense from the intended one in the matching document. This problem is more serious if the polysemous word is an important keyword in the query.

Not surprisingly, the TREC query collection contains a number of polysemous words. A few examples are:

- TREC query 350: health and computer terminals. The word terminal may be used as an adjective; it may also refer to an airport terminal.
- TREC query 355: ocean remote sensing. Remote may also be used as a noun (as in a “television remote”).
- TREC query 397: automobile recalls. Recall may be used as a verb, or (less commonly) as the name of a metric.

Sanderson [2008] points out that a large class of ambiguous words viz., words and phrases that are proper nouns, or are used as such, occur rarely in traditional, TREC-like query collections. The query “apple” is a typical example. The word apple may refer to the fruit, or the computer company, or a number of other entities. The term ‘Jaguar’ is another typical example. It could refer to the “big cat” that is formally named Panthera onca, but it could also refer to other objects / entities of more recent origin such as cars, bands, pens or one of several

http://en.wikipedia.org/wiki/Apple_(disambiguation)
http://en.wikipedia.org/wiki/Jaguar_(disambiguation)
http://www.jaguarpen.com
companies. Acronyms with multiple expansions (e.g., “SVM” discussed in Section 3.1), and acronyms that are also valid words (e.g., FIRE, acronym for Forum for Information Retrieval Evaluation) constitute another frequently occurring class of ambiguous queries. These examples show that polysemy in a language generally increases over time, as new concepts may be tagged with words from the existing vocabulary. However, these classes of polysemous queries have not been seriously studied in past research on polysemous queries.

The performance of a system on an ambiguous query depends on the target collection. Naturally, ambiguity is a concern only if the collection actually contains the word used in multiple senses. If the word is used in only one sense in the target collection, then the query is effectively unambiguous for that collection. This may happen, for example, in domain-specific search engines (Section 3.10).

Approaches to handling polysemous query words can broadly be divided into three groups.

**Word sense disambiguation (WSD).** A very large number of studies have focused on the general problem of word sense disambiguation [Navigli, 2009]. A significant body of work has also been done on WSD for IR. Queries may be *explicitly* disambiguated by tagging each polysemous query word with a sense code which is utilised when computing query-document scores. Schütze and Pedersen [1995] showed that a word sense disambiguation algorithm can improve retrieval effectiveness by 7–14%. Their WSD algorithm was applied in conjunction with the standard vector space model for IR. The approach was evaluated using the Category B TREC-1 corpus (WSJ subcollection).

In a later critique, Ng [2011] argues that the question of how effective WSD is for IR remains an unresolved question, with different researchers reporting contradictory findings. He showed that many studies that have demonstrated a positive impact of WSD on IR have made use of small datasets, or weak baselines. It is generally agreed, however, that polysemy is a more serious problem for short queries; it is also generally agreed that in situations where WSD helps IR, an increase in WSD accuracy has a positive impact on IR effectiveness [Sanderson, 2000, Navigli, 2009].

**Implicit WSD.** Retrieval methods may also make use of *implicit* disambiguation methods. For example, consider the query “erosion of river banks caused during rainy season”. Even though the term “bank” is polysemous, a document \( D \) that contains the word in its intended sense is more likely to also contain the terms erosion, river, or rain, as compared to a document \( D' \) that uses the word in the sense of a financial institution. Most reasonable retrieval models will favour \( D \) over \( D' \), thus automatically “selecting” the correct sense of bank. In other words, the intended sense of a polysemous word within a long query may be automatically favoured because of the additional context provided by the other query terms (this point was also discussed in Section 3.1).

Additional context may also be provided by the earlier queries issued by the user within the same session. Cao et al. [2009b] use Conditional Random Fields to model this context, and show that incorporating session information often improves query disambiguation.

**Search result diversification.** The third approach to handling ambiguity, specially in the case of short queries, is search result diversification (SRD) [Santos et al., 2013]. In SRD, the goal of a system is to present a result list that contains documents grouped according to the various possible interpretations of the given query. This allows the user to select the results corresponding to the appropriate sense of the query. The user’s feedback may be used to expand the query, keeping in mind its intended sense.

YIPPY [11] is an example of a real-life search engine that attempts a form of SRD. It presents a

---

10 [http://www.jaguarind.com/aboutus/aboutus.html](http://www.jaguarind.com/aboutus/aboutus.html) [http://www.jaguarltg.com/](http://www.jaguarltg.com/)

11 [http://yippy.com](http://yippy.com)
### Table 7: Examples of underspecified queries. The interpretation in italics is the one specified in a longer version of the query (specifically, in the description field).

| TREC query # | Query title | Possible interpretations |
|--------------|-------------|--------------------------|
| Q260         | Evidence of human life | during a particular period in history? in some geographical locations (e.g., desert islands)? |
| Q364         | Rabies | particular cases and corrective action? which animals are carriers? signs, symptoms, prevention, treatment? overview / encyclopedic entry? |
| Q376         | mainstreaming | of children with physical or mental impairments? of physically disabled persons in general? of tribal / marginalised communities? of juvenile delinquents? |

3.3.2 Underspecified queries

A user strongly focussed on a particular aspect of a topic may be temporarily oblivious to other aspects of the topic when searching for information. Thus, the user may not specify which particular aspect related to the search keyword(s) she is interested in. Alternatively, she may not be able to think of a precise formulation for her information need on the spur of the moment, and may provide only a broad specification of the topic of interest. In such cases, the user’s information need may remain unclear from the query words, even when these words are not polysemous. Table 7 shows a few possible interpretations of some TREC queries that are of this kind.

Given that such queries are open to multiple interpretations by humans, some level of user interaction or true relevance feedback is likely to be unavoidable in order to obtain satisfactory results from a search engine. If the documents retrieved in response to the initial query turn out to be satisfactory, simple PRF is likely to be beneficial (but, of course, no further QE may be necessary). Otherwise, the best option for the retrieval system may be to present a diversified set of results (as discussed in Section 3.3.1). The user can then provide some feedback by marking document sets or individual documents, or may simply select one of the sets, if appropriate.

Recent research has explored the possibility of obtaining implicit feedback via eye tracking or other neuro physiological signals [Eugster et al., 2015, Gonzlez-Ibez and Shah, 2015]. In a
research setting, this may involve placing potentially intrusive / bothersome sensors, but with progress, non-intrusive means of obtaining feedback are likely to emerge. In such cases, a system may be able to obtain feedback directly from the natural neuro physiological signals emitted by the user (e.g., her facial expressions), without requiring any explicit action on her part.

3.4 Context implicit in queries

Quite often, when a person in a particular situation converts an information need to an actual query, e.g., “national elections”, she may not be consciously aware that the query may have a very different interpretation for someone in a different situation. Thus, the intent of such queries becomes clear only when additional information (e.g., nationality, gender, location, time at which query was submitted, demographic information) about the user is known. We refer to this additional information as context; such queries may be termed implicit-context queries. Bai et al. [2007] further differentiate between context around and context within a query. In their terminology, a user’s domain of interest, her background knowledge and preferences comprise the context around a query. In this section, we use the word context in this sense. In contrast, the context within a query refers to the sense-disambiguating effect of the query words when taken together (as discussed in Section 3.3 under Implicit WSD). For example, this “internal” context determines that the word program in the query Java program is related to the word computer, but this relationship does not hold if the query is TV program. Bai et al. show how both kinds of context information may be integrated into a language modeling approach to IR. They report promising experimental results on the TREC collections.

Table 8 lists examples of such queries taken from the TREC query collection. The persons who create the TREC topics are based in the USA. Thus, the context implicit in Q269 implies that ‘foreign’ means countries other than the USA. The same query would be interpreted differently if it were to occur in the CLEF / FIRE / NTCIR query collections. Since implicit-context queries admit of multiple valid interpretations, they are related to ambiguous queries.

