Approaches to Using Word Collocation in Information Retrieval

Olga Vechtomova

Thesis submitted for the degree of Doctor of Philosophy

Department of Information Science
City University
London

November 2001

Contents

List of Figures

List of Tables

Acknowledgements

I am thankful to everyone who contributed to this project and helped to make it possible. In particular I would like to thank:

Steve Robertson and Susan Jones for their inspiring ideas, wise supervision, support and guidance.

Steve Walker for his advice and help with Okapi.

My parents and friends for their support and encouragement.

Declaration of Copyright

Abstract

The thesis explores long-span collocation and its application in information retrieval. The basic research question of the thesis is whether the use of long-span collocates can improve performance of a probabilistic model of IR. The model used in the project is the Robertson & Sparck Jones probabilistic model.

The basic research question was explored by investigating three different ways of integrating collocation information with the probabilistic model:

1. Global collocation analysis. The method consists in expanding the original query with long-span global collocates of query terms. Global collocates of a query term are selected from large fixed-size windows around all occurrences of a term in the corpus and ranked by statistical measures of Mutual Information (MI) and Z score. A fixed number of top-ranked collocates is used in query expansion.

   Query expansion with global collocates did not show to be superior to the original queries, the possible reason being the fact that query terms often have a fairly broad meaning and, hence, a rather semantically heterogeneous pattern of occurrence.

2. Local collocation analysis. This method is a form of iterative query expansion following relevance or pseudo-relevance (blind) feedback. The original query is expanded with the query terms' collocates which are extracted from the long-span windows around all occurrences of query terms in the known relevant documents, and selected using statistical measures of MI and Z. Some parameters whose effect was systematically studied in this experiment set are: window size, measure of collocation significance for collocate ranking, number of query expansion collocates and categories of terms in the expanded queries.

   Some results showed a tendency towards performance gain over relevance feedback in the probabilistic model, however it was not significant enough to conclude that this method is superior to the existing relevance feedback used in the model.

3. Lexical cohesion analysis using local collocations. This experiment set aimed to explore whether the level of lexical cohesion between query terms in a document can be linked to the document's relevance property, and if so, whether it can be used to predict documents' relevance to the query. Lexical cohesion between different query terms is estimated from the number of collocates they have in common.

   The experiments proved that there exists a statistically significant association between the level of lexical cohesion of the query terms in documents and relevance. Another set of experiments, aimed at using lexical cohesion to improve probabilistic document ranking, showed that sets re-ranked by their lexical cohesion scores have similar performance as the original ranking.

1 Introduction

The project presented in this thesis explores the phenomenon of word collocation and its application in information retrieval. Specifically the project was aimed at studying long-span collocations and their use in a probabilistic model of IR.

Long-span topic-level collocations are motivated by lexical-semantic relations, as opposed to short-span collocates – motivated by lexical-grammatic or habitual factors Long-span collocates have certain lexical-semantic relatedness, causing their more than random co-occurrence in the same thematic environments – topics – in text. The motivation for studying this group of collocations in their application to IR is to explore ways of obtaining more relevance-discriminating evidence from the contextual environments of query terms in texts.

In IR long-span collocations received relatively little attention, in contrast to short-span collocation, used in phrase identification, or document-wide co-occurrence, used for a wide range of tasks like term weight calculation or relevance feedback. Much of the research on document-wide co-occurrence dates back to the time before the wide use of full-text. However, it is still very widely used, for example in the relevance feedback process of the Robertson & Sparck Jones probabilistic model – the testbed for this project. Accounting for document-wide co-occurrence is reasonable with the abstracts collections, where the contents of an abstract is rather semantically homogeneous. However, it no longer seems adequate for full-text collections, containing multi-topic documents. It was hoped that accounting for term dependencies within smaller and more semantically homogeneous text constructs would lead to improved retrieval performance.

The basic research question of the project is whether the use of long-span collocates of query terms can improve performance of the probabilistic model. The research question was explored by studying three different ways of integrating collocation information into the probabilistic model: global collocation analysis, local collocation analysis and lexical cohesion analysis using local collocations. Four hypotheses, examined over the course of the project, as well as an overview of the above three sets of studies forming the project are given in Chapter 5.

Some of the information presented in sections *5.4 Choice of statistical measures of collocation significance* and *6.3 Lexical-semantic analysis of global collocates* was originally presented in a paper by Vechtomova and Robertson – [Vechtomova2000].

The structure of the thesis is described below.

Chapters 2, 3 and 4 are largely theoretical and contain literature review and discussions.

Chapter 2. *Linguistic perspective on text and lexis* gives a theoretical background of the linguistic aspects of the objects of study. The first section contains a discussion on the organisation of text, what defines a text and what properties differentiate a text from unconnected sequences of sentences. Following [Halliday76] it is argued that a

text is a semantic unit of a non-structural nature, held together by a property of cohesion. A brief description of the hierarchies of linguistic elements forming the plane of expression and the plane of contents of a text is also given. The first section also contains a discussion of the text-forming property of cohesion.

The second section is devoted to lexis as a stratum of the language-system. Specifically, it discusses the notions of a lexical unit, lexical meaning, lexical/semantic relations and syntagmatic/paradigmatic relations between lexical units.

The third section centres around the notion of lexical cohesion and the mechanisms of its realisation in text: lexical links, lexical chains and bonds. A review of the most influential research projects on lexical cohesion in linguistics is also given

The fourth section focuses on linguistic perspectives on collocation. Two different views of collocation are described: collocation as a topic-forming relationship [Halliday76, Hoey91] and collocation due to lexical-grammatical or habitual relations [e.g., Manning99, Palmer81, Robins89]. This section also briefly introduces the branch of linguistics – corpus linguistics – where the methodology for statistical identification of collocation in text was formed. Subsection *2.4.2 Patterns of collocation in text* discusses such aspects of collocation as the distance and consistency of a word's influence on its environment in text.

Chapter 3. *Natural language text in IR context* contains literature reviews and discussions on a number of project-related topics. The chapter starts with a brief discussion of two major approaches to document and query representation in IR: controlled language and natural language indexing.

The second section contains a discussion of two approaches to IR – statistical and natural language processing (NLP). The first half of this discussion is of a more philosophical nature – talking about compatibility of IR with NLP, on the one hand, and with statistical approaches, on the other. The second half compares statistical and NLP approaches to defining complex indexing units (phrases) in IR.

The third section contains literature review of approaches to word collocation in IR, which are divided into three groups: short-span collocations, document-wide co-occurrences and long-span collocations. The later group, being the focus of this thesis, is given the maximum coverage.

The chapter also contains discussions and reviews of IR research on lexical cohesion and document passages.

Chapter 4. *Probabilistic information retrieval* presents the IR model used in this project – the Robertson & Sparck Jones model of probabilistic IR. The chapter focuses on the two major aspects of the model: term relevance weighting and relevance feedback. A separate section talks about term independence assumptions, the model is based on, and a relevance-based dependence, implied by these assumptions.

Chapter 5. *Collocation analysis in this project* starts with the section *5.1 Research question*, containing the formulations of the basic research question and the hypotheses examined in the project. Section *5.2 Overview of the experiments* contains a brief overview of the three stages of experiments, constituting this project: global collocation analysis, local collocation analysis and lexical cohesion analysis using local collocations. These sections were included into chapter 5 instead of the beginning of the thesis, because it was considered important to present the theoretical background first before introducing the experimental part.

This chapter also contains detailed descriptions of the methodologies used in all or most of the experiments in the project, as well as the material used in the experiments: document collection and topics.

Chapters 6, 7 and 8 describe three stages of experiments carried out over the course of the project. A brief overview of these experimental stages is given in chapter 5, section *5.2 Overview of the experiments*.

Chapter 9 *Conclusions and Recommendations* sums up the main findings of the project and presents suggestions for further research.

2. Linguistic perspective on text and lexis

2.1 Natural language text

2.1.1 Text organisation

The definition of text and the nature of its organisation has been a matter of considerable debates in linguistics. Linguists disagree not only about the factors that determine internal organisation of text, but also what kind of language activity should be considered text and which - non-text. Text analysis has become a focus of some areas of linguistic study in the recent decades. As a linguistic discipline it is still at a stage of active research where controversies about key objects of research are not uncommon. It is not the aim of this chapter to give a complete coverage of all research activities in the area of text linguistics; instead the most influential approaches to the nature of text and its organisation will be introduced, some of which determined the view of the notion of text adopted in this work.

Lyons gave examples of two polar approaches to the nature of text [Lyons77]. The first is the view of the text as a "product of more or less conscious and controlled literary composition" [Lyons77, vol.2, p.631]. The second approach is shared by linguists who adopted a broader understanding that the text can be the product of everyday language-behaviour including phonologically transcribed conversations [Lyons77, vol.2]. However neither of these approaches to defining the notion of text clearly indicate the criteria for distinguishing text from non-text. As Halliday and Hasan pointed out, native speakers intuitively know whether any product of language activity is a text or not. This means there should be some objective characteristics of text that determine our perception of it as a text and not as a sequence of unrelated sentences.

One possible way of defining a text is to attribute to it the idea of a *whole*. The view of a text as a whole gained substantial recognition among linguists working within different frameworks of linguistic study: text grammars, discourse analysis, semantics. Halliday and Hasan emphasised this idea, defining the text as "any passage spoken or written, of whatever length, that does form a unified whole" [Halliday76]. They recognise however that distinction between a text as a whole and a sequence of disconnected sentences is a matter of degree, which depends on the stylistic factors such as literary genre and the author's language ability and competence. Nevertheless they point out that if a piece of written or spoken language activity is to be considered a text, it must have some internal features that differentiate it from a collection of unrelated sentences.

Although many linguists agree that the pivotal characteristic of text is its being a whole unity, their opinions diverge considerably as to what exactly holds a text together. The differences are rooted in the overall approach to text: it is viewed either as a *grammatical entity*, or as a *semantic unit*.

Linguists who look at a text as a grammatical entity come from the structuralist background and study text within the framework of the so-called "text grammar".

Most text grammarians use some aspects of the sentence grammar in the analysis of the text. They claim that text has a rigid structure comparable to that of the sentence. Influential work in this area of text study belongs to Van Dijk [Dijk77]. He claims that the sentences are interrelated structurally in text just as clauses are interrelated within the sentence, and that both sentences and texts can be similarly described at some level using the same methods. However the global structure of a text, according to Van Dijk, also requires another level of description, which must be formulated as a systematic set of rules to derive the semantic structure of a text from the semantic structures of sentences. He introduces the notion of *macro-structures* to describe the semantic representation of a text or its part. Macro-structures are a formal equivalent to the intuitive notion of a topic of discourse. Van Dijk assumes that a topic of discourse, which denotes what a text or a part of it is about, can be described as a complex logical proposition which is derived from a set of propositions contained in the sequences of sentences. The semantic structure of a text can be described using several levels of macro-structures, starting from the simple macro-structure of a single sentence to the macro-structures of paragraphs and the entire text. [Dijk77]

Such a formalist approach to a text has been criticised by some linguists [Halliday76, Brown83]. Brown and Yule argue that an attempt to formulate general rules for deriving the semantic structure of a text objectively is an illusion. What Van Dijk attempts is to produce a sentence representing the underlying topic of the text and transform it into a set of logical propositions which would have a relation of entailment with the set of propositions derived from the sentences in the text. Brown and Yule argue that this approach is a way of determining just one of the possible topics in the text and cannot claim objectivity [Brown83].

Some linguists do not recognise a text as an object of grammatical study, claiming that the sentence is the highest grammatical unit [Halliday76, Lyons77]. Halliday and Hasan argue that a text is not a structural unit. It cannot be said that it is similar in kind to a sentence, and it does not consist of sentences in the same way as sentences consist of clauses. While it is possible to specify a limited number of structural relations within a sentence, it is not possible to draw up a list of structural relations within a text, or to categorise sentences into classes which perform certain roles in text. They doubt that sentences can enter into generalised structural relationships to realise some function in the text as a unity [Halliday76].

Many opponents of the structural approach to text suggest that a text, rather than having a structure, has some organisation [Halliday76, Hoey91]. Instead of considering text as a grammatical unit that permits structural description, they view it as a semantic unit that has some patterns of organisation. An interesting distinction between a structural and organisational approaches to text was drawn by Hoey. He states that a structural description is "one that permits one to make predictive statements about the data under examination" [Hoey91, p.13]. Structural descriptions predict what is possible or impossible to occur, whereas patterns of organisation depict what has occurred. Therefore if a structural stance is adopted, the description must account for which combinations are possible and under what circumstances. Description as patterns of organisation, on the other hand, does not suggest the impossibility, but the probability of certain combinations. Hoey emphasises that " text analysts can describe what is common, the culturally popular patterns of organisation,

but they are not convincing when they seek to make predictive statements about what can and cannot occur" [Hoey91, p.204].

Halliday and Hasan describe text as a unity brought together not by a structure but by the organisation they term texture [Halliday76]. They introduce the notion of "texture" to emphasise the fact that a text is a whole unit, but the factors that contribute to its unity are not of structural nature. The property of the text that gives it texture is *cohesion*. Texture is formed by the relationships that bind together items in text, which they call cohesive ties. The concept of cohesion will be described in greater detail in section *2.1.3 Cohesion in text*.

The following section will give an overview of the elements of text as seen from two different areas of linguistic study: grammar and semantics.

2.1.2 Text elements

Like any unit of language in use, text exists in two indivisible planes: *the plane of contents* and *the plane of expression*. The linguistic disciplines that study means of language expression are phonology and grammar, while the discipline which concerns itself with the study of meanings is semantics.

Linguistic elements can be described as organised hierarchically: the smaller units are combined to form larger units of language, which are different in kind (figure 2.1).

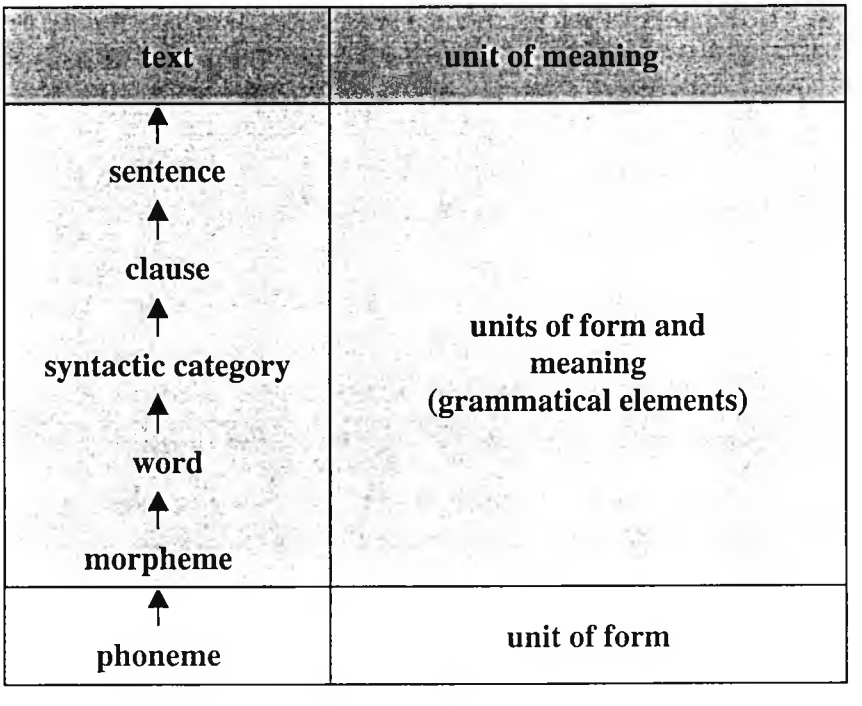


Figure 2.1. Stratal organisation of language elements

The smallest unit of the spoken language is the *allophone*, which is a sound variant of a structural element of *phoneme*. The linguistic discipline which studies the sound system of language is phonology. The phoneme, however, does not carry any meaning in itself, but serves only as a building block for higher ranked elements.

The next element in the hierarchy is a *morpheme*, which is a component of a word structure. The essential feature of a morpheme as contrasted to a phoneme is that the morpheme is a unit of meaning. It is defined as "the smallest unit of language that carries information about meaning or function" [O'Grady97, p.133]. Morpheme is the smallest grammatical element and is an object of the branch of grammatical study - morphology. Morphology is commonly defined as "the system of categories and rules involved in word formation and interpretation" [O'Grady97, p.132]. A *word* - the next element in the hierarchy of linguistic elements - can consist of one or more morphemes. Complex words normally comprise a root - a morpheme that carries the major part of the word's meaning and one or more affixes - morphemes carrying grammatical meaning. For example word *'walked'* consists of a root morpheme *'walk'* and of an affix *'ed'* which conveys the grammatical meaning of past. The word as a primary object of lexical study will be covered in a greater detail in the subsequent part *2.2 Lexis.*

The highest grammatical element in a text is a *sentence*. The branch of grammar that studies the sentence structure is syntax. Syntax can be formally defined as "the system of categories and rules that underlies sentence formation" [O'Grady97, p.181]. A sentence, however, does not simply consist of sequences of words. There are several levels of structural components that are combined together to form sentences. The smallest syntactic unit is called a *syntactic category*, which is defined as "a group of words or sequences of words in a given language that can replace one another in any sentence of the language whatsoever without affecting grammaticality" [Culicover76, p.13]. A syntactic category can either contain single words, in which case it is known as a lexical category, or it can contain sequences of words: a phrase category [Culicover76]. The most studied lexical categories are noun, verb, adjective and preposition. Phrase categories are based on lexical categories, and are structured as head + specifiers, where the head is the lexical category which the phrase is built on [O'Grady97]. Thus, if the head of a particular phrase is represented by a lexical category of noun, the category the phrase belongs to is a noun phrase.

Syntactic categories are combined into *clauses*. A clause is centred around a predicate - a verb or a verb phrase, and can also have other syntactic categories functionally related to the predicate, such as subject and complement. A sentence can consist either of only one clause - a *simple sentence*, or of more than one clause – *coordinate* and *complex sentences*. In coordinate sentences the clauses have a coordinate relationship, whereas in complex sentences one clause is grammatically subordinate to another.

The linguistic elements presented above function each at their own level to build the plane of expression of text, which serves in its entirety to realise the underlying contents of text. The discipline devoted to the study of meaning is semantics. Semantics as a study of meaning expressed by any sign system is not confined to linguistics, however language, being a major sign system is a dominant object of semantic study. Language semantics concerns itself with the meanings expressed by

language elements on different levels, such as word meanings and sentential meanings. Word meaning will be an object of a separate discussion in the later section of this work devoted to lexis (*2.2.1 Lexical unit and lexical meaning*).

Textual content is not a direct combination of the meanings conveyed by individual sentences. As described in the previous section, text is a more complex semantic unit than a simple combination of sentences. Therefore the meaning on the sentential level will only be touched upon briefly in this section, with the main focus on the larger semantic units existing in text.

Most discussions about the textual content centre around the intuitive notion of *topic* or 'aboutness' of a text or its parts. The notion of topic has proved difficult to define in linguistic study. Primarily, it is important to differentiate sentence topic and discourse topic. The topic, as it is understood in sentence grammar, is "its centre of attention" [Finegan94, p.200], in contrast to the notion of "comment", which is new information introduced in the sentence about the topic. The definition of the notion of discourse topic is a more difficult task than the definition of a sentential topic. Some linguists suggest that a topic is a proposition. Among such approaches is the rather formalist definition of a topic as a complex logical proposition suggested by Van Dijk [Dijk77], which was discussed in the previous section of this work. The proposition-based approach implies the idea that there exists a phrase or a sentence that is a single correct representation of the topic of the whole text or its part. Brown and Yule criticize this approach, arguing that for any text there can be several ways to represent the topic, each of which will reflect a particular judgement about the text contents [Brown83]. To determine the set of correct judgements about the discourse topic Brown and Yule suggest that extra-textual factors need to be taken into account. They describe the factors that are external to the text but necessary for the correct interpretation of its contents as the "activated topic framework" [Brown83].

The linguists whose approaches to topic have been discussed above try to define this notion from the point of view of its contents. Others, in contrast, try to derive its definition by formulating methods of locating boundaries between topics in text. Such approaches are based on the assumption that a shift from one topic to another is marked in some way in the text. The most straightforward way is to look for the boundary between topics in the orthographic boundaries between paragraphs. However, as some linguists point out, orthographic paragraphs often have stylistic rather than semantic functions [Longacre, Hinds cited in Brown83, p.95]. Therefore it is possible that the beginning of a new paragraph coincides with the start of a new topic, but this is not necessarily so [Brown83]. Another approach is to try to derive the information about topic boundaries from some other information which can be elicited from text. The information in question is various means of expression of cohesion in text, in particular its type - lexical cohesion, discussed in detail in section *2.3 Lexical cohesion*. Some implications of the analysis of lexical chains and bonds for the discovery of topic structure in text will be touched upon in sections *2.3.2* and *2.3.3*.

2.1.3 Cohesion in text

Following linguists who adopted an organisational approach to text [Halliday76, Hoey91], it is assumed in this thesis that text is a non-structural unity, whose entirety is dependent not on grammatical rules, but on non-structural relations existing between its elements. Halliday and Hasan introduced the concept of 'textual' or 'text-forming' component of the linguistic system, which they define as a "set of resources in a language whose semantic function is that of expressing relationship to the environment" [Halliday76, p.299]. They claim that it is the meaning realised through text-forming component of the language that creates text, and distinguishes it from the unconnected sequences of sentences. Cohesion constitutes the major unit of text-forming resources of the language, and its role is to express continuity throughout the parts of text. As Halliday and Hasan point out that the continuity created by cohesion consists in "expressing at each stage in the discourse the points of contact with what has gone before." [Halliday76, p.299]

It is important to emphasise that cohesion is a semantic concept which exists in a set of semantic relations between elements in text called cohesive relations or 'ties' in Halliday and Hasan's terms. These cohesive relations are realised through a set of devices, which will be covered later in this section, but the means of realisation of the cohesive relations are only of secondary importance. The meaning of the cohesive relation does not depend on the means of its realisation [Halliday76].

A cohesive relation occurs between two items in text when for the complete interpretation of one item the presence of the other is required. These two cohesively related items can occur, either within a sentence, or in different sentences throughout the text. In the former case, cohesion is realised in the syntactical structure of the sentence; in the latter it is realised through non-structural text-forming devices. A sentence, being subject to strict syntactic rules of structuring, imposes restrictions on how a cohesive relation can be expressed. For example there are rules of pronominalisation which require the use of a pronoun to name the entity mentioned for the second time in the same sentence [Halliday76]. Cohesion, however, is not divided into two distinct types: sentential and textual; it is first of all a semantic and not purely formal relation between two items, whose position in text is not governed by grammatical structure. "Cohesion is a general text-forming relation or a set of such relations, certain of which when incorporated within a sentence are subjected to certain restrictions." [Halliday76, p.9]

Cohesive relations are realised in the plane of expression of text through grammatical and lexical devices. Halliday and Hasan distinguished five categories of cohesive relations according to the means of their realisation in text [Halliday76]:

- Reference
- Substitution
- Ellipsis
- Conjunction
- Lexical cohesion

The first four categories are realised through grammatical structures, whereas the fifth category is realised through lexis.

Reference occurs when certain items do not refer to a concept directly, but refer to another item in text which in turn refers to that concept. In English such reference items are pronouns and determiners. Reference is a semantic relation, i.e. a relation between the meanings of the two items in the context and not between the linguistic units.

Substitution is a grammatical relation whereby certain words stand for other lexical units which took place earlier in text. The examples of such words are 'one' standing for a noun and 'do' standing for a verb which occurred earlier in text. The cohesive relation which takes place in substitution is the relation between two lexical items, as distinct from reference - relation between meanings.

Ellipsis can be defined as "substitution by zero" [Halliday76, p.143]. Substitution and ellipsis are the same process, only in case of ellipsis what stands for the item mentioned earlier in text is nothing.

Conjunction consists in using adjunct-like elements like 'although', 'however', 'nevertheless'. Conjunction is different from the above three types of cohesive relations, as conjunctive elements are not cohesive themselves in the sense that they are not directly related to any preceding item or its meaning. The role of conjunctive elements is to express certain meanings which presuppose other semantic components in text before them [Halliday76].

Lexical cohesion is achieved through semantic connectedness between lexical items in text. It is principally different from the above four types in the fact that cohesive relations falling into this category are in the first instance lexical relations and only secondarily textual. But as Hoey points the relation between lexical and textual is bi-directional, i.e. "the text provides the context for the creation and interpretation of lexical relations, just as the lexical relations help create the texture of the text" [Hoey91, p.8].

Lexical cohesion will be the subject of a more detailed discussion in section *2.3*. Since lexical cohesion is a relationship between words as lexical units and not as grammatical items in their functional roles in text, it is first necessary to illuminate another stratum of language organisation – *lexis*, which will be the focus of the discussion in the next section.

2.2 Lexis

2.2.1 Lexical unit and lexical meaning.

The question of what object in language should be defined as a lexical unit is controversial in linguistics. It is outside the scope of this thesis to give a comprehensive overview of all approaches to this problem. Therefore, first, the factors contributing to the complexity of the task of defining a lexical unit will be outlined and, secondly, the perception adopted in this thesis will be described.

Attributing the role of a lexical unit to a pre-theoretical notion of a 'word' as a sequence of letters delimited by spaces in text, will leave many questions unanswered.

Language in use is characterised by a complex interplay between grammatical and lexical factors. A word in text can incorporate both grammatical and lexical properties. Each word functioning as a grammatical element in a sentence has certain grammatical meaning or a set of meanings, but not every word has lexical meaning, i.e. referring to a concept. It is widely accepted to classify words into content words and function words. Function words like prepositions and articles have only grammatical meaning, which can be explained only in relation to other elements in the sentence. Content words, on the other hand, also have a lexical meaning, which is based on the reference to a concept. Function words are, therefore, the object of grammatical study, whereas content words are the object of both grammatical and lexical study. Each occurrence of a content word has a combination of a lexical and grammatical meaning, for example word *travelled* has a lexical meaning referring to the concept of 'travelling' and a grammatical meaning of past. *Travelled, travels, travelling* are not considered as different words, but different word forms of the same lexeme *to travel*. The lexeme of a word is a unit of lexis devoid of grammatical characteristics, representing all grammatical forms of this word in use. And it is a lexeme as a unit of a language structure of lexis that will be considered as a lexical unit in this thesis.

Another controversial question is whether to consider multiword entities like compound words or compound terms as lexical units. So far there is no reliable method for isolating such entities in text and distinguishing them from combinations of two or more lexical units. Compounding is recognised as a word forming mechanism in a language, whereby two or more lexemes are put together to form a new lexeme. While there is little doubt that words like *highranking* constitute a single lexical unit, items such as *search engine*, composed by the same word forming mechanism could be viewed as two lexical units or one. The essential characteristic of a compound word is that as a whole it denotes a new concept, different from the concepts denoted by any of its compounds. In contrast, a noun phrase like *powerful engine* does not refer to a new concept. Therefore, a compound word which as a whole denotes a new concept can be considered as a lexical unit, i.e. part of the lexical structure of language.

The definition of a lexical meaning is no less controversial than that of a lexical unit. The point of view that the lexical meaning of a word is a concept referred to by it is common in linguistics. Saussure's idea [Saussure16] that a word as a linguistic sign is a two-sided entity consisting of a sound-form and a concept referred to by it has received a wide acclaim in structural linguistics. In this work a related but somewhat different approach to defining a concept and a lexical meaning is adopted. A concept representing our knowledge about a real-world object is viewed as having an almost infinitely large number of attributes characterising this object from all facets: its characteristics, relations with other objects, etc. Lexical meaning is a subset of these attributes, including only the most characteristic attributes of the concept, necessary for differentiating it from other concepts [Apresjan74, Levkovskaja62, Novikov83]. This is evident in the existence of synonyms particularly in terminologies, whereby although all members of the synonym group are related to the same concept, they have different lexical meanings, because each of them highlights a particular aspect of this concept.

Another important point about the nature of lexical meaning, postulated in structural linguistics [Saussure16], is that lexical units do not exist in isolation in language. They are part of the systems of lexical units existing in two dimensions - paradigmatic and syntagmatic (see the next section). Therefore the complete meaning of the lexical unit can be derived only by considering the relationships it has with other lexical units in these two spheres of the language.

To summarise, the lexical meaning is established, on the one hand through its relatedness to extra-linguistic entities - concepts, and on the other hand through its relations with other lexical meanings in both the lexical structure of the language (the paradigmatic sphere) and in the language in use (the syntagmatic sphere).

Before moving on to the discussion of syntagmatic and paradigmatic relations it is necessary to clarify two concepts: *semantic relations and lexical relations*, which will be used further in this work. Semantic relations are relations between concepts and are extra-linguistic in their nature. Lexical relations exist between lexical meanings of the lexical units. [Evens88] They are however more complex than relations between concepts due to the fact that on the one hand they reflect the relations between the concepts underlying lexical meanings (e.g. *hyponymy*), and on the other hand, being part of the language, they include purely linguistic relations between lexical meanings (e.g. *synonymy*). Because lexical relations between words encompass not only the linguistic but also the conceptual sphere it is not uncommon to refer to them as *lexical-semantic relations* or even *semantic relations* to emphasise the relations between underlying concepts. In this work all three terms will be used to refer to relations between lexical meanings of words, the choice of the term depending on the desired emphasis: either on the purely linguistic relations or on the underlying conceptual relations. Where the distinction is unimportant the term *lexical-semantic relations* will be used.

2.2.2 Syntagmatic and paradigmatic relations

Saussure, the founder of modern structural linguistics, formulated the idea that any linguistic element exists and derives its value through relationships with other linguistic elements [Saussure16]. This became the central thesis of Saussurean structuralism. Linguistic elements at any level of analysis are interrelated in two dimensions - in the language as a system and the language in use (*'langue'* and *'parole'* in Saussurean terms or *'language-system'* and *'language-behaviour'* in Lyon's terms [Lyons77]). Relationships in the language-system are known as paradigmatic and in the language-behaviour as syntagmatic. Robins summarised the differences between the two types of relations as follows:

> "Syntagmatic relations are those holding between elements forming serial structures, or 'strings' (...) at a given level ... " [Robins89, p.47]
>
> "Paradigmatic relations are those holding between comparable elements at particular places in structures ..." [Robins89, p.47]

Syntagmatic relations between words as lexical units are relations between the lexical meanings of words co-occurring in the linearity of text. Saussure stressed that

syntagmatic relations exist between words consecutively occurring in text: "words acquire relations based on the linear nature of language because they are chained together" [Saussure16, p.123]. From this it follows that those relationships between words that realize lexical cohesion in text - repetition and collocation (see section *2.3*) - are of syntagmatic type.

Paradigmatic relations exist between lexical units as part of the lexical structure of the language-system. The lexical structure of the language can be viewed as an abstraction: the mental organisation of the lexicon, on the one hand mirroring the mental organisation of concepts, and on the other hand holding relations specific to the language itself.

Relations commonly recognised as holding within the lexical structure of the language are:

- Antonymy;
- Synonymy;
- Hyponymy;
- Meronymy (part-whole relationship).

Antonymy is a name for a wide range of relations, all of which are broadly characterised by the idea of 'oppositeness' of meaning between two lexical units.

Synonymy is a relationship between two or more lexical units which can be characterised as having 'sameness of meaning'. It is doubtful, however, that language keeps true synonyms, i.e. words with exactly the same lexical meanings. Usually synonymy in language is not absolute but relative, i.e. words have certain degree of similarity in meaning, but they have some differences - stylistic, emotive, collocational or dialectal [Palmer81].

Hyponymy is a hierarchical relation between more specific lexical units - hyponyms and more general units - hypernyms. There is a widely though not universally accepted point of view that lexicon of the language is organised hierarchically and can be represented as tree-diagrams [Lyons77]. The lexical meaning of a word in many cases can be specified by relating this word to its hypernym and contrasting it to its co-hyponyms.

Meronymy or part-whole relation is the second type of hierarchical relation. Though the part-whole relation does not organise the lexical structure like the hyponymy relation, it is important in the sense that lexical meanings of some words are specified through their relation to other words by the part-whole relation.

2.3 Lexical cohesion

Each lexical item used in a text acquires its own textual history, i.e. the lexical environment or context in which it is interpreted. By the time a word is encountered in text, the context for its instantial interpretation has already been built up. At the same time the fact that the previous lexical items which constitute the context of this word are necessary for the realisation of its 'instantial meaning' shows that it forms relationship of lexical cohesion with them [Halliday76]. It can be said that there is a

bi-directional relationship between the word in question and preceding lexical items which constitute its context: on the one hand the preceding lexical items provide a background for the correct interpretation of this word in the current stretch of text, and on the other hand the word itself through its cohesion with the earlier items functions as an element which contributes to the continuity of text.

The main feature distinguishing lexical cohesion from grammatical types of cohesion is the fact that lexical items that cohere in text are independent of each other in the realisation of their lexical meaning, though they rely on their common environment for the realisation of their *'meaning potential'* [Hoey91, p.71], or *'instantial meaning'* [Halliday76, p.289]. Grammatical items in grammatical cohesion, in contrast, not having a lexical meaning of their own, rely entirely on their lexical counterparts for the realisation of their contextual meaning.

It is notable that, in Halliday and Hasan's analysis of various types of cohesion in texts of various genres, lexical cohesion accounts for more than 40% of all cohesive relations [Halliday76].

The third stage of our experiments (chapter 8) explores the concept of lexical cohesion between query terms, realised through the similarity of their collocates. In particular, it studies the relation between documents' lexical cohesion and their property of relevance. It was, therefore, considered important to give in the following subsections a more detailed discussion of lexical cohesion and linguistic mechanisms for its realisation in text.