Unlike the creators of TREC topics, the overwhelming majority of Web search engine users are not trained information-seeking professionals. Thus, implicit-context queries are encountered far more frequently by Web search engines. In order to improve retrieval effectiveness for such queries, researchers have focused on personalising search [Jeh and Widom, 2003, Liu et al., 2004], and the use of contextual information during search [Coyle and Smyth, 2007]. While some systems explicitly capture or ask for contextual information [Bharat, 2000, Glover et al., 2001], others guess the context from a user’s actions [Budzik and Hammond, 2000, Finkelstein et al., 2001], or from query logs [Huang et al., 2003].

3.5 Queries involving common nouns or named entities

Generally, user-queries contain a significant proportion of nouns [Xu and Croft, 2000]. These nouns may be either named entities (NEs) — names of persons, places, organisations, etc. — or common nouns.

Queries containing named entities. Many TREC queries contain NEs, e.g., King Hussain (Q450), babe ruth (Q481), baltimore (Q478), Antarctica (Q353), AT&T (Q028), and Smithsonian
Table 9: Examples of queries containing common nouns.

Institute (Q686). These are usually an important (often the most important) component of the query. Thus, it may generally be assumed that an article should contain the NE in order to be relevant. Conversely, the presence of the NE in a document is a reasonable indicator of its relevance. Queries that are focussed on an NE are often relatively easy. If the query is expanded nevertheless, the relative importance of the NE with respect to other query terms should be maintained in the expanded query.

However, if the NE itself is ambiguous (e.g., Michael Jordan could refer to one of several distinct well-known persons), then the issues discussed in Section 3.3 need to be addressed. An additional issue that may arise is the following. A document containing the NE will usually also contain a number of pronouns referring to the NE. Anaphora or coreference resolution — the process of identifying pronominal references or alternative names for a named entity — may therefore be useful.

Queries containing common nouns. It may be much harder to obtain satisfactory results if an important aspect of the query is specified via a common noun. Table 9 shows a few examples of TREC queries belonging to this category.

The words “Companies” and “Writers” are common nouns. It is entirely likely that relevant documents for these queries will contain the names of specific companies or authors, rather than the corresponding common nouns in their surface forms. Thus, during expansion, such queries should be handled differently from queries containing named entities. In some cases, expanding common nouns in the original query using names of specific instances may be useful. For example, the term ‘writers’ may be expanded by adding the names of some popular writers. The expanded query should be appropriately structured, for example, by including the names as a list of disjuncts along with the term writer.

This presupposes access to appropriate ontologies or gazetteer lists that provide, for example, a list of author or company names. If such resources are available, it would be more efficient to use these during indexing, i.e., documents that contain specific author names could be tagged with the terms writer or author.

The system also needs to address the additional issue of selecting which common nouns are to be expanded, since expanding any common noun present in the query may not be a good idea. Interestingly, Buckley [2009] provides an example of a query that belongs to this category even though it contains an NE. TREC topic 398 (Identify documents that discuss the European Conventional Arms Cut as it relates to the dismantling of Europe’s arsenal.) turns out to be problematic because the word ‘Europe’ is too general; relevant documents are likely to discuss moves made by specific European countries towards disarmament.

---

1[https://en.wikipedia.org/wiki/Michael_Jordan_(disambiguation)]
3.6 Queries containing negative terms

Sometimes, users may be able to anticipate the types of irrelevant documents that may be retrieved in response to a given query. In such situations, a user may want to provide a detailed statement of her information need that also explicitly specifies what the user is not looking for. Any keywords that are used to characterise irrelevant information are referred to as negative terms.

Consider the query “terrorist attacks on the US other than 9/11”. Since the user has explicitly specified that she is not looking for information about the 9/11 attack, this term should be counted as a negative term for this query. Likewise, if a user is looking for local restaurants besides those that serve Chinese food, she may submit “restaurants not serving Chinese food” as her query. For this query, Chinese would count as a negative term. This example is more complex, however, since serving and food should probably not be counted as negative terms, even though the negation qualifies these terms syntactically. Table 10 shows some examples of TREC / INEX queries that contain negative terms.

During expansion, queries that contain negative terms need to be handled carefully. If the negating qualifiers are ignored (as they usually are), QE is likely to add terms related to topics that are explicitly designated as irrelevant, leading to a drop in performance. If the negative terms can be identified, then they may simply be removed from the original query. A more aggressive approach would be to include the negative terms in a NOT clause within a structured query. Naturally, for this method to work, negative terms have to be identified with high accuracy.

To the best of our knowledge, approaches that try to address what the user does not want have so far focused only on the initial (verbose) queries. For example, Pramanik et al. [2015] propose a method to automatically identify negative terms in verbose queries and to remove them before initial retrieval. This method is reported to yield improvements across a number of collections and various retrieval models. We expect that these improvements will also lead to post-QE improvements.

3.7 Multi-aspect queries

A multi-aspect query is one that seeks information about a particular aspect of a broader topic.

Multi-aspect queries are best understood via examples. Consider the query “Terrorist attacks on Amarnath pilgrims.” One could regard “Amarnath pilgrims” as the primary topic of the query [Mitra et al., 1998; Buckley, 2009].

| Query # | Query title | Narrative |
|---------|-------------|-----------|
| TREC Q124 | Alternatives to Traditional Cancer Therapies | ... any attempt to experiment with or demonstrate the efficacy of any non-chemical, non-surgical, or non-radiological approach to preventing or curing cancer ... |
| INEX Q419 | film starring +"steven seagal" | ... films played by Steven Seagal, not produced by him. |

Table 10: Examples of queries containing negative terms / aspects.
query. There are various sub-topics of this general topic: travel routes taken by the pilgrims, places for pilgrims to stay along the way, etc. In this query, the user is interested in one specific sub-topic related to Amarnath pilgrims.

TREC query 203, on the economic impact of recycling tires, is a similar example. The broad topic of this query is recycling, but the user is only interested in the recycling of tires (rather than other material), and more specifically in the economic impact thereof (rather than, say, the technology involved). Table 11 lists a few more examples of multi-aspect queries from the TREC query set.

Sometimes, a user may designate multiple sub-topics of a topic as interesting. For a user who is interested in “causes and effects of railway accidents”, documents exclusively discussing either the causes or the effects of a railway accident are generally regarded as relevant. Such queries that are “disjunctive” in a sense (but possibly conjunctive in form) have a broader scope than the examples discussed above, and are expected to be easier to handle. Multi-aspect queries are usually hard when the multiple aspects are combined in a conjunctive sense. Buckley [2009] contains a detailed analysis of why automatic IR systems frequently find multi-aspect queries hard.

Quite often, AQE methods add terms that are mostly related to the general topic of the original query (e.g., recycling for TREC Q203 discussed above). This overemphasises one aspect of the query at the expense of the others, and usually leads to query drift. Ideally, during expansion, multi-aspect queries should be expanded in a balanced way, i.e., using terms related to all (or most) of the multiple aspects. This requires systems to be able to (i) recognise the various aspects of a query, and (ii) to identify which aspect(s) of the query a candidate expansion term is related to. Mitra et al. [1998] studied some preliminary methods (both manual and automatic) that try to prevent query drift by ensuring that the query is expanded in a balanced way. AbraQ, an approach described by Crabtree et al. [2007], attempts balanced query expansion in a Web search setting by first identifying the different aspects of the query, identifying which aspects are under-represented in the result set of the original query, and finally, identifying expansion terms that would strengthen that particular aspect of the query.