2.3.1 Types of lexical cohesion

Halliday and Hasan in their earlier work [Halliday76] distinguished two broad categories of lexical cohesion: *reiteration* and *collocation*.

Reiteration, as Halliday and Hasan understand it, refers to a broad range of relations between a lexical item and another word occurring before it in text, where the second lexical item can be an exact repetition of the first, a general word, its synonym or near-synonym or its superordinate. Lexical reiteration is different from the grammatical cohesion type - reference by several major factors. First of all the second word in this relation is a member of an open set, i.e. it is a lexical and not a grammatical unit. Secondly, it can not only refer to the previous item, but be related to it through a lexical-semantic relation, for example synonymy. And thirdly, it does not need to refer to the same referent as the first word. As Halliday and Hasan point out the idea of reference is irrelevant to lexical cohesion, since it is not a referential relation. The second word can be identical, inclusive, exclusive or unrelated to the referent expressed by the first word [Halliday76, p.283].

Another broad category that Halliday and Hasan distinguish in lexical cohesion - collocation - is defined rather vaguely in their earlier work [Halliday76]. Collocation will be discussed in section *2.4* of this thesis, however at this point it is necessary to introduce the notion by giving the earliest established understanding of collocation in linguistics formulated by Firth [Firth58]: collocation is a relationship between items which co-occur in text with more than random probability. Although Halliday and

Hasan admit that collocation is a relationship between lexical items that occur in the same environment, they, nevertheless, define this category rather generally, namely as lexical cohesion which is not reiteration [Halliday76, p.287].

Thus, the suggested categorisation of lexical cohesion in Halliday and Hasan's earlier work is as follows (taken from [Halliday76, p.288]):

Type of lexical cohesion:

1. Reiteration
    (a) same word
    (b) synonym (or near synonym)
    (c) superordinate
    (d) general word
2. Collocation

In her later work Hasan [Hasan84] admitted the intuitive character of their earlier category of collocation and left it out from her later categorisation of lexical cohesion altogether, introducing in its place a set of narrowly-defined lexical relations. The following categorisation is taken from [Hasan84, p.202]:

A	*General*	i. repetition	leave, leaving, left
		ii. synonymy	leave, depart
		iii. antonymy	leave arrive
		iv. hyponymy	travel, leave (including co-hyponyms, leave, arrive)
		v. meronymy	hand, finger (including co-meronyms, finger, thumb)
B	*Instantial*	i. equivalence	the *sailor* was their *daddy*; *you* be the *patient*, *i*'ll be the *doctor*
		ii. naming	the dog was called *Toto;* they named the *dog Fluffy*
		iii. semblance	the *deck* was like a *pool*; all my pleasures are like *yesterdays*

Hoey [Hoey91] in his categorisation of lexical cohesion omitted the category of collocation. Instead he categorised a rather broad range of relationships under the heading of *repetition*. He distinguished the following sub-categories:

1. Simple lexical repetition
2. Complex lexical repetition
3. Simple partial paraphrase
4. Simple mutual paraphrase
5. Complex paraphrase
6. Superordinate, hyponymic and co-reference repetition

By simple lexical repetition Hoey understands a relationship between two words in text which are the same lexical units, belonging to the same lexical category, but possibly having different grammatical characteristics, such as tense or number. Hoey points out that the seemingly uncomplicated nature of this relationship is deceptive. The major problem in identifying this relation lies in the possible polysemy of a word.

If only form is taken into account, then any two words with identical forms will be treated as an instance of repetition relationship. First of all Hoey points out that repetition does not presuppose identical meaning of its components. Their meanings can be related to different degrees. The major requirement, however, is that two words must have a similar collocational environment. If two occurrences of a word in a text have similar meanings, their collocational environment will be similar as well [Hoey91]. For more detailed discussion of collocation see section *2.4*.

Complex repetition occurs when two words have the same root morpheme, but their other word-forming morphemes are different (e.g., write, writing; visible, invisible), or when they are identical formally, but belong to different lexical categories, e.g., animal *(noun)*, animal *(adjective)*.

Both simple and complex repetitions cover the cases when two words have identical root morphemes. Words which enter paraphrase relations are different lexical units. The main criteria for defining a relation as a paraphrase is that one word could be used as a substitute for another word and its meaning must have the same level of generality. It could be compared to Hasan's category of synonymy, but, as Hoey points out, he relies on the context in identifying the relation as a simple paraphrase, whereas Hasan's synonymy is a lexical relation, i.e. that of the lexical structure of the language, rather than a textual relation which is interpreted in the context [Hoey91]. Hoey also includes into this category antonyms which have different morphological form (e.g., wet, dry), again stressing the point that this is primarily a textual relation, realised in context, and not lexical in the first place.

Hoey further distinguishes the cases of simple paraphrase as partial and mutual. A paraphrase is partial if only one of the participants of the relation can be substituted by the other, it is mutual when either of them can be used in place of each other [Hoey91].

Complex paraphrase is a relationship requiring three lexical items. It occurs when one item is a simple paraphrase of another item and a complex repetition of the third. According to Hoey the relationship between the second and the third items is a complex paraphrase [Hoey91, p.64]. For example, *publication* and *article* are in a relation of simple paraphrase and *publication* and *publish* are in a relation of complex repetition, hence there is a relation of complex paraphrase between *publish* and *article*.

Recognising superordinate, hyponymic and co-reference relationships as types of repetition, Hoey narrowed down their definition to only those cases where the identity of reference is retained. This is obvious in the case of co-reference, where any two words which point to the same referent are considered to be co-referential. In the cases of superordinate and hyponymical relation, Hoey argued that if identity of reference is not taken into account then, any words denoting a physical inanimate object, for example, must be considered repetition cases of an earlier occurrence of such general word as *thing* [Hoey91, p.69].

2.3.2 Lexical links and lexical chains

A single instance of a cohesive relation between two items was termed a *'tie'* by Halliday and Hasan. Since Halliday and Hasan were studying all types of cohesion, the term 'tie' refers to both grammatical and lexical cohesive relations. Hoey used the term *'link'* to denote an instance of repetition in the broad sense he defined in [Hoey91]. The difference between Hoey's definition of a 'link' and Halliday and Hasan's definition of a 'tie' is twofold: first, 'tie' is a more general notion than 'link', but more importantly, 'tie' implies directionality in a relation. The term 'link' has been used more extensively than 'tie' in other research to denote a lexical cohesive relation [Morris91, Hirst97, Ellman2000]. Since the study of grammatical types of relations is beyond the scope of this project, the term 'link' will be used throughout this work to refer to instances of lexical cohesion.

Lexical cohesion in text is normally realised through sequences of linked words - lexical chains. The term *'chain'* was first introduced by Halliday and Hasan to denote a relation where an element refers to an earlier element, which in turn refers to an even earlier element and so on. The notion of 'chain' in Halliday and Hasan's sense, again like 'tie', includes all types of cohesive relations. To distinguish the notion of a chain composed from lexical cohesive relations the term *'lexical chain'* will be used in this thesis.

Morris and Hirst [Morris91] define lexical chains as sequences of related words, which have distance relations between them. One of the prerequisites for the linked words to be considered units of a chain is that they should co-occur within a certain span.

While Hoey suggested using only information derivable from text to locate links in text, Morris and Hirst used Roget's thesaurus in identifying lexical chains. According to their algorithm two words are connected in a lexical chain if they have one of the following [Morris91]:

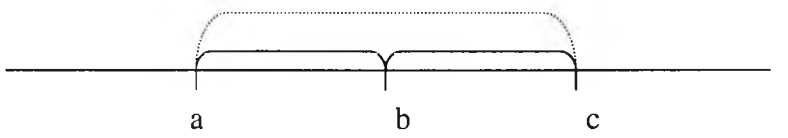
1. Their index entries point to the same thesaurus category or to adjacent categories;
2. The index entry of one points to a thesaurus category that contains the other;
3. The index entry of one contains the other;
4. The index entry of one points to a thesaurus category that in turn contains a pointer to a category pointed to by the index entry of the other;
5. The index entries of each point to thesaurus categories that in turn contain a pointer to the same category.

Morris and Hirst discovered that the first two are the most common types of relationships found in lexical chains, accounting for 90 percent of all relationships between elements of lexical chains built in their experiments.

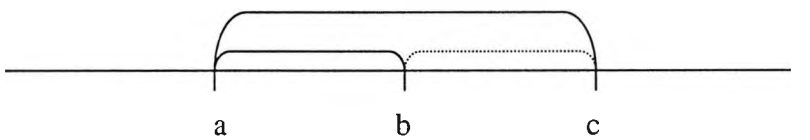
The use of the above criteria for the inclusion of a lexical item into a lexical chain results in the chain elements being connected with a diverse range of semantic relations. This prompted a question whether transitive connections can be considered as chain-forming relations. Morris and Hirst's analysis of sample texts showed that more than one transitive link weakens the relationship between words sharply. Therefore they considered only transitivity of one link for building lexical chains

[Morris91]. Two types of one-level transitive relations were counted when building lexical chains:

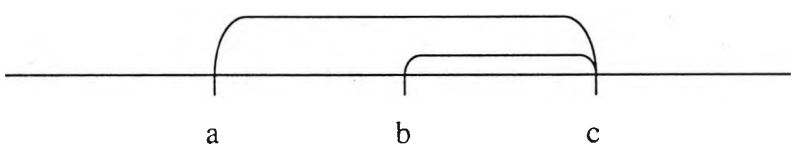
1. If word *a* is related to word *b*, and word *b* is related to word *c*, then word *a* is related to *c*.



2. If word *a* is related to word *b*, and word *a* is related to word *c*, then word *b* is related to *c*.



However, if two words are transitively linked through the third word which occurs later in the text, their relationship is not included into the chain as at the time of interpretation they were not related [Morris91].



In contrast Hoey, identifying only the cases of repetition where the range of semantic relations between words in a chain is less diverse, assumed that if a lexical item is related to one of the preceding elements in the chain, it is treated as being related to all other elements as well [Hoey91].

An important parameter characterising lexical chains, pointed out by Morris and Hirst, is the distance between chain elements. A text segment focusing on a particular subtopic is expected to have high density of related words. Morris and Hirst [Morris91] experimentally discovered that there can be up to two or three sentences between the chain elements. Subtopic development in the text reflects the flow of author's reasoning or the description of a certain pragmatic situation, therefore it is reasonable to expect that the author returns to the discussion of the same thing after digressing from it. Morris and Hirst called the resumption of the previous lexical chain after certain gap as *chain return*. Chain returns are indicators that the author resumes the subtopic entity he/she unfolded before. If the distance between chain elements is four or more sentences (no more than 19), than it is a signal of a chain return [Morris91].

Other parameters that characterise lexical chains and contribute to their strength are:

- Reiteration - the more words are repeated throughout the lexical chain, the stronger it is;

- Density - the more elements the lexical chain links within a certain text unit, the stronger is the chain;
- Length - the larger is the expanse of text that the lexical chain stretches through, the stronger it is [Morris91].

The motivation of Morris and Hirst's [Morris91] research of lexical chains was to see if lexical chains correspond to the intentional structure of discourse (proposed by Grosz and Sidner) and if they can be used to detect it. Experimentally they determined separately lexical chains and the intentional discourse structure. The comparison suggested strong correspondence between the two.

Morris and Hirst however did not automate their algorithm due to the unavailability of the electronic version of Roget's thesaurus. Therefore no large-scale testing of the algorithm was performed.

Later Hirst and St-Onge [Hirst97] attempted to adapt Morris and Hirst's algorithm for determining lexical chains, using WordNet relationships instead of Roget's thesaurus relationships.

Their method consists in attempting to link all the synsets associated with the two words being compared and ruling out those synsets which remain unconnected. They define three kinds of relations between words in the chain which are weighted according to their strength [Hirst97]:

- Extra-strong relation - between a word and its literal repetition;
- Strong relations
    (a) both words occur in the same synset;
    (b) there is a horizontal link between the two synsets associated with two different words;
    (c) there is any kind of link between a synset associated with each word if one word is a compound term that includes the other.
- Medium-strong relation occurs when there is an *allowable path* between two synsets associated with each word. The path must contain 2-5 links between synsets. The weight of such relation depends on the length of the path and the number of path direction changes.

The number of allowable paths between two words in Wordnet is restricted not only by the number of links that connect them, but also by the type of each lexical relation in the path. Since the semantics of the multilinked relationship is composed from the semantics of each of the links, their type and order is important. Hirst and St-Onge specified three rules according to which the multilink paths between words in Wordnet are acceptable for lexical chaining:

- No other relation may precede an upward link;
- No more than one change of direction is allowed;
- There can be a horizontal link between an upward and a downward relationships.

Hirst and St-Onge recognise the limitations of Wordnet, which caused problems such as including the wrong word into the chains or failing to include the word into the chain where it belongs. The major problem is that of a limited set of relations in Wordnet, which links words only by means of the most generic conceptual relations and does not provide for nonsystematic associative relations. Another problem is inconsistency in the semantic distance between words in Wordnet. This is evident when some words are semantically close but are separated by a large number of links in Wordnet and vice versa [Hirst97].

Hearst [Hearst94] in her research of the role lexical chains can play in text segmentation, discovered that locating individual chains in text does not give enough evidence of the topic boundaries. This is true especially for long texts with dense discussions, where it is not uncommon for multiple chains to overlap in the same stretch of text. Therefore it is important to analyse bunches of chains, eliciting information about topic boundaries from the points in text where one bunch of chain ends and another begins.

Interesting evidence about the distribution of link types in different types of texts was gathered by Ellman and Tait [Ellman2000]. Their algorithm for computing lexical chains is based on the algorithms by Morris and Hirst [Morris91] and Hirst and St-Onge [Hirst97]. The link types they compared were the following [Ellman2000]:

1. Two identical words;
2. Two words are members of the same Roget's thesaurus category;
3. Two words are members of the same group of categories in Roget's thesaurus;
4. Two words are linked through one level of thesaurus pointers.

The analysis of the link distribution showed that the most common is the link between two identical words, followed closely by the link between words belonging to the same category. The fourth type of link proved to be rather rare [Ellman2000]. This finding provides support for the methods which rely only on word repetition in the detection of lexical cohesion (cf. [Hoey91]). However the fact that the distribution of the link type 2 follows closely the distribution of the link type 1 means that similarity comparison between words belonging to the same thesaurus category can improve lexical chaining.

In our experimental study of lexical cohesion between query terms (chapter 8) we identify lexical links by detecting only the cases of simple lexical repetition, i.e. the identical stems (see section *8.2.1*). Ellman's findings suggest that our method identifies a large share of all lexical links.

2.3.3 Bonds between sentences and nets of bonds

Hoey pointed out that text cohesion is built not only of links between words, but also of semantic relationships between sentences. If sentences are not related as whole units, even though there are some lexically linked words found in them, they are no more than a disintegrated sequence of sentences sharing lexical context [Hoey91]. Following Winter [in Hoey91], he emphasised that it is important to interpret cohesion by taking into account the sentences where it is realised. For example, two

sentences in text can enter the relation, where the second one exemplifies the statement expressed in the previous sentence. Sentences do not have to be adjacent to be related, and relation can connect several sentences.

A cohesive relation between sentences is termed as *bond* [Hoey91]. Hoey defines bond between sentences as a sufficient number of lexical links between them. The number of lexical links the sentences must have to be bonded is a relative parameter, depending indirectly on the relative length and the lexical density of the sentences [Hoey91, p.92]. An empirical method for estimating a minimum number of links the sentences must have to form a bond is to rely on the proportion of sentence pairs that form bonds in text. If the proportion of sentences linked by any given number of links is too high, then it is important to increase the cut-off point, until the degree of connection is not above average [Hoey91].

It is notable that in Hoey's experiments, only 20% of bonded sentences were adjacent pairs. Analysing non-adjacent sentences, Hoey made and proved two claims about the meaning of bonds. The first claim is that the significance of bonds is greater than simply statistical evidence of the connection by repetition. Bonds between sentences are indicators of semantic relatedness, which is more than a sum of relations between linked words. The second claim is that a large number of bonded sentences are intelligible without recourse to the rest of the text. It is true that being semantically related, some bonded sentences are also coherent and can be interpreted on their own [Hoey91].

Our method of estimating the level of lexical cohesion between query terms was inspired by Hoey's method of detecting bonds between sentences. However, the two approaches have a number of fundamental differences in their aims and methodologies (for more detail see section *8.2.1*).

As already mentioned above, the relation between sentences can be multiple. All bonded sentences in a text or their subsets form what Hoey termed *nets* [Hoey91]. The analysis of nets of bonded sentences can throw light on the role each bonded sentence plays in text. Each sentence can be characterised by the number of preceding and following sentences in text it forms bonds with. Depending on these two sentence coordinates, Hoey distinguished four categories of sentences:

- Marginal sentences;
- Central sentences;
- Topic opening sentences;
- Topic closing sentences.

Sentences which form no bonds with other sentences are termed marginal, for their contents "neither builds lexically upon what has gone before nor provides the lexis for subsequent statements" [Hoey91, p. 105]. It is important, however, to note that although they do not directly contribute to the topic development, they can provide additional information, necessary for its understanding.

Central sentences are those that have exceptionally high number of bonds with other sentences. The minimum number of bonds required to define a sentence as central is relative, what is similar to the relative definition of the minimum number of links for bonding.

Sentences which are related to more following than preceding sentences are topic-opening, and those related to more preceding than following sentences are topic-closing.

An interesting study of nets was made earlier by Skorochod'ko [Skorochod'ko72], who formed a classification of nets, or "semantic networks" in his terminology. Identifying semantic relatedness between sentences through occurrence of related or repeated words, he built the semantic network of the text by representing it as a graph, where the sentences and the relations between them are shown as nodes and arcs respectively. He distinguished four types of nets that represent semantic organisation of text (figure 2.2).

The relatedness between sentences in Skorochod'ko's approach is relative. One of the factors determining strength of the relationship is the number of lexical links which sentences have between them.

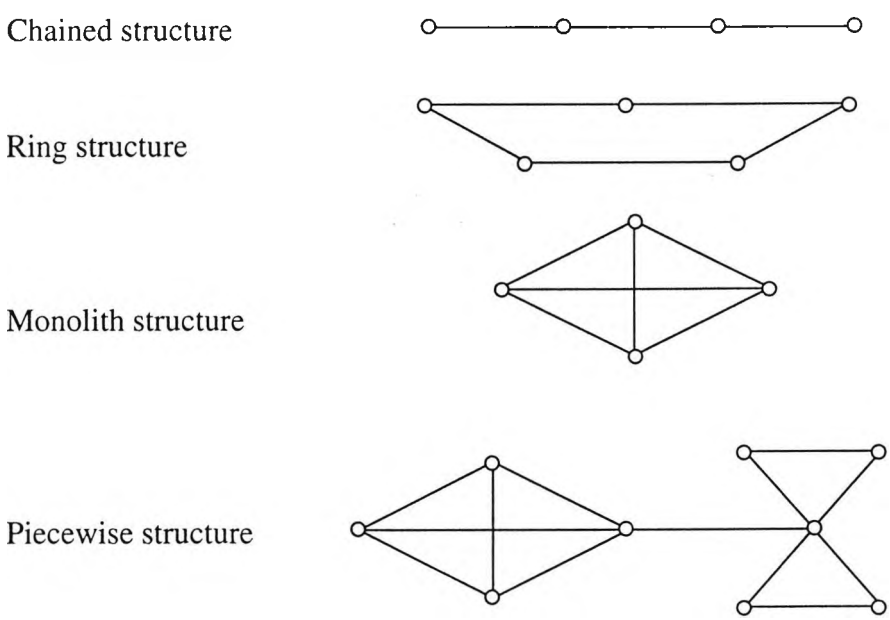


Figure 2.2. Semantic structure types by Skorochod'ko [Skorochod'ko72, p.1180]

In texts with nets of chained and ring type predominantly adjacent sentences are bonded, distance relations being uncommon. In monolith type of nets, each sentence is connected to the majority of others. Piecewise nets consist of groups with densely interconnected sentences, all groups being connected in turn.

The types of semantic networks presented above determine the semantic organisation of the text, in other words provide some information on the topic development in text, whether it is presented through sequential narration like in chain or monolith type, or

through dense discussion with frequent recourse to what was said before or what will follow, like in monolith or piecewise structures.

As can be seen from the research findings described in the above three subsections, cohesion is a text-organising component. By identifying cohesive relations it is possible to uncover (at least with some degree of accuracy) topic boundaries in text and ultimately the overall semantic organisation of text. However, lexical cohesion can also throw light on smaller scale relations existing between text elements. In our third group of study (chapter 8) we explored lexical cohesion between query terms and its association with relevance.

2.4 Collocation

2.4.1 Definitions of collocation

The earliest understanding of collocation, as mentioned in section *2.3.1*, was formed by Firth [Firth58]. Collocation in the Firthian sense is a name given to lexical items which co-occur with more than random probability in text. This is a rather broad understanding of the phenomenon of co-occurring words. Later many linguists attempted to narrow down the meaning of the term collocation or build a completely new vision of what collocation is. Many proposed definitions often reflect the background the linguists come from and are formulated in the light of the subject of their research.

Both Halliday and Hasan [Halliday76] and Hoey [Hoey91] who focused on the study of cohesion emphasised that collocation is a realisation of lexical cohesion in text. As mentioned earlier (section *2.3.1*) Halliday and Hasan gave a rather vague definition of collocation:

> "Here we shall simply group together (...) all lexical cohesion that is not covered by what we have called 'reiteration' - and treat it under the general heading of collocation, or collocational cohesion ..." [Halliday76, p.287]

They emphasise that there must be a lexical-semantic relation between words to be considered collocates. Although they recognise that cohesion between two collocates in text is not so much due to the existence of a lexical-semantic relation, as to the fact that they have a tendency to occur in the same environment, they claim that words co-occur because they are in some kind of lexical-semantic relation. Because they were unable to give a systematic classification of lexical relationships holding in collocations, Hasan later suggested to take out collocation from linguistic terminology until such classification is formed [Hasan84]. It would be more reasonable, perhaps, to admit the variety of lexical-semantic relationships existing between collocates, but to center the definition around the idea of typicality or regularity of their co-occurrence.

Hoey also interpreted the notion of collocation in the light of cohesion studies. He asserts that "collocation is the direct result of bonding (and of similar relations

between sentences in and across texts)." [Hoey91, p.219]. Unlike Halliday and Hasan he avoided including the undefined range of lexical-semantic relations into his understanding of collocation, focusing solely on the recurrence of words' appearance together in text: "Words only have collocations because they appear in recurring combinations in text ..." [Hoey91, p.219].

The recognition of collocation as the mechanism for creating text cohesion by [Hoey91] and [Halliday76] means that collocation in their sense is a topic-forming relationship, whose elements - collocates - can be separated by some stretches of text. Their co-occurrence is due not to grammatical patterns and restrictions, but to belonging to the same topic. They indeed exhibit certain semantic relations which are reflected in their usage in the same thematic environment, in the same topic. Below another widely accepted approach to collocation, based on a rather different stance, is given.

Some linguists understand by collocation a more than random use of words next to each other either due to referential association of their meanings, or due to habitual or customary patterns which evolved in the language in use [Robins89, Palmer81]. Robins distinguished collocations which are related to the situational and referential meaning of the collocates, e.g. *bright day*, from collocations less tightly connected with the reference, but whose use is more habitual, e.g. *white coffee*, where the word 'white' does not refer to the colour of the referent [Robins89, p.65]. Palmer drew a distinction between these types of collocates based on the idea of collocational restriction. Collocations can have different degrees of restriction. He distinguished three types of collocational restriction [Palmer81, p.79]:

1. Collocational restriction based entirely on the meanings of the words (e.g., *bright day*, *deep sea*);
2. Collocational restriction based on a range of words. A word can collocate with a range of words that have some common semantic feature (e.g., *pack of wolves*, *pack of hounds*, *pride of lions* but not *pack of lions* or *pride of wolves*);
3. Collocational restriction wholly due to habitual and customary use of the words (e.g., *rancid butter, rancid bacon*, *sour milk* but not *rancid milk* or *sour butter*).

It is possible to explain collocations of the first type entirely through the meaning of their elements. Collocations of the second and third types can or cannot be explained semantically through the meanings of their elements. For example, Palmer suggested that restricted collocations like 'white wine' and 'white coffee' could possibly be semantically explained by saying that 'white' means "with the lightest of the normal colours associated with the entity" [Palmer81, p.77]. But he admitted that it is impossible to draw a clear line between those collocations which can be semantically explained and those which cannot.

Palmer [Palmer81] and Robins [Robins89] recognised idioms as a special kind of collocations, whose meaning is not deducible from the meanings of their components and often has nothing to do with them semantically (e.g., *red herring*). Palmer also argued that the third type of restricted collocations can be referred to as partial idioms, where one word is used in its usual meaning (*'coffee'* in *white coffee*), but the other acquires a meaning specific to the collocation (*'white'*) [Palmer81].

Some scientists narrowed down the meaning of collocation to refer only to restricted type of collocations, whose meaning cannot be completely derived from the meaning of their elements. Manning and Schuetze [Manning99] defined collocation as grammatically bound elements occurring in a certain order which are characterised by limited compositionality, by which they mean the impossibility of deriving the meaning of the total from the meanings of its parts. They admit the existence of syntagmatic word associations across larger expanses of text, bound topically, not grammatically, but they suggest calling such associations 'co-occurrences' and to reserve the term 'collocation' only for grammatically bound combinations.

Manning and Schuetze highlighted the following categories among the collocations in the narrow sense as grammatically bound constructions [Manning99, p. 174]:

1. Light verbs (e.g., *make a decision, do a favour*);
2. Phrasal verbs (e.g., *to check in, to cut down*);
3. Proper names;
4. Terminological expressions (e.g., *fission reactor fuel*)

Defining collocation through the notion of limited compositionality, they note that collocations can have different degrees of invariability. Some collocations allow certain substitutions of its elements (this co-relates with Palmer's second and third type of collocational restriction), but some not - proper names, idioms, some terminological expressions.

In the light of the approaches described above in this thesis it is recognised that there are two major types of collocation:

1. Collocation due to lexical-grammatical or habitual restrictions. These restrictions limit the choice of words that can be used in the same grammatical structures with the word in question.
2. Collocation due to a typical occurrence of a word in a certain thematic environment. Two words hold a certain lexical-semantic relation, i.e. their meanings are close semantically, therefore they tend to occur in the same topics in texts.

Lexical-grammatical/habitual restrictions and lexical-semantic relations are the linguistic factors that cause the phenomenon of word collocation. The nature of these linguistic factors is quite complex and the development of a general method to identify collocations in text through the analysis of these factors is not an easy task, as evident from the failed attempt to impose strict classification on the phenomenon of collocation by [Halliday76] (see earlier section *2.3.1*). Instead it is possible to isolate this phenomenon in text empirically, not by looking for what causes it, but by identifying what characterises it most typically - namely, more than random probability of its occurrence.

The empirical approaches to identification of collocates from text were developed within the branch of linguistics based entirely on the analysis of empirical data - corpus linguistics. Corpus linguistics which developed in the last 30 years, emerging from the work of such linguists as Kucera and Francis [Kucera67] and Sinclair [Sinclair74], concerns itself with extrapolating general conclusions about various linguistic phenomena through the empirical analysis of corpora. Corpora are large

collections of texts, sampled to be maximally representative of a general/specific type of language in use [Wilson96]. Empirical identification of collocations is based on using various statistical parameters of lexical units elicited from the corpora. The statistical methods of finding collocations in text used in this project will be described in chapter 5.

Sinclair and Jones [Sinclair74] were the first to attempt corpus-based analysis of collocations. Although the size of the corpus they analysed was relatively small - 135000 words in spoken corpus and 12000 words in written corpus - compared to modern corpora typically several million words long, their work is of significant value as it defined the framework for corpus-based collocation analysis and yielded some information about patterns of collocation behaviour in text.

The major notions of collocation analysis introduced in [Sinclair74] and systemised further in [Sinclair91] are those of 'node', 'collocate', 'significant collocate' and 'span'. A 'node' is defined as "an item whose total pattern of co-occurrence with other words is under examination" [Sinclair74, p. 22]. While a 'collocate' is "any item which appears with the node within a specified environment." [ibid.].

Sinclair and Jones use the term 'collocation' to refer to any co-occurrence of two words within certain environment in text. To refer to those collocations which occur with more than random probability in text they use the term 'significant collocation'. "'Significant' collocation is regular collocation between items, such that they co-occur more often than their respective frequencies and the length of text in which they appear would predict." [Sinclair74, p. 25].

The term 'span' is used to refer to the stretch of text around the node within which words are considered to be its collocates [Sinclair74, p.27]

The above terms have been adopted in later corpus-based collocation research and will be also used in these senses throughout this thesis.

2.4.2 Patterns of collocation in text

This section will cover some patterns discovered by corpus linguists using statistical analysis of collocation and word distribution in text. The probability of occurrence of each word in text is affected by the presence of other words around it. More specifically, it is possible to say that due to the effect of collocation the appearance of each word in text influences to a varying degree the statistics of word occurrences around it [Beeferman97]. The major questions in this context are the following:

1. How far the influence of each word extends in text?
2. Is the influence of a word consistent throughout the word's environment?

The first question, concerning the size of the environment throughout which a word exerts its influence, remains arguable. Most researchers so far have made empirical decisions about the span of a word's environment, based mainly on the practical considerations of the particular research task. The span is measured either in syntactic units, such as phrases, sentences, paragraphs or even entire texts [Harper78], or by the number of words to the left and right of the node, for example, 4 words [Sinclair74], 5

words [Church90], 400 words [Beeferman97], 4,10, 50 words [Edmonds97]. Such lack of uniformity can be partly attributed to different interpretations of the concept *the environment of a word.* As already mentioned in the previous section of the thesis we distinguish two types of collocations: (1) those subject to lexical-grammatical restrictions and (2) collocations influenced by lexical-semantic or topical relations. The first type of collocations occur within a *short-span* environment, whereas the second type - within a *long-span* environment. Our experimental study of statistical parameters of the environment of a word indicated a marked difference between word distributions in the immediate span of the node (4-5 words either sides) and the bigger span, which correlates with the results of a similar experiment but on a smaller span of 10 words by [Sinclair74]. Our experiment consisted in counting the number of observed token types for each slot within 50 words both sides of the node. We analysed windows around all instances of 40 randomly selected indexing terms in the stemmed[1] TREC AP collection. The results showed that the variability of collocates within the immediate span was much lower than further away from the node. Immediate collocates maintained stable joint co-occurrence with the node, indicative of lexical-grammatical factors (e.g. phrasal verbs, compound terms). Term variability further away from the node was distinctly higher[2] (figure 2.3).

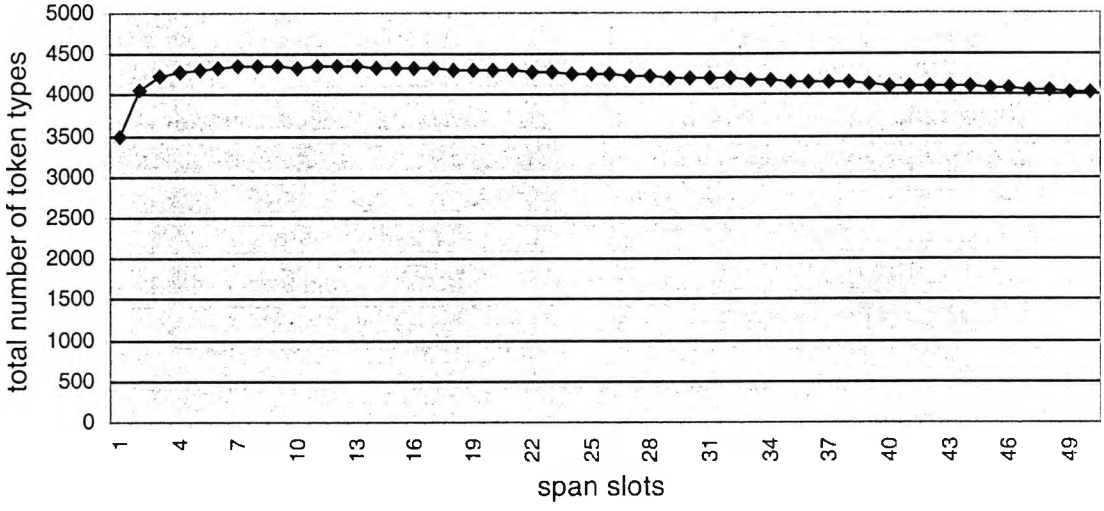


Figure 2.3. Distance distribution of token types in the windows of 50 words either sides of the nodes.

This experiment allowed us to identify only the boundary between the short-span and the long-span influences of the word. Beeferman et al. [Beeferman97] in their study of the effect distance has on triggering and prediction capabilities of collocates (or "trigger-words" in their terminology) obtained empirical evidence about the long-distance influence patterns of words. They analysed distance distributions in two groups of collocates: self-collocates - repetition of the same lexeme, e.g. (*gene, gene*), and non-self collocates, where collocate 1 and collocate 2 are different lexemes, e.g. (*gene, chromosome*). The technique they used consisted in calculating for each

[1] Okapi parsing algorithm was used to stem words and filter out stop words

[2] The gradual fall of the curve is due to the fact that windows are frequently truncated by hitting the document boundary.

distance k the probability that two trigger words are separated by exactly $k \pm 2$ words. The results showed that a word's influence on statistical distribution of words around it stretches as far as several hundred words, levelling off by about 400 words.