Zhao and Callan [2012] also identify “problematic” query terms — terms that are probably not present in relevant documents — on the basis of the term’s idf, or by the predicted probability of that term occurring in the relevant documents. These query terms are selectively expanded. The final expanded query is a structured query in Conjunctive Normal Form (CNF), with each conjunct expected to correspond to a query term (or aspect) and its synonyms. The authors

| TREC Query # | Query title | Aspects |
|--------------|-------------|---------|
| Q100 | Controlling the Transfer of High Technology | 1. High Technology 2. Transfer 3. Controlling |
| Q294 | Animal husbandry for exotic animals | 1. Animal husbandry 2. exotic animals |
| Q299 | Impact on local economies of military downsizing | 1. military downsizing 2. local economies 3. Impact |
| Q321 | Women in Parliaments | 1. Women 2. Parliaments |

Table 11: Examples of queries containing multiple aspects.
argue that the use of CNF ensures balanced expansion, minimises topic drift, and yields stable performance across different levels of expansion.

Wu et al. [2012] propose a different approach within a true relevance feedback framework that may also be regarded as being targeted towards balanced expansion. This approach attempts to diversify the set of documents judged by a user. Instead of simply letting the user judge the top-ranked results returned in response to the initial query, the system partitions the initially retrieved documents into sub-lists, and reranks the documents on the basis of the query term patterns that occur in them (i.e., whether a document contains only a single term, multiple terms occurring as a phrase, or in close proximity, etc.). The documents are then presented iteratively to the user for judgment.

### 3.8 “High-level” query

Some queries, such as those shown in Table 12, contain terms that correspond to abstract or “high-level” concepts. These terms may not themselves be present in relevant documents; instead, other more concrete terms may be used to convey specific instances of the same concept. If one or more such abstract terms form an important component of an information need, we refer to the corresponding query as a high-level query.

‘Impact’ and ‘effect’ are typical examples of such high-level terms. Consider the query “effect of tsunami”, for example. Here ‘effect’ is a high level term, and refers to anything that happened as a result of a tsunami. A relevant document may not contain the term ‘effect’; instead, it may describe the effect of a tsunami using words such as ‘death toll’, ‘property damage’, etc.

TREC query 389 (“illegal technology transfer”) is another example. The description field of the query asks: “What specific entities have been accused of illegal technology transfer such as: selling their products, formulas, etc. directly or indirectly to foreign entities for other than peaceful purposes?” ‘Technology transfer’ is thus an abstract concept. Relevant documents may or may not contain this term. Instead, they may contain terms like ‘sell’, ‘license’, that describe concrete methods of technology transfer.

**Difference with queries involving common nouns (Section 3.5).** There is a subtle difference between high-level queries and queries involving common nouns (discussed in Section 3.5). Consider an example from Table 9: *writers*. The ‘instantiation’ of writers, i.e., the set of persons who are writers, is not dependent on the query context. In contrast, an abstract term may be instantiated via different sets of keywords, depending on the subject or domain of the query. The ‘effects’ or ‘impact’ of a natural disaster, a foreign tour by a head of state, or of substance
abuse are likely to be described using different words. Thus, finding ‘bag-of-word’ equivalents of such concepts, being context-sensitive, is more difficult. As a result, SEs often fail to retrieve an adequate number of relevant documents in response to high-level queries. For the same reason, correctly automatically expanding such queries is also challenging. Roussinov [2010] shows that external corpora may be mined to obtain words or word sequences (conditionally) related to high-level query terms. For example, in TREC query 353, the notion of exploration may be indicated by the word station, provided it occurs along with the word Antarctica, but not as a part of a phrase such as train station.

3.9 Recall-oriented queries

In certain situations, recall is of paramount importance to the user. Queries issued by a user in such situations can be termed recall-oriented. The TREC million query track [Allan et al., 2007] defines recall-oriented queries as “looking for deeper, more open-ended information whereas precision-oriented queries are looking for a small, well contained set of facts”. Some typical recall-oriented search tasks are:

- E-discovery: searching for documents required for disclosure in a legal case [Oard et al., 2010, Oard and Webber, 2013].
- Prior-art patent search: looking for existing patents which might invalidate a new patent application.
- Evidence-based medicine: finding all prior evidence on treatments for a medical case.

For these tasks, having to examine several irrelevant documents may be an acceptable overhead, but the penalty for missing a relevant document is likely to be high.

The TREC legal track models a recall-oriented task. Query 100 from this track reads: “Submit all documents representing or referencing a formal statement by a CEO of a tobacco company describing a company merger or acquisition policy or practice”. Note that the query explicitly requires all relevant documents to be retrieved. This is in contrast to casual, ad hoc searches, in which users are generally satisfied by a small number of relevant documents retrieved at the top ranks.

Table 13 shows that recall generally increases with the number of terms added to a query during QE. Thus, for recall-oriented queries, massive query expansion, i.e., expansion by adding a very large number of potentially useful terms that occur in at least one relevant document, may be a good idea. However, the risk of query drift significantly increases if massive expansion is based on PRF. Relevance feedback involving some user interaction may be necessary to ensure high recall without a concomitant loss in precision. Ghosh and Parui [2015] have recently proposed a method that uses the Cluster Hypothesis to effectively leverage only a modest amount of user interaction for high recall.

3.10 Domain specific queries

Queries which are related to and need information from one specific domain (e.g., sports, medicine, law) are called domain specific queries. Earlier work on classifying queries according to their domain has been discussed in Section 2.1. Over the years, TREC has offered a number of tasks that address IR from specific domains / genres. Table 14 provides a non-exhaustive list of some of these tasks.

Domain-specific queries constitute a special case of Vertical search, where the system caters to users interested in a particular type of online content. Vertical searches may focus not only on...

\[^{14}\text{http://www.isical.ac.in/~fire/2011/slides/fire.2011.robertson.stephen.pdf}\]
\[^{15}\text{https://en.wikipedia.org/wiki/Vertical_search}\]
| #Term | #rel-ret(among top 1000) | recall@1000 | MAP |
|-------|--------------------------|-------------|-----|
| 10    | 8273                     | 0.6686      | 0.2452 |
| 20    | 8442                     | 0.6784      | 0.2525 |
| 30    | 8530                     | 0.6851      | 0.2561 |
| 40    | 8556                     | 0.6891      | 0.2574 |
| 50    | 8551                     | 0.6901      | 0.2586 |
| 60    | 8562                     | 0.6906      | 0.2586 |
| 70    | 8587                     | 0.6922      | 0.2595 |
| 80    | 8589                     | 0.6927      | 0.2601 |
| 90    | 8602                     | 0.6938      | 0.2605 |
| 100   | 8605                     | 0.6943      | 0.2611 |

Table 13: Effect of increasing the degree of expansion on recall on the TREC678 collection (expansion method used: KLD, no. of top documents: 40).

| Track name       | Years | Domain                                              |
|------------------|-------|-----------------------------------------------------|
| Legal            |       |                                                     |
| Enterprise search|       | Searching an organisation’s data                    |
| Genomics         |       | Genomics data (broadly construed to include not just gene sequences but also supporting documentation such as research papers, lab reports, etc. |
| Chemical         |       | Information retrieval and extraction tools for chemical literature |
| Medical records  |       | Free-text fields of electronic medical records       |

Table 14: TREC tracks that focus on domain-specific IR.

A particular domain or topic, but also on a specific media type or genre of content, e.g., image or video search, shopping, travel, and scholarly literature.

Expansion strategy. For some domains, it should be possible to leverage domain-specific lexical resources for expansion. For example, MeSH or the UMLS metathesaurus may be used to expand queries in the biomedical domain. Hersh et al. [2000] have reported on the effectiveness of using the UMLS metathesaurus for QE. Similarly, Lu et al. [2009] have studied expansion of PubMed queries using MeSH. Naturally, in order to utilise such domain-specific ontologies, a system should be able to identify the target domain of user queries with reasonable accuracy. On the other hand, if the user explicitly indicates the domain of the query, this not only eliminates the query-classification step, but should also help to reduce any ambiguity that might be present. In recent work, Macias-Galindo et al. [2015] have confirmed that the domain of interest is important when quantifying the semantically relatedness between words. Even though their experiments were not directly related to QE, their findings are expected to be applicable when estimating the semantic relation between the query and candidate expansion terms.