The main objective of Beeferman's research, however, was to prove the hypothesis that statistical parameters within a word's environment are not static, but are directly dependent on the distance away from the node. This leads us to the second question raised earlier about the consistency of the word's prediction capabilities. Beeferman's major discovery was that the influence of the word decays exponentially. Another observation was characteristic of the group of self-collocates and demonstrated that the word-triggering effect increases from the distance of 1 word to 25 words, but then falls gradually. They termed this phenomenon *the lexical exclusion principle*, due to the fact that lexical-syntactic restrictions prevent self-collocates or repetitions to appear within short distances in text [Beeferman97].

Mason [Mason97] discovered interesting patterns of the word's influence on its environment which he called *lexical gravity*. His corpus-based study of collocation proved among others the claims that (1) each word has an individual pattern of influence and (2) lexical and grammatical items have different patterns of influence on their environment. To test these claims he measured the variability of words in each slot for arbitrary distances using type-token ratio (TTR). Experiments showed that lexical gravity is not symmetrical, i.e. a word can have different degree of influence on its preceding and following neighbours and, most importantly, that each word has different patterns of influence on its immediate environment. These findings concern only short-span collocations, and can be theoretically explained by the fact that each word imposes different degrees of lexical-grammatical restrictions on its short-span environment, where, especially in restricted type of collocations, some words tend to occupy fixed slots relative to the node. Whereas, in the long-span environment of the node, words related to it semantically can occupy any slot.

While lexical units have a generally similar type of pattern, grammatical items have strikingly different patterns. Another feature characteristic of some grammatical items is negative gravity, i.e. higher than average variability. This is due to the fact that grammatical items form patterns with grammatical classes and not individual words. Articles, for example, are followed by any noun or noun phrase, i.e. they restrict the selection of the following word only grammatically, but not lexically or semantically.

Not all words in the environment of the node are influenced or predicted by its occurrence. The environment of a word consists of several types of units: high-frequency words (e.g. prepositions, articles, auxiliary verbs), which co-occur frequently with any word; one-off collocates, with low joint frequency and significant collocates with more than random probability of co-occurrence with the node. In the long-span environment, significant collocates are those that are related to the node semantically, their non-random co-occurrence being due to the fact that both belong to the same or related semantic fields and consequently tend to be used in the same topics. The number of such topic-discriminating words is relatively small in comparison to the total number of words in a stretch of text describing a topic [Renouf93]. Renouf argued that the majority of words in text are topic-independent or related to more than one topic. She distinguished eight types of such words: very common words, discourse organising words, homonyms, semi-technical words, words

with several technical senses, metaphors, typographically ambiguous words [Renouf93]. Genuine topically-dependent words, as opposed to high-frequency or context-independent words, are expected to have significant amount of association with the node. Significant collocates which can predict each other's occurrence are distinguished from chance word pairs by using various statistical measures of association. The statistical methods used in this project will be detailed in chapter 5. Section *3.3 Approaches to collocation in IR* will also cover other research of collocations in the IR context.

3. Natural language text in IR context

3.1 Document and query representation

The aim of the traditional ad-hoc information retrieval task is to deliver documents relevant to users' queries, representing their information needs. Both documents and queries are natural language texts, which, as demonstrated in the previous chapter, are complex entities, realised through several levels of syntagmatically combined linguistic elements and whose meaning as a whole is more than the direct combination of these elements' meanings. Therefore the relationship between the underlying semantic meaning of the text and the linguistic elements at various levels explicating it cannot be easily specified. This is due to the inherent flexibility and richness of linguistic means of expression, making it possible to express the same idea in different ways by exploiting different levels of linguistic expression. The linguistic ambiguity is twofold: on the one hand the same idea or concept can be described through different means of expression, and on the other hand – the same linguistic element or combination of elements can refer to different ideas or concepts.

The major operation in IR is matching the query with the document. Due to the linguistic ambiguity, and the complexity of the relationship between the plane of expression and the plane of contents of text, this task is not easy to perform. The characteristics of language are not the only factors to make the IR task non-trivial; others which involve extra-linguistic cognitive factors are: the accurateness with which the user's information need is expressed linguistically, the vagueness of the information need itself and the amount of knowledge the user has about his information need. But these factors are subjects of a separate research field and are outside the scope of this thesis.

Because of the above characteristics of NL texts, direct comparison of the document and the query, and identification of the degree of their match on a fine scale, is impossible. This task requires a reduction of the text dimensionality, and a normalisation of the means of linguistic expression, in both the collection of documents and the incoming queries. Therefore in IR the matching is performed between *document* and *query representations*. The document and query representations consist of some attributes and possibly relations between these attributes which ideally should characterise texts unambiguously, so that uniform comparisons between them become possible. Document/query representations must accurately reflect the contents of the text, be sufficiently discriminating, but at the same time they must be normalising and summarising. The decisions of what attributes should constitute representations, what relationships they should hold, and where these attributes should be taken from are fundamental in IR. The following summary non-exhaustively illustrates the broad spectrum of approaches to building document/query representations:

1. What are the attributes characterising documents/queries?
   - Single words;
   - Phrases (linguistically or statistically motivated);

- Semantic concepts (assumptions about what a concept is are model-specific);
- Artificial formalisms.

2. Where are the attributes taken from?
   - Derived from a document/query;
   - Assigned to it from a controlled language structure.
3. What assumption about the relationships between the attributes are made?
   - None ("bag of words" approaches);
   - Syntagmatic (elicited from the documents statistically or using natural language processing - NLP);
   - Paradigmatic (taken from knowledge bases, e.g. thesauri).

An additional factor which must be taken into account when building document representations – indexing – is whether the representation should be created as static at document file time, or tuned to a particular request at search time. The former belongs to a *pre-coordinate* approach to indexing, wherein, for example, phrases are entered into document representation at file time, and the query must contain exactly the same phrases as the document representation to match it. The latter is of *post-coordinate* type, where the phrases are not anticipated at file time, but terms are coordinated into phrases at search time depending on the phrases used in the query. Document representations in pre-coordinate systems are characterised by high specificity, which can be detrimental to performance [Paijmans93]. In modern IR the emphasis has shifted from pre-coordinate to post-coordinate paradigm.

At the most general level there are two main approaches to creating document representations: to assign the indexing units – single terms or phrases – to the document representation from existing knowledge bases, such as thesauri or semantic nets, or to derive the indexing units from the text of the document itself. The former type of indexing is known as *controlled-language* or *assigned indexing*, the latter – *natural-language*, *free-text* or *derived indexing*. Two following sub-sections will give a brief overview of these two approaches and their implications.

3.1.1 Controlled language indexing

Controlled language indexing consists in analysing the contents of the document (manually or automatically) and assigning terms to the document representation from external knowledge bases such as thesauri or lexical nets. The knowledge bases contain "controlled vocabulary" – a limited set of terms standardised lexically and syntactically and usually holding a set of relationships.

There are different kinds of knowledge bases, whose use in IR at present extends far beyond controlled language indexing. Traditional knowledge bases are thesauri, which were first used predominantly as a controlled language structure, but then were also applied to other IR tasks like query expansion. Thesauri were defined by Guinchat et al. as "tools consisting of a controlled set of terms linked by hierarchical or associative relations, which mark any needed equivalence relations (synonyms) with terms from the natural language and concentrate on a particular area of knowledge" [in Chowdhury99, p.125]. Thus two major functions of a thesaurus are: first, to control equivalence relations, i.e. to suggest a single preferred term for a set of

synonymically related terms, and second, to link related terms together via a set of relationships. Three major types of relations in manually engineered thesauri are:

- Equivalence relations;
- Hierarchical relations;
- Associative relations.

Most manually engineered thesauri like Inspec or MESH (Medical Subject Headings) contain equivalence and hierarchical relations, which in turn can be generic relationships (hyponymy) and/or whole-part relationships (meronymy). Association between terms can be specified either through a single associative relationship or broken down into a set of more specific conceptual relations.

All relationships holding between terms in manually engineered thesauri are *paradigmatic* (for definitions of *paradigmatic and syntagmatic relations* in linguistics see section *2.2.2*). In contrast, automatically constructed thesauri feature *syntagmatic* relations between terms, since terms and their relations are derived from texts.

Other notable types of knowledge bases are lexical nets, the most famous of which is WordNet [Miller90]. WordNet is organised by paradigmatic lexical and semantic relations. Words are grouped by the synonymical relation into *synsets*. Each synset is claimed to represent a concept, by comprising all synonymical terms that express it. Polysemous words can belong to more than one synset depending on the concept each of their meanings represents. Synsets have unique identification codes and are linked by a set of relations such as hyponymy, meronymy and antonymy.

The main motivation behind using controlled vocabulary in indexing was to reduce the negative effect caused by language flexibility, exhaustivity and ambiguity. Controlled language indexing claims to bring synonyms and semantically related terms together, normalising or reducing the explication of a concept/idea to a single standard unit of expression - phrase or term [Foskett96, Lancaster79]. Foskett argues that controlled-language indexing is essentially a concept indexing opposed to non-normalised term indexing, which he claims is weakened by the fact that the same idea can be expressed differently across the collection, in the same document or in the user's query [Foskett96].

However the very aim the controlled-language is trying to achieve – remove the negative effects of natural language – arguably has many disadvantages in the IR task of matching document and query representations. The main problem with the controlled language approach is its inherent limitation of the indexing vocabulary. It is not specific enough to reflect many semantic aspects of the document contents and equally the user's information need. Moreover it erases the fine distinctions between the natural language texts which is detrimental to precision and is not suitable for highly specific searches. Similarly a query formulated in very specific terms, if translated to controlled vocabulary, will lose its specificity.

The problem with insufficient specificity of controlled languages became even more serious with the arrival of large full-text collections, where the dimensionality of the document contents increased dramatically in comparison with abstracts or bibliographic databases. Moreover long multi-topic documents presented an even

bigger challenge to the indexing task. These factors contributed to a shift of focus in IR from controlled vocabulary to natural-language or free-text indexing.

3.1.2 Natural-language indexing

Deriving indexing units from the natural language texts of the documents has several major advantages. The richness of the means of expression in the natural language suggests that representations can be very specific, reflecting the subtleties of the document contents. Similarly the user can formulate his/her information need at any level of specificity.

The effectiveness of controlled language indexing is only as good as the applied knowledge base, its coverage, accuracy of representation of the concept space and currency. One potential problem is that any document which contains a new concept/idea not recorded in the knowledge base, will be incorrectly represented in the IR system. Another hazard is that the assignment of controlled vocabulary terms to the document relies on the accurate interpretation of the document contents. Natural-language indexing overcomes the first problem by relying only on the contents of the document in its indexing, therefore it makes no difference in building document representations whether the document conveys new previously uncovered knowledge or deals with known concepts. The second problem is not unique to controlled vocabulary approaches. In natural-language approaches inaccurate interpretation of the text may bring into the document representation, for example, incorrectly coordinated complex indexing units and wrong associations between terms. This is less of a problem in post-coordinate type of systems where binding of terms into more complex units is under the control of the request formulation. This not only reduces the problem of false coordination, but also tunes representation to the user's need. Such flexibility of matching built in post-coordinate natural-language indexing cannot be offered by controlled-language systems.

The obvious disadvantages of natural-language indexing are no or limited control of lexical and syntactical ambiguity. This however is partially compensated for by the redundancy of natural-language indexing. Natural language is inherently redundant, i.e. any topic in a text is expressed by a large number of linguistic units (words, phrases) many of which are taken from the same semantic domain, and thus potentially have the ability to disambiguate each other through their collocation in the same stretch of text. When a user's query, also expressed in natural language, contains several query terms (or phrases) there is a chance for automatic disambiguation to take place in cases where documents match on two/more query terms, in particular if they have these words as collocates within a certain window.

Extensive research of various indexing approaches carried out in the last thirty years showed that indexing the documents by deriving single terms occurring in them yields results not worse than controlled-language indexing by terms or phrases [Lancaster79, Lewis96]. Currently there is a strong tendency to combine controlled-vocabulary with natural-language indexing. One way in which the two approaches can complement each other was pointed out by Lancaster: including controlled vocabulary units into document representation provides for generic search and sets the context for the

interpretation of natural-language units, the inclusion of which, in turn, brings the desired specificity in representing the document contents [Lancaster79].

3.2 Statistical approach vs. natural language processing (NLP) approach to IR

3.2.1 Limitations and benefits of statistical methods and NLP in IR

Since the shift from assigned to derived indexing in IR, statistical word-based approaches to indexing and matching tasks have formed dominant IR models which remain consistently effective to the present day. Statistical models rank documents on the basis of the estimated probability of their relevance to the query (probabilistic models), or by their similarity to the query (vector-space models). The general characteristic of statistical approaches is that they represent both documents and queries as "bags of words" – sets of single terms derived from documents/queries. The terms are assigned some statistical characteristics - weights, which characterise their ability to describe and distinguish the contents of the document. The greater the term's content-discriminating ability, the higher will be its weight, hence the higher will be its influence on the document score. The document-query matching consists in counting the number of matching terms, combining their weights, and modulating them by other parameters, for example relevance feedback data. In probabilistic and vector-space approaches to IR the result of the document-query matching process is the assignment of scores to documents, which are used in their ranking. The retrieval principles of the probabilistic model used in this project will be covered in more detail in chapter 4.

Within the currently dominant post-coordinate paradigm in IR, the major principle of statistical models is that the representation of the document is tuned to the particular query at query time. Lewis and Sparck Jones stress three main benefits that statistical word-based approach brings to IR [Lewis96]:

- Late binding. Complex concepts are not composed at indexing stage, but are controlled by the user during querying;
- Redundancy. Indexing terms are taken from the text of the documents, reflecting the variability and abundance of the natural language lexicon;
- Derived representations. Representations built entirely from the lexical means of the document are likely to preserve differences and similarities between texts.

Statistical models so far have demonstrated the most steadily effective performance, tested in a range of environments, under various system parameters and using different evaluation scenarios. However, the typical effectiveness achieved is still rather low (in the range of 30% to 60% recall or precision), therefore leaving much scope for improvement [Lewis96].

The main philosophical problem with statistical word-based IR is that it deals only with the surface plane of expression of language at a symbolic level. Smeaton pointed four assumptions implicit in statistical word-based IR [Smeaton97]:

- Users can find precise explication for their information need;
- Users' information need is static throughout the search;
- Authors of documents can express their concepts and ideas precisely and accurately;
- Users know what terminology is used by authors to express concepts and ideas in the documents.

These assumptions are clearly not justified in reality, which prompts the conclusion that the representation of documents/queries as sets of independent terms by statistical IR models is not an entirely adequate solution for the primary IR task of satisfying users' information needs [Smeaton97].

It is true that natural language text is more than a collection of words, as demonstrated in the previous chapter of the thesis. The complex nature of text and the fact that the major objective in IR is to compare two natural language texts – document and query – gave rise to the idea that what is needed is a deeper analysis of textual means of expression, which lead to the plane of contents. The need to investigate ways for deeper text analysis in IR became widely acknowledged. The discipline that has been developing methodologies for linguistically motivated text analysis for several decades is Natural Language Processing (NLP). NLP has emerged and grown into a separate discipline from such activities as Machine Translation (MT) and Natural Language (NL) interfaces. These applications require deep semantic interpretation of text and unambiguous, accurate and complete representation of its conceptual level. Therefore the ultimate goal of NLP research was the development of methods for achieving conceptual representation of text.

When NLP techniques as developed for tasks like MT were borrowed by IR, a very relevant philosophical question was asked by some IR researchers: is full conceptual text representation really needed for IR [Smeaton97, Sparck Jones99]. An interesting analysis of IR and NLP compatibility was given by Smeaton [Smeaton95, Smeaton97]. He characterised IR and NLP as activities with rather different characteristics. IR has a high degree of imprecision and vagueness associated with it: the search is performed in the conditions of a limited knowledge about the users and their information needs: the type of the information need, its motivation and domain it belongs to, the users' background knowledge, their knowledge of the collection, and changes in their information need throughout the search. IR has a high degree of noise tolerance in document indexing and matching processes, in the sense that with much less than 100% recall and precision operating IR systems will still satisfy users' information needs. This is because the ultimate relevance judgement is always made by the user, who can tolerate the imprecision of the system.

On the contrary, the original tasks NLP was applied for have no tolerance of errors: in machine translation, for example, even one incorrectly interpreted word or syntactic construction can lead to an inaccurate perception of the output text by the reader.

The problems NLP faces in its original spheres of application and in IR are quite different. What is a major challenge for NLP – a complete unambiguous understanding of texts – is simply irrelevant for IR [Smeaton97, Sparck Jones99]. Smeaton argues that IR need not concern itself with meaning of documents at all. The complexity of its task is quite limited: it only needs to distinguish one document from another in the context of a given query [Smeaton97]. This is obviously a much simpler task than general context-free interpretation of contents.

Smeaton argues that the fundamental difference between NLP and IR in handling imprecision will be a major obstacle in their successful integration. But another approach to processing natural language text, which, according to Smeaton, has more in common with IR – corpus linguistics – possibly holds a bigger potential for successful integration into IR. Indeed, corpus linguistics (overviewed in part *2.4.1 Definitions of collocation*) is based on statistical and probabilistic approaches to language processing, which suggests that there is some related methodological ground between corpus linguistics methods and statistical approaches to IR.

There is a growing interest in applying corpus linguistics methods to IR (see part *3.3 Approaches to collocation in IR*). However, the dominant trend in current linguistic approaches to IR is still NLP, borrowed from its traditional spheres of application - MT and NL interfaces [Smeaton95].

The specificity of environment that IR operates in imposes certain limitations on successful application of full-scale NLP analysis. Many NLP techniques, in particular those operating at the semantic level of text analysis, rely heavily on knowledge bases. Modern IR often deals with large-scale domain-independent collections and limited knowledge about the domain of the query. Either manual or automatic construction of domain-independent knowledge bases for complete semantic text analysis is not a realistic response to these problems at present.

The application of NLP techniques to domain-specific small-scale IR has demonstrated some high precision results. For example systems like SCISOR [Jacobs and Rau in Smeaton95] and FERRET [Mauldin in Smeaton95] perform domain-specific searches. Successful operation of NLP techniques in such systems requires large domain-specific knowledge bases.

In domain-independent systems such complex NLP knowledge-based approaches are not feasible. What works well in domain-independent large-scale IR are low-level NLP techniques, rather than full-scale NLP text analysis. Examples are the CLARIT system [Evans *et al.* in Smeaton95] and Phrasefinder, an extension to INQUERY [Jing94], both of which use low-level NLP techniques to pre-process the corpus or the sample of the corpus to identify phrases. After statistical normalisation these are recorded as a collection-specific vocabulary (CLARIT) or a term-term association thesaurus (Phrasefinder) and used for document/query representation.

Smeaton argued that all successful NLP methods in IR (like phrase extraction, part-of-speech tagging, proper noun recognisers, techniques based on machine readable dictionaries, domain-dependent or domain-independent thesauri) have in common the fact that they use NLP as a black box tool to produce more complex means of document/query representation than single terms. Such means of representation can

be phrases, word senses and co-ordinated words. The only change NLP brought forth is that it has "replaced the bag of words/stems with a bag of word senses/phrases/co-ordinated terms as representation, albeit for an improvement in retrieval effectiveness." [Smeaton95]

To summarise the main points made above, presently there are three broad avenues that have been or are being considered for exploring in linguistically motivated IR:

- *Full-scale NLP-based semantic-level analysis of texts*; first, possibly not needed at all, and, secondly, at present prohibitively expensive and resource-demanding to implement, due to the requirement of large domain independent knowledge bases for successful operation.

- *Low-level NLP techniques*; with a more modest aim, for example extraction of more complex units than single words, such as phrases, demonstrating some successful applications in IR.

- *Corpus linguistics methods*; in principle having much in common with statistical IR. And as will be demonstrated later in section *3.3* there is some experimental evidence that they can be a viable alternative to knowledge-based NLP techniques and can be integrated successfully into statistical IR models.

Shallow language processing techniques are, therefore, more viable and suitable for combining with IR processes, than full-scale in-depth NLP. However doubts have been expressed as to whether low-level language techniques can offer significantly more than what is already offered by statistical IR methods. For instance, the word sense disambiguation that NLP can support is already implicitly present in the document/query matching in statistical IR (see section *3.1.2*). Another example – query expansion with paradigmatically related words from knowledge bases – is achieved to some extent in relevance feedback, where the terms taken from the document and added to the query can be in paradigmatic relations with the query terms, since many words in the same context tend to belong to the same domain [Lewis96]. Although in long multi-topic documents the chances for relevance feedback to return words paradigmatically related to the query term are much less than in abstracts or short documents.

The IR task which has received most attention from NLP is the process of building document/query representations. The application of NLP techniques to index term identification and representation formation is known as *linguistically-motivated indexing (LMI)*. It is contrasted to *non-linguistic indexing (NLI)*, based on statistical methods. The main thrust of LMI is to identify multi-word units and characterise their internal structure [Sparck Jones99]. The next section will discuss statistical and linguistically-motivated approaches to multi-word index terms (phrases) and also their value in document representation as compared to single terms.

3.2.2 Phrases. Statistical and linguistically-motivated approaches.

Claims that automatically derived complex terms are better contents discriminators than simple terms have been made since the beginning of research on automated IR in the 60s. Simple statistical co-occurrence based techniques for identification of compound terms have always been rivalled by NLP-based techniques. The main considerations in favour of NLP were: (1) it may have better tools to uncover meaningful linguistic phrases and (2) it can capture the relationships between words.

To avoid terminological confusion, following Sparck Jones, *complex terms* will be used to denote compound terms defined by LMI and *joined terms* – for statistically defined phrases using NLI [Sparck Jones99].

Joined terms are short-span collocations extracted from text using different modulations of their frequency parameters. Complex terms are identified using a variety of NLP methods ranging from low-level techniques such as part-of-speech tagging, aimed at identifying word-sequences of a certain syntactic pattern like adjective + noun, to more complex methods like extended N-grams and syntactic parsing, attempting to discover uniform semantic units underlying various forms of expression.

At the early stages the motivation for research on automatic phrase generation came from the determination to emulate human indexing. The belief was that complex normalising descriptions of the kind assigned to documents by human indexers are more useful than simple terms. One of the early experiments on phrase indexing was carried out by Bely [Bely et al. 1970 in Sparck Jones99], who used very elaborate NLP techniques to identify instantiations of thesaurus concepts and their semantic relationships in documents. Despite the fact that no retrieval evaluation was conducted, the research suggested that the relational structure of the descriptions was not flexible enough for sufficient matching. Another historically important piece of research was undertaken by Salton [Salton 1968 in Sparck Jones99], whose technique consisted in identification of thesaurus terms in text supported by syntactic analysis. The comparison of performance results for syntactic phrases and for statistical phrases, defined as within-sentence co-occurrences of thesaurus descriptor constituents, showed that there is no performance improvement in using syntactical phrases over simple statistical phrases.

One of the most comprehensive evaluations of phrases was undertaken by Fagan [Fagan 1987,1989 in Sparck Jones99]. He analysed the results of his own experiments, relating them to the previous works on phrases, thus drawing rather large-scale conclusions. The main focus of his experiments was systematic evaluation of joined terms under different parameter settings like distance between their constituents and their frequency values. The evaluation results showed that performance for joined terms was in general better than for simple terms. He then compared performance for joined terms with performance for complex terms, which he obtained using syntactic parsing, stemming and normalisation to head-modifier pairs. The evaluation showed that NLP-based complex terms gave results similar to or worse than statistically joined terms. When he analysed earlier work taking into account his findings, he concluded that the same pattern: joined terms $\geq$ complex terms $\geq$ simple terms, was evident in all the experiments.

He concluded that complex terms gave poor performance because queries and documents did not share exactly the same phrases. For this reason complex terms simply did not contribute much to the matching scores. Among the reasons for the systems' inability to match documents and queries by complex phrases Fagan pointed out the low collection frequency of the best compounds and the fact that the documents involved were of abstract length. However even in full-text setting the problem of insufficient matching by complex terms remained. Strzalkowski et al. pointed to another main reason for this, namely, the limited amount of information about the user's information need conveyed by the queries [Strzalkowski99(a)].

Croft et al. [Croft et al. 1991 in Sparck Jones99] again experimentally confirmed that syntactic phrases perform no better than statistical co-occurrences, even with relaxed proximity constraints to document-size windows. They also pointed out the potential usefulness of smaller window sizes for long documents. It should be stressed at this point that large collocation windows capture topical relations between words, therefore in the strict sense collocates separated by a distance greater than their immediate lexical-syntactic environment cannot be referred to as phrases or joined terms. The relations that bind long-distance collocates are topic-motivated lexical-semantic relations which are different from lexical-syntactic relations that spread only within a short-span distance around the node and hold together various lexical-syntactic constructions, compound terms among them (see section *2.4.1*).

The difficulty in analysing the efficiency of specific LMI or NLI methods for phrase-based indexing is attributed by many researchers [Sparck Jones84, Fagan89, Lewis91] to the fact that the range of environment variables is very large and each system combines a range of indexing and matching devices, making it difficult to evaluate the effect a specific LMI or NLI technique has on performance. It is also not yet entirely clear how different system devices interact and how phrase-based indexing can be better combined with other devices such as weighting, relevance feedback, query modification.

For instance the decision whether a document is indexed by phrases at file time or at search time has a direct impact on the relevance feedback process. If documents are indexed at file time by single terms with the final phrase-based document representation being built at search time with respect to a particular query, then relevance feedback will only be able to add single terms [Sparck Jones99].

Another rather complex issue of phrase-based techniques is related to weighting. Phrases like single terms vary in their contents-discriminating ability; however weighting formulas for single terms seem to be unsuitable for phrases [Strzalkowski99(a)]. It is true that a weighting model is one of the central devices of any IR system that uses weighting and therefore it is important that with the introduction of such new devices as phrase indexing the weighting model is still working correctly. At present there is no weighting method for compound terms that has proved consistently effective. Some common methods which calculate the phrase weight from the weights of its components did not demonstrate consistent results [Fagan 1987, Lewis and Croft 1990 cited in Strzalkowski99(a)]. Strzalkowski concludes with respect to LMI that the lack of an appropriate weighting model for

complex terms can be one of the reasons why LMI did not yield any positive results [Strzalkowski99(a)].

Another aspect that proved to have an impact on the efficiency of LMI techniques was the type and length of the query. Strzalkowski et al. focused their NLP research on the analysis of the interrelationship between NLP methods and the type/length of the query. They moved their attention from the issues of document representation to query representation and proved that long and more descriptive queries worked better with NLP methods than short ones. They designed an expansion method whereby summaries of top ranked documents were manually or automatically selected for addition to the initial query. The expanded query takes the form of a meta-document, covering different aspects of the request, thereby increasing the chances for the match on linguistically-motivated phrases. In TREC-7 they used rather low-level linguistic tools for query expansion, which they tested on two systems: SMART and InQuery. The results were quite optimistic for interactive query expansion, which improved average precision by at least 40%; automatic query expansion did not yet show consistently better performance [Strzalkowski99(b)].

Analysing the research on phrases in IR to date, Sparck Jones made some general conclusions in respect to phrase indexing and NLP-based LMI in particular. She pointed that "statistical facts about term occurrences help as much to make joint terms linguistically legitimate as NLP can, (...) as well as helping to make them effective through weighting." [Sparck Jones99, p. 20].

Another conclusion is that single term indexing is likely to stay even with the introduction of phrases into indexing. "Treating phrases, even normalised ones, as fixed and undecomposable units is an extremely stringent retrieval strategy, and the default strategy that automatically adds all the single terms into a description is much more flexible." [Sparck Jones99, p.21] The advantage this strategy brings is multiple term matches due to single term redundancy.

LMI was not shown so far to be superior to NLI on its own [Sparck Jones99]. Given that the interrelationship of LMI with other system parameters is not clear yet, some researchers like [Strzalkowski99(b)] shifted their attention to the investigation of more specific conditions where LMI can be useful and how it can be better integrated with other IR devices.

3.3 Approaches to collocation in IR

There have been a wide range of approaches towards using word co-occurrence information in IR. They differ not only in the way co-occurrence information is used in retrieval and the motivation behind its usage, but also in the understanding of the phenomenon of collocation. As section *2.4.1* demonstrated, there is no uniform understanding of the notion of collocation. It was decided in the same section to distinguish short-span collocations, motivated by lexical-syntactic relations, and long-span topic-level collocations, motivated by lexical-semantic relations.

At the broadest level the work done in IR involving word co-occurrences can be divided into:

1. Approaches that use short-span collocation information for phrase identification;
2. Approaches using document-wide co-occurrences of terms;
3. Approaches using long-span collocations to obtain some context information.

The first type of approaches were touched upon in the above section *3.2.2*. Statistically defined short-span collocations are used by NLI approaches to generate composite indexing units – joined terms or statistical phrases, which are statistical surrogates of the genuine linguistic phrases.

Short-span lexical-syntactic relations are not the focus of this thesis. The focus is on topic-level relations between words, relations which form the lexical cohesion of the text/part of text, and that can reveal some semantic characteristics of the context in which they occur. For this reason and because they were covered in the previous section, short-span collocation-based NLI methods will not be touched upon in this section.

The second research direction has been rather intensively explored, with much of the research undertaken before the wide use of full-text. The general motivation behind research on document-wide co-occurrence was to understand the effect that information about the presence of more than one term in a document can have on retrieval performance. Approaches to document-wide co-occurrence account only for the presence or absence of two/more terms in documents, and do not make use of contextual information about co-occurrence of terms within some limited stretches of text. This research area is only indirectly related to the thesis, therefore only a selective coverage of the previous work in this direction will be made here.

The third type of approaches deals with co-occurrences in large-scale subtextual constructs like windows. With the arrival of full-text collections containing long multi-topic documents, accounting for document-level term dependencies no longer seems adequate. Instead, exploitation of term dependencies within more homogeneous subdocument semantic units – topics – may lead to improvements in retrieval performance. Approaches of this type focus on studying term dependencies within *limited* spans of text, and attempt to capture statistical evidence of relations pertaining to a topic in a document. The thesis follows this research direction, therefore a maximum coverage of past approaches to long-span collocations will be given in this section.

Document-wide term dependencies will be referred to as *co-occurrences*, and only dependencies within limited spans of text will be referred to as *collocations*. Many authors also refer to window-delimited term dependencies as *co-occurrences*, therefore their original terminology will be retained when talking about their work.

The limited spans of text for identifying collocations will be called *windows*. As described in section *2.4.2* different researchers adopt various ways of measuring spans of text constituting a window. In this thesis a *window* is understood as a wordcount span to the left and right of the node (for definition of *node* see section *2.4.1*

Definitions of collocation). Other definitions adopted by authors of the reviewed works will be specified.

3.3.1 Document-wide co-occurrences

Attempts to incorporate document-wide co-occurrence data date back to the work by Spark Jones [Sparck Jones68, Sparck Jones70, Sparck Jones71(a), Sparck Jones73], Van Rijsbergen & Harper [Van Rijsbergen77, Harper78] and Smeaton & Van Rijsbergen [Smeaton83].

Sparck Jones used document-wide co-occurrence information in automatic index term classification. She experimented with different clustering techniques to group terms showing document-wide co-occurrence into clusters. The performance was not very different for various clustering techniques; it seemed more to be affected by other factors, such as frequency of the terms included in the clusters. Sparck Jones argued that inclusion of low-frequency strongly connected terms into clusters and exclusion of high-frequency terms were some of the major factors why the method worked well [Sparck Jones71(a)]. Later experiments however [Sparck Jones73] could not prove the robustness of co-occurrence term clustering, yielding significant improvements only on one collection.

The work done by Van Rijsbergen [Van Rijsbergen77] consisted in modifying a probabilistic term weighting scheme, which was originally built assuming term independence (see section *4.2*) to account for term dependence. His model derives pairwise term dependencies from the distribution of co-occurrences in the whole collection or in the sets of relevant and nonrelevant documents. "Co-occurrence" he defined as the presence of both words in the same document. The strength of association between two co-occurring words was measured using expected mutual information measure (EMIM), on the basis of which the best dependence tree – the maximum spanning tree (MST) was constructed, connecting the terms with the most significant dependencies.

Initially Van Rijsbergen derived a non-linear term weighting function, for which the dependency parameters were taken from the co-occurrence based MST [Van Rijsbergen77]. Later Harper and Van Rijsbergen [Harper78] carried out an extensive evaluation of the dependence model. In the first stage of testing they compared the independence model with the dependence model for both original and expanded queries. These experiments assumed that all relevance information is known (upper bound experiments). Expanded queries were constructed by adding closely-related terms from the MST to the original query. The results showed that the dependence model with expanded queries gave better performance than the independence model with expanded queries, which in turn was better than either models with the original queries. Thus, it was concluded that two factors in combination are necessary to significantly contribute to the upper bound performance: query expansion via the MST and term weighting assuming term dependence.

Terms taken from the MST are weighted not according to their similarity to the original term but according to their ability to discriminate relevant from non-relevant documents. Further, Harper and Van Rijsbergen derived another method of evaluating

a term's relevance discriminating ability – by modifying EMIM as term weight. They suggested calculating EMIM between the term assignment and the relevance assignment rather than between two terms as in its original use. The new term weighting model uses a linear weighting function together with EMIM term weights summed over the query set.

The evaluation of the EMIM weighting model showed that with the original queries it performed better than the independence model and the previously defined strict dependence model. With expanded queries it gave similar performance to the dependence model.

Harper and Van Rijsbergen concluded that the EMIM weighting model (modified dependence model) is preferable to the strict dependence model because, first, EMIM weights require half as many parameters as the dependence weights and, second, its performance results are not worse than those of the dependence model. Though term weighting in the modified dependence model is based on a linear function, the term dependence is still accounted for by the fact that expansion terms are taken from the MST.