3.11 Short answer type queries

Some queries need very specific and ‘to the point’ answers that comprise a few words, a single sentence, or a short passage. In such cases, the user does not want to read a full document, or a long passage to find the answer. Most queries starting with ‘what’, ‘who’, ‘where’, ‘when’,
‘which’, ‘whom’, ‘whose’, ‘why’ etc. fall in this category. There are few examples of such queries in the TREC adhoc dataset, but the query sets for the Question-Answering (QA) tasks at TREC, CLEF and NTCIR consist of these types of queries.

Systems that effectively address such queries usually have the following architecture [Prager, 2006]. The question is first analysed to determine the answer type, and to generate an appropriate keyword query. The keyword query is used to retrieve a set of passages (or documents) from a corpus. The retrieved passages are analysed to generate a list of candidate answers. The candidate answers are further processed to generate the final ranked list of answers.

Query expansion can, and often does, play a role in retrieving passages or documents in response to the keyword query. In one of the best-known QA systems [Pasca and Harabagiu, 2001, Moldovan et al., 2003], some of the question words are selected as keywords (using mainly part of speech information). The original question is parsed to determine dependencies between the question words, which are in turn used to order the list of selected keywords. These keywords are also spell-checked; spelling variants are added to the query if necessary. The most important of these keywords are used to retrieve documents using the Boolean model. From these documents, the system extracts paragraphs or smaller text passages containing all keywords in close proximity of one another. If too many paragraphs are retrieved, the query is expanded by including additional terms from the list of keywords; if too few paragraphs are retrieved, some of the keywords from the initial query are dropped. The system also employs QE in a more traditional way by using WordNet to expand the query keywords with morphological, lexical and semantic alternatives.

3.12 Queries that need special handling during query processing

Query processing generally includes some (or all) of the following steps: stopword removal, stemming, case normalisation, treatment of acronyms and numbers, handling spelling errors, etc. In this section, we consider queries that need special handling during query processing, i.e., queries for which the general (or “standard”) query processing methods would result in a loss of some important information, which in turn would lead to poor retrieval effectiveness. Note that this special processing must be done on the initial query; the question of whether (or how) to expand the query arises later. Indeed, without this special processing, initial retrieval effectiveness may be so poor that any subsequent expansion of the query would be pointless.

- **Stopword removal.** Articles, conjunctions, prepositions and other frequently occurring words are discarded as stopwords because they usually have a grammatical function, and are not indicative of the subject matter of documents and queries. The words before and after are two examples of such words that are included in the default stopword list used by TERRIER. However, in a query like “increased security measures after 9/11”, the word “after” is an important qualifier. Discarding it as a stopword during indexing of documents and queries may cause problems.

- **Case normalisation.** Many IR systems reduce all alphabets to their lowercase forms during indexing. Since proper nouns can be identified by a starting capital letter, this case normalisation may result in loss of information in some cases. In TREC query 409 (legal, Pan Am, 103), the word Am is not actually used as a stopword; however, through a combination of case normalisation and stopword removal, many systems would incorrectly discard this word from the query. This problem would also arise if the acronym U.S. were written as US.

The Smart system ran into a related problem during the initial years of TREC: because the ampersand in query 028 (AT&T’s Technical Efforts) was treated as a word delimiter, AT&T’s was tokenised as at, &, t, and ’s after case normalisation, and all four tokens were discarded, resulting in very poor performance for this query. This problem might...
also occur with TREC query 391 (*R&D drug prices*), with *R&D* being tokenised as *r* and *d*.

- **Identification of numbers.** A user query may contain numbers denoting a year, a flight number or something similar which is an integral part of the information need. Simply ignoring numbers during indexing of either documents or queries (such as TREC query 409 discussed above) may have a significant detrimental effect.

- **Stemming.** Stemming is used to conflate morphological variants of a word to a canonical form, so that a keyword in a query matches a variant occurring in a document. Whether a query word should be stemmed or not often depends on the query, and more specifically the sense of the query word. For example, in a query about Steve Jobs, the word ‘Jobs’ should not be stemmed to ‘job’. Similarly, the word ‘apples’ occurring in a document about the fruit should not be stemmed to match the word ‘Apple’ in a query about Apple’s marketing strategy for the iPhone. [Paik et al., 2013] show that a query-specific stemming approach is significantly more effective than applying a generic stemmer uniformly to all queries and documents in a collection. To achieve this effect, documents should not be stemmed at the time of indexing. Instead, a given query should be expanded by adding to it only the desirable variants of query keywords.

- **Indexing phrases.** The question of whether to use phrases — multiple words that occur contiguously or in close proximity and constitute a single semantic unit, e.g., blood cancer, machine learning — during indexing and retrieval has been investigated in a number of studies [Fagan, 1987, Mitra et al., 1997]. This question is also tied to the issue of whether to use phrases during QE. The use of phrases has been found to generally improve performance, though its effect is not always significant. [Song et al., 2006] show that keyphrases extracted from retrieved documents may be useful as expansion terms. Their keyphrase extraction algorithm makes use of the occurrences of stopwords in the documents. Thus, in order to use their method in a practical SE, documents and queries need special handling during indexing and retrieval.

### Multi-lingual query

Multilingual queries, i.e., queries that make use of words from more than one language (say, \(L_1, L_2, \ldots, L_k\)), are a particular class of queries that need special handling. Such queries [Mustafa et al., 2011] are common in multilingual countries or communities like India or the EU. A number of factors lead to the creation of multilingual queries.

- The amount and variety of native language content on the Web is still rather low for many languages, e.g., Assamese or Punjabi. An Assamese user may be able to read English fluently, and is thus likely to know the most important English keywords related to her information need. At the same time, she may be unable to find appropriate English words to completely formulate her query in English. For example, consider a user who is looking for the differences between interpreters and assemblers. For such a user, it would be natural to submit a query that mixes the English words *interpreters* and *assemblers* with the Assamese equivalents of *difference* and the remaining words.

- In a country like India, where the language used at work is often English, users may not be familiar with the local equivalents of all technical terms. If such a user is specifically interested in a technical or official document in her native language, her natural tendency would be to search using a mix of English and native language words.

- Some English terms are very commonly used in non-English-speaking regions. For example, in Bengali documents, the term ‘recipe’ is more likely to be used than the Bengali
equivalent (randhanpronali). In addition, documents may use either the Bengali transliteration of ‘recipe’, or the original Roman form of the word. An experienced user who is aware of this may include all three terms in her query for better recall.

When processing a multilingual query, a system needs to address the following problems.

- **Source language identification.** If words from multiple languages are present in the query, then their respective languages have to be identified. This is trivial if the languages use distinctive scripts, but if any of the languages involved shares its script with other languages, the language identification problem becomes harder. If different inverted indices are maintained for different languages, the system also needs to determine which target collections need to be searched for a given multilingual query.

- **Transliteration.** For a very long time, native language keyboards were a rarity for many languages. Users of these languages were habituated to using the Roman script when writing in their language. Such habits die hard, and many users continue to prefer using the Roman script to write in their language. In order to retrieve documents in the original language, the system needs to first back-transliterate words from Roman to the native language.

Moreover, if a query entered by such a user is multilingual, word-level language identification may be harder, since the Roman script is used for all words. The problem is compounded further if, after transliteration, words in the user’s native language match valid English words. For example, More is both a valid English word, and a reasonably common surname in Marathi. Similarly, Shulk is a fictional character and the main protagonist in a popular video game; it also means tax in Hindi and other Indian languages.

Some of these problems are being studied within the “Search in the Transliterated Domain” track at FIRE. This track involves two subtasks: (i) given a multilingual query, label each word with its language; and (ii) reverse-transliterate non-English words written in Roman script into their native script. If these problems are not properly addressed, QE may hurt performance. If an important expansion term in one language happens to be a valid word in another language, the system also needs to carefully consider the net benefit of including such an ambiguous term in the expanded query when retrieving documents from a multilingual collection.