Experiments on relevance feedback also showed that the modified dependence model performed better with relevance feedback than the independence model. However the experiments were conducted on rather small collections, and as the authors themselves admit, more testing with larger and more heterogeneous collections should be undertaken before making the final conclusions.

Van Rijsbergen [Van Rijsbergen77] also suggested other uses for the co-occurrence based MST. It can be used as a classification of index terms, being transformed into a hierarchy of single-link inter-term relationships. Another application is in interactive searches, where users can have access to MST as a resource with information on related terms, and use it to formulate their queries.

Related research by Smeaton & Van Rijsbergen [Smeaton83] experimented with automatic query expansion on three types of terms:

- MST terms;
- Nearest neighbours (NN) – terms most strongly related statistically;
- Index terms from the relevant documents.

The three query expansion methods were comparatively evaluated against no query expansion and expansion with randomly selected terms. The results were rather pessimistic: no query expansion gave best results, followed by expansion with random terms, followed by terms from relevant documents, which in turn were better than MST terms and NN terms. Moreover, more added terms meant more decrease in performance. The authors attributed these results to several possible factors, one of which was poor probability estimations.

Later Peat and Willet [Peat91] suggested that the cause of the problems of using document-level co-occurrence information in automatic query expansion lies in the methods of estimating similarity between terms co-occurring in documents. They pointed that terms considered 'similar' by these methods tend to have corresponding frequency parameters; since query terms often have high collection frequencies, terms

estimated to be 'similar' to them will also tend to have high collection frequencies. Being high frequency terms, 'similar' terms are poor relevance discriminators, and hence will not improve performance by being added to the original query.

3.3.2 Long-span collocations

There have been much less active research in the area of long-span collocations within a limited text area. Some of this research has been directed at integration of collocation data into probabilistic models: Losee [Losee94] used the Bahadur Lazarsfeld expansion method and Mittendorf et al. [Mittendorf2000] conducted experiments on including different levels of collocation information into the Robertson & Sparck Jones model.

Losee [Losee94] attempted to incorporate term dependence information into probabilistic retrieval, limiting term dependence to a certain window within a document. The results he obtained, however, cannot be extended without further testing to full-text retrieval since he used abstracts only for evaluation.

Losee conducted his experiments using Bahadur Lazarsfeld expansion with varying degrees of truncation to estimate probabilities. He chose retrospective type of experiments, using full relevance information. The documents were ranked by the Expected Precision (EP) of the document, which is calculated from the ratio of the probability that the feature occurs in a relevant document to the probability that it occurs in a nonrelevant document, the latter being estimated by the probability of the feature's occurrence in the collection. The performance measure used in the experiments was Average Search Length (ASL) – the average number of documents retrieved when retrieving any relevant document.

Results of the experiments on different degrees of document-wide term dependence showed the general tendency of performance increase corresponding to increase of the degree of dependence. However, a significant increase in performance was observed only for pairs and triples of words, used in estimation of the feature's occurrence probability in relevant documents. A greater level of dependence gave little performance improvement.

The second stage of Losee's experiments was aimed at testing his hypothesis that limiting term dependence to a span within a document could improve performance. Losee defined a *span* as a maximum number of words between two terms whose dependencies are computed. The experiments showed that there is a general tendency of increase in performance (decrease in the ASL) corresponding to the increase of the span of dependence. The major performance increase was observed for the span of 3-5 words.

Losee's experiments on span-limited dependence were correlated with a parallel research on text windows by Haas and Losee [Haas94]. The motivation behind this research was to identify the optimum window size in a document that could be useful in information retrieval tasks. By *window* the authors understand a group of words in contiguous positions in text. This understanding of *window* is different from the one assumed in this thesis as left and right spans of text around a centre – node. Haas and

Loose's understanding of *window* has no idea of a centre, instead it is seen as a range of words within which lexical-syntactic relations are at their strongest. They also focused mainly on short-span relations between words, whereas in this thesis the focus is on long-span topical relations.

Haas and Losee conducted retrieval experiments on an abstracts collection, using windows of different sizes (from 2 to 17 words) as a range within which query terms must appear. In each document the best window (i.e. the window with the maximum concentration of query terms) is identified and its similarity to the query is calculated by the number of common token types – Coordination Level Match (CLM). The performance is evaluated by Average Search Length.

The results corresponded to those by Losee [Losee94]: the ASL decreased gradually with the increase in window length, the sharpest decrease being observed for window sizes 3 to 5.

Mittendorf et al. [Mittendorf2000] applied the Robertson-Sparck Jones probabilistic weighting formula to word collocations (co-occurrences in their terminology) as indexing features, as distinct from the original word-based features. They hypothesise that accounting for word co-occurrences by simply adding complex indexing units like phrases into the word-based indexing vocabulary in weighted retrieval was not yielding significant improvement due to the resulting probabilistic inconsistencies. To avoid such inconsistencies, they suggest using only one type of indexing features – either single terms, or co-occurrence based complex indexing units.

Mittendorf et al. aimed to test whether accounting for positional closeness of query terms in documents in term weighting has any effect on retrieval performance. They defined second-order features as opposed to first-order features – the single words that the traditional Robertson-Sparck Jones model uses. Their second-order features are order-sensitive word co-occurrences within the pre-defined set window sizes. The window ranges they defined for their experiments were aimed to approximate to various NL text constructs:

- window range (from –1 to 0 words) corresponds to standard indexing units used in Robertson-Sparck Jones weighting;
- window range (from 0 to 1 words) corresponds to phrases;
- window range (from 1 to 10 words) approximates to the size of a sentence;
- window range (from 10 to 30 words) approximates to the size of an average paragraph;
- window range (from 30 to 200 words) corresponds to document size.

The motivation behind using co-occurrences within windows of limited size was to capture the local information that they claim will be lost if the upper bound of the window range is too large or if the co-occurrence is only accounted for on a document-wide basis.

They chose the routing task as the testbed for their evaluation. The co-occurrence based weighting was tested with different query sizes: from 3 to 60 terms. The results showed that only for query sizes larger than 10 terms did the sentence- and paragraph-wide co-occurrences yield a significantly better average precision than the original

model. Co-occurrences corresponding to phrases were better than single terms only when the query size was very large. The general tendency observed for phrase-, sentence- and paragraph-wide co-occurrences was an increase in average precision with the increase of the query size. The use of large window sizes with the upper bound of 200 gave very similar average precision to the original model, with the query length having similar influence on both methods. The conclusion the authors arrive at is that large window sizes do not provide sufficiently local information, apart from the information on the presence or absence of terms in a document.

Further, Mittendorf et al. attempted to derive a new ranking formula which would combine the ranking of documents on different feature sets (co-occurrences in different window sizes) by query-specific logistic regression. The same regression analysis did not work for all queries, therefore for each query a different combination of co-occurrence types is needed, which the authors explain by the fact that the types of co-occurrences that better describe one information need may not work for another. The evaluation of the query-specific combined ranking in the routing environment showed an acceptable performance (in terms of average precision) for a routing method, however they admit that more tuning should be done.

Other attempts to use collocation information were targeted either at interactive search, by composing a collocation-based term resource and offering it to users for interactive query expansion [McDonald97], or at automatic query expansion [Jing94, Xu96]. Another factor by which these approaches can be categorised is the scope of collocation analysis, according to which there can be distinguished *global* analysis techniques [e.g. Jing94] and *local* analysis techniques [e.g.Xu96]. *Global* techniques consist in processing the entire document collection to extract term collocations, whereas *local* techniques use either known relevant documents or top ranked documents for deriving collocations.

Work done by McDonald et al.[McDonald97] was intended to minimise the effort on the part of users in refining their queries following an initial search. They attempted to substitute the traditional relevance feedback mechanism, requiring the users to examine documents, by offering them more structured information about the database contents. This information is structured in the form of collocation (here termed *co-occurrence*) based term networks, which the authors claim have two major advantages over document output for relevance feedback: first, the networks maximise the database content coverage and, second, they minimise effort required by the users in familiarising themselves with the database contents.

During the pre-processing stage a subset of indexing terms across the database is selected, consisting of the most content-discriminating terms. Within this subset term collocations are identified based on within-sentence co-occurrence, and selected using Dice's coefficient. The window size as one sentence was an arbitrary decision, which was not comparatively evaluated against other possible window sizes. The term networks are generated from the co-occurrence data using their *Pathfinder* algorithm which filters out insignificant term associations based on frequency of term co-occurrence.

At search stage following the initial query submission, the user is presented with a graphical representation of the portion of the network, related to the query terms. The user then can substitute the existing query terms, add new ones, remove or negatively weight inappropriate terms.

Although no large-scale evaluation of the method using global co-occurrence based network was performed, the authors observed that associations were not always topic-related because networks captured term associations throughout the collection and across topics. Therefore no distinction was made between different uses of a term. To mitigate this negative effect they suggested that networks should be constructed from the retrieved sets of documents, i.e. using a local rather than global technique.

The evaluation of the second method was conducted using TREC-6 interactive track topics. During the pre-processing stage, sets of documents were obtained by using a Boolean search on terms from topic titles. A separate database was constructed for each set and was invoked when a query derived from the corresponding topic was submitted. The results were inconclusive. The authors admit the limitations of using Boolean search for the task of retrieving relevant documents. Also it is not clear what would be the mechanism for index-time construction of topic-specific networks for real-life operability of the model, or whether search-time network construction is feasible.

Jing and Croft [Jing94] developed an approach for the automatic construction of an association thesaurus through the global analysis of collocation (*co-occurrence* in their terminology) data. The program they designed for automatic thesaurus construction – *PhraseFinder* – works as an adjunct to the *INQUERY* retrieval system. Index units can be either single or composite terms, identified in the text through a set of phrase rules. Co-occurrences between index units are identified within window sizes of 3-10 sentences, as these approximate the size of an average paragraph. After filtering out co-occurrences that are too frequent or too rare, each index unit is recorded with the list of its co-occurrences in the thesaurus.

The co-occurrence thesaurus is used in query expansion. It is implemented as an INQUERY database, with each entry being a separate pseudo-document. When a query is submitted, the system retrieves a corresponding pseudo-document for each query term and outputs a ranked list of collocates from these pseudo-documents. The top N collocates are then used for query expansion.

The evaluation of the above query expansion technique was aimed at testing several parameters, among which are the following:

- How expansion with different indexing units (duplicate/nonduplicate) affects performance;
- What phrase rules result in better performance;
- What is the optimum window size.

Expansion with two types of index units was tested: duplicates and nonduplicates. *Duplicates* are indexing units each component of which is present in the original query. *Nonduplicates* must consist of components that are not part of the original query. The purpose of adding duplicates was mainly to test how thesaurus data can be used to reweight the original query. The purpose of the expansion with nonduplicates

was to evaluate the thesaurus by the usefulness of new terms it provides for request formulation.

Experiments with phrase rules showed that among single word index units the use of only nouns for query expansion resulted in better performance than the use of any other part of speech alone or when no part of speech distinction is made. Among complex units again the use of phrases containing only nouns resulted in better performance. The maximum performance improvement was achieved by using the phrase rule: one noun, two adjacent nouns and three adjacent nouns.

The window sizes for collocates extraction tested in the experiments were 3, 5 and 10 sentences. Three thesauri based on different window sizes were constructed. The experiments were conducted on TIPSTER sample collection, with the queries derived from the concept field of the topics. The results showed that a 10 sentence window size gave better performance than 3 and 5, however the difference was not significant. The conclusion the authors came to was that the method was rather insensitive to the window size. The window must be of paragraph size order, but the exact number of sentences in the hypothetical paragraph does not affect performance significantly.

The evaluation of the global co-occurrence based query expansion technique, implemented as PhraseFinder, proved to be rather robust, improving the average performance of queries. Another advantage is that it constructs a term association resource that can also be offered to users for interactive query expansion.

There have also been somewhat marginally related, formalistic global methods applied to building automatic thesauri for the use in IR. They encode word co-occurrence statistics into inference models, such as Bayesian networks [Han93, Park96] and use other techniques such as term clustering. Han et al. [Han93] suggested the method of building *collocation maps* – inference models encoding statistical evidence on word co-occurrence in delimited text windows in the collection. Park et al. [Park96] later developed a method of building automatic hierarchical thesauri by extracting term pairs within limited spans, encoding them into the collocation map, building thesaurus classes using term clustering methods and linking similar clusters. They evaluated the thesaurus constructed using the proposed method in query expansion. Only abstract databases were used for thesaurus construction, also they manually selected indexing terms from one of them. The results showed some improvement for the collocation map built from automatically extracted terms and more considerable improvement for the collocation map built from manually selected terms.

Xu and Croft [Xu96] compared the global co-occurrence technique used by PhraseFinder [Jing94] with their own technique of *Local Context Analysis* (LCA) also implemented with INQUERY. The collocates of query terms are defined as noun groups, taken from the retrieved N top ranked passages of fixed size of 300 words. The authors point at the downside of using whole documents for term extraction, especially in long documents, where the co-occurrence of terms over a very long span may not reflect any topical relationship.

The noun groups from the top-ranked passages are ranked according to a variant of *tf/idf* measure - a function of the individual frequencies of occurrence of query terms

and noun groups, their co-occurrence frequencies in the retrieved passages and their inverse passage frequencies in the entire collection. The ranking function penalises units with very high frequency, rewards units co-occurring frequently with the query term, and emphasises co-occurrence of a unit with all query terms.

The evaluation carried out on three collections: TREC-3, TREC-4 and WEST, showed that LCA improves performance significantly for both TREC-3 and TREC-4 collections. Performance for WEST is improved only with downweighted expansion terms, which is explained by the fact that the original query terms are good and should be emphasised.

Comparison with PhraseFinder showed that LCA gives a better performance improvement, for example on TREC-3 Phrasefinder is 7.8% better than the baseline, while LCA, using 100 top passages, is 23.3% better. Among the observed downsides of the global technique, the authors pointed out the fact that frequent terms which nevertheless can be good content indicators are inevitably filtered out, whereas in local technique they can be added to the query. Another negative factor observed for the global technique used in Phrasefinder is that it does not account for co-occurrence with all query terms, which can increase the chance of adding unrelated terms.

The authors also compared LCA with local feedback, which uses top ranked documents for query expansion. The performance of local feedback for different collections indicated that it was sensitive to the number of documents used for feedback, which appeared to depend on the number of relevant documents in the collection for the query. LCA was relatively insensitive to the number of passages. Another downside of local feedback is that it is very sensitive to the number of relevant documents in the top ranked documents, LCA on the contrary is not so sensitive.

Ishikawa et al. [Ishikawa98] also suggested an approach based on the local analysis technique for blind query expansion in the routing task. Their method uses top ranked passages to extract terms (noun phrases), for which a strength of association with query terms occurring in the same passages is calculated. A modified mutual information measure is used for this purpose. Terms with a mutual information score above the set threshold are selected for query expansion. Expanded terms are assigned the weights of the query terms with which they have the strongest association level. For term weighting they use Robertson's term relevance weighting formula [Robertson76]. The evaluation of runs with expanded queries did not show any improvement over initial queries.

3.4 Applications of lexical cohesion in IR

Research undertaken by Stairmand [Stairmand97] was motivated by the analysis of lexical cohesion by Morris and Hirst [Morris91] (see section *2.3.2 Lexical links and lexical chains*). In Stairmand's method the lexical contents of documents is mapped into WordNet synsets. Then two main types of constructs are identified in each document: lexical clusters and lexical chains. Lexical clusters consist of the related synsets co-occurring in the document and representing a distinct textual context,

which can manifest itself in different parts of the document. To identify where in text such context manifests itself, the method locates lexical chains, using Morris and Hirst's algorithm (see section *2.3.2*). To eliminate spurious synsets, only those synsets are kept in clusters which are also members of lexical chains. The clusters in each document are then ranked by their 'strength', which is derived from the distribution of cluster members throughout the document. The document representation consists of synsets from the 'strongest' or dominant clusters. Each synset in the representation is assigned a weight which reflects the dominance of its cluster, and hence its context of occurrence in the document.

At search time, each query term, mapped into a WordNet synset, is matched against the weighted synsets representing the documents. Since the weight of each synset in the document representation reflects the score of its cluster, the matching implicitly takes into account the whole context of occurrence of the query term, and determines how pertinent to the document is this context, and hence the query term belonging to it.

Stairmand hypothesised that accounting for the context of occurrence of the query term in indexing and matching could be superior to term based methods. However he recognises that representing document contents with WordNet synsets has severe limitations due to WordNet's restricted coverage. The evaluation based on rather simple queries compared the above method with a term-based system – SMART. Top three documents retrieved by each system were given to users for relevance judgement. The results demonstrated improved performance for simple queries.

A further extension to the above method is aimed at capturing global statistical co-occurrences of synsets in document representations, and bringing them together in synset groupings. The author points at the inadequately specific nature of relations holding between synsets in WordNet, which do not cover all possible semantic relations between words in text, the exact nature of which is often difficult to establish. For this reason he points at the potential benefits of exploiting statistically significant co-occurrence relations between synsets for both document indexing and query expansion.

The co-occurrences were identified globally on a document-wide basis. The strength of association was measured using mutual information score. For each synset a set of significantly associated collocates was created, forming a cluster of associated terms around it. Finally only those clusters were selected, whose nodes attracted a high number of strongly associated synsets. The method creates a resource of co-occurrence based clusters of synsets, which can be used as a supplement to WordNet. The intuitive observation suggested that the majority of generated clusters were semantically coherent, however no formal evaluation of this resource was reported.

Another use of lexical cohesion in IR is in summarisation of documents and retrieval only of the information relevant to the user's request. Manabu et al. [Manabu2000] conducted a research on query-biased summarisation of the documents in retrieval, using lexical chains. Their method is a variant of passage retrieval (see section *3.5 Document passages*). Many query-biased summarisation methods produce document summaries by identifying best sentences. Manabu et al. argue that such summaries are likely to have low cohesion, since adjacent sentences may not be related. Instead they

claim that identifying the best passages using information from lexical chains can result in more coherent summaries.

The building of lexical chains requires the use of some knowledge about the relatedness of terms in a stretch of text. As described in section *2.3.2* one method is to use a knowledge base like a thesaurus [Morris91] or WordNet [Hirst97] to obtain information about term relatedness. Another method that Manabu et al. adopt is to estimate statistical similarity between words, for which they use semantic similarity score, based on the degree of co-occurrence of two words in the same document. Similarity scores are then used to group words into clusters. Lexical chains are built from words belonging to the same clusters.

At search time query terms are matched against the members of lexical chains. Lexical chains containing query terms are identified and the passages are extracted as the text segments covered by overlapped lexical chains. The boundaries of the passage correspond to the left- and rightmost members of the overlapping chains. The identified passages are output to the user, ranked by the scores based on the length of the chains composing each passage and the degree of their overlap.

The query-biased chain method was evaluated by recall, precision, F-measure, time required for the task and the number of times the users referred to full-text. The authors comparatively evaluated this method with full document retrieval and retrieval using three other summarisation methods:

- location method - first N sentences of the document);
- term frequency method - top sentences ranked on the basis of *tf/idf* measure of words each sentence contains;
- query-biased term frequency method - similar to the previous method, but the score of words is calculated to bias towards the query terms (for more detail on sentence-based summarisation methods see section *3.5*).

F-measure results for the chain method were significantly better than location and query-biased frequency methods, but no significant improvement against frequency method and full documents. Recall levels for both query-biased methods (chain and query-biased frequency) were better than the rest. The authors admit that allowing the users access to full text may result in better performance than display of summaries only. After adjusting the accuracy by the number of times full documents were accessed, full text retrieval became the best, however the results for the chain method were still better than other summarisation methods.

3.5 Document passages

Passage retrieval, although a rather different technique from collocation methods, has one major aspect in common with it, namely capturing the *locality* of the text. It is motivated by the same idea – that a local context is more semantically homogeneous than an entire document and that lexical units occurring within a liamited span of text are more likely to be semantically connected than words in different parts of a document.

Many authors recognise the problems associated with document retrieval from full-text collections [Hearst93, Salton93, Hawking96, Kaszkiel97], namely that full-length documents often contain only some parts that are relevant to the user's need, and that these documents can be ranked low because of the overall higher proportion of unrelated terms in non-relevant parts. Traditional document retrieval approaches could not isolate relevant passages in full-length documents, therefore what was needed was a more granular approach to document contents.

The advantages that passage retrieval offers are:

- more convenient for the users, since they do not have to process large amounts of non-relevant information contained in full-length multi-topic documents [Salton93, Kaszkiel97];
- relevant passages are easier to retrieve than relevant full-length multi-topic documents, which may have high concentration of non-relevant items, and therefore, can be rejected [Salton93];
- fixed-length passages can alleviate the problem of document normalisation when they are used as a mechanism for document retrieval [Kaszkiel97];
- arguably a document containing a passage with a high concentration of matching terms is more likely to be relevant than a document with matching terms scattered throughout its length [Kaszkiel97].

At the same time Kaszkiel et al. pointed several problems that can be associated with passages [Kaszkiel97]:

- retrieval can be more computationally expensive as it involves more items to rank;
- passages with varying length do not solve length normalisation problem;
- relevant documents with no high-scoring passage can be ignored.

There are different ways passages can be used in information retrieval. Robertson et al. [Robertson95] pointed out three uses passages can have in IR:

- The score of the best passage(s) can be used in calculating the document score;
- The best passage(s) can be displayed to the user instead of the whole document;
- Only relevant passages are used for relevance feedback.

Different methods for passage definition either at file-time or query-time have been proposed. One of the earliest approach to passage definition and text summarisation is sentence-based [Luhn58]. Sentences in the text are ranked according to the combined weights of the terms they contain, optionally modulated by the frequency and concentration of terms in these sentences. The higher the weights of the terms in the sentence, and the higher the concentration and closeness of these terms in the sentence, the greater the score. Sentences with top scores are combined together to form a passage. A modification of this approach takes account of the query term presence in the sentences for ranking – query-biased approaches [Tombros98]. Here term weighting depends on whether a word is a query term or not. Other modifications of the sentence scoring approach summarised by Salton et al. [Salton93] are:

- the location of sentences in the text;
- the occurrence of special "clue" words in the sentences;
- the use of syntactic relations between words in different sentences.

The problem with sentence-based passages is their often poor readability, which is due to the fact that rather crude term frequency measures are used to estimate sentence importance [Salton93] and that sentences are taken from all over the document and are not necessarily coherent [Manabu2000].

Another approach is to use other larger natural language constructs like paragraphs or sections of long documents. Wilkinson [Wilkinson94] experimented with the use of text segments as passages for document retrieval. Salton [Salton93] suggested a top-down approach for gradual retrieval of successively smaller document parts: full-texts, text sections, text paragraphs or sets of adjacent sentences. Initial search retrieves full-texts by query-document matching, then local structures (sections, paragraphs and sentences) in the retrieved documents are compared to the query, and documents are re-ranked by the similarity of their substructures. A passage with the highest score, given it is also higher than the document's score, is presented to the user. The user is also given an option to view smaller or larger text structures. The evaluation of this method demonstrated significant performance improvements over document retrieval.

Passage experiments with Okapi [Robertson95, Sparck Jones98] also use passages built from natural text paragraphs. Paragraphs are identified in text by their orthographic delimiters: indentation and/or blank lines. Passages are arbitrary length windows, consisting of at least one paragraph and a maximum of twenty consecutive paragraphs with the default distance between passages as one paragraph. The entire document is also considered for the best passage selection.

Passages are identified at query time and only the top 10000 ranked documents are analysed, based on experimental evidence that further down the ranked set the chances of getting a document with a good passage are very low. Passages are used in Okapi for document retrieval, i.e. the documents receive the scores of their best passages and are ranked according to them. Experiments demonstrated performance improvements with the use of passages, but they are not always consistent. Average precision and recall/precision at higher cutoff points are often increased by 2-10 percent, however at smaller cutoff points precision decreases [Robertson99].

Another group of approaches uses some lexical-semantic evidence from a text to determine the best passages [Hearst93, Manabu2000, Mittendorf94].

Hearst [Hearst93, Hearst94] developed a technique called *TextTiling* to identify best passages from the lexical-semantic evidence that signals topic shifts. The pairs of adjacent fixed-size blocks of text, each consisting of N fixed-size word sequences, are compared according to how similar they are using a cosine similarity measure. Boundaries of a semantically homogeneous passage are identified via changes in the sequence of similarity scores and are adjusted to the nearest paragraph breaks. In other words a passage is built from a maximum number of adjacent semantically similar word sequences.

Another approach to passage retrieval is based on lexical chaining technique by [Manabu2000] and was described in detail in the previous section.

Mittendorf et al. [Mittendorf94, Knaus95] developed a method of inferring passage boundaries from text using Hidden Markov Models. Passages are of variable length and are identified at query-time by evaluating the query against the entire document. TREC experiments demonstrated the capabilities of the method to improve the initial ranking of the documents.

Kaszkiel et al. [Kaszkiel97] noted that methods using variable-length passages are susceptible to problems of length normalisation. They argue that fixed length passages are simpler to define, more robust and effective. There have been numerous experiments that use fixed-length passages. Callan [Callan94] proposed defining passages at query-time as windows of fixed wordcount length. The windows overlap, so that the beginning of the next window is in the middle of its predecessor. The beginning of the first window is set to the first occurrence of a query term. Other approaches that use fixed-size windows, each overlapping with its neighbouring windows, were used in TREC for an ad hoc task by [Buckley95] and for a routing task by [Yochum96], both of whom use passage scores for calculating the final document scores. The window sizes defined in these approaches range between 100 and 200. The evaluations of these methods on TREC data demonstrated some improvements.

Kaszkiel et al. [Kaszkiel97] conducted extensive experiments comparing their technique of fixed-length arbitrary passages with some other methods. They experimentally confirmed that fixed-length arbitrary passages with the minimum size of 150 and heavily overlapping give substantial performance improvements, especially for long document collections. They demonstrate that their method is robust in contrast to some other methods like the use of natural text constructs, which in their experiments were not always reliable.

Another approach to identifying passages for pseudo relevance feedback was suggested by Hawking et al. [Hawking98]. They define a passage or a *hotspot* as a contiguous stretch of text within n characters of a query term occurrence. This approach to defining a window of text is similar to window definition in some collocation methods including our method, however this approach cannot be regarded as a collocation method, since it does not estimate the strength of association between the query term and its neighbours in text. Terms extracted from all hotspots in the top N ranked documents are weighted using a variation of the Offer Weight formula designed by Robertson [Robertson90]. The results showed considerable improvement in recall, but no significant precision gains.

4. Probabilistic information retrieval

Early attempt to apply probabilistic theory to information retrieval dates back to the work by Maron and Kuhns [Maron60] on *probabilistic indexing*. Their work made substantial contribution to the formulation of probabilistic approaches to such principal concepts as relevance and ranking. However the idea of probabilistic indexing is essentially different from *probabilistic retrieval*, the models of which were developed by Robertson and Sparck Jones [Robertson76], Van Rijsbergen [Van Rijsbergen79] and Croft and Harper [Croft79]. Beginning from the eighties there has also been attempted integration of probabilistic indexing and retrieval approaches into a *unified theory* of information retrieval [Robertson82].

Before moving on to the discussion of the probabilistic model used in this project – Robertson & Sparck Jones model – it is worthwhile to mention the main principles characteristic of all probabilistic retrieval theories. The aim of probabilistic theories is to rank documents in descending order of the probability of their relevance to the query. Since each document is unique and it is impossible to estimate probabilities of unique events, the models derive these probabilities from the characteristics of non-unique events constituting the documents – terms. The estimation of the term's usefulness in discriminating relevant from non-relevant documents is done through *term weighting*. The probability of relevance of each document to the query is estimated through the corresponding weighting matching function. The main differences between probabilistic models are in the weighting functions they use, and the ways these functions are derived.

4.1 Robertson & Sparck Jones model of probabilistic IR

The probabilistic model developed by Robertson and Sparck Jones [Robertson76] approaches the general problem of information retrieval by asking for each document and query the *Basic Question* [Sparck Jones98, p. 5]:

> "What is the probability that *this* document is relevant to *this* query?"

There are two assumptions about relevance implied by the Basic Question:

1. Relevance is a binary attribute;
2. Relevance can be attributed to a document irrespectively of other documents in the collection.

These assumptions are of course simplifications, but they are essential for the model. The main idea of the model is to estimate the probability of relevance of each document in order to use this estimation in ranking the documents for presentation to the user. This idea of the model is known as a *Probability Ranking Principle* [Robertson77], quoted from [Sparck Jones98, p.7]:

> **P1** : If retrieved documents are ordered by decreasing probability of relevance on the data available, then the system's effectiveness is the best to be gotten for the data.

The probability of relevance of each document is estimated from the relevance-predicting characteristics of attributes constituting a document – terms, specifically

query terms. Relevance-based term weighting developed by Robertson & Sparck Jones, and which will be covered in more detail in the next subsection, has been derived from a formal *theory of relevance weights* [Robertson76]. The theory's aim is to achieve document ranking through weighting of terms given some known information on their behaviour in relevant and non-relevant documents. There are two principal assumptions underlying the theory [Robertson76, p.132]:

Independence assumption: The distribution of terms in relevant documents is independent and their distribution in non-relevant documents is independent.

Ordering principle: That probable relevance is based on both the presence of search terms in documents and the absence from documents.

For more detailed discussion of the independence assumptions see section 4.2.

4.1.1 Term relevance weighting

The theory of relevance weights based on the above assumptions, also known as the binary independence retrieval model, is represented on the general level by the formula:

$$w_i = \frac{p_i(1-\overline{p}_i)}{\overline{p}_i(1-p_i)} \qquad (4.1)$$

where p_i is the probability of occurrence of term i in a relevant document;
$\overline{p}_i$ is the probability of term i occurrence in a non-relevant document.

The estimation of the above probabilities requires presence of information about occurrence of terms in relevant and non-relevant documents. The relevance information for a query need not be complete, and in practice it is usually available only to some extent. This information is obtained in the form of user relevance judgements. The following contingency table of term incidence has been proposed as a basis for a more specific representation of the model [Robertson76]:

	Relevant	Non-relevant	
Containing the term	r	$n-r$	n
Not containing the term	$R-r$	$N-n-R+r$	$N-n$
	R	$N-R$	N

where R is the number of known relevant documents
r is the number of known relevant documents with term i
N is the number of documents in the collection
n is the number of known documents containing term i

Given the term incidence information in the contingency table, general term weighting function (4.1) can be re-written as following:

$$w = \log \frac{r(N-n-R+r)}{(R-r)(n-r)} \qquad (4.2)$$

Before any relevance information is obtained, the estimation of p_i and $\overline{p}_i$ is performed on the assumption that the number of relevant documents R in the collection is much smaller than the number of non-relevant, therefore $\overline{p}_i$ is estimated as the number of documents in the collection with term i, i.e. n/N, while p_i is assumed to be a constant. In these conditions the term weighting function (4.2) gives the same result as a collection frequency weight (*CFW*) function, introduced first in [Sparck Jones71(b)]:

$$CFW = \log \frac{N}{n_i} \tag{4.3}$$

The problem with function (4.2) is that when no or very little relevance information is known, i.e. when the value of any of the central cells in the contingency table is zero, it results in the infinite weights. To avoid this problem 0.5 was added to each of the four parameters of the function. The resulting Relevance Weight (*RW*) formula is:

$$RW = \log \frac{(r+0.5)(N-n-R+r+0.5)}{(R-r+0.5)(n-r+0.5)} \tag{4.4}$$

The matching score (MS) for a document is simply the sum of the weights of the present query terms:

$$MS - RW = \sum_i \log \frac{(r+0.5)(N-n-R+r+0.5)}{(R-r+0.5)(n-r+0.5)}$$

Further elaboration of the term weighting function was targeted at exploiting the information on term frequencies within documents [Robertson94] and document length normalisation. Robertson and Walker suggested a formula which modulates the weight of term i by its within-document frequency - TF_i:

$$W(TF_i) = \frac{TF_i(k_1+1)}{k_1+TF_i} w_i \tag{4.5}$$

The formula has the following behaviour: it returns zero for term absence, i.e. $TF_i = 0$, and it increases monotonically with term frequency, but only to a certain point. The weight increase is adjustable by the tuning constant k_1: if $k_1 = 0$, no account of TF_i is taken, whereas a high value for k_1 results in the linear increase of weight with TF_i. The most effective value was experimentally determined to be from 1.2 to 2 [Sparck Jones98]. From this evidence the authors infer that TF_i has a non-linear effect on performance, i.e. after a certain point, taking account of more term occurrences contributes very little.

Inclusion of term frequency into weight estimation in turn requires a mechanism to equally handle documents of different lengths. To develop document normalisation

function Robertson and Sparck Jones made a simple assumption that "where there are two documents about the same topic but of different lengths, this is just because the longer is more wordy" [Sparck Jones98, p.25]. The normalisation factor they suggested can be tuned using the constant b to vary the strength of the above assumption:

$$NF = ((1-b) + b\frac{DL}{AVDL}) \tag{4.6}$$

where DL is document length
$AVDL$ is average document length
b is a tuning constant

If b is set to 1, the formula will normalise document length on the above assumption, i.e. that longer documents are simply more verbose. Whereas smaller values will reward longer documents on the premise that their length is not entirely due to higher verbosity, but to a greater topic development. Experimental evidence suggests an optimum value of 0.75 for b [Sparck Jones98].

Inclusion of the document normalisation factor (4.6) into function (4.5) leads to the following formula:

$$W(TF_i) = \frac{TF_i(k_1+1)}{k_1 * ((1-b) + b\frac{DL}{AVDL}) + TF_i} w_i \tag{4.7}$$

The function (4.7) can be instantiated either with, or without relevance information. The function where w_i is CFW formula (4.3), i.e. without relevance information, is known as Combined Weight (CW). The function with RW (4.4) as w_i is known as Combined Iterative Weight (CIW).

There have also been developed variations of function (4.7) to take account of query term frequency for long queries: query adjusted combined weight ($QACW$):

$$QACW = CW * QTF \tag{4.8}$$

and query adjusted combined iterative weight ($QACIW$):

$$QACIW = CIW * QTF \tag{4.9}$$

where in both formulas QTF is query term frequency.

4.1.2 Relevance feedback

The mechanism for reweighting/reformulating the original query by means of the terms in the retrieved relevant documents is known as *relevance feedback*. Since first suggested by [Rocchio65], the idea of reformulating the query by adding selected terms from the relevant documents is widely considered to be more effective than reweighting the original terms alone.

In the Robertson & Sparck Jones probabilistic model, relevance feedback is implemented through selection of terms from the relevant documents as candidates for the new expanded query. Robertson and Sparck Jones emphasise the *selective* approach to query expansion, as opposed to an alternative strategy of *massive* query expansion, advocated by some research groups, e.g. SMART [Buckley95], where any term is added to the query, for which there is some positive/negative evidence of usefulness. Robertson and Sparck Jones argue that selection is useful, since it makes possible adapting query expansion to the environment conditions of the search, i.e. varying the number of added terms according to the size of the original query. The size of the expanded query is, therefore, predictable, and not affected by the size of the relevant documents, which, if massive expansion is used, can yield a very large number of terms.

They suggest ranking terms in the relevant documents and applying a threshold for term selection, which varies with the size of the original query. The ranking function used for term selection should be different from the function used for term weighting. Robertson and Sparck Jones argue that the question answered by relevance weight:

> **Q1:** "How much evidence does the presence of this term provide for the relevance of this document?"

is different from the question that should be answered by the term selection value (TSV):

> **Q2:** "How much will adding this term to the request benefit the overall performance of the search formulation?" [Sparck Jones98, p.30]

The term selection value should, therefore, reward those terms which are likely to have a strong impact on bringing relevant documents to the top of the ranked list, and not those, which though being good relevance indicators for one document are unable to contribute more because they are too infrequent. To achieve this kind of term selection Robertson [Robertson90] derived the following term selection formula, known as offer weight (OW):

$$OW = rRW \qquad (4.10)$$

Terms selected by OW are then weighted using the term weighting function.

A set of Okapi experiments on TREC data [Sparck Jones98], aimed at identifying the size of the optimum query expansion set for different length types of original queries, established that the following sizes work reliably well: 32 terms for long queries (title, description and narrative fields of TREC topics), 24 for medium (title and description) and 16 for very short queries (title field only). The experiments also suggested that small changes in the sizes of expansion sets did not affect performance.

Another question put forward by Robertson et al. is whether original query terms should be rewarded by the term selection value. They conducted experiments, which emphasised original query terms by assuming the existence of 20 hypothetical documents, with each query term occurring in 19 of them. The results showed no performance improvement from rewarding original query terms.

A query expansion strategy related to relevance feedback, and which was extensively tested with Okapi, is *blind* query expansion. Instead of known relevant documents, it uses documents *assumed* to be relevant through the information implied by their ranking position in the initial ranked set. The documents for query expansion are taken from the top of the ranked set and terms are selected from them on the same principle as for relevance feedback described above. Okapi experiments showed that blind feedback, though not always as good as relevance feedback, can be better than no query expansion.

4.2 Term independence assumptions

There are three fundamental assumptions commonly underlying probabilistic retrieval theories:

I1: Terms occur independently within the whole collection.

$$P(A,B) = P(A)P(B)$$

I2: Terms occur independently within the set of relevant documents.

$$P(A,B \mid R) = P(A \mid R)P(B \mid R)$$

I3: Terms occur independently within the set of nonrelevant documents.

$$P(A,B \mid \overline{R}) = P(A \mid \overline{R})P(B \mid \overline{R})$$

Robertson and Sparck Jones [Robertson76] preferred assumptions I2 and I3 over I1 and I2, since the latter contradict each other, as was recognised by the authors, and later demonstrated by [Cooper90]. Because of I2 and I3 assumptions, the model developed by Robertson and Sparck Jones became known as *binary independence model*.

Though independence within relevant and non-relevant documents is a less stringent assumption than independence within the whole collection, it is still an oversimplification of the data known about the real word behaviour in texts. Independence assumptions are needed however for the following benefits they bring [Sparck Jones98]:

- They simplify the task of model development;
- Model instantiation as an operable system is made possible;
- They result in performance improvements over simple term matching.

If no term independence was assumed, the implementation of such model as a working system would be unrealistic, since the probabilities would need to be estimated for every possible combination of terms in the documents.

However it has been argued [Sparck Jones98] that estimation of all term dependencies may not be needed under the model, since the assumptions it is built on – I2 and I3 – imply term dependencies within the whole collection. Specifically, if it is known that two terms occur more often in relevant documents than in non-relevant, then they can be considered to have some degree of dependency in the whole collection.

Cooper [Cooper90] suggested a weaker assumption, which he called *linked dependence* assumption (I4):

I4: $$\frac{P(A,B \mid R)}{P(A,B \mid \overline{R})} = \frac{P(A \mid R)}{P(A \mid \overline{R})} \frac{P(B \mid R)}{P(B \mid \overline{R})}$$

This assumption associates the relevance-implied dependency between A and B with the nonrelevance-implied dependency. As Cooper argued this is not an independence assumption, but assumption of linked dependencies. What important is that it leads to exactly the same equation as that of binary independence model.

All this leads to the conclusion that the binary independence model already accounts to some extent for relevance-implied document-level term dependencies. In the early days after the appearance of Robertson & Sparck Jones model, there have been many formal/less formal attempts to compensate for independence assumptions in probabilistic retrieval [e.g., Van Rijsbergen77, Harper78, Smeaton83] (see section *3.3.1*), which hardly led to any significant or consistent improvements. Cooper argues that present cooling of interest to such modifications is due to the wider recognition of the above-described implied dependencies. Moreover, he hypothesises that document-level co-occurrence data may not in itself be an important factor in retrieval performance [Cooper90].

5. Collocation analysis in this project

5.1 Research question

The basic research question of the project is whether the use of long-span collocates of query terms can improve performance of the probabilistic model. We aimed to answer this research question by investigating different ways of integrating collocation information with the probabilistic model. Over the course of the project four main hypotheses were examined and tested.

Initially we hypothesised that query expansion with collocates of query terms, extracted globally, i.e. from the entire corpus, can lead to performance improvement. Formal hypothesis statement is as follows:

Hypothesis 1: Expansion of the initial query with statistically significant global collocates of query terms results in significant performance improvement over the initial query evaluated under the same conditions.

Experiments to explore and test the first hypothesis formed the first experiment set: global collocation analysis. The detailed description of these experiments, their methodologies and results are given in chapter 6. The results of the exploration of global collocations and their contribution to retrieval performance through query expansion, led us to the study of a related technique – the use of local collocations, i.e. collocates of query terms extracted from relevant documents only. This technique was called local collocation analysis. The aim of this study was to explore query expansion with local collocates of query terms following relevance feedback, and to compare it to the existing Okapi query expansion technique. We hypothesised that query expansion with local collocates can perform better than Okapi query expansion.

Hypothesis 2: Expansion of the initial query with statistically significant local collocates following relevance feedback results in significant performance improvement over Okapi relevance feedback under the same conditions.

We explored this hypothesis in the second experiment set: local collocation analysis, described in chapter 7.

From the study of collocates of query terms in two different scenarios: global and local collocation analysis, we moved on to the investigation of another parameter: lexical cohesion between query terms, estimated through their local collocates. Initially we hypothesised that relevant documents tend to have a higher level of lexical cohesion between query terms than non-relevant documents. This hypothesis was based on the following premise: query terms are used together to describe the topic the user is interested in, hence in a relevant document, i.e. containing this topic, the query terms are likely to be also used to describe the relevant topic. If terms occur in the same topic, they tend to cohere with each other and have similar collocation

environments. In a non-relevant document, the occurrence of these query terms is not motivated by the presence of a relevant topic, but due to other factors, therefore they are less likely to occur in the same semantic context.

We aimed to explore whether the level of lexical cohesion between query terms in a document can be linked to the document's relevance property, and if so, whether it can be used to predict the document's relevance to the query.

Initially we formulated a hypothesis to investigate whether there is any statistically significant relation between two document properties – its relevance to a query and lexical cohesion between query terms occurring in it.

Hypothesis 3: There exists statistically significant association between the level of lexical cohesion of the query terms in documents and relevance.

We conducted a series of statistical analyses, which formed the first half of the third set of experiments: lexical cohesion analysis using local collocations (Chapter 8). Statistical dependency, discovered between documents' lexical cohesion scores and their relevance property, suggested the next step: retrieval evaluation of the usefulness of lexical cohesion in predicting documents' relevance. We hypotheised that re-ranking the initial Okapi document sets by the documents' lexical cohesion scores can yield better performance results than the initial Okapi ranking. Formal statement of this hypothesis is:

Hypothesis 4: Re-ranking of Okapi document sets by lexical cohesion scores results in significant performance improvement over initial Okapi ranking.

Retrieval experiments exploring this hypothesis formed the second half of the third set of experiments, also presented in chapter 8.

5.2 Overview of the experiments

Any IR system is a very complex mechanism, whose performance is a result of interplay of a multitude of variables. Some of these variables are characteristics of the core of the system – its indexing and retrieval mechanisms. Others are determined by the environment, the system operates in: the documents, the users, their requests, type of their interaction with the system, etc. The scope of our research is the core of the system, specifically its retrieval mechanism. By hypothesising that introducing certain changes into the retrieval system will improve its performance, we aim to understand how each of these changes affects the dependent variables of the system.

The two types of experimental environments – laboratory and operational – offer different conditions for variable testing. In the laboratory environment, it is easier to maintain control over all system variables, while testing the effect of one variable on the dependant variables. The same experimental conditions can be replicated as many times as needed, facilitating a uniform comparison of different variable values. In the

operational environment, the picture is quite different. Introduction of such unpredictable and changeable variable as the user makes it impossible to maintain control over the experimental parameters. However it is understood that the ultimate test of any system's effectiveness lies in user satisfaction, and should be performed with real users and their genuine needs. But first it is necessary to form a sound understanding of the system's internal mechanism, which is the aim of this project, real-life testing being outside its scope.

The work done within this project is grouped into three sets of experiments:

- **I set:** Global collocation analysis
- **II set:** Local collocation analysis
- **III set:** Lexical cohesion analysis using local collocations

Each set of experiments will be discussed in detail in subsequent chapters 6, 7 and 8. The details of methodologies specific to each of these experiments, and motivation of their choice will also be described in the corresponding chapters. In the following three subsections a concise overview of the experiment sets will be presented. This chapter also contains sections with the detailed description of some methodologies which are common to most or all of the experiments in this project.

All coding for the experiments was done by the author. Perl scripting language was used to write the programs. Being an interpreted language, Perl works more slowly than compiled languages like C or C++. But, first, because the author does not have C/C++ skills and, secondly, because system speed was not an issue in this project, Perl was used instead. Many experiments were associated with a large-scale processing of textual data. Perl is very suitable for text processing tasks, since it is well integrated with the Unix operating system and fully supports regular expressions and pattern matching/replacement.

The testbed for the experiments was the Okapi experimental IR system, largely developed at City University. It implements the Robertson-Sparck Jones probabilistic model, described in chapter 4. The core of Okapi is the Basic Search System (BSS), which implements the probabilistic retrieval mechanism. It is accessible in the form of the command line interface i1+, either directly to the user from Unix shell, or from scripts for batch processing. The author wrote Perl scripts to interact with i1+ in batch mode, submitting all test queries, and receiving ranked document sets.

5.2.1 Global collocation analysis

There are two stages to these experiments. The first stage concerns itself with testing query expansion with global collocates. The second stage consists of statistical analyses of measures of collocation association – Mutual Information (MI) and Z score (see sections *5.4.1* and *5.4.2* below).

The first stage of this set of experiments was directed towards testing automatic query expansion with global collocates of query terms. The global collocates of a term are all words occurring within a fixed-size window (see section *5.3*) around every occurrence of a term in the corpus. Significance of association of these collocates with

the node term in question is estimated using statistical measures, in our experiments MI and Z statistics.

Global collocation analysis requires a pre-processing stage to extract global collocates of terms from the corpus. For a fully operational system that uses global collocates, a resource must be built, where each index term is associated with a list of its significant global collocates. For the purposes of testing query expansion with global collocates, we did not need to have lists of collocates for every indexing term. Instead we built a test resource only for terms from the test queries. Each query term was associated with all collocates from windows of 200 words around each of its occurrences in the collection. These collocates were then ranked separately by MI and Z statistics. Query expansion was tested under a range of conditions, with different numbers of top MI-ranked and Z-ranked global collocates.

The second stage of the experiments consisted of a range of regression analyses of MI and Z measures, in order to achieve a clearer understanding of their ability to select relevance-discriminating terms. Starting from the assumption that the Offer Weight (OW) (see section *4.1.2*), estimated given some relevance information, is the best method available in the probabilistic model for selecting good relevance-discriminating terms, we used regression analysis to test if it is possible to achieve term rankings similar to OW by using MI, Z and other term frequency data, i.e. without any relevance information.

5.2.2 Local collocation analysis

The second set of experiments was targeted at iterative query expansion with local collocates. Local collocation analysis can be used as a form of either relevance, or pseudo-relevance (blind) feedback. Because relevance feedback in the probabilistic model on the whole yields better results than blind feedback, it was decided first to test local collocation analysis as relevance feedback mechanism, then if it showed improvement over Okapi relevance feedback results, it could be tested as blind feedback.

In relevance feedback experiments we simulated user's relevance judgments by using 5 known (from TREC relevance judgments) relevant documents found among the top 1000 Okapi ranked documents following the initial run.

Local collocates were extracted from the windows around every occurrence of the query terms in relevant documents. Local and global variants of Z and MI scores (see *5.4.2*) were applied for collocate ranking. Expanded queries were built from a fixed number of the top ranked collocates of each query term.

A method of query expansion with local collocates and Okapi relevance feedback terms (i.e. selected using Offer Weight function) was also tested.

Relevance feedback was evaluated with a range of values for the following variables:

- Window size;
- Measure of collocation significance for ranking collocates;

- Number of top ranked collocates used in the query;
- Number of Okapi relevance feedback terms used in the query.

Another experiment involved deriving and testing a new collocation weighting measure – Collocate Relevance Weighting (CRW). It was derived from a contingency table containing collocation frequency parameters, on an analogy with the contingency table used by Robertson & Sparck Jones for the derivation of the Relevance Weight (RW) function. CRW was used in term selection on the analogy to the use of Offer Weight (OW) in the probabilistic model. (For OW and RW see sections *4.1.1* and *4.1.2*).

All above experiments were run within two types of experimental searching scenarios:

- *Retrospective searching*. 5 relevant documents for local collocation analysis are taken from the same document collection, which is then searched with the expanded queries;
- *Predictive searching based on half collections*. 5 relevant documents in the even half of the collection are used for the extraction of local collocates. Searching with the expanded queries is done on the odd half of the collection.

The expanded queries constructed in the experiments consisted of different types of terms. Retrieval evaluation does not give us information on how each category of terms contributes to performance. In order to gain a deeper insight into the influence each category has on performance we undertook the following analysis.

The categories of terms that could be found in the queries expanded by local collocates and Okapi relevance feedback terms are:

- Collocate;
- Collocate of 2/more query terms;
- Collocate of 1 query term;
- Okapi relevance feedback term;
- Original query term;
- Collocate of 2/more query terms and an Okapi relevance feedback term;
- Collocate of 1 query term and an Okapi relevance feedback term;
- Collocate and an Okapi relevance feedback term;
- Collocate and an original query term;
- Okapi relevance feedback term and an original query term;

The analysis was done by excluding each term in turn from each query, running this query and recording the difference in average precision between the complete query and the query without the term in question. According to this difference, the term was considered to improve, degrade or not affect performance. Results summed for each term category gave information on the level of its contribution to performance.

5.2.3 Lexical cohesion analysis using local collocations

The broad aim of this set of experiments was to understand the relationship between two document properties: its lexical cohesion and relevance to a query. More specifically, we are interested in the lexical cohesion between different query terms in a document, and whether the strength of lexical cohesion can predict documents' relevance. Our approach towards measuring the level of lexical cohesion between query terms is based on the method of lexical bonds analysis, whereby sentences are considered to be bonded if they have a minimal number of links between them (see section *2.3.3*).

The aim of sentence bonds analysis, however, is different from ours. Sentence bonds analysis aims to identify sentences in text which are semantically related. We want to identify if, given the query $Q = (q_1, q_2, q_3)$, query terms q_1, q_2, q_3, occurring in a document, are semantically related. We do this by identifying the similarity between their collocation environments. We build a collocation environment of query term q_i by merging the fixed-size windows around all occurrences of this term in a document. Collocation environments of the query terms are then compared using two criteria:

- The number of links they have;
- The number of token types they share;

Using either of these two methods, for each document we get a number of links or types shared by two or more query terms occurring in it. A *lexical cohesion score* (LCS) is then calculated for each document.

Our next step was to test whether the lexical cohesion scores have any relation to documents' relevance. We designed an experiment, wherein we compared sets of relevant documents with sets of non-relevant documents on the basis of their lexical cohesion scores.

Finally, a set of retrieval experiments was conducted, wherein we tested the use of local cohesion scores in document ranking. We evaluated a set of runs, in which documents retrieved by Okapi in the initial run were re-ranked by a linear combination function of LCS and Okapi document score.

5.3 Windowing technique

We define collocates of a single instance of the term in question as all words that occur within a fixed-length window surrounding this term. Each window is centred around a *node* term (see section *2.4.1*). A node can be any word in text whose collocates we want to identify. In our experiments collocates were identified for query terms, therefore a window is defined for each instance of each query term in a set of relevant documents (local analysis) or corpus (global analysis).

A window is defined as a fixed number of words to the left and right of the node. Ideally left and right sides of the window are of equal lengths, but, as will be described later, in practice it is not always the case. The choice of defining windows by counting the number of words, instead of using natural language constructs like sentences and paragraphs was made, first, because the lengths of the latter are highly

irregular, which will complicate the estimation of association strength between collocates. The second reason is that NL structures require more computational effort to delimit in text. For some corpus linguistics applications the order in which collocates occur in text is important, for example in machine translation, speech recognition, optical character recognition. Order-sensitive collocation statistics in such applications is necessary for making correct lexical choices in short-span lexical-syntactic structures. Church et al. [Church90, Church91, Church94], for example, were studying ordered collocations. Here, on the contrary, the order in which collocates occur together is unimportant, since we are interested in semantically related collocates that can occur anywhere within the large area surrounding the node. In other words, collocates of a term instance are all words that have either backward co-occurrence with it, i.e. occur within *S* word span to its left, or forward co-occurrence, i.e. occur within *S* word span to the right (figure 5.1).

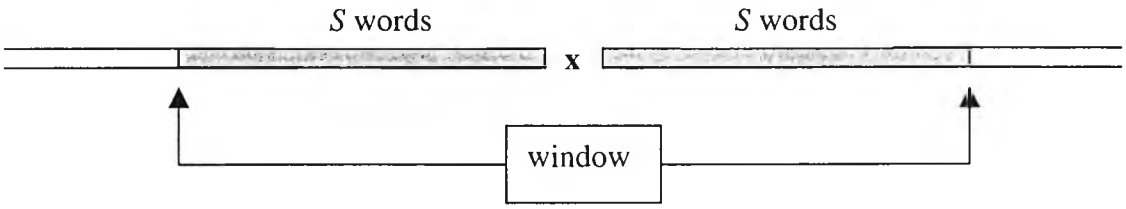


Figure 5.1. Window around node *x*, defined as spans of *S* words to the left/right of *x*

However, for two reasons, the windows actually used are often smaller than suggested by this distance. A window around term *x* may be truncated if either (a) it hits a document boundary (figure 5.2), or (b) it hits another occurrence of term *x* (figures 5.3 and 5.4). The latter truncation of the window is necessary to avoid duplicate extraction of the same word as a collocate of two instances of *x* when they occur near each other, i.e. when the distance between them is less than *S* words. If another instance of *x* is found after the node *x*, we truncate the window at this point (figure 5.3), if another *x* is found before the node *x* we ignore the left-hand half of the window altogether (figure 5.4).

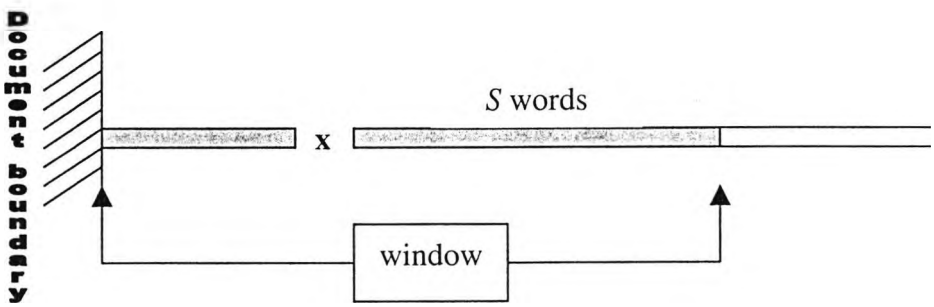


Figure 5.2. Window truncated by hitting the document boundary

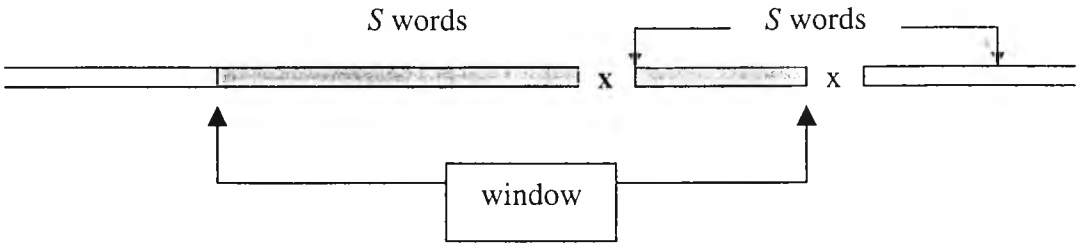


Figure 5.3. Right-hand half of the window truncated by hitting another occurrence of x after the node

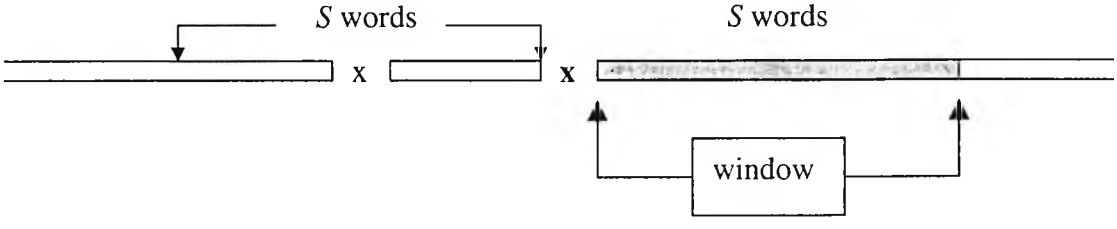


Figure 5.4. Left-hand half of the window ignored when another occurrence of *x* is found before the node

A decision must be made in the windowing technique regarding the size of the span (*S*) to the left/right of the node. As described in sections *2.4.2* and *3.3.2* the span sizes used in various projects vary widely from 1 (adjacency) to 400 words. The choice of the span size in collocational analysis is usually determined by which syntactical or semantic constructs of the NL text are under analysis, e.g. phrases, sentences, paragraphs, topics. Since we are interested in topical or semantic relations between words, the span size must be of the scale of a topic in text. A topic, as discussed in section *2.1.2* is a rather nebulous entity, which often cannot be indisputably delimited even by humans. Moreover it is not necessarily present in the form of an uninterrupted stretch of text, but can re-surface throughout the text, being interwoven with other topics. As discussed in sections *2.3.2* and *2.3.3* more complex approaches for topic detection have been proposed by using, for example, lexical chains. It is not the aim of this project to develop precise topic delimitation method for collocate selection, which can be rather computationally demanding, and hence not suitable for search-time use. Instead, a more crude and fast technique of collocate extraction from fixed-length windows is used, which is complemented by the second stage – selection of significant collocates via statistical measures.

The initial span chosen for the global collocation analysis experiments (chapter 6) was 100 words. This decision was motivated by the research of Beeferman et al. [Beeferman97] described in section *2.4.2*, who established that a word's influence on its environment stretches as far as several hundred words, leveling off at 400 words. In other words, they were able to prove that the distribution of words within several hundred words of the node is different from the global distribution of words in corpus.

The factors affecting this distribution at such big distances from the node cannot be lexical-syntactical, but semantic and topical. In subsequent experiments on local collocation analysis (chapter 7) other smaller span sizes were tried: 50, 25, 15 and 10 all of which were still sufficiently larger than short-span environment, where lexical-syntactic factors dominate word relations. The decision of not using span sizes larger than 100 was made, first, because the monotonic decay of the word distribution curve in Beeferman's experiments suggests that the influence of the node weakens with the distance, therefore increasing the chance of noise terms. Secondly, the spans used would still yield a sufficiently large number of terms needed for the experiments.

Since an ideal window is symmetrical, its size is $S + S + 1$, where S is the span size. However as the observed window sizes around instances of a given term in a document/corpus are variable (figures 5.2, 5.3 and 5.4), we calculate the average window size - v_x - around the term x. Average window sizes, as will be described in the next section, are needed to calculate collocation significance scores – MI and Z. Also, as will be described later, we defined two variants of Z and MI statistics: for the global collocation analysis, where collocates are found for every occurrence of x in the collection, and for the local collocation analysis, where collocates are identified for the instances of x in relevant documents.

In the global method, v_x is estimated by summing the observed window sizes around all instances of x in the corpus and dividing them by the frequency of occurrence of x in the corpus $f(x)$:

$$v_x = \frac{\sum_{i=1}^{f(x)} W_i}{f(x)} \qquad (5.1)$$

where W_i is the observed window around ith instance of x in the corpus;
$f(x)$ is the frequency of x in the corpus.

In the local method, v_x is estimated similarly by dividing the sum of observed windows of x in relevant documents by $f_r(x)$ – the frequency of x in relevant documents.

5.4 Choice of statistical measures of collocation significance

There exist several statistical measures for collocation selection. It is out of the scope of this project to evaluate all statistical measures. Instead we have chosen two statistics which are most commonly used in corpus linguistics [McEnery96] – mutual information (MI) and Z score. Moreover, these statistics are also used together, as in lexical analysis research by [Church91, Church94] and in [CobuildDirect] – a concordance/collocation on-line service, based on "The Bank of English" corpus.

Mutual information originated in the field of information theory [Fano61], and since then has been used extensively in a wide variety of applications, e.g., speech

recognition [Jelinek90], information retrieval (e.g. see [Van Rijsbergen77] in section *3.3.1*) and various uses of corpus linguistics like lexicography [CobuildDirect], lexical analysis, sense discrimination, and analysis of aligned corpora [McEnery96].

Z score is a statistic for hypothesis testing, i.e. for assessing whether a certain event is due to chance or not. When used for collocation selection, Z score tests whether the co-occurrence of two words is due to other factors than chance. It is very similar to a *t* score, the difference lying in the fact that Z is used with the data distributed normally. As will be described in more detail later, the large size of the corpus used in this project warrants normal distribution, hence *Z* score instead of *t*.

Both Church et al. [Church94] and Clear [Clear99] pointed out that *t* score and mutual information tend to bring to the top different kinds of collocations. *t* score tends to pick high frequency word combinations, and may have a drawback of showing syntactical collocations with functional words like '*by post*'. Mutual information highlights less frequent word combinations that are specific to both words, e.g. fixed phrases, some compound terms and proper names, e.g. '*Laurens van der Post*' (examples taken from [Clear99]). The drawback of MI is that it can reward very low-frequency corpus-specific collocates, that are not easily generalised across corpora.

Church et al. [Church91, Church94] used MI and *t* score in their study of synonymy and lexical substitutability. Specifically they were interested in the possibilities of identifying differences between near-synonyms from the patterns of their use in text, i.e. from the regularities of their co-occurrence with other words. They argued that MI is a better tool for finding associations between words, while *t* is good for identifying dissimilarities in the use of near synonyms.

Church et al. [Church94] and Clear [Clear99] suggested the combined use of MI and *t*, by intersecting the results obtained by both measures and selecting collocates scored highly by both of them. They claim that such selection will return the most significant collocates. More specifically, Church et al. pointed that it will reduce the number of unwanted terms, that MI and *t* can bring up, i.e function words (*t* score) or very low-frequent words (MI).

The following two subsections will describe MI and Z measures and our modified formulas, some of which were originally presented in [Vechtomova2000].

5.4.1 Mutual information (MI)

The mutual information score between a pair of words or any other linguistic units "compares the probability that the two words are used as a joint event with the probability that they occur individually and that their co-occurrences are simply a result of chance" [McEnery96, p.71]. The mutual information score grows with the increase in frequency of word co-occurrence. If two words co-occur mainly due to chance their mutual information score will be close to zero. If they occur predominantly individually, then mutual information will be a negative number.

The standard formula for calculating mutual information score is:

$$I(x,y) = \log_2 \frac{P(x,y)}{P(x)P(y)} \tag{5.2}$$

where $P(x,y)$ is the probability that words x and y occur together;
$P(x)$ and $P(y)$ are the probabilities that x and y occur individually.

MI statistic is usually applied where term x immediately follows term y in text, e.g. as used in [Church91, Church94]. The probability that two words occur as a joint event $P(x,y)$ is estimated as $f(x,y)/N$, where joint frequency - $f(x,y)$ denotes the number of times that y appears immediately after x. In this thesis a different interpretation of $f(x,y)$ is assumed – as the frequency with which y occurs either sides of x within the maximum distance of S words (where S is the span size). Therefore the standard MI formula was modified to provide for unordered co-occurrence within a distance more than one word. But the most important difference from the standard MI is the asymmetry of our approach. Standard MI is a symmetrical measure, i.e. $I(x,y) = I(y,x)$ as joint probabilities are also symmetrical: $P(x,y) = P(y,x)$. The asymmetry of our approach arises due to the use of average window sizes. As described above in section *5.3*, the actual window sizes around instances of a term x are often smaller than the ideal window size of $(S + S + 1)$. For this reason we use the average of all windows around term x – v_x to estimate the probability of occurrence of y in the windows around x – $P_v(x,y)$. However, if we were to start with y and consider the occurrences of x in the windows around y, we would replace v_x in the formula with v_y. In general these two are different.

The modified MI formula for the global method is:

$$I_v(x,y) = \log_2 \frac{P_v(x,y)}{P(x)P(y)} = \log_2 \frac{\frac{f(x,y)}{Nv_x}}{\frac{f(x)f(y)}{N^2}} \tag{5.3}$$

where $f(x,y)$ - joint frequency of x and y in the corpus;
$f(x)$ and $f(y)$ - frequencies of independent occurrence of x and y in the corpus;
v_x - average window size around x in the corpus;
N - corpus size.

The modified MI formula for the local method is:

$$Local\ I_v(x,y) = \log_2 \frac{\frac{f_r(x,y)}{Rv_x(R)}}{\frac{f_r(x)}{R}\,\frac{f_c(y)}{N}} \tag{5.4}$$

where $f_r(x,y)$ - joint frequency of x and y in the relevant documents;
$f_c(y)$ - frequency of y in the corpus;
$f_r(x)$ - frequency of x in the relevant documents;

$v_x(R)$ - average window size around x in the relevant documents;
N - corpus size;
R - size of the relevant set (in tokens).

While mutual information is useful in filtering out pairs of words whose joint probability of occurrence is greater than chance, it gives very limited information as to how far joint probability differs from chance. Very high mutual information scores generally indicate strong bond between two words, whereas lower scores can be misleading, especially with low frequencies. Therefore it is not safe to make assumptions about the strength of words' association without knowing how much of that association is due to chance.

5.4.2 Z score

Z score is a more reliable statistic: it gives us an indication with varying degrees of confidence that an association is genuine by measuring the distance in standard deviations between the observed frequency of occurrence of y around x, and its expected frequency of occurrence given the null hypothesis. For a chance pair of words in the conditions of low word frequencies we may misleadingly get a high mutual information score, whereas their Z score will not be high since the variances of probabilities will be large.

Our approach to measuring the significance of collocations with Z score is somewhat similar to Church's et al. use of a t statistic [Church91, Church94]. However, there are three main differences:

(a) We are interested in collocations within a substantial window around the starting node;
(b) The argument on which the measure is based is asymmetric: it considers, given a word x, the probability that word y will occur within the window. (The resulting formula is also asymmetric, for the same reasons as those leading to the asymmetry of the MI formula, discussed above);
(c) Because we are dealing with collocations over a large corpus, the small-sample characteristics which lead to the choice of the t statistic do not apply – we use the Z statistic instead.

We take as null hypothesis that the presence of x does not predict the presence or absence of y in the windows – that any location in these windows is exactly as likely to contain y as any other location in the corpus.

In the global method the total number of locations which might contain term y collocated with x is $v_x f(x)$. Under the null hypothesis, the probability that any given one of these locations contains y is $f(y)/N$. Thus the expected number of occurrences of y in these locations is the mean of a binomial distribution, $v_x f(x) f(y)/N$. Also, because the probability $f(y)/N$ is in general very small, the mean square error of this expected value (the variance of the binomial distribution) is approximately also $v_x f(x) f(y)/N$.