**Noisy queries**

Finally, we consider queries that need special handling because of the presence of errors or noise. Such noise can be introduced because of spelling errors committed by the user, or because queries are submitted via a noise-inducing interface, e.g. spoken queries, querying via mobile messaging, and queries written by hand using a stylus.

Since queries in most test collections are methodically created by experienced or professional users of IR systems, such queries are usually free from noise. TREC query 464 — nativityscenes — is one of the rare TREC queries that contain an error. However, query logs of practical search engines are likely to have large numbers of such examples.

Noisy queries also need special handling, usually spelling correction. A fair amount of work has recently been done on spelling correction in queries [Gao et al, 2010, Duan and Hsu, 2011, Li et al, 2012], and a large number of patents exist for such techniques. Some of these methods are employed in many practical Web search engines that are often able to suggest or even automatically provide corrections for such noisy queries.
4 Conclusions and future work

Query expansion is a standard technique for addressing the well-known vocabulary mismatch problem faced by IR systems. Over the years, a number of effective QE techniques have been proposed. However, the effect of different QE techniques for individual queries can vary greatly. Our long-term goal is to improve overall performance by applying QE techniques tailored to a given query, rather than applying the same general QE method to all queries. To this end, we have proposed a taxonomy of query classes. Not all proposed query categories are new. However, we have specifically considered query categorisation from a QE perspective.

We have discussed the properties of each query class with examples. We have also proposed some QE strategies that might be effective for each query category. We believe that there is significant scope for future work in a careful investigation of the most effective QE techniques for each query class.

Our next step will be to come up with more precise formulations of QE techniques for the various categories and to test these proposed techniques using standard datasets. While for many query categories, such testing can be done using TREC datasets, a few categories pertain specifically to Web queries.

An additional challenge will be to automatically detect the type of a given query. This is likely to be straightforward for some query types, but we will need to systematically study automatic query classification approaches in future work.

To conclude, we believe that in the recent future, as the Web continues to grow, and search becomes a more and more frequent activity, IR systems will need customised methods for individual queries and users. The work described in this report is an initial step in this direction.

A TREC: Text REtrieval Conference

Experimental IR, like any other experimental discipline, depends heavily on the existence of standardised benchmark datasets, or test collections. A test collection in IR is a collection of documents along with a set of test queries. The set of relevant articles for each query is also known. To measure the effectiveness of a technique, documents are retrieved using that technique for each test query in the collection. Using the relevance information for the queries, the average precision value can be computed for each query. The mean average precision for the entire query set is then calculated. Different techniques can be compared using the average precision figures they yield on a given test collection. Obviously, techniques that perform well across a wide variety of test collections can be regarded as robust.

For our experiments, we use parts of the TREC collection [Voorhees and Harman, 2005]. TREC (Text REtrieval Conference) is an ARPA and NIST co-sponsored effort that brings together information retrieval researchers from around the world to discuss their systems and to evaluate them on a common test platform. The documents and queries in this collection are described below.

A.1 Documents

The TREC document collection consists of a large number of full-text documents drawn from a variety of sources. The documents are stored on CD-ROMS, called the TREC disks. The disks are numbered, and a combination of several disks can be used to form a text collection for experimentation. Some statistics about the data on various disks is listed in Table A.1 (adapted from [Voorhees and Harman, 1998]). The sources for the data are:

- Disk 1
A.2 Queries

The queries are natural-language queries supplied by users. Most queries consist of 3 parts:

- **Title**: a few keywords (usually 2–3) related to the users query,
- **Desc** (description): a short, natural-language statement of the user’s information need,
- **Narr** (narrative): a more detailed specification of what makes a document relevant for the corresponding topic.

The queries have varied widely from year to year. At the first two conferences, TREC–1 and TREC–2, the queries were quite long and represented long-standing user information needs. Reflecting a trend towards realistic user queries, the queries for TREC–3 were considerably shorter and the queries for TREC–4 were just a sentence or two. Some characteristics of the query sets are shown in Table 16 (the training queries were provided to help train systems for TREC–1).

Users also provide relevance judgments (i.e. they specify which documents are useful and which are non-relevant) for the documents in the collection. These judgements enable us to measure the retrieval effectiveness (using average precision figures) of our algorithms.
### Table 15: TREC Document Statistics

| Source | Size (Mb) | Number of articles | Median number of terms/article | Average number of terms/article |
|--------|-----------|--------------------|--------------------------------|-------------------------------|
| Disk 1 |           |                    |                                |                               |
| WSJ    | 270       | 98,732             | 182                            | 329                           |
| AP     | 259       | 84,678             | 353                            | 375                           |
| ZIFF   | 245       | 75,180             | 181                            | 412                           |
| FR     | 262       | 25,960             | 313                            | 1017                          |
| DOE    | 186       | 226,087            | 82                             | 89                            |
| Disk 2 |           |                    |                                |                               |
| WSJ    | 247       | 74,520             | 218                            | 377                           |
| AP     | 241       | 79,919             | 346                            | 370                           |
| ZIFF   | 178       | 56,920             | 167                            | 394                           |
| FR     | 211       | 19,860             | 315                            | 1073                          |
| Disk 3 |           |                    |                                |                               |
| SJMN   | 290       | 90,257             | 279                            | 337                           |
| AP     | 242       | 78,321             | 358                            | 379                           |
| ZIFF   | 349       | 161,021            | 119                            | 263                           |
| PAT    | 245       | 6,711              | 2896                           | 3543                          |
| Disk 4 |           |                    |                                |                               |
| FT     | 564       | 210,158            | 316                            | 413                           |
| FR94   | 395       | 55,630             | 588                            | 645                           |
| OR     | 235       | 27,922             | 288                            | 1374                          |
| Disk 5 |           |                    |                                |                               |
| FBIS   | 470       | 130,471            | 322                            | 544                           |
| LAT    | 475       | 131,896            | 351                            | 527                           |

### Table 16: Query Statistics

| Query Id. | # of Queries | Min | Max | Mean |
|-----------|--------------|-----|-----|------|
| TREC-1    | 50           | 44  | 250 | 107.4|
| TREC-2    | 50           | 54  | 231 | 130.8|
| TREC-3    | 50           | 49  | 180 | 103.4|
| TREC-4    | 50           | 8   | 33  | 16.3 |
| TREC-5    | 50           | 29  | 213 | 82.7 |
| TREC-6    | 50           | 47  | 156 | 88.4 |
| TREC-7    | 50           | 31  | 114 | 57.6 |
| TREC-8    | 50           | 23  | 98  | 51.8 |
In recent years, the TREC collection has emerged as a standard test collection for experimental IR. At TREC–6, the sixth in this series of conferences, thirty-eight groups including participants from nine different countries and ten companies were represented. Given the participation by such a wide variety of IR researchers, a large and heterogeneous collection of full-text documents, a sizeable number of user queries, and a set of relevance judgments, TREC has rightfully become a standard test environment for current information retrieval research.

### Bibliography

James Allan, Ben Carterette, Blagovest Dachev, Javed A. Aslam, Virgil Pavlu, and Evangelos Kanoulas. Million query track 2007 overview. In *TREC*, 2007.

Giambattista Amati, Claudio Carpineto, and Giovanni Romano. Query difficulty, robustness, and selective application of query expansion. In *ECIR*, pages 127–137, 2004.

Azin Ashkan and Charles L. A. Clarke. Characterizing commercial intent. In *CIKM*, pages 67–76, 2009.