But we actually observe $f(x,y)$ occurrences of y within these windows around x. Therefore we can calculate a normal deviate (Z score) as

$$Z = \frac{f(x,y) - \frac{v_x f(x) f(y)}{N}}{\sqrt{\frac{v_x f(x) f(y)}{N}}} \qquad (5.5)$$

This score can be compared with normal distribution tables in the usual way.

Under the null hypothesis as formulated above, for small samples this could be interpreted as a *t*-score with $v_x f(x)$-1 degrees of freedom. In our case this will always be large enough to warrant the normal approximation. However, it does flag up one problem: suppose $f(x)$ were only one. It would then appear that we artificially inflated our sample by considering a window of size 100 words (say) either side, when the locations we are considering all relate to a single instance of x. Our response to this problem is simply to avoid using the method on terms with very small frequencies ($f(x) < 30$).

Church and Hanks use a different estimate for the variance, involving the co-occurrence frequency $f(x,y)$. Our asymmetric argument and explicit formulation of the null hypothesis suggest the variance based on the individual frequencies.

For the local method, we modified the above global *Z* function as:

$$Local\ Z = \frac{f_r(x,y) - \frac{f_c(y)}{N} f_r(x) v_x(R)}{\sqrt{\frac{f_c(y)}{N} f_r(x) v_x(R)}} \qquad (5.6)$$

where $f_r(x,y)$ - joint frequency of x and y in the relevant documents;
$f_c(y)$ - frequency of y in the corpus;
$f_r(x)$ - frequency of x in the relevant documents;
$v_x(R)$ - average window size around x in the relevant documents;
N - corpus size.

According to Church et al. [Church91] the threshold of significance of association between two collocates measured by *t* score should be no less than 1.65 standard deviations. In our experiments we adopted the same threshold for filtering out insignificant associations in both global and local analyses.

5.5 Document collection and topics

5.5.1 Choice of the database

The database chosen for the experiments in this project was FT 96 from TREC Disk 4. The collection contains news articles from the *Financial Times* newspaper from 1991 to 1994 inclusive. The size of the database is 210,158 documents, 565 MB.

The size of documents ranges from 1-2 sentences (articles of "News in brief" type) to several hundred words length. However many documents are several hundred words long.

The choice of the database was motivated by the fact that it contains many documents of reasonable length, what is essential to our experiments on long-span topic related collocations. It is hypothesised that the use of long-span collocational methods on short documents is not needed, since, first, short-span documents are more likely to be semantically homogeneous, i.e. contain just one topic, and secondly, large window sizes that capture semantic relations are likely to approximate or exceed the boundaries of short documents, in effect leading to document-wide term analysis.

Long documents are more likely to cover more than one topic, hence terms separated by large distances in such documents have more chance to be semantically unrelated than terms within shorter distances. As it is assumed that a user's query term more or less precisely describes the topic the user is interested in, the environment around the occurrence of this term in a document has more chance of being related to the user's request, than environments of many other terms in a document. Therefore restricting term selection by large-size windows around query terms in long documents will reduce the chances of getting unwanted terms describing unrelated topics.

The reason for choosing a database from TREC collection is the availability of topics and relevance judgements, which were used in evaluation.

5.5.2 Indexing the database

Since the Okapi retrieval system was used in the experiments, the FT 96 collection was indexed as an Okapi database.

There are two indexes in FT 96 database: dn and kw. The first index – dn (document number) is built from the "DOCNO" fields of database records, containing numbers of the documents. The second index – kw (keyword) is built from the "TEXT" fields, containing the full texts of documents.

Each term in the "TEXT" fields of records before being included into the index was subjected to two processes:

- Checking against the go-see-list (GSL);
- Stemming.

The GSL file used was compiled by Steve Walker for Okapi TREC runs. In the GSL file there are several categories of terms, which should be handled differently from simple index terms. Specifically, there are four categories:

1. Phrases;
2. Stopwords;
3. Semi-stopwords;
4. Synonyms.

Under the first category, word combinations which should be indexed as single index units are specified. In total the GSL file used contained 59 phrases. They include the following types of phrases:

- Some proper nouns like geographical names (e.g., Las Vegas, Saudi Arabia);
- Terminological expressions (e.g., global warming, genetic engineering);
- Idioms (e.g., black Monday, fat cat);
- Phrasal verbs (e.g., catch up, dress up).

The category of stopwords specifies terms that should be excluded from the index. There are 213 stopwords in the list, which are either function words (like articles and prepositions), or very frequent content words (e.g., go, longer, look, say, tell). Function words do not have lexical meaning of their own, therefore cannot be used as contents discriminators. Highly frequent content words have a lexical meaning, but it is usually a very general one. Such words can occur in virtually any context, what makes them incapable to act as content discriminators.

Semi-stopwords are indexed, but not used in relevance feedback. There are 107 semi-stopwords, which include numbers, years, some frequent content words (e.g., obtain, order, problem, time), names of months and days of week. Such terms could be useful in specific queries, therefore their exclusion from the index would limit search flexibility. However, due to their high frequency and multicontextual nature they are not likely to bring benefit to the relevance feedback process.

The fourth category contains sets of words that should be treated interchangeably in retrieval. Not all of them are synonyms in linguistic sense; specifically there are the following types of interchangeable sets of words:

- part-of-speech variants of the same lexeme (e.g., administrative, administration);
- alternative spellings (e.g., auto immune, autoimmune);
- abbreviations and full variants (e.g., CEO, chief executive officer);
- name of a country and nationality (e.g., Denmark, Danish, Dane);
- synonyms (e.g., visual display unit (VDU), visual display terminal (VDT) or star wars, strategic defense initiative (SDI)).

In the GSL file used, there are 443 synonym sets. During indexing Okapi assigns a unique code to each set, for example @0012 to represent [*Alps*, *Alpine*].

It is reasonable to question whether the results of collocation analysis with and without using lists of phrases and synonym sets would be the same. If the system supported full-scale phrase indexing, then the results can be expected to be significantly different from the system using single-term indexing. However, here, the sizes of phrase and synonym lists are small compared to the overall index size of 193,550, which leads us to assume that they are not likely to lead to significant differences in performance. Besides, the results of all collocation runs were compared to Okapi runs performed under the same conditions, what makes the comparative evaluations uniform.

The second process each index term undergoes is stemming. There are two stemming functions implemented in Okapi indexing software: weak and strong. Both are based on Porter's stemming algorithms [Porter80] and differ in the degree of reducing word forms to stems. This is a rather crude process, which does not necessarily bring words to their linguistically correct root morphemes; however this is not important for the retrieval task.

We used strong stemming, simply because it was also used in Okapi TREC runs.

In all our experiments, the collocates extracted from the corpus (in global analysis) or individual documents (local analysis) were subjected to both GSL look-up and stemming. Therefore the collocates we analysed are handled uniformly to index terms: they are stemmed, stopwords in the GSL file are discarded, and some collocates are recorded as GSL phrases or GSL synonym set codes.

There is another effect the GSL look-up has on windowing technique. In all experiments windows are built around query term instances after all terms were looked up in GSL file, i.e. stopwords were eliminated and (possibly compound) members of synonym sets were changed to codes. Therefore, we build windows by counting these normalised terms, and not the original words in the unprocessed texts.

5.5.3 Topics

50 TREC topics 251-300 were used as queries in our experiments. We created short requests from the contents of title fields of the topics. Such requests correspond to the type of briefly formulated queries, which are frequently submitted by average users in practice. Below is an example of a TREC topic:

> **Topic title:** Exportation of Industry
>
> **Description:**
> Documents will report the exportation of some part of U.S. Industry to another country.
>
> **Narrative:**
> Relevant documents will identify the type of industry being exported, the country to which it is exported; and as well will reveal the number of jobs lost as a result of that exportation.

In our experiments, queries were composed from the contents of title fields. The query for the above topic would be "exportation industry". Long queries, for example in previous Okapi experiments, are composed from the contents of all fields in a topic.

The first two sets of experiments explore the use of collocates for query expansion. Short queries are good candidates for query expansion, as there is more scope for expansion, than with long queries. However, it is not obvious from previous Okapi experiments that they perform better after query expansion than, say, medium or long queries [Sparck Jones98]. Therefore other query types could be also evaluated, if there is evidence for any significant performance improvement for short queries.

In the third set of experiments – lexical cohesion analysis using local collocates – short queries were believed to be more suitable for the technique we developed. A document's lexical cohesion score is estimated from the number of types or links shared by collocational environments of two or more query terms. If we compare collocational environments of a large number of query terms, then we are more likely to get a high number of topic-neutral words as shared links or types, which do not provide evidence that query terms are lexically coherent. In this case we would need to apply some statistical criteria for the selection of collocates, as in the first two experiment sets.

Request formulations were automatically built from the title fields of topics by applying the same two-stage process – GSL look-up and stemming, as was used in collection indexing.

6. Global collocation analysis experiments

6.1 Introduction

The aim of this set of experiments was to examine the hypothesis that global long-span collocates of query terms, used for query expansion, can lead to performance improvements over the initial query.

Initial stage was to prepare the necessary platform for the experiments, i.e. to build a resource of global collocates. Then two stages of experiments were conducted:

1. Retrieval experiments on query expansion with global collocates;
2. Statistical analysis of collocation association measures MI and Z, which were used for selecting collocates for query expansion.

The first stage of experiments is directly targeted at testing the above hypothesis. The objective was to evaluate the retrieval performance of the expanded queries and to compare it to the performance of the initial queries. The objective of the second experiment stage was to get a statistical evidence about the capability of collocation association measures (MI and Z) applied to global collocates, of predicting their usefulness as relevance-discriminating terms. The second stage is seen as a deeper quantitative analysis of the global collocates, which throws light on the interrelationship of their statistical characteristics, such as Offer Weight, MI, Z and collection frequency.

6.2 Construction of the database of global collocates

The first step in the construction of the database of global collocates was to create a corpus from FT 96 collection. FT 96 has been indexed as an Okapi database. The corpus was built by extracting every record from the database and parsing it using the Okapi parsing algorithm, consisting of GSL look-up and stemming (see section *5.5.2* above). The corpus was recorded as a single text file, consisting of concatenated records. Each record, delimited with record boundaries, is represented as a sequence of tokens – stems and GSL synonym codes of the original wordforms appearing in text. The order of tokens is the same as the order of their corresponding wordforms in the text of each record.

If we were to build a fully operational system supporting query expansion with global collocates, we would have to identify global collocates for each index term in the database. Since our task was laboratory evaluation of this technique, in which we used 50 request formulations, we had to identify global collocates only for the query terms from these request formulations. This significantly reduced our time for building the global collocates database, yet did not impose any limitations on the experiments conducted using this collocation database.

Request formulations were constructed from 50 TREC topics as described in section *5.5.3*. Each request formulation is represented as a list of stems and/or GSL synonym codes of the corresponding query terms. For convenience we will refer to stems/GSL codes of words in the request formulation simply as query terms.

The next step was to extract for every occurrence of every query term in the corpus all its collocates using the windowing technique presented in section *5.3*. The window size set for global collocates extraction was 201 (for rationale of this decision also see section *5.3*). For each identified collocate of a query term instance we recorded a collocation pair: [*query term*, *collocate*].

After all collocation pairs have been recorded, the next step was to rank collocates of each query term by two association measures – MI and Z (see sections *5.4.1* and *5.4.2* above). MI and Z scores were calculated only for those collocates which have individual collection frequency greater than 30 (see section *5.4.2*). Terms with smaller frequencies were discarded from the collocation lists. For each query term two files were created – one with its collocates ranked in descending order of their MI scores, the other – containing collocates in descending order of their Z scores. For samples of MI- and Z-ranked collocation lists see Appendix A.1.

In the MI-ranked lists, collocates with $MI > 0$ are considered to have association greater than chance. The higher the collocate's MI score, the stronger is its association with the node. It was observed that on average the number of collocates with $MI > 0$ is quite large for most query terms, well exceeding 100 terms. For most query terms, top 100 collocates ranked by MI also tended to have significant Z scores. As noted in section *5.4.2* we consider $Z > 1.6$ as significant.

In the Z-sorted list, again the number of significant collocates was quite large – most query terms had more than 100 collocates with $Z > 1.6$. Z scores in the top of the list tended to be quite high, top collocates of some query terms had $Z > 100$. However, some terms in the top 100 of the Z-ranked lists had rather low MI scores ($MI \approx 0$).

The next decision to be made was whether to use a fixed number of top-ranked collocates in MI- and Z-ranked lists for query expansion, or collocates with scores above a certain threshold. The problem with the latter approach is that the scores are highly variable for collocates of different query terms. Top collocates of some query terms have, for example, Z scores well above 100 (table 6.1), others have much lower scores (table 6.2). MI scores have smaller variance than Z, but the values of top scores and the numbers of collocates with MI above a certain value are also too variable. This strategy would result in a huge number of collocates selected for some terms, and none for others. For this reason we decided to select a fixed number of top ranked collocates for query expansion from both MI- and Z-ranked lists. A very large margin of significant collocates in both MI- and Z-ranked will still exceed the fixed number of top collocates we need for query expansion. This means that all collocates we choose either from Z-, or MI-ranked lists will have significant values of the corresponding statistic.

collocate	Z score
tobacco	383.515
marlboro	218.875
smoke	178.732
bat	172.081
smoker	158.420
morris	154.348
nicotin	145.943
rjr	145.569
seita	136.464
@0018	118.345
filip	111.361
rothman	107.858
nabisco	103.962
brand	102.078
bppc	88.9071
morriss	80.7002
packet	70.8662
rj	66.7835
habgood	65.9124
bunzl	64.0986

Table 6.1 Top 20 collocates of the query term *cigarette* (topic 257) ranked by Z score

collocate	Z score
narva	43.6465
curzon	39.3772
seti	33.4198
estonia	30.8962
estonian	30.5858
immigr	28.7626
parti	27.7749
hindus	24.8815
camacho	24.8815
kashmiris	24.5734
voter	23.8587
zoe	23.3030
karajan	23.1119
baird	22.8774
haider	22.7896
@0111	21.9208
poe	21.4189
thian	21.1312
boarder	21.0979
cultur	20.5486

Table 6.2 Top 20 collocates of the query term *alien* (topic 252) ranked by Z score

6.3 Lexical-semantic Analysis of Collocations

To understand the tendencies MI and Z statistics show in the collocates ranking a comparative lexical-semantic analysis of collocations has been carried out. Specifically we looked into the following aspects:

- differences between MI and Z in selecting collocates for general and more specific terms;
- collocates of polysemantic words;
- differences between collocates and terms found in manually engineered term structures.

6.3.1 Differences between collocates selected with MI and Z statistics

Many top collocates ranked by Z score tend to be related to the nodes semantically and there is a large number of collection-independent collocates. This tendency was observed in collocation lists for both general and specific terms. The picture is distinctly different in the MI-ranked term lists. For general terms MI tends to pick rare collection-dependent collocates, which are predominantly proper names, whereas for more specific terms the results yielded by MI statistic were resembling those of Z score, i.e. it selected more general conceptually associated terms. For example the types of significant collocates for the term *acquire* in MI-ranked and Z-ranked lists have marked difference (table 6.3).

MI	Z
Noorda 4.19 (surname)	**acquisition** 141.64
Huntsman 4.18 (company name)	**pound** 114.25
Nextel 4.18 (company name)	**stake** 104.267
Gartland 4.14 (surname)	**company** 102.65
Revco 4.08 (company name)	**group** 99.77
Viglen 4.07 (company name)	**purchase** 84.33
Tampella 4.02 (company name)	**share** 84.17
Cinzano 4.02 (company name)	**profit** 66.67
Conspress 3.98 (company name)	**operation** 61.70
CCL 3.91 (company name)	**business** 59.141

Table 6.3. Lists of top collocates for the term *acquire* sorted by MI and Z statistics

The table illustrates that all top MI-ranked terms are proper names. The inspection of documents containing the instances of the node term together with the listed company names or surnames showed that their topics are all related to the idea of acquisition/purchase of companies by other enterprises. All listed company names and surnames denote parties in company acquisition transactions.

The Z-sorted list in contrast contains more general terms, of the type expected to be found in manually engineered term structures. They are all related to the sense of *acquire* as *obtaining by means of a financial transaction*. This is explained by a large number of economy/finance related documents in the FT 96 collection.

Another example is collocations for the synonym group @0104 (*environment, environmental*). It illustrates similar tendency of MI to select specific and rare terms and of Z to select higher frequency general terms (table 6.4).

MI	Z
tribal 5.98 **GEF** 4.34 (Global Environment Facility) **ecolabel** 4.22 **Lalonde** 4.17 (B. Lalonde, Environment Minister, France) **VOC** 4.15 (Volatile Organic Compounds) **Meana** 4.14 (Ripa di Meana. EC Environment Commissioner) **Topfer** 4.13 (C.Topfer, Environment Minister, Germany) **CPRE** 4.13 (The Council for the Protection of Rural England) **Ripa** 4.11 (Ripa di Meana) **DSD** 4.03 (Duales System Deutschland - scheme adopted by companies in Germany to recover waste from households and reuse the raw materials) **UNEP** 4.01 (UN Environment Programme) **deforest** 4.00 **LRB** 3.98 (London Residuary Body)	**waste** 152.12 **pollution** 150.42 **emission** 146.35 **recycle** 131.15 **energy** 101.45 **carbon** 100.39 **water** 95.49 **pollute** 93.37 **dioxide** 92.26 **Gummer** 86.27 (R.Gummer, the Minister of Agriculture) **forest** 86.10 **Greenpeace** 84.20 **clean up** 79.43

Table 6.4. Lists of top collocates for the synonym group @0104 (*environment, environmental*) sorted by MI and Z statistics

Here MI-sorted list is also dominated by low-frequency collocates, most of which are proper names, specific to this particular collection. All of them, as evident from the comments, are topically related to the sense of the term *environment* as:

> *the complex of physical, chemical, and biotic factors (as climate, soil, and living things) that act upon an organism or an ecological community and ultimately determine its form and survival*[*]

Z score highlights more general terms with higher collection frequency for the above sense of the node. Many of these terms, as seen from the table, are conceptually related to this sense and could be considered key terms in the textual topics about environmental protection issues.

Similar pattern is evident in the collocation lists for the term *education* (table 6.5).

MI	Z
Ofsted (The Office for Standards in Education) 5.00365 **GNVQ** (General National Vocational Qualifications) 4.98525 **Blatch** (Lady Blatch, schools minister) 4.91188 **Natfhe** (The University and College Lecturers' Union) 4.85370 **Gruchy** (Nigel de Gruchy, general secretary of the National Association of Schoolmasters/Union of Women Teachers) 4.79819 **GCSE** 4.79289 **Educationalist/educationally** 4.78483 **truancy** 4.76496 **A-level** 4.72806 **Nasuwt** (National Association of Schoolmasters/Union of Women Teachers) 4.71952	**school** 351.413 **teacher** 220.438 **student** 169.629 **pupil** 155.444 **curriculum** 155.220 **university** 147.055 **college** 144.692 **A-level** 141.422 **vocation** 130.109 **Patten** (Christopher Patten, chairman, Conservative party) 120.885

Table 6.5. Lists of top collocates for the term *education* sorted by MI and Z statistics

In contrast to the previous examples of general terms, specific terms have a different distribution of MI and Z significance scores, with more overlap between the top collocates. Typical examples are collocation lists of the terms *nitrogen* (table 6.6) and *gene* (table 6.7).

[*] the definition is taken from Merriam-Webster Online Dictionary (http://www.m-w.com)

MI	Z
NOx* (nitrogen oxides) 9.26984	**oxide*** 200.956
monoxide* 9.24756	**emission** 181.221
particulate* 9.20253	**monoxide*** 168.745
superconductor* 9.18177	**particulate*** 162.550
legume 8.68129	**dioxide** 154.997
gasification 8.65821	**NOx*** 154.909
urea 8.56510	**sulphur** 152.579
ammonia* 8.24317	**superconductor*** 115.374
autocatalyst 8.17606	**carbon** 111.171
superconduct(ing/ivity) 8.11193	**diesel** 102.248
soot 8.05053	**pollution** 102.000
oxide* 7.99091	**fertiliser** 94.1899
co-generation 7.90757	**hydrocarbon** 89.7964
phosphor 7.90368	**pollute** 89.7602
nitrate* 7.85498	**nitrate*** 84.3515
lolly 7.81349	**ammonia** 81.3761

Table 6.6. Lists of top collocates for the term *nitrogen* sorted by MI and Z statistics (* terms are top ranked in both lists)

MI	Z
genome* 9.11	**genetic*** 426.08
PCR 8.86 (polymerase chain reaction)	**DNA*** 261.99
transgenic* 8.79	**genome*** 187.77
NIH* 8.77 (National Institute of Health)	**therapy** 165.13
Venter 8.77 (Craig Venter, one of America's leading gene researchers)	**protein** 156.88
	cell 145.23
fibrosis* 8.77	**fibrosis*** 144.57
cystic* 8.77	**cystic*** 144.57
chromosome* 8.68	**transgenic*** 132.89
DNA* 8.65	**chromosome*** 131.04
Lockhart 8.60 (Gene Lockhart)	**NIH*** 123.71 (National Institute of Health)
genetic* 4.39	

Table 6.7. Lists of top collocates for the term *gene* sorted by MI and Z statistics (* terms are top ranked in both lists)

As it can be seen from the two above tables, 7 out of 11 top collocates of the term *nitrogen* and 8 out of 11 collocates of the term *gene* occur in both MI and Z lists. The majority of collocates in MI and Z lists of both nodes represent domain-independent semantic associations. The tendency of MI to give high scores to low-frequency collocates like proper names applies, though to a less extent, to specific terms too (as evident from table 6.7). Selection of low-frequency associations has, however, its downside. One of the collocates picked by MI for the term *gene* – *Lockhart* – is a surname of a person whose first name is '*Gene*'. Although words *gene* and *lockhart* co-occur in the collection only nine times, the term *gene* is very specific to *lockhart,* i.e. the latter term has a very distinct pattern of co-occurrence with the former among its collocates. Z also ranked *lockhart* high, but its position is relatively low compared to the position in the MI-ranked list.

6.3.2 Collocates of polysemantic words

The example mentioned in the previous passage spotlights a significant problem of automatic term extraction, namely, multiple word senses. Statistical methods of

collocate extraction do not provide for word sense disambiguation, therefore both MI- and Z-ranked collocation lists for polysemantic query terms can feature collocates related to different word senses of these terms. Collocates of the term *pyramid* provide a good illustration of the co-occurrence patterns of polysemantic words (table 6.8).

MI	Z
Amway[1] 9.24 (the name of the pyramid selling scheme) **Caritas**[1] 9.06 (the name of the Romanian pyramid scheme) **Mavrodi**[1] 8.97 (surname of the head of the Russian pyramid company MMM) **Cluj**[1] 8.63 (hometown of Caritas) **MMM**[1] 8.32 (the name of the Russian pyramid company) **Projet**[2.2] 8.17 (Grands Projets) **Alchemy**[1] 8.12 (the name of the pyramid company) **Louvre**[2.2] 7.84 **Angkor**[2.1] 7.79 **Tyzack**[1] 7.75 (surname of the head of a pyramid selling scheme)	**MMM**[1] 184.46 **Caritas**[1] 156.58 **Mavrodi**[1] 143.59 **Amway**[1] 125.19 **Louvre**[2.2] 122.75 **Alchemy**[1] 88.09 **Cluj**[1] 84.28 **Projet**[2.2] 71.98 **Luxor**[2.1] 50.94 **Egypt**[2.1] 50.31

Table 6.8. Lists of top collocates for the term *pyramid* sorted by MI and Z statistics

The collocates listed in table 6.8 refer to the following senses of the word *pyramid*:

1. financial scheme or company
2. architectural construction (2.1. Egyptian pyramids; 2.2. Glass pyramids in Louvre courtyard)

Because many documents in the FT 96 collection are financial and business newsarticles, the predominant number of significant collocates refer to the first sense of the word *pyramid* – financial scheme or company.

6.3.3 Comparison of collocates with terms from engineered term networks

Comparison of statistically formed collocation lists with engineered term structures – thesauri and lexical networks – showed a rather insignificant overlap of terms. We compared MI- and Z-ranked collocation lists with WordNet term relations and INSPEC thesaurus entries. For a brief description of WordNet see section 3.1.1 earlier in this thesis. WordNet contains general lexicon of the language, INSPEC thesaurus, however, is limited to the physics domain. In our analysis we had to select only those terms which are present in both INSPEC and WordNet.

First we analysed a polysemantic word *pressure*. All top collocates of *pressure* in both MI and Z lists (table 6.9) are related to either of its two general lexicon senses:

1. **pressure** - a force that compels;
2. **pressure** - imperativeness, insistence, press (the state of urgently demanding notice or attention)*.

[1, 2.1, 2.2] collocates related to the corresponding senses of the term *pyramid*

* definitions are taken from WordNet

MI	Z
PWR 3.28301	**rate** 111.394
IDO 2.94352	**inflationary** 95.2335
blocker 2.76148	**inflation** 76.6912
inflationary 2.65797	**Bundesbank** 73.6852
VVER 2.62802	**under** 71.3450
EAP 2.61010	**ERM** 66.2900
UAC 2.57057	**dmark** 65.5720
VOC 2.55431	**currency** 60.1104
nonft 2.54556	**cut** 59.3700
micromachine 2.54556	**price** 54.8346

Table 6.9. Lists of top collocates for the term *pressure* sorted by MI and Z statistics

Sense 1	Sense 2
Synonyms	
force	imperativeness, insistence, press urgency
Coordinate terms (terms with the common hypernym)	
force duress lifeblood wheels	urgency hurry, haste criticality, criticalness, cruciality
Hypernyms	
force influence power, powerfulness, potency quality attribute abstraction	urgency necessity need, demand condition, status state

Table 6.10. Terms related to the above two senses of the term *pressure* in WordNet

There are no terms in common with either INSPEC, or WordNet. The most striking fact about the selected collocates (table 6.9) is that most of them belong to the subject domain of economy/finance, as was already noted for some other general terms in previous sections. Since INSPEC contains terms in the physics domain, the lack of any common terms with the collocation lists is self-explanatory. As for the lack of common terms with WordNet, one of the reasons is that WordNet contains a very restricted set of relations – synonymy, hyponymy, meronymy and antonymy (table 6.10). The majority of collocates do not fit into these categories. They are related to the node situationally through a wide range of lexical-semantic relations, which are often difficult or impossible to classify (cf. the attempt to classify collocates by [Halliday76] in section *2.4.1*). Moreover, engineered term networks do not contain proper nouns, which are common especially in MI lists.

Comparison of the collocation lists and WordNet/INSPEC entries for technical terms, e.g. *fuel, plutonium, uranium* also showed little overlap. There were no matching terms for the top 12 collocates of *plutonium* and *uranium* in INSPEC and hyponymical, holonymical and synonymical relations in WordNet. Collocates of the term *fuel*, top ranked by MI, did not have any matching terms in either INSPEC, or WordNet , however the top 12 collocates selected by Z score for the term *fuel* contained 1 word which matched an entire term in INSPEC and 3 words which matched parts of the compound terms in INSPEC or WordNet (table 6.11). This

highlights one more important difference between engineered term networks and collocates – each term in either INSPEC, or WordNet represents a concept, therefore complex concepts are represented by compound terms. Collocation lists built in this project, in contrast, always contain single words.

Collocates of the term **fuel** selected by Z score	Completely/partially matching terms in INSPEC and WordNet
diesel 156.83 **energy** 138.17 **reactor** 137.04 **coal** 118.01	**diesel oil** (hyponym) WordNet **energy resources** (broader term) INSPEC **fission reactor fuel** (narrower term) INSPEC **coal** (narrower term) INSPEC **coal gas** (hyponym) WordNet

Table 6.11. Top terms in Z-ranked collocation list matching INSPEC and WordNet terms related to the term *fuel*

6.4 Retrieval experiments

6.4.1 Experimental design

The experiments were designed to evaluate separately query expansion with Z- and MI-ranked collocates. In section *6.2* the rationale was given for using a fixed number of top ranked collocates for query expansion, instead of setting a threshold on collocation scores. The general mechanism of query expansion with global collocates we used is to take the top *n* collocates of each query term ranked by one of the statistics, merge them together and remove duplicates. The original query terms were kept in the expanded query.

First it was decided to evaluate top 8 and 16 ranked collocates for each query term. If the results suggest any significant improvement, then other numbers of collocates could be considered.

Query expansion with global collocates was implemented using the Okapi system by creating databases of collocates in Okapi format. Four separate collocation databases were built:

top8-mi - top 8 MI-ranked collocates;
top16-mi - top 16 MI-ranked collocates;
top8-z - top 8 Z-ranked collocates;
top16-z - top 16 Z-ranked collocates.

Each database contains one record for each query term. Each record has three fields: dn (record number), kw (node) and co (collocates). The kw field of a record contains a node, and the co field contains a list of its collocates.

Two indexes are created for each database: dn – index of the record number field (dn), and te – index of the node field (kw). Each record was, thus, indexed by a single node term in the kw field.

Okapi collocation databases serve as an intermediate layer in the existing querying technique. When the initial query is submitted, it is searched first against one of the

collocation databases. Each query term matches a single record by the indexable node term in the kw field. The contents of both kw and co fields of all found records is then merged together and, after removing duplicates, submitted to FT 96 Okapi database as the expanded query (fig.6.1). For the Perl script implementing this algorithm see Appendix D.1.2.

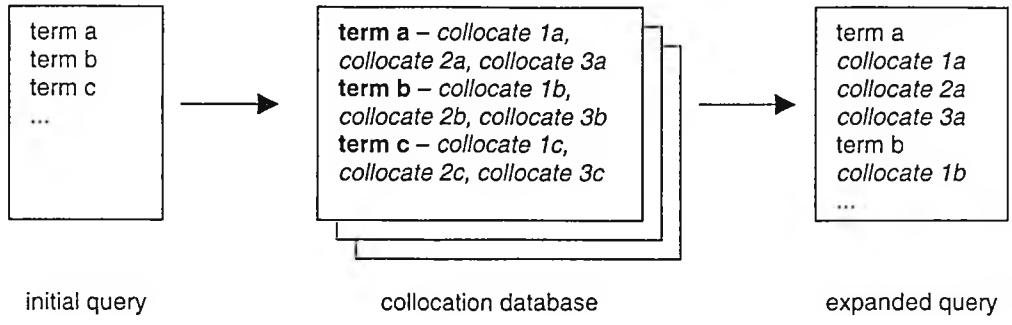


Figure 6.1. Query expansion with global collocates

The collocation database was searched using a simple unweighted search, since there is only one record corresponding to each query term. The expanded query was searched against FT 96 using weighted function bm2500 without relevance information, which instantiates Robertson-Sparck Jones' Combined Weight formula (see section *4.1.1*). The value for the tuning constant k_1 (section *4.1.1*), controlling the effect of within-document frequency (TF_i), was set to 1.2. Another tuning constant b, controlling document length normalisation, was set to 0.75. These values proved to work well in the previous Okapi runs.

First 1000 documents of each ranked set were evaluated using the trec_eval program and a file with TREC relevance judgements. Out of 50 topics (250-300), 6 topics (262, 270, 276, 279, 281, 296) have no relevant documents, and were ignored by trec_eval. For this reason all evaluation results are based on 44 topics.
We evaluated 4 query expansion runs:

top 8 MI – query expansion with top 8 MI collocates per query term;
top 8 Z – query expansion with top 8 Z collocates per query term;
top 16 MI – query expansion with top 16 MI collocates per query term.
top 16 Z – query expansion with top 16 Z collocates per query term.

The results of expanded query runs were compared to the results of Okapi unexpanded weighted search with the initial queries. The search was done also with bm2500 with the same settings for tuning constants.

6.4.2 Analysis of results

All query expansion runs – top 8 MI, top 8 Z, top 16 Z and top 16 MI – performed worse than Okapi run without query expansion, as the summary of trec_eval results shows in table 6.12.

	no expansion	top 8 MI	top 8 Z	Top 16 MI	top 16 Z
Retrieved	42686	43534	44000	44000	44000
Relevant	1583	1583	1583	1583	1583
Relevant retrieved at 1000	632	573	520	526	504
Average precision (non-interpolated) for all rel docs (averaged over queries)	0.1310	0.0432	0.0375	0.0344	0.0340

Table 6.12. Summary of retrieval results for query expansion with global collocates

The analysis of runs top 8 MI and top 8 Z by query shows that top 8 Z improved 7 queries, did not affect 3 and hurt 34 queries. top 8 MI improved 3, did not affect 2 and hurt 39 queries.

A large difference between the performance of all expanded runs and the unexpanded Okapi run suggested that fine tuning of variables such as the number of query expansion terms will not result in significant performance improvements over Okapi search with initial queries. Therefore, to decide whether further retrieval experiments with different variable values are necessary, it was decided to gain a deeper understanding of the potential of collocates to retrieve relevant documents. To achieve this we undertook a series of statistical analyses, described in the following section.

6.5 Statistical analysis

6.5.1 Presence of global collocates in relevant documents

We decided first to find out how many collocates occur in relevant documents. We did this by calculating for each collocate its term selection value in the probabilistic model – Offer Weight (OW), described in section *4.1.2*. Offer Weight below or equal to zero indicates that the term has no occurrence in relevant documents. To calculate OW scores of collocates, first their Relevance Weights (RW) have to be determined. We used the Okapi term weighting operation, which implements RW formula (section *4.1.1*). Weighting was done with complete relevance information, i.e. r_i = the number of all relevant documents with term i for a given query, R = total number of relevant documents for a given query. Relevance information was obtained from TREC relevance judgements. Offer Weight was then calculated by multiplying RW with r_i (the same as for RW).