Ricardo A. Baeza-Yates, Liliana Calderón-Beñavides, and Cristina N. González-Caro. The intention behind web queries. In Fabio Crestani, Paolo Ferragina, and Mark Sanderson, editors, *SPIRE*, volume 4209 of *Lecture Notes in Computer Science*, pages 98–109. Springer, 2006. ISBN 3-540-45774-7. URL [http://dblp.uni-trier.de/db/conf/spire/spire2006.html#Baeza-YatesCG06](http://dblp.uni-trier.de/db/conf/spire/spire2006.html#Baeza-YatesCG06).

Jing Bai, Jian-Yun Nie, Guihong Cao, and Huges Bouchard. Using query contexts in information retrieval. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 15–22. ACM, 2007.

Steven M. Beitzel, Eric C. Jensen, Abdur Chowdhury, David Grossman, and Ophir Frieder. Hourly analysis of a very large topically categorized web query log. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR ’04, pages 321–328, New York, NY, USA, 2004. ACM. ISBN 1-58113-881-4. doi: [http://doi.acm.org/10.1145/1008992.1009048](http://doi.acm.org/10.1145/1008992.1009048). URL [http://doi.acm.org/10.1145/1008992.1009048](http://doi.acm.org/10.1145/1008992.1009048).

Steven M. Beitzel, Eric C. Jensen, Ophir Frieder, David D. Lewis, Abdur Chowdhury, and Aleksander Kolcz. Improving automatic query classification via semi-supervised learning. In *Proceedings of the Fifth IEEE International Conference on Data Mining*, ICDM ’05, pages 42–49, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2278-5. doi: [http://dx.doi.org/10.1109/ICDM.2005.80](http://dx.doi.org/10.1109/ICDM.2005.80). URL [http://dx.doi.org/10.1109/ICDM.2005.80](http://dx.doi.org/10.1109/ICDM.2005.80).

Michael Bendersky and W Bruce Croft. Discovering key concepts in verbose queries. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 491–498. ACM, 2008.

Michael Bendersky, Donald Metzler, and W Bruce Croft. Parameterized concept weighting in verbose queries. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 605–614. ACM, 2011.

Krishna Bharat. Searchpad: explicit capture of search context to support web search. *Computer Networks*, 33(1):493–501, 2000.

J. Bhogal, A. Macfarlane, and P. Smith. A review of ontology based query expansion. *Inf. Process. Manage.*, 43(4):866–886, July 2007. ISSN 0306-4573. doi: [10.1016/j.ipm.2006.09.003](http://dx.doi.org/10.1016/j.ipm.2006.09.003). URL [http://dx.doi.org/10.1016/j.ipm.2006.09.003](http://dx.doi.org/10.1016/j.ipm.2006.09.003).

Andrei Z. Broder. A taxonomy of web search. *SIGIR Forum*, 36(2):3–10, 2002.
Chris Buckley. Why current IR engines fail. Inf. Retr., 12:652–665, December 2009. ISSN 1386-4564. doi: 10.1007/s10791-009-9103-2. URL http://dl.acm.org/citation.cfm?id=1644394.1644417

Jay Budzik and Kristian J Hammond. User interactions with everyday applications as context for just-in-time information access. In Proceedings of the 5th international conference on intelligent user interfaces, pages 44–51. ACM, 2000.

Bin Cao, Jian-Tao Sun, Evan Wei Xiang, Derek Hao Hu, Qiang Yang, and Zheng Chen. PQC: personalized query classification. In CIKM, pages 1217–1226, 2009a.

Huanhuan Cao, Derek Hao Hu, Dou Shen, Daxin Jiang, Jian-Tao Sun, Enzhong Chen, and Qiang Yang. Context-aware query classification. In SIGIR, pages 3–10, 2009b.

David Carmel and Oren Kurland. Query performance prediction for IR. In Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’12, pages 1196–1197, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1472-5. doi: 10.1145/2348283.2348540. URL http://doi.acm.org/10.1145/2348283.2348540

David Carmel and Elad Yom-Tov. Estimating the query difficulty for information retrieval. Synthesis Lectures on Information Concepts, Retrieval, and Services, 2(1):1–89, January 2010. ISSN 1947-945X, 1947-9468. doi: 10.2200/S00235ED1V01Y201004ICR015.

Claudio Carpineto and Giovanni Romano. A survey of automatic query expansion in information retrieval. ACM Computing Surveys, 44(1):article 1, January 2012.

Claudio Carpineto, Renato de Mori, Giovanni Romano, and Brigitte Bigi. An information-theoretic approach to automatic query expansion. ACM Trans. Inf. Syst., 19(1):1–27, 2001.

Ben Carterette, Virgil Pavlu, Hui Fang, and Evangelos Kanoulas. Million query track 2009 overview. In TREC, 2009.

Francesco Colace, Massimo De Santo, Luca Greco, and Paolo Napoletano. Improving relevance feedback-based query expansion by the use of a weighted word pairs approach. Journal of the Association for Information Science and Technology, 2015.

Maurice Coyle and Barry Smyth. Information Recovery and Discovery in Collaborative Web Search. In Advances in Information Retrieval (ECIR 2007), volume 4425 of LNCS, pages 356–367, 2007. ISBN 0501182047. doi: 10.1007/978-3-540-71496-5_33. URL http://dx.doi.org/10.1007/978-3-540-71496-5_33

Daniel Wayne Crabtree, Peter Andreea, and Xiaoying Gao. Exploiting underrepresented query aspects for automatic query expansion. In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 191–200. ACM, 2007.

Steve Cronen-Townsend, Yun Zhou, and W. Bruce Croft. Predicting query performance. In Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’02, pages 299–306, New York, NY, USA, 2002. ACM. ISBN 1-58113-561-0. doi: 10.1145/564376.564429. URL http://doi.acm.org/10.1145/564376.564429

Steve Cronen-Townsend, Yun Zhou, and W. Bruce Croft. A framework for selective query expansion. In Proceedings of the thirteenth ACM international conference on Information and knowledge management, CIKM ’04, pages 236–237, New York, NY, USA, 2004. ACM. ISBN 1-58113-874-1.

Huizhong Duan and Bo-June Paul Hsu. Online spelling correction for query completion. In Proceedings of the 20th international conference on World wide web, pages 117–126. ACM, 2011.
Manuel J. A. Eugster, Tuukka Ruotsalo, Michiel M. Spap, Ilkka Kosunen, Oswald Barral, Niklas Ravaja, Giulio Jacucci, and Samuel Kaski. Predicting Relevance of Text from Neuro-Physiology. In Proceedings of the Neuro-Physiological Methods in IR Research - a SIGIR Workshop. ACM, 2015. URL https://sites.google.com/site/neuroir2015/papers

Joel L Fagan. Experiments in automatic phrase indexing for document retrieval: a comparison of syntactic and non-syntactic methods. Technical report, Cornell University, 1987.

Hui Fang. A re-examination of query expansion using lexical resources. In Proceedings of ACL-08: HLT, pages 139–147, Columbus, Ohio, June 2008. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P/P08/P08-1017

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. Placing search in context: The concept revisited. In Proceedings of the 10th international conference on World Wide Web, pages 406–414. ACM, 2001.

Evgeniy Gabrilovich, Andrei Broder, Marcus Fontoura, Amruta Joshi, Vanja Josifovski, Lance Riedel, and Tong Zhang. Classifying search queries using the web as a source of knowledge. ACM Trans. Web, 3(2):5:1–5:28, April 2009. ISSN 1559-1131. doi: 10.1145/1513876.1513877. URL http://doi.acm.org/10.1145/1513876.1513877

Jianfeng Gao, Xiaolong Li, Daniel Micol, Chris Quirk, and Xu Sun. A large scale ranker-based system for search query spelling correction. In Proceedings of the 23rd International Conference on Computational Linguistics, pages 358–366. Association for Computational Linguistics, 2010.