Occurrence of collocates in relevant documents was calculated for top 30 MI, top 30 Z and top 100 Z collocates (table 6.13).

	Total number of collocates	Collocates with OW ≤ 0	Percentage of collocates present in relevant documents
Top 30 MI collocates	4080	3750	8%
Top 30 Z collocates	4080	2589	36.5%
Top 100 Z collocates	13261	9277	30%

Table 6.13. Presence of global collocates in relevant documents

We then conducted a correlation analysis between MI and OW on the top 100 MI collocates, and Z and OW on the top 100 Z collocates. It was then decided that a more suitable statistical method for the analysis of these variables would be regression analysis, described in the following section.

6.5.2 Regression analysis

Offer Weight is a standard term selection value in Okapi query expansion following relevance feedback. Previous experiments showed that Okapi relevance feedback achieves significant performance improvements over unexpanded runs. Based on these results we assumed that Offer Weight is the best available method in the probabilistic model to identify good relevance-discriminating terms for query expansion. The higher the Offer Weight, the stronger is the term's relevance discriminating capability. OW, however, is calculated when some relevance information is obtained. We carried out regression experiments in order to find out whether it is possible to predict OW from the statistical information on collocates available to us prior to relevance feedback.

We have the following variables characterising each collocate:

1. **MI** – MI score;
2. **Z** – Z score;
3. **JF** – joint frequency of the collocate and the node term ($f(x,y)$ in MI and Z formulae);
4. **NOPOS** – number of postings (documents) containing the collocate in question.

Another variable – individual term frequency – was considered to give similar information as NOPOS, therefore was not used in the regression.

Multiple linear regression is a suitable technique to examine whether the above variables can be used to predict Offer Weight. The SPSS package was used for this purpose.

To explore the relationship between the independent variables above and the response variable OW, we first ran a regression on all four variables. All significant collocates (with $Z > 1.6$) of all query terms in 50 topics were used in the analysis. The summary statistics of this regression run are presented in table 6.14.

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.211(a)	.045	.045	7.9502
a Predictors: (Constant), NOPOS, Z, MI, JF				

ANOVA

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	1915569.482	4	478892.370	7576.645	.000(a)
	Residual	41011024.834	648843	63.206		
	Total	42926594.316	648847			
a Predictors: (Constant), NOPOS, Z, MI, JF						
b Dependent Variable: OW						

Coefficients

Model		Unstandardized Coefficients		t	Sig.
		B	Std. Error		
1	(Constant)	.181	.018	10.153	.000
	MI	-.309	.010	-30.805	.000
	Z	.152	.001	115.846	.000
	JF	1.787E-04	.000	26.518	.000
	NOPOS	1.090E-04	.000	49.193	.000
a Dependent Variable: OW					

Table 6.14. Summary for regression using MI, Z, JF and NOPOS to predict OW

The significance values in the table above show that all four regression coefficients are significantly different from 0. Z coefficient indicates that Z is the best predictor variable, while negative coefficient for MI indicates that it has negative correlation with OW. The positive intercept value shows that this four-variable model has some explanatory power.

Since Z appeared to be the best predictor variable, we re-ran the regression dropping all other variables in turn. The following regression models were tried:

1. Z, JF, NOPOS
2. Z, JF
3. Z, NOPOS
4. Z, MI
5. Z

The statistical characteristics of these models are given in Appendix A.3.

Comparison of the squared multiple correlation (R Square) values for all models showed that the initial four variable model (MI, Z, JF, NOPOS) has the best explanatory power. It was then decided to conduct a retrieval experiment with a four-variable regression equation as a collocate-ranking function. The following regression equation was used to calculate predicted Offer Weights (*PREDICTED OW*) for collocates in the analysis:

$$PREDICTED\ OW = .181 - .309 * MI + .152 * Z + 1.787E\text{-}04 * JF + 1.090E\text{-}04 * NOPOS$$

Collocates of each query term were then ranked by their PREDICTED OW scores. Since top 8 collocates per query term showed better results in the initial query expansion runs, than top 30, we also used top 8 PREDICTED OW-ranked collocates.

The algorithm for query expansion and the values for bm2500 tuning constants were the same as in the previous runs. The summary of results for this run is given in table 6.15.

	PREDICTED OW
Retrieved	44000
Relevant	1583
Relevant retrieved	554
Average precision (non-interpolated) for all rel docs (averaged over queries)	0.0364

Table 6.15. Summary of retrieval results for PREDICTED OW run

The recall/precision values in the table show that PREDICTED OW performs similarly to top 8 MI and top 8 Z runs.

Following these results, we conducted two more regression analyses. The first regression was run to predict OW on only those terms whose JF $\geq$ 30. Four explanatory variables were used as before. The second regression was run to predict RW on the data consisting of both collocates and original query terms. The explanatory variables included four variables as before plus another variable QT (query term) – a binary variable with values: 1 – the term is a query term and 0 – the term is a collocate. The characteristics of these models are given in Appendix A.3. All these models had weaker explanatory power than the initial four-variable model to predict OW. Retrieval run on the scores predicted by one of them also proved to be worse than PREDICTED OW run.

6.6 Concluding remarks

The hypothesis examined in this set of experiments was:

Hypothesis 1: Expansion of the initial query with statistically significant global collocates of query terms results in significant performance improvement over the initial query evaluated under the same conditions.

The experiment results achieved do not support this hypothesis. Retrieval performance of the top ranked collocates was significantly worse than no query expansion. Regression on the statistical parameters of collocates to predict their value as relevance-discriminating terms showed some explanatory power, although retrieval

results of collocates ranked by the best regression equation were very similar to using Z or MI alone.

Regression analysis also suggested that Z overall is a better statistic for predicting collocates' relevance discriminating power, than MI. The fact that the top 30 MI ranked terms only occur in 8% of relevant documents, in contrast to 36.5% for top 30 Z collocates, also suggests that Z is more suitable for ranking collocates in query expansion.

One conclusion was that the use of global collocates alone without any other information did not look promising. As mentioned earlier in this section, in view of such big difference in performance between expanded and unexpanded queries, fine tuning of different parameters, like the number of query expansion terms or ranking function, was not expected to lead to significant improvements.

A possible reason why global collocates performed badly was the fact that collocates come from the many different contexts in which query terms occur throughout the collection. Since the number of non-relevant documents is much larger than relevant, only a small proportion of these contexts occur in relevant documents and have anything to do with the query topic. Even occurrences of the same sense of the query term can be used to describe a wide variety of topics. If, for example, a term has very broad lexical meaning and occurs in a wide range of topics, then its significant collocates can relate to any of these topics, not necessarily to the one meant in the query. And, of course, the problem is aggravated further if the query term is polysemantic.

Many of the query terms we used have fairly broad lexical meanings. To a certain extend they narrow each other's meaning through combined use in the query. If many collocates of the query terms happen to be from unrelated topics, their addition simply introduces more noise to the query and degrades its performance.

It was realised that some solution is needed to reduce the number of collocates from unwanted contexts. One direction to follow is to extract collocates not from the corpus, but from a limited number of documents, for which there exists some relevance information, implied (in the case of blind feedback) or asserted via a relevance judgement. The next set of experiments, described in the following chapter, was designed to explore this research direction.

7. Local collocation analysis experiments

7.1 Introduction

The aim of this experiment set was to examine the hypothesis that local collocation analysis – query expansion with local collocates of query terms following relevance feedback – can result in performance improvements over Okapi query expansion from relevance feedback.

The hypothesis was explored through a systematic retrieval evaluation of the local collocation analysis technique with different values for the following key variables:

- Window size;
- Measure of collocation significance for ranking collocates;
- Number of top ranked collocates in the expanded query;
- Number of Okapi relevance feedback terms in the expanded query.

A large number of combinations of different values for the above variables is possible. We did not replicate runs for every possible combination of variable values. Instead a more selective approach has been adopted: those runs which showed best results with a certain value for one variable, were replicated with a range of values for other variables.

The objective of this set of experiments was to evaluate the performance of the local collocation analysis technique with a range of variable values, and to understand how the performance is affected by the changes in each of these variables. The methodologies and results of retrieval experiments undertaken are presented in sections *7.2* and *7.3*.

Following the retrieval evaluation experiments, a complementary study was conducted, which was targeted at evaluating the influence of each category of terms found in the expanded queries on performance. Its objective was to get a better understanding of each category's contribution to performance, what was not evident from the main retrieval experiments. The study of query terms by category is detailed in section *7.4*.

7.2 Query expansion with local collocates ranked by Z and MI scores

Before starting full-scale retrieval experiments on all 50 topics, we decided first to explore whether local collocation analysis holds any potential for performance gains over global collocation analysis technique and standard Okapi query expansion, and hence, whether it is a feasible direction to pursuit. We did this by trying out the technique on a single query. The results appeared to be somewhat promising and suggested full-scale systematic testing under a range of system parameters.

First, we conducted a series of evaluations on all topics using retrospective searching technique, whereby expanded queries are applied to the same set of documents (the

whole collection), from which the documents used for query expansion were derived. Retrospective runs were later replicated in a less idealistic setting of predictive searching based on half collections. The advantage of using predictive searching is that it allows us to see whether expansion terms have any predictive value. In retrospective searching very rare collocates, occurring, say, only in the known relevant documents, could perform well simply because they retrieve the same relevant documents form which they are derived. In predictive searching it would be evident that such terms have no or little predictive value to retrieve new relevant documents.

7.2.1 Experimental design

Retrospective relevance feedback searches were run on the whole FT 96 collection. Relevance information for relevance feedback was obtained using the following method: FT 96 database was searched with the initial queries using Okapi weighted search – bm2500 without relevance information. First 5 relevant documents from each retrieved set of 1000 documents were extracted and recorded as relevant documents to be used in relevance feedback runs. Information about the relevance of the documents in the Okapi sets was taken from TREC relevance judgements file.

To run predictive searches based on half collections, FT 96 database was divided into two equal halves: the odd half – containing all records with odd internal record numbers (IRN), and the even half – containing all records with even internal record numbers. IRN is an integer number assigned to each record in the Okapi database by Okapi BSS. The even half of the collection was used for deriving relevant documents. The expanded queries were searched against the odd half.

To obtain relevant documents for relevance feedback in predictive evaluation, the same technique as for retrospective experiments was used, with the relevant documents taken from Okapi sets retrieved from the even half of the collection instead of the full database.

A range of retrospective and predictive query expansion runs using local collocates was conducted with different values of variables, listed in section *7.1*. The specific runs conducted will be presented in the next section *7.2.2 Analysis of results* together with their performance results. In this section the methodology for these runs will be presented.

Local collocates for all runs were extracted from the relevant documents using the same technique, described below.

For every topic all occurrences of each query term are located in the relevant documents used for relevance feedback. Collocates are extracted from the windows around every occurrence of query terms in these documents using the windowing technique described in *5.3*. Window size is one of the variables, whose effect on performance was tested in the experiments. All runs were performed with the window size 200. The best performing runs were replicated with other window sizes: 100, 50, 30 and 20.

In more detail, the algorithm for extracting local collocates of query terms (Perl scripts listed in Appendix D.2) consists of the following stages:

1. 5 relevant records per topic obtained for retrospective/predictive searching (see earlier in this section) are parsed using Okapi parsing algorithm: GSL look-up and stemming, the same as applied to the collection during indexing (see section *5.5.2*). The result is a sub-corpus, consisting of concatenated records with marked record boundaries. Each record is represented as a sequence of stems and GSL codes in the original order of their corresponding wordforms.
2. Each query term is looked up in the sub-corpus, and for each of its occurrence collocates are extracted using the windowing technique (section *5.3*).
3. Extracted collocates are recorded in a separate file for each query term.

After all local collocates of query terms are extracted from the relevant documents, they are ranked by the significance of their association with the node. The measures used for collocate ranking were Z and MI scores, specifically their global and local variations (described in sections *5.4.1* and *5.4.2*).

The next step is the construction of the expanded query. The main questions at this stage are: should the original query terms be retained in the query, and how many top ranked collocates should be selected into the expanded query.

Original query terms were kept in the queries expanded with their local collocates. To ensure consistent comparison of results it was decided to keep original query terms in the expanded queries of Okapi relevance feedback (RF) runs as well.

It was decided to use a fixed number of top-ranked collocates for query expansion, instead of setting a threshold on the ranking scores. The rationale for this was the same as in the global collocation analysis technique (section *6.2*), i.e. high variability of scores for collocates of different query terms.

The initial experiment on a single topic suggested that merging local collocates with Okapi relevance feedback (RF) terms in the expanded query can possibly lead to performance gains over Okapi query expansion from relevance feedback. It was decided to evaluate this technique on all topics with different variable settings.

The algorithm for building expanded queries from both local collocates and Okapi RF terms consists of the following stages:

1. Collocates are extracted from the relevant documents (as in the previous method of query expansion with local collocates only) and ranked by either global, or local Z or MI.
2. Okapi RF terms are extracted from the relevant documents and ranked by OW as in the standard Okapi relevance feedback.
3. Top N collocates per query term are added to the expanded query.
4. Top I Okapi RF terms are added to the expanded query.
5. Duplicates are eliminated from the expanded query.

OW was calculated for Okapi RF terms using formula 4.10 (p. 59) with r_i = the number of documents containing term i in the set of R ($R \leq 5$) – relevant documents used for query expansion for the topic in question.

The numbers of Okapi RF terms added to the query were 10 and 20. We evaluated the combinations of different numbers of local collocates per query and 10 or 20 Okapi RF terms.

Expanded queries in retrospective runs were searched against FT_96 collection, and in predictive runs – against the odd half of the collection. Searching was done using Okapi weighted function bm2500 with relevance information (the relevant documents used for query expansion). bm2500 with relevance information instantiates Robertson-Sparck Jones' Combined Iterative Weight (CIW) formula (see section *4.1.1*). R and r values used in term weighting were the same as in the OW calculation above. Tuning constant k_1 (controlling the effect of within-document frequency) was set to 2 and b (controlling document length normalisation) was set to 0.75.

In all runs top 1000 documents of each ranked document set were evaluated using trec_eval program and a file with TREC relevance judgements.

7.2.2 Analysis of results

Retrospective evaluation

Each run will be referred to by an abbreviated name which reflects the query expansion method. Names for retrospective runs are preceded with the abbreviation 'RETRO'.

Performance of local collocation analysis runs was evaluated against Okapi relevance feedback (RF) runs with the same sets of relevant documents and using comparable numbers of Okapi RF terms for query expansion. Retrospective performance results of Okapi runs are presented in table 7.1. Trec_eval summaries for all retrospective runs are given in Appendix B.1.

Run name	Query description	Average precision
RETRO UNEXPANDED	Original query terms	**0.1310**
RETRO OK 20	20 Okapi RF terms	**0.4945**
RETRO OK 25	25 Okapi RF terms	**0.5096**
RETRO OK 30	30 Okapi RF terms	**0.5184**
RETRO OK 35	35 Okapi RF terms	**0.5259**

Table 7.1. Retrospective performance results of Okapi runs

Expansion with Z-ranked collocates

Initially we evaluated query expansion with local collocates ranked by global Z. Expansion with 8 local collocates per query term, extracted from windows of size 200 and ranked by global Z (run 'RETRO 8 GLOBAL Z COL' in table 7.2) turned out to be worse than all retrospective Okapi relevance feedback runs. Evaluation on a single topic suggested that merging in the expanded query 8 local collocates per query term ranked by global Z with 20 Okapi RF terms could lead to performance gains over Okapi relevance feedback. We evaluated this method on all topics, conducting a retrospective run 'RETRO 8 GLOBAL Z COL + 20 OK' The result of this run was similarly worse than all RETRO Okapi RF runs.

Run name	Query description	Average precision
RETRO 8 GLOBAL Z COL	8 collocates/query term ranked by global Z (only terms with term frequency > 30)	**0.3220**
RETRO 8 GLOBAL Z COL + 20 OK	8 collocates/query term ranked by global Z (only terms with term frequency > 30) + 20 Okapi RF terms	**0.3148**

Table 7.2. Retrospective performance results of query expansion runs with local collocates ranked by global Z

We then conducted a similar run 'RETRO 8 Z COL + 20 OK' (top 8 collocates/query term from 200 window size + 20 Okapi RF terms) but this time ranking collocates by local Z. The performance of this run (RETRO 8 Z COL + 20 OK in table 7.3) was better than 'RETRO 8 GLOBAL Z COL + 20 OK' and some of the Okapi RF runs. The gain over 'RETRO 8 GLOBAL Z COL + 20 OK' was significant – 39.4%.

As this performance result was more promising than those of collocates ranked by global Z, we decided to explore this technique further with different parameter values. Table 7.3 lists all conducted retrospective runs with collocates ranked by local Z. First we tried different numbers of collocates per query term without Okapi RF terms (8, 12, 16, 21) extracted from window size 200. The table shows that average precision grows slowly with the increase in the number of collocates used: from 0.4758 in 'RETRO 8 Z COL' to 0.5029 in 'RETRO 21 Z COL'. There can also be observed a very slow growth in average precision with the decrease in the window size: from 0.4758 – in 'RETRO 8 Z COL (window size 200)' to 0.4810 – in 'RETRO 8 Z COL (window size 20)'.

Run name	Window size / Query description	200	100	50	30	20
RETRO 8 Z COL	8 Z collocates/query term	**0.4758**	**0.4720**	**0.4733**	**0.4857**	**0.4810**
RETRO 12 Z COL	12 Z collocates/query term	**0.4896**				
RETRO 16 Z COL	16 Z collocates/query term	**0.5034**				
RETRO 21 Z COL	21 Z collocates/query term	**0.5029**				
RETRO 8 Z COL + 20 OK	8 Z collocates/query term + 20 Okapi RF terms	**0.5194**	**0.5230**	**0.5245**	**0.5258**	**0.5263**
RETRO 16 Z COL + 10 OK	16 Z collocates/query term + 10 Okapi RF terms	**0.5257**				
RETRO 16 Z COL + 20 OK	16 Z collocates/query term + 20 Okapi RF terms	**0.5171**	**0.5264**	**0.5271**	**0.5313**	**0.5316**
RETRO 21 Z COL + 10 OK	21 Z collocates/query term + 10 Okapi RF terms	**0.5219**				

Table 7.3. Retrospective performance results (in average precision) of query expansion runs with local collocates ranked by local Z

We then tried several combined runs – combinations of collocates and Okapi RF terms in the expanded queries (16 collocates + 10 Okapi terms, 16 collocates + 20 Okapi terms, 21 collocates + 10 Okapi RF terms). Addition of Okapi terms to collocates improves performance to a varied extent. For example average precision of 'RETRO 8 Z COL + 20 OK (200 window size)' is 8.4% better than 'RETRO 8 Z COL (200 window size)', while average precision of 'RETRO 16 Z COL + 20 OK (200 window size)' is only 2.6% better than 'RETRO 16 Z COL (200 window size)'.

In the combined runs there cannot be seen a pattern of precision growth with the increase in the number of collocates, which was observed in the expansion runs with collocates only. The differences between combined runs are marginal, therefore they can be considered to be relatively unaffected by either the number of collocates, or the number of Okapi RF terms within these ranges.

Next, we tested the effect of different window sizes on performance in combined runs. Two runs: 'RETRO 8 Z COL + 20 OK' and 'RETRO 16 Z COL + 20 OK'were evaluated with window sizes: 100, 50, 30 and 20. Here also, like in the runs 'RETRO 8 Z COL (window sizes 200-20)' we can observe a tendency towards increase in average precision with the decrease in the window size. However, the difference in precision between the runs with different window sizes is negligible, for example run 'RETRO 16 Z COL + 20 OK (20 window size)' is only 2% better than 'RETRO 16 Z COL + 20 OK (200 window size)'. This suggests that window size does not have significant influence on performance of combined runs.

Overall combined runs yielded similar performance results to Okapi RF runs. Performance gains of the best combined runs 'RETRO 16 Z COL + 20 OK (30 window size)' and 'RETRO 16 Z COL + 20 OK (20 window size)' over the best Okapi RF run 'RETRO OK 35' are not statistically significant.

Expansion with MI-ranked collocates

Results of query expansion with local collocates ranked by global MI (table 7.4) are similar to the results of query expansion with local collocates ranked by global Z (table 7.2).

Run name	Query description	Average precision
RETRO 8 GLOBAL MI COL	8 collocates/query term ranked by global MI (only terms with term frequency > 30)	**0.3177**
RETRO 8 GLOBAL MI COL + 20 OK	8 collocates/query term ranked by global MI (only terms with term frequency > 30) + 20 Okapi RF terms	**0.3104**

Table 7.4. Retrospective performance results of query expansion runs with local collocates ranked by global MI

Next, we tested local MI for collocates ranking (table 7.5), which resulted in noticeably better performance than global MI.

Run name	Window size / Query description	200	100	50	30	20
RETRO 8 MI COL	8 MI collocates/query term	**0.4458**	**0.4610**	**0.4551**	**0.4690**	**0.4733**
RETRO 12 MI COL	12 MI collocates/query term	**0.4688**				
RETRO 16 MI COL	16 MI collocates/query term	**0.4877**				
RETRO 21 MI COL	21 MI collocates/query term	**0.4991**				
RETRO 8 MI COL + 20 OK	8 MI collocates/query term + 20 Okapi RF terms	**0.5227**	**0.5251**	**0.5240**	**0.5249**	**0.5274**
RETRO 16 MI COL + 10 OK	16 MI collocates/query term + 10 Okapi RF terms	**0.5274**	**0.5260**	**0.5292**	**0.5291**	**0.5270**
RETRO 16 MI COL + 20 OK	16 MI collocates/query term + 20 Okapi RF terms	**0.5264**	**0.5267**	**0.5282**	**0.5301**	**0.5290**
RETRO 21 MI COL + 10 OK	21 MI collocates/query term + 10 Okapi RF terms	**0.5272**				

Table 7.5. Retrospective performance results (in average precision) of query expansion runs with local collocates ranked by local MI

Runs with different numbers of local MI collocates alone (in table 7.5) were slightly worse than corresponding runs with local Z collocates (in table 7.3), for example 'RETRO 8 MI COL' is 6% worse than 'RETRO 8 Z COL'. The pattern of results is similar to Z runs: precision increases with the increase in the number of collocates – run 'RETRO 21 MI COL (200 window size)' is 10.6% better than 'RETRO 8 MI COL (200 window size)'. Also, similarly to Z runs, precision grows slowly with the decrease in the window size: 'RETRO 8 MI COL (20 window size)' is 5% better than 'RETRO 8 MI COL (200 window size)'.

Combined runs of local MI collocates with Okapi RF terms (in table 7.5) yielded similar performance results to combined runs with local Z collocates and Okapi RF terms (in table 7.3). Similarly to Z combined runs, performance was only marginally

affected by the numbers of added Okapi terms (10 or 20) or collocates used (8, 16 or 21). Window sizes similarly do not have noticeable effect on performance.

Predictive evaluation

Following the retrospective evaluation, the same runs were replicated using predictive searching based on half collections. Predictive runs presented below are marked as 'PRED'. Trec_eval summaries for all predictive runs are given in Appendix B.2.

Performance results of Predictive Okapi runs are presented in table 7.6.

Run name	Query description	Average precision
PRED UNEXPANDED	Original query terms	**0.0799**
PRED OK 10	10 Okapi RF terms	**0.1343**
PRED OK 20	20 Okapi RF terms	**0.1400**
PRED OK 25	25 Okapi RF terms	**0.1520**
PRED OK 30	30 Okapi RF terms	**0.1483**
PRED OK 35	35 Okapi RF terms	**0.1533**

Table 7.6. Predictive performance results of Okapi runs

Expansion with Z-ranked collocates

Query expansion with 8 collocates per query term ranked by global Z – 'PRED 8 GLOBAL Z COL' (in table 7.7) is worse than all Okapi RF runs. Addition of 20 Okapi RF terms (run 'PRED 8 GLOBAL Z COL + 20 OK' in table 7.7) improves performance drammatically – by 28%.

Run name	Query description	Average precision
PRED 8 GLOBAL Z COL	8 collocates/query term ranked by global Z (only terms with term frequency > 30)	**0.0974**
PRED 8 GLOBAL Z COL + 20 OK	8 collocates/query term ranked by global Z (only terms with term frequency > 30) + 20 Okapi RF terms	**0.1360**

Table 7.7. Predictive performance results of query expansion runs with local collocates ranked by global Z

PRED expansion runs with local collocates ranked by local Z (runs 'PRED 8 Z COL', 'PRED 12 Z COL', 'PRED 16 Z COL' and 'PRED 21 Z COL' in table 7.8) are not better than the best PRED Okapi RF run 'PRED OK 35'(in table 7.6). Some of them show similar performance to Okapi runs with comparable number of terms, for example, run 'PRED 8 Z COL (20 window size)', in which the average length of queries is 15.5 terms, has similar average precision to 'PRED OK 20'.

The tendency of precision growth with the increase in the number of collocates in PRED runs expanded with collocates only (200 and 100 window sizes) is similar to the corresponding previous retrospective runs. For example, 'PRED 21 Z COL (200

window size)' is 8.8% better than 'PRED 8 Z COL (200 window size)'(in table 7.8). However this tendency was not observed in the runs with window sizes 50, 30 and 20: 'PRED 21 Z COL (20 window size)' is even slightly worse than 'PRED 8 Z COL (20 window size)'. This suggests that terms from smaller window sizes, situated lower in the ranked list, have weaker relevance-predicting power than terms from larger window sizes, in the same ranking positions.

Tendency of precision increase with the decrease in the window size is not very consistent: while in 'PRED 8 Z COL' and 'PRED 12 Z COL' runs the difference in precision between 200 and 20 window sizes is 11%, in 'PRED 16 Z COL' and 'PRED 12 Z COL' the difference is very small.

The results suggest that either increasing the number of collocates, or shortening the window size tends to improve precision to approximately the same extent.

Expansion with 12 collocates taken from windows of size 20 – run 'PRED 12 Z COL (20 window size)' – performs best among runs with collocates only.

The gain from adding 20 Okapi RF terms to collocates in the combined runs was, similarly to combined RETRO runs, not consistent: while 'PRED 8 Z COL + 20 OK (100 window size)' is 19% better than 'PRED 8 Z COL (100 window size)', 'PRED 16 Z COL + 20 OK (20 window size)' is only 3% better than 'PRED 16 Z COL (20 window size)' (in table 7.8). Addition of 10 Okapi RF terms yields similar results to the corresponding runs with collocates only (compare 'PRED 16 Z COL + 10 OK' with 'PRED 16 Z COL' or 'PRED 21 Z COL + 10 OK' with 'PRED 21 Z COL' in table 7.8)

Run name	Window size / Query description	200	100	50	30	20
PRED 8 Z COL	8 Z collocates/query term	**0.1268**	**0.1294**	**0.1302**	**0.1376**	**0.1433**
PRED 12 Z COL	12 Z collocates/query term	**0.1346**	**0.1383**	**0.1459**	**0.1401**	**0.1518**
PRED 16 Z COL	16 Z collocates/query term	**0.1386**	**0.1456**	**0.1362**	**0.1367**	**0.1432**
PRED 21 Z COL	21 Z collocates/query term	**0.1391**	**0.1480**	**0.1325**	**0.1396**	**0.1384**
PRED 8 Z COL + 20 OK	8 Z collocates/query term + 20 Okapi RF terms	**0.1536**	**0.1602**	**0.1549**	**0.1568**	**0.1553**
PRED 16 Z COL + 10 OK	16 Z collocates/query term + 10 Okapi RF terms	**0.1388**				
PRED 16 Z COL + 20 OK	16 Z collocates/query term + 20 Okapi RF terms	**0.1527**	**0.1561**	**0.1495**	**0.1485**	**0.1477**
PRED 21 Z COL + 10 OK	21 Z collocates/query term + 10 Okapi RF terms	**0.1413**				

Table 7.8. Predictive performance results (in average precision) of query expansion runs with local collocates ranked by local Z

The window sizes in the combined PRED runs (similarly to RETRO) do not have a noticeable effect on performance.

The best PRED run in this group is 'PRED 8 Z COL + 20 OK (100 window size)' with the average precision of 0.1602, which is slightly better than the best Okapi RF run 'PRED OK 35' with the average precision of 0.1533.

Expansion with MI-ranked collocates

Predictive expansion runs with local collocates ranked by global MI (table 7.9) are worse than all PRED Okapi RF runs.

Run name	Query description	Average precision
PRED 8 GLOBAL MI COL	8 collocates/query term ranked by global MI (only terms with term frequency > 30)	**0.1155**
PRED 8 GLOBAL MI COL + 20 OK	8 collocates/query term ranked by global MI (only terms with term frequency > 30) + 20 Okapi RF terms	**0.1099**

Table 7.9. Predictive performance results of query expansion runs with local collocates ranked by global MI

Predictive runs with different numbers of collocates ranked by local MI alone (in table 7.10) are either worse or similar to the corresponding PRED runs with local Z collocates (in table 7.8). For example, run 'PRED 8 MI COL (200 window size)' is 20% worse than 'PRED 8 Z COL (200 window size)', while run 'PRED 21 MI COL (200 window size)' yields the same results as 'PRED 21 Z COL (200 window size)'.

Run name	Window size / Query description	200	100	50	30	20
PRED 8 MI COL	8 MI collocates/query term	**0.1006**	**0.0971**	**0.1158**	**0.1399**	**0.1371**
PRED 12 MI COL	12 MI collocates/query term	**0.1222**	**0.1202**	**0.1365**	**0.1435**	**0.1524**
PRED 16 MI COL	16 MI collocates/query term	**0.1302**	**0.1437**	**0.1341**	**0.1372**	**0.1380**
PRED 21 MI COL	21 MI collocates/query term	**0.1390**	**0.1440**	**0.1295**	**0.1355**	**0.1359**
PRED 8 MI COL + 20 OK	8 MI collocates/query term + 20 Okapi RF terms	**0.1455**	**0.1426**	**0.1501**	**0.1565**	**0.1528**
PRED 16 MI COL + 10 OK	16 MI collocates/query term + 10 Okapi RF terms	**0.1398**				
PRED 16 MI COL + 20 OK	16 MI collocates/query term + 20 Okapi RF terms	**0.1515**	**0.1626**	**0.1522**	**0.1486**	**0.1487**
PRED 21 MI COL + 10 OK	21 MI collocates/query term + 10 Okapi RF terms	**0.1477**				

Table 7.10. Predictive performance results (in average precision) of query expansion runs with local collocates ranked by local MI

Precision of PRED expansion runs with MI collocates only, taken from window sizes 200 and 100, similarly to PRED Z runs (in table 7.8) demonstrates the tendency of growth with the increase in the number of collocates: 27% difference between 'PRED 8 MI COL (200 window size)' and 'PRED 21 MI COL (200 window size)'. However, again like in PRED Z runs, this tendency is less evident in the runs with window sizes 30 and 20.

The best run using collocates only is (like in local Z runs) expansion with 12 collocates taken from windows of size 20 'PRED 12 MI COL (20 window size)'.

Combined runs with local MI collocates and Okapi RF terms showed similar results to the corresponding combined Z runs. The best run in this group is 'PRED 16 MI COL + 20 OK (100 window size)' with the average precision of 0.1626, which is 5.7% better than the average precision 0.1533 of the best PRED Okapi RF run 'PRED OK 35'.

7.3 Collocation relevance weighting

7.3.1 Experimental design

In this experiment we derived and evaluated a new weighting function for the selection of collocates from relevance feedback – Collocation Relevance Weighting (CRW) function. The CRW function is the adaptation of Robertson-Sparck Jones' RW function (see section *4.1.1*). In probabilistic model each term is weighted by RW, based on the relevance data about this term, which is usually represented as a contingency table (p. 56). The unit of data in the contingency table is a document. We decided that instead of counting the number of documents the term is present in, we can count the number of term-slots in the documents containing or not containing the term in question. Each term-slot is classified as belonging to either relevant, or non-relevant windows. By *relevant windows* we mean fixed-size windows around all occurrences of any query term in the relevant document(s). A *non-relevant window* covers all remaining term-slots in the collection. Each term-slot in either relevant, or non-relevant windows is classified as either containing, or not containing the term in question. The resulting contingency table is as following:

	Positions in relevant windows	Positions in non-relevant windows	
Containing the collocate	c	$n-c$	n
Not containing the collocate	$C-c$	$N-C-n+c$	$N-n$
	C	$N-C$	N

where:

c - number of occurrences of term i in the relevant windows;
n - total number of occurrences of term i in the collection;
C - total number of term-slots in the relevant windows;
N - total length of the collection (in term-slots).

If a relevant window was set to the entire document, then we would end up with a method similar to the existing weighting method in the probabilistic model, but of course, with different numbers in the contingency table (term occurrences, instead of numbers of documents). What we get from our method, is the contextual information about term occurrence. The method rewards those terms, which occur not in any random part of the relevant documents, but in the windows around query terms in these documents.

The CRW function (7.1) derived from our contingency table is analogous to RW function derived from the contingency table in the probabilistic model.

$$CRW = \log \frac{(c+0.5)(N-n-C+c+0.5)}{(C-c+0.5)(n-c+0.5)} \qquad (7.1)$$

We evaluated this technique in both retrospective and predictive searching scenarios on the same relevance information as in the previous experiments with local Z-ranked collocates.

The collocate extraction method is slightly different in this technique, than in the previous ones. In this method we define one set of windows for all query terms. If we were extracting collocates with our usual windowing technique (section *5.3*), then we would come across the following problem:

Two different query terms can occur close to each other, i.e. their windows overlap, as shown in figure 7.1.