Susan Gauch, Jianying Wang, and Satya Mahesh Rachakonda. A corpus analysis approach for automatic query expansion and its extension to multiple databases. ACM Trans. Inf. Syst., 17(3):250–269, July 1999. ISSN 1046-8188. doi: 10.1145/314516.314519. URL http://doi.acm.org/10.1145/314516.314519

Kripabandhu Ghosh and Swapan Kumar Parui. Clustered semi-supervised relevance feedback. In CIKM, 2015.

Eric J Glover, Steve Lawrence, Michael D Gordon, William P Birmingham, and C Lee Giles. Web search—your way. Communications of the ACM, 44(12):97–102, 2001.

Roberto Gonzalez-Ibez and Chirag Shah. Affective Signals as Implicit Indicators of Information Relevancy and Information Processing Strategies. In Proceedings of the Neuro-Physiological Methods in IR Research - a SIGIR Workshop. ACM, 2015. URL https://sites.google.com/site/neuroir2015/papers

Manish Gupta and Michael Bendersky. Information Retrieval with Verbose Queries. Foundations and Trends in Information Retrieval, 9(3-4):209–354, 2015a. ISSN 1554-0669. doi: 10.1561/1500000050. URL http://www.nowpublishers.com/article/Details/INR-050

Manish Gupta and Michael Bendersky. Information retrieval with verbose queries. In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’15, pages 1121–1124, New York, NY, USA, 2015b. ACM. ISBN 978-1-4503-3621-5. doi: 10.1145/2766462.2767877. URL http://doi.acm.org/10.1145/2766462.2767877

D. Harman. Towards interactive query expansion. In Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR ’88, pages 321–331, New York, NY, USA, 1988. ACM. ISBN 2-7061-0309-4. doi: http://doi.acm.org/10.1145/62437.62469. URL http://doi.acm.org/10.1145/62437.62469
Donna Harman and Chris Buckley. Overview of the Reliable Information Access Workshop. *Information Retrieval*, 12(6):615–641, July 2009. ISSN 1386-4564. doi: 10.1007/s10791-009-9101-4. URL [http://link.springer.com/10.1007/s10791-009-9101-4](http://link.springer.com/10.1007/s10791-009-9101-4)

Claudia Hauff, Djoerd Hiemstra, and Franciska de Jong. A survey of pre-retrieval query performance predictors. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 1419–1420. ACM, 2008. URL [http://dl.acm.org/citation.cfm?id=1458311](http://dl.acm.org/citation.cfm?id=1458311)

Ben He and Iadh Ounis. Query performance prediction. *Inf. Syst.*, 31(7):585–594, November 2006. ISSN 0306-4379. doi: 10.1016/j.is.2005.11.003. URL [http://dx.doi.org/10.1016/j.is.2005.11.003](http://dx.doi.org/10.1016/j.is.2005.11.003)

William Hersh, Susan Price, and Larry Donohoe. Assessing thesaurus-based query expansion using the umls metathesaurus. In *In Proc. of the 2000 American Medical Informatics Association (AMIA) Symposium*, pages 344–348, 2000.

Chien-Kang Huang, Lee-Feng Chien, and Yen-Jen Oyang. Relevant term suggestion in interactive web search based on contextual information in query session logs. *Journal of the American Society for Information Science and Technology*, 54(7):638–649, 2003.

Glen Jeh and Jennifer Widom. Scaling personalized web search. In *Proceedings of the 12th international conference on World Wide Web*, pages 271–279. ACM, 2003.

Y. Jing and W.B. Croft. An association thesaurus for information retrieval. In *Proceedings of the Intelligent Multimedia Information Retrieval Systems (RIAO ’94, New York, NY)*, 1994, pages 146–160, 1994.

Oren Kurland, Anna Shtok, Shay Hummel, Fiana Raiber, David Carmel, and Ofri Rom. Back to the roots: A probabilistic framework for query-performance prediction. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM ’12, pages 823–832, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1156-4. doi: 10.1145/2396761.2396866. URL [http://doi.acm.org/10.1145/2396761.2396866](http://doi.acm.org/10.1145/2396761.2396866)

Matthew Lease. An improved markov random field model for supporting verbose queries. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 476–483. ACM, 2009.

Yanen Li, Huizhong Duan, and ChengXiang Zhai. A generalized hidden markov model with discriminative training for query spelling correction. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 611–620. ACM, 2012.

Yinghao Li, Wing Pong Robert Luk, Kei Shiu Edward Ho, and Fu Lai Korris Chung. Improving weak ad-hoc queries using Wikipedia as external corpus. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR ’07, pages 797–798, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-597-7. doi: 10.1145/1277741.1277914. URL [http://doi.acm.org/10.1145/1277741.1277914](http://doi.acm.org/10.1145/1277741.1277914)

Fang Liu, Clement Yu, and Weiyi Meng. Personalized web search for improving retrieval effectiveness. *Knowledge and Data Engineering, IEEE transactions on*, 16(1):28–40, 2004.

Zhiyong Lu, Won Kim, and W. John Wilbur. Evaluation of query expansion using mesh in pubmed. *Inf. Retr.*, 12(1):69–80, February 2009. ISSN 1386-4564. doi: 10.1007/s10791-008-9074-8. URL [http://dx.doi.org/10.1007/s10791-008-9074-8](http://dx.doi.org/10.1007/s10791-008-9074-8)

Daniel Macias-Galindo, Lawrence Cavedon, John Thangarajah, and Wilson Wong. Effects of domain on measures of semantic relatedness. *Journal of the Association for Information Science and Technology*, 2015.
George A. Miller. Wordnet: a lexical database for English. Commun. ACM, 38:39–41, November 1995. ISSN 0001-0782. doi: http://doi.acm.org/10.1145/219717.219748. URL http://doi.acm.org/10.1145/219717.219748

Mandar Mitra, Chris Buckley, Amit Singhal, Claire Cardie, et al. An analysis of statistical and syntactic phrases. In RIAO, volume 97, pages 200–214, 1997.

Mandar Mitra, Amit Singhal, and Chris Buckley. Improving automatic query expansion. In SIGIR ’98, pages 206–214, 1998.

Dan Moldovan, Marius Paca, Sanda Harabagiu, and Mihai Surdeanu. Performance issues and error analysis in an open-domain question answering system. ACM Transactions on Information Systems, 21(2):133–154, 2003. ISSN 10468188. doi: 10.1145/763693.763694.

Mohammed Mustafa, Izzedin Osman, and Hussein Suleman. Indexing and weighting of multilingual and mixed documents. In Proceedings of the South African Institute of Computer Scientists and Information Technologists Conference on Knowledge, Innovation and Leadership in a Diverse, Multidisciplinary Environment, SAICSIT ’11, pages 161–170, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0878-6. doi: 10.1145/2072221.2072240. URL http://doi.acm.org/10.1145/2072221.2072240

Roberto Navigli. Word sense disambiguation: A survey. ACM Comput. Surv., 41(2):10:1–10:69, February 2009. ISSN 0360-0300. doi: 10.1145/1459352.1459355. URL http://doi.acm.org/10.1145/1459352.1459355

Hwee Tou Ng. Does word sense disambiguation improve information retrieval? In Proceedings of the fourth workshop on Exploiting semantic annotations in information retrieval, ESAIR ’11, pages 17–18, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0958-5. doi: 10.1145/2064713.2064724. URL http://doi.acm.org/10.1145/2064713.2064724

Douglas W Oard and William Webber. Information retrieval for e-discovery. Information Retrieval, 7(2-3):99–237, 2013.

Douglas W Oard, Jason R Baron, Bruce Hedin, David D Lewis, and Stephen Tomlinson. Evaluation of information retrieval for e-discovery. Artificial Intelligence and Law, 18(4):347–386, 2010.

Jiaul H. Paik and Douglas W. Oard. A fixed-point method for weighting terms in verbose informational queries. In Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014, pages 131–140, 2014. doi: 10.1145/2661829.2661957. URL http://doi.acm.org/10.1145/2661829.2661957

Jiaul H Paik, Swapan K Parui, Dipasree Pal, and Stephen E Robertson. Effective and robust query-based stemming. ACM Transactions on Information Systems (TOIS), 31(4):18, 2013.

Marius A Pasca and Sandra M Harabagiu. High performance question/answering. In Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, pages 366–374. ACM, 2001.

John Prager. Open-domain question-answering. Found. Trends Inf. Retr., 1(2):91–231, January 2006. ISSN 1554-0669. doi: 10.1561/1500000001. URL http://dx.doi.org/10.1561/1500000001

Rahul Pramanik, Sukomal Pal, and Manajit Chakraborty. What the user does not want?: Query reformulation through term inclusion-exclusion. In Proceedings of the Second ACM IKDD Conference on Data Sciences, CoDS ’15, pages 116–117, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3436-5. doi: 10.1145/2732587.2732606. URL http://doi.acm.org/10.1145/2732587.2732606
Yonggang Qiu and Hans-Peter Frei. Concept based query expansion. In Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR ’93, pages 160–169, New York, NY, USA, 1993. ACM. ISBN 0-89791-605-0. doi: 10.1145/160688.160713. URL [http://doi.acm.org/10.1145/160688.160713](http://doi.acm.org/10.1145/160688.160713)

Dmitri Roussinov. Aspect presence verification conditional on other aspects. In Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval, pages 865–866. ACM, 2010.

M. Sanderson. Retrieving with good sense. Information Retrieval, 67:47–67, 2000. ISSN 1386-4564. doi: 10.1023/A:1009933700147. URL [http://dx.doi.org/10.1023/A:1009933700147](http://dx.doi.org/10.1023/A:1009933700147)

Mark Sanderson. Ambiguous queries: test collections need more sense. In Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, pages 499–506. ACM, 2008.

Rodrygo LT Santos, Craig Macdonald, and Iadh Ounis. Search result diversification. Foundations and Trends in Information Retrieval, 9(1):1–90, 2015.

Hinrich Schütze and Jan O. Pedersen. Information retrieval based on word senses. In Proceedings of the fourth workshop on Exploiting semantic annotations in information retrieval, SDAIR, pages 161–175, 1995.

Dou Shen, Jian-Tao Sun, Qiang Yang, and Zheng Chen. Building bridges for web query classification. In Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR ’06, pages 131–138, New York, NY, USA, 2006. ACM. ISBN 1-59593-369-7. doi: 10.1145/1148170.1148196. URL [http://doi.acm.org/10.1145/1148170.1148196](http://doi.acm.org/10.1145/1148170.1148196)

Anna Shtok, Oren Kurland, and David Carmel. Predicting query performance by query-drift estimation. In Leif Azzopardi, Gabriella Kazai, Stephen Robertson, Stefan Rüger, Milad Shokouhi, Dawei Song, and Emine Yılmaz, editors, Advances in Information Retrieval Theory, volume 5766 of Lecture Notes in Computer Science, pages 305–312. Springer Berlin Heidelberg, 2009. ISBN 978-3-642-04416-8. doi: 10.1007/978-3-642-04417-5\_30.

Anna Shtok, Oren Kurland, David Carmel, Fiana Raiber, and Gad Markovits. Predicting query performance by query-drift estimation. ACM Trans. Inf. Syst., 30 (2):11:1–11:35, May 2012. ISSN 1046-8188. doi: 10.1145/2180868.2180873. URL [http://doi.acm.org/10.1145/2180868.2180873](http://doi.acm.org/10.1145/2180868.2180873)

Mor Sondak, Anna Shtok, and Oren Kurland. Estimating query representativeness for query-performance prediction. In Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’13, pages 853–856, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2034-4. doi: 10.1145/2484028.2484107. URL [http://doi.acm.org/10.1145/2484028.2484107](http://doi.acm.org/10.1145/2484028.2484107)

Min Song, Il Yeol Song, Robert B Allen, and Zoran Obradovic. Keyphrase extraction-based query expansion in digital libraries. In Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries, pages 202–209. ACM, 2006.

Olga Vechtomova, Stephen Robertson, and Susan Jones. Query expansion with long-span collocates. Inf. Retr., 6(2):251–273, April 2003. ISSN 1386-4564. doi: 10.1023/A:1023936321956. URL [http://dx.doi.org/10.1023/A:1023936321956](http://dx.doi.org/10.1023/A:1023936321956)

Vishwa Vinay, Ingemar J. Cox, Natasa Milic-Frayling, and Kenneth R. Wood. On ranking the effectiveness of searches. In SIGIR 2006: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, USA, August 6-11, 2006, pages 398–404, 2006. doi: 10.1145/1148170.1148239. URL [http://doi.acm.org/10.1145/1148170.1148239](http://doi.acm.org/10.1145/1148170.1148239)
Ellen Voorhees and D. Harman. Overview of the Sixth Text REtrieval Conference (TREC-6). In *Proceedings of the Sixth Text REtrieval Conference (TREC-6)*, pages 1–24. NIST Special Publication 500-240, 1998.

Ellen M. Voorhees. 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, 1994. Springer-Verlag New York, Inc. ISBN 0-387-19889-X. URL [http://dl.acm.org/citation.cfm?id=188490.188508](http://dl.acm.org/citation.cfm?id=188490.188508)

Ellen M. Voorhees. Overview of the TREC 2003 robust retrieval track. In *TREC*, pages 69–77, 2003a.

Ellen M. Voorhees. Overview of the TREC 2003 robust retrieval track. In *TREC*, pages 69–77, 2003b.

Ellen M. Voorhees and Donna K. Harman, editors. *TREC Experiment and Evaluation in Information Retrieval*. MIT Press, 2005.

HC Wu, Robert WP Luk, Kam-Fai Wong, and Jian-Yun Nie. A split-list approach for relevance feedback in information retrieval. *Information Processing & Management*, 48(5):969–977, 2012.

Bo Xu, Hongfei Lin, and Yuan Lin. Assessment of learning to rank methods for query expansion. *Journal of the Association for Information Science and Technology*, 2015.

Jinxi Xu and W. Bruce Croft. Query expansion using local and global document analysis. In *SIGIR*, pages 4–11, 1996.

Jinxi Xu and W. Bruce Croft. Improving the effectiveness of information retrieval with local context analysis. *ACM Trans. Inf. Syst.*, 18(1):79–112, 2000.

Y. Xu, G.J.F. Jones, and B. Wang. Query dependent pseudo-relevance feedback based on Wikipedia. In *SIGIR 2009*, pages 59–66, 2009.

Elad Yom-Tov, Shai Fine, David Carmel, and Adam Darlow. Learning to estimate query difficulty: Including applications to missing content detection and distributed information retrieval. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR ’05, pages 512–519, New York, NY, USA, 2005. ACM. ISBN 1-59593-034-5. doi: 10.1145/1076034.1076121. URL [http://doi.acm.org/10.1145/1076034.1076121](http://doi.acm.org/10.1145/1076034.1076121)

Le Zhao and Jamie Callan. Automatic term mismatch diagnosis for selective query expansion. In *Proceedings of the 35th international ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR ’12, pages 515–524, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1472-5.

Yun Zhou and W. Bruce Croft. Ranking robustness: A novel framework to predict query performance. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, CIKM ’06, pages 567–574, New York, NY, USA, 2006. ACM. ISBN 1-59593-433-2. doi: 10.1145/1183614.1183696. URL [http://doi.acm.org/10.1145/1183614.1183696](http://doi.acm.org/10.1145/1183614.1183696)

Yun Zhou and W. Bruce Croft. Query performance prediction in web search environments. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR ’07, pages 543–550, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-597-7. doi: 10.1145/1277741.1277835. URL [http://doi.acm.org/10.1145/1277741.1277835](http://doi.acm.org/10.1145/1277741.1277835)