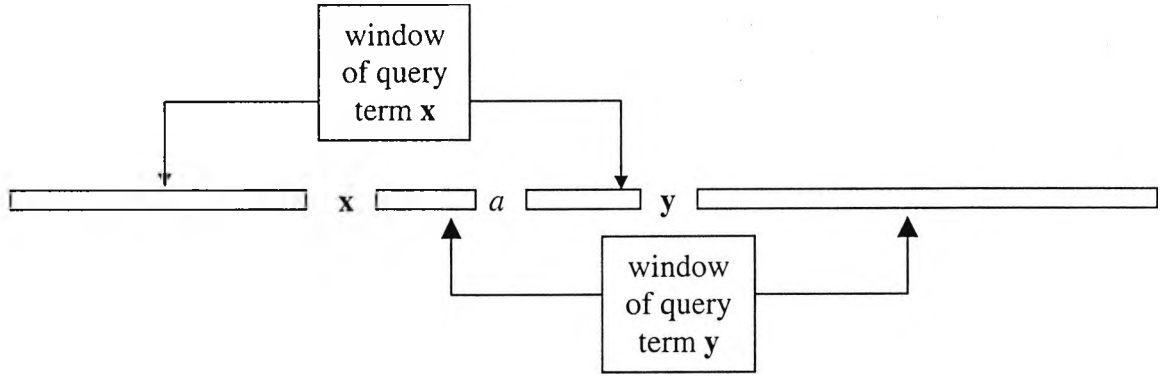


Figure 7.1. Overlapping windows of two different query terms *x* and *y*

In our usual windowing technique collocates of each query term are extracted independently of other query terms. Therefore, in a case like this we would record two collocation pairs:

x - *a*
y - *a*

In CRW method, we identify only one set of relevant windows – for all query terms, therefore, if we were using the usual windowing technique, the term-slot containing collocate *a* would be counted twice in *c* (number of occurrences of a term in the relevant windows). For this reason, we record collocates in overlapping windows of different query terms only once.

One possible drawback of this method is that we cannot account for closeness of a collocate to more than one query term.

After all collocates are extracted, they are ranked for query expansion. On the analogy with the use of Offer Weight (*RW* * *r*) for the selection of terms from relevance feedback in the probabilistic model, we used *CRW* * *c* for collocation ranking.

To ensure consistent comparison with the previous expansion runs with MI and Z collocates, here the original query terms were similarly kept in the expanded queries

7.3.2 Analysis of results

CRW query expansion was evaluated using both retrospective and predictive searching. We tested query expansion with top 20 and 35 ranked collocates. Results of retrospective runs are given in table 7.11, results of predictive runs – in table 7.12. More detailed trec_eval summaries for CRW runs are given in Appendix B.3.

Run name	Window size / Query description	200	100	50	30
RETRO 20 CRW	20 CRW collocates per query (retrospective)	**0.3917**		**0.3845**	
RETRO 35 CRW	35 CRW collocates per query (retrospective)	**0.4000**		**0.4048**	

Table 7.11. Retrospective performance results (in average precision) of query expansion runs with local collocates ranked by *CRW* * *c*

Run name	Window size / Query description	200	100	50	30
PRED 20 CRW	20 CRW collocates per query (predictive)	**0.1082**		**0.1228**	
PRED 35 CRW	35 CRW collocates per query (predictive)	**0.1085**		**0.1221**	

Table 7.12. Predictive performance results (in average precision) of query expansion runs with local collocates ranked by *CRW* * *c*

Both RETRO and PRED CRW sets of runs were worse than corresponding Okapi runs. They did not suggest either that CRW is superior to using local Z or MI for the ranking of local collocates.

7.4 Evaluation of performance by categories of terms in the expanded query

Retrieval experiments with local collocates ranked by local measures of MI and Z (section *7.2* above) showed that in both retrospective and predictive evaluations combined queries with local Z-ranked collocates and Okapi terms, though did not demonstrate significant performance gains, suggested some possible tendency towards improving performance over Okapi RF. Because combined queries consisted of several categories of terms, it was interesting to see what impact on performance

each category of terms has. The retrieval experiments did not give us such information. Knowing which category/ies contributes most to performance gain or loss, would let us adopt a more focused approach, and develop a more selective technique of query expansion. We could emphasize the terms of the category, which has evidence to contribute most to performance improvement, and remove/ downweight terms of the category deteriorating performance.

For analysis we chose one of the best runs in the predictive experiments – 'PRED 8 Z COL + 20 OK (100 window size)'. Expanded queries in such combined runs can include the following categories of terms:

1. Collocate;
2. Collocate of 2 or more query terms;
3. Collocate of 1 query term;
4. Okapi relevance feedback term;
5. Original query term;
6. Collocate of 2 or more query terms and an Okapi relevance feedback (RF) term;
7. Collocate of 1 query term and an Okapi relevance feedback term;
8. Collocate and an Okapi relevance feedback term;
9. Collocate and an original query term;
10. Okapi relevance feedback term and an original query term.

Since we did not want original query terms to interfere with the evaluation of performance of categories 1, 2, 3, 4, 6, 7 and 8, we did not include them into these categories.
The experiment methodology consisted of the following stages.

First, each term in all expanded queries for 50 topics was tagged in the following format:

term <*r*> <*a*> <*b*> <*c*>

where:

r – number of relevant documents the term occurs in;
a – number of original query terms the term co-occurs with [0-n];
b – binary variable indicating if a term is an Okapi RF term or not [0, 1];
c – binary variable indicating if a term is an original query term or not [0, 1].

The influence of each term on performance was identified by taking out the term from the query, running this query and recording its performance in average precision. Average precision of the query without the term in question was compared to average precision of the complete query. If, for example, average precision of the former is greater, then the term is considered to degrade the performance.

The influence of each term on performance is classified as:

- Improving;
- Indifferent;
- Degrading.

After performance data for each query term is obtained, we total the number of query terms in each term category for each performance group.

The results are summarised in tables 7.13 and 7.14 and presented as a chart in figure 7.2.

Category	Improve	Indifferent	Degrade	Total
1. Collocate	83	181	151	415
2. Collocate of 2 or more query terms	21	62	39	122
3. Collocate of 1 query term	62	119	112	293
4. Okapi RF term	187	170	302	659
5. Original query term	84	51	25	160
6. Collocate of 2 or more query terms and an Okapi RF term	11	23	15	49
7. Collocate of 1 query term and an Okapi RF term	22	30	40	92
8. Collocate and an Okapi RF term	33	53	55	141
9. Collocate and an original term	11	1	2	14
10. Okapi RF term and an original query term	43	2	5	50

Table 7.13. Influence of categories of terms in the expanded queries 'PRED 8 Z COL + 20 OK (100 window size)' on average precision

Term category	Improve	Indifferent	Degrade
1. Collocate	20%	44%	36%
2. Collocate of 2 or more query terms	17%	51%	32%
3. Collocate of 1 query term	21%	41%	38%
4. Okapi RF term	28%	26%	46%
5. Original query term	52%	32%	16%
6. Collocate of 2 or more query terms and an Okapi RF term	22%	47%	31%
7. Collocate of 1 query term and an Okapi RF term	24%	33%	43%
8. Collocate and an Okapi RF term	23%	37%	40%
9. Collocate and an original term	79%	7%	14%
10. Okapi RF term and an original query term	86%	4%	10%

Table 7.14. Influence of categories of terms in the expanded queries 'PRED 16 Z COL + 20 OK (200 window size)' on average precision (in percentage)

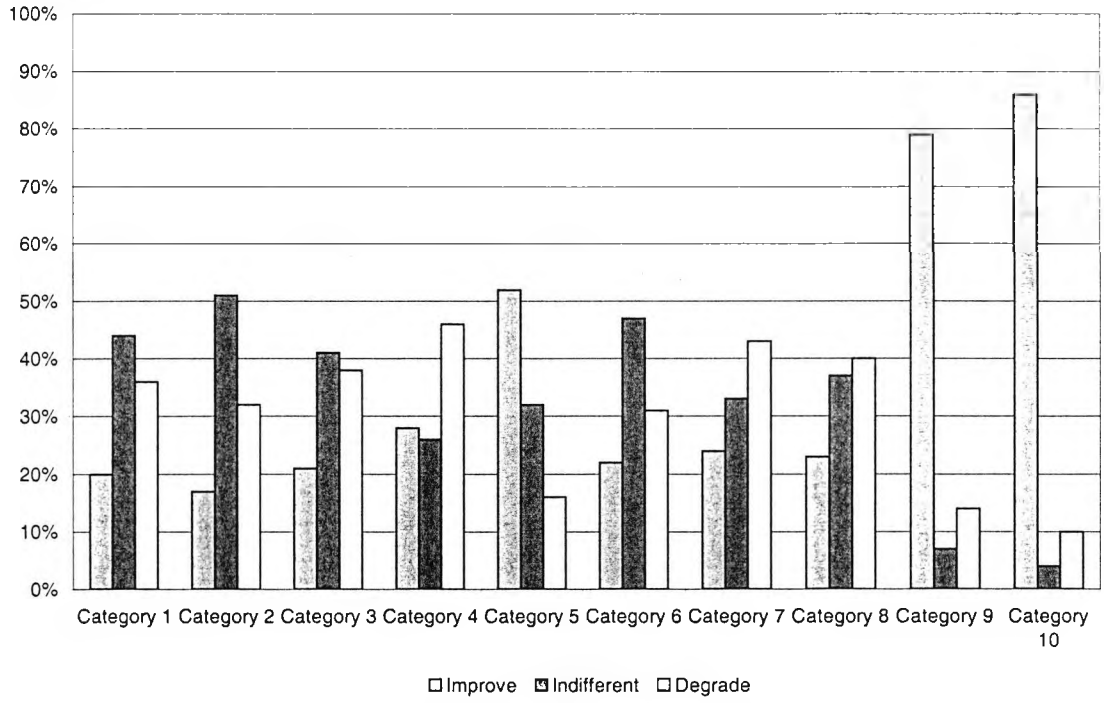


Figure 7.2. Influence of categories of terms in the expanded queries 'PRED 16 Z COL + 20 OK (200 window size)' on average precision

The results, rather surprisingly, showed that either collocates, or Okapi RF terms hurt precision in a larger number of cases, than improve it. More Okapi RF terms (category 4) than collocates (category 1) improve precision, however even more of them hurt precision.

It was expected that collocates of more than 1 query term would be better relevance discriminators than collocates of 1 query term only. The results showed that fewer collocates of 2 or more query terms (category 2) improve precision, than any collocates (category 1) or collocates of 1 query term (category 3). On the other hand, fewer collocates of 2 or more query terms hurt precision in comparison with categories 1 or 3.

Terms which are both collocates and okapi terms (categories 6, 7 and 8) improve precision in a larger number of cases than terms of categories 1, 2 and 3. Although, except category 6 (collocates of 2/more query terms), they hurt precision in a larger number of cases either.

A category that suggested greater improvement than degradation of performance is category 5: *Original query terms*. The term's status as an original query term plus either a collocate (category 9), or an Okapi RF term (category 10), indicates a much higher relevance discriminating ability. There is, however, a rather low number of terms in these categories: 14 in category 9 and 50 in category 10.

It is not clear, however, from this data to what degree terms in these categories improve or hurt precision. For example, a small number of terms could strongly improve precision, while a larger number of terms could hurt precision insignificantly. To test whether this could be the case, we created histograms on the data of differences between the precision of a complete query and the precision of a query without a term of the category in question (figures 7.3 – 7.9).

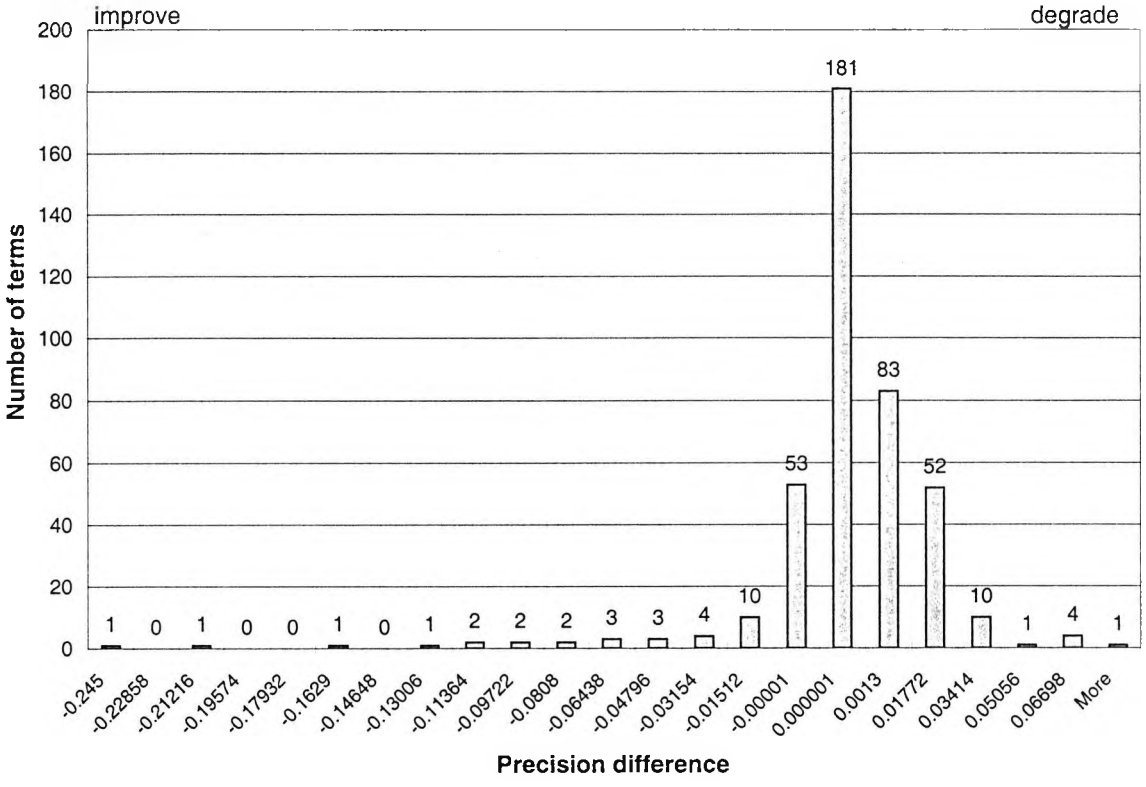


Figure 7.3. Distribution of precision differences for category 1. *Collocate*

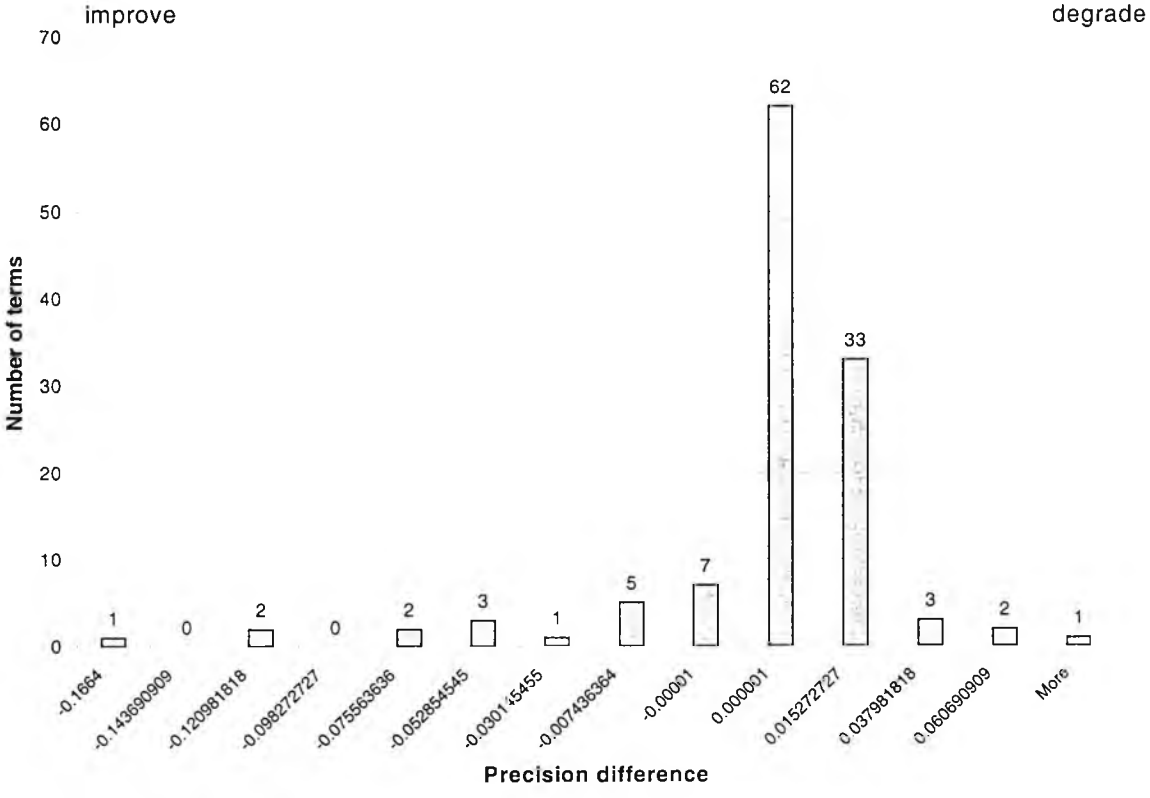


Figure 7.4. Distribution of precision differences for Category 2:
Collocate of 2 or more query terms

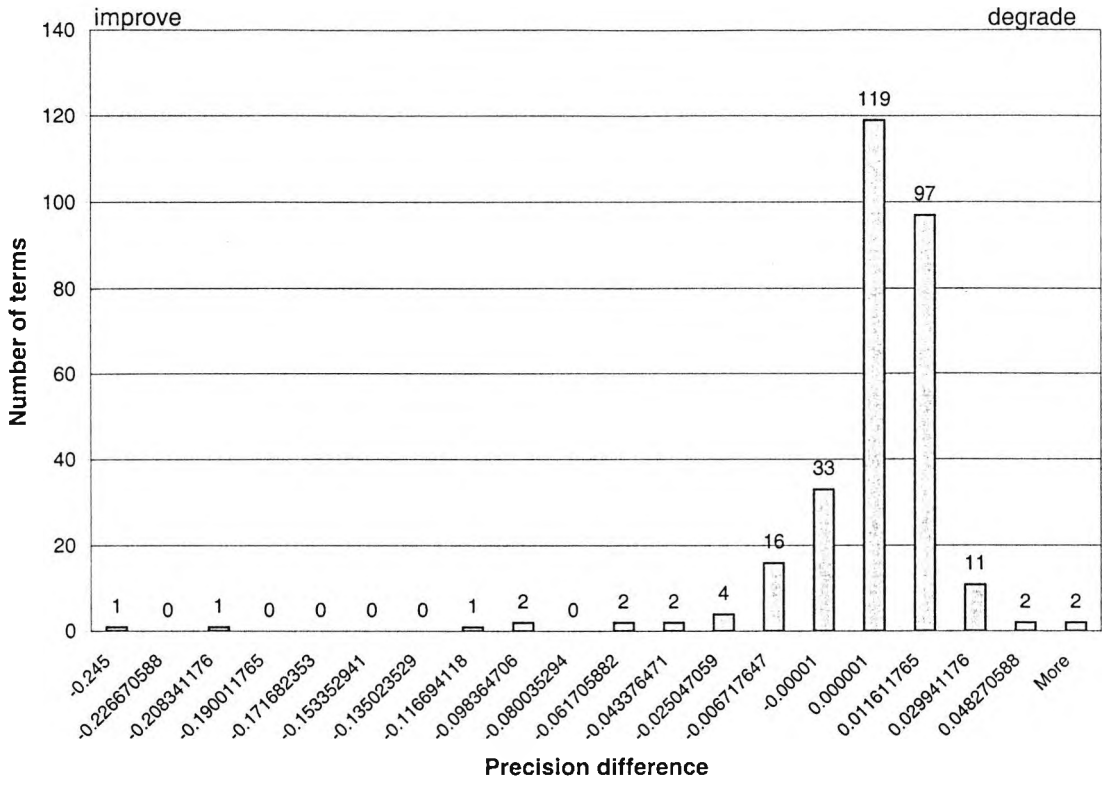


Figure 7.5. Distribution of precision differences for Category 3:
Collocate of 1 query term

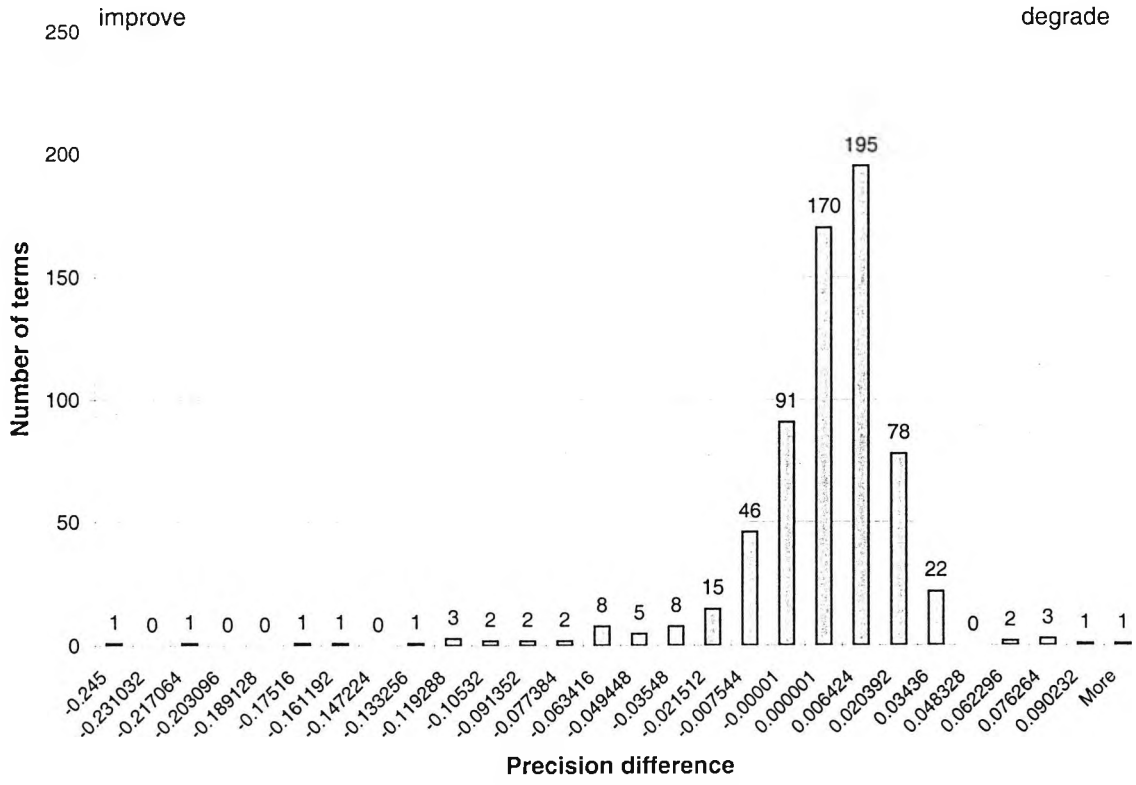


Figure 7.6. Distribution of precision differences for Category 4:
Okapi RF term

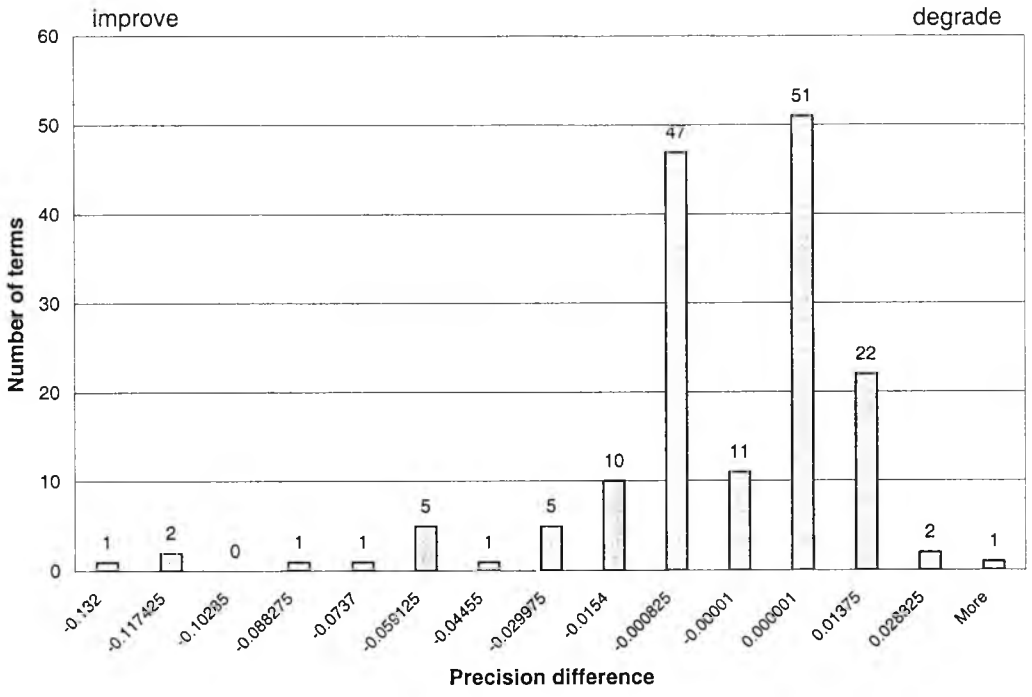


Figure 7.7. Distribution of precision differences for Category 5: *Original query term*

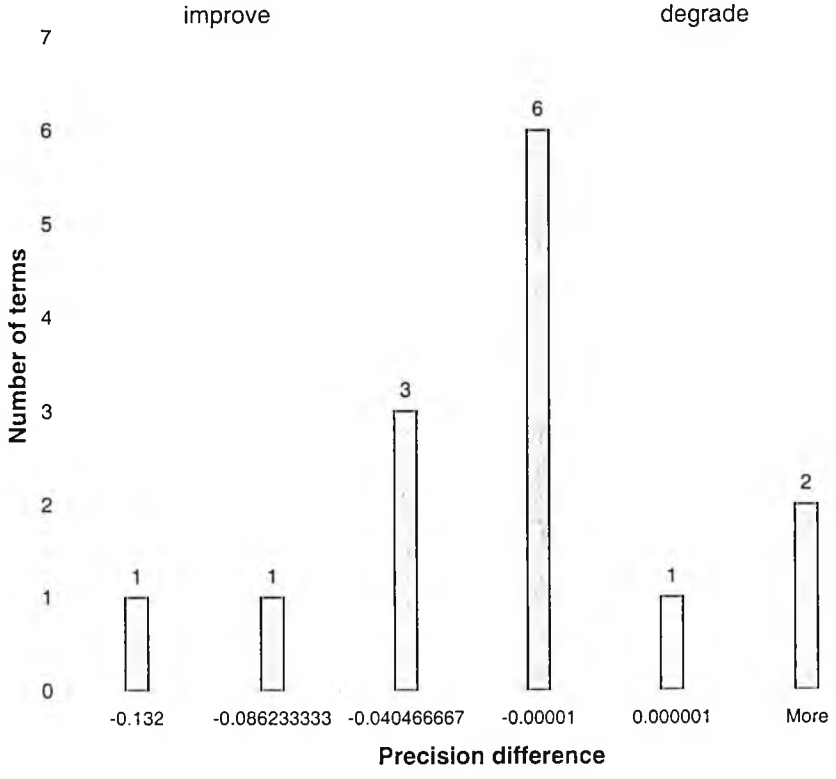


Figure 7.8. Distribution of precision differences for Category 9: *Collocate and an original query term*

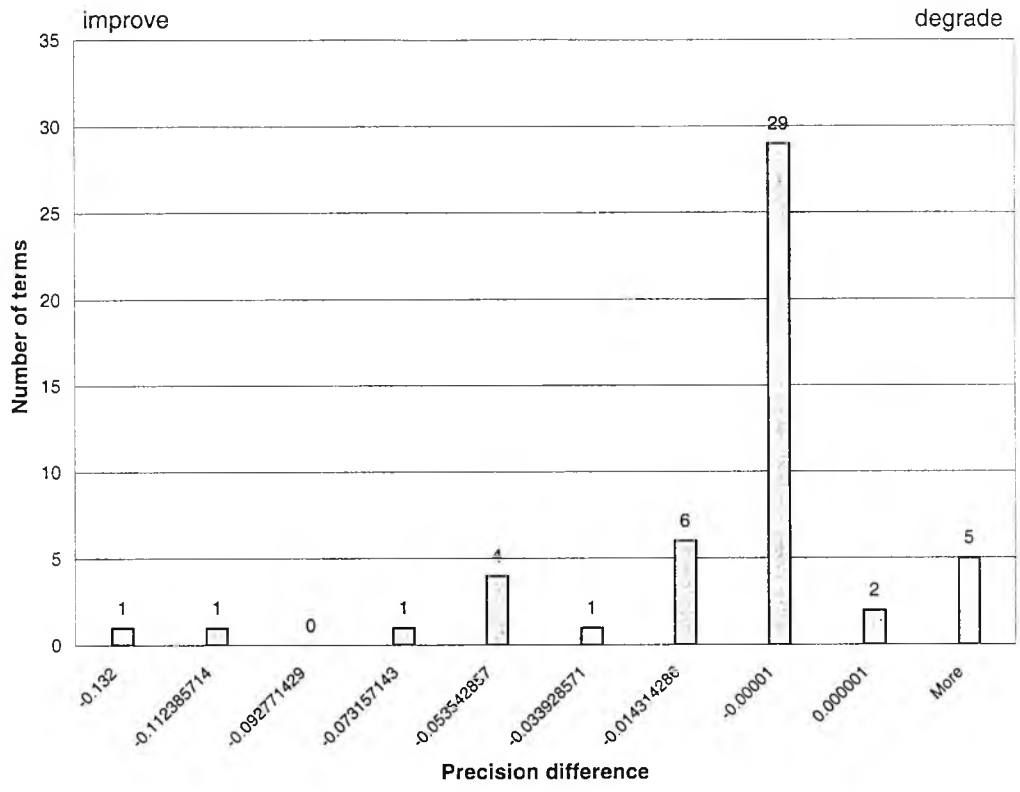


Figure 7.9. Distribution of precision differences for Category 10: *Okapi RF term and an original query term*

The histograms show that in all categories individual contribution of the majority of terms to performance increase is to a very small degree. It appeared, for example, that in the category 10. *Okapi RF term and an original query term* (figure 7.9) 29 out of 43 terms improving performance, resulted in the negligible increase of precision by up to 0.00001. The influence of the majority of terms on performance loss is also rather low.

The same analysis was done on the best run with local MI collocates – 'PRED 16 MI COL + 20 OK (100 window size)'. The patterns of term influence on average precision are similar to those of the run 'PRED 8 Z COL + 20 OK (100 window size)', analysed in this section. Results of the analysis of the run 'PRED 16 MI COL + 20 OK (100 window size)' are given in Appendix B.5.

7.5 Concluding remarks

In this set of experiments we examined the following hypothesis:

Hypothesis 2: Expansion of the initial query with statistically significant local collocates following relevance feedback results in significant performance improvement over Okapi relevance feedback under the same conditions.

Two types of retrieval experiments were conducted – query expansion with local collocates ranked by Z or MI, and query expansion with local collocates ranked by CRW. Also, a study of contribution of terms to performance by categories was carried out.

Query expansion with local collocates ranked by local variants of MI and Z was better than expansion with local collocates ranked by global variants of MI and Z.

There has been observed a strong tendency in the runs with local MI/Z collocates taken from windows 200 and 100 of precision growth with the increase in the number of collocates in the queries. This tendency weakened, however, with the decrease in the window sizes. This implies that collocates from smaller window sizes, ranked lower by MI or Z, have weaker relevance-predicting power than terms from larger window sizes, ranked similarly.

Another tendency, observed in MI and Z runs with 8 and 12 collocates per query term, consisted in precision growth with the decrease in the window sizes. It was less evident in the predictive runs with other numbers of collocates per query term.

An interesting fact is that precision of the lowest performing runs '8 MI COL (200 window size)' and '8 Z COL (200 window size)' can be improved in two ways: either by decreasing the size of windows, or by increasing the number of collocates per query term. The upper limits of both ways of precision improvement are, however, very similar.

The best performance among both MI and Z runs with collocates only was achieved by using 12 collocates from windows of size 20.

Combining of local Z collocates with Okapi RF terms for query expansion showed to perform reasonably well both retrospectively and predictively. Often addition of 20 Okapi RF terms results in precision gains over the corresponding runs with collocates only. Combined runs suggested some improvement over Okapi relevance feedback: combined run 'PRED 8 Z COL + 20 OK (100 window size)' is 4.3% better than 'PRED OK 35' and run 'PRED 16 MI COL + 20 OK (100 window size)' is 5.7% better than 'PRED OK 35'. Both gains are not statistically significant though.

Window size has no consistent effect on the performance of combined runs. Combined runs did not demonstrate the pattern of precision growth with the increase in the number of collocates either.

MI on the whole turned out to perform similarly or in some cases slightly worse than Z. Z can be considered a somewhat better statistic than MI for the selection of query expansion terms, however the difference between the performance results of MI and Z collocates is usually very narrow.

Query expansion with collocates ranked by our derived measure – Collocation Relevance Weighting – did not prove to be superior to either Okapi relevance feedback, or combined expansion with local Z/MI collocates and Okapi terms.

Overall, the results of retrieval experiments suggested that we are gaining some precision increase from including both Okapi terms and local Z/MI collocates in the expanded queries. But, these results were not consistent enough, and precision increase was not substantial enough to lead to a firm conclusion that this is a more efficient query expansion method than using Okapi RF terms alone.

To understand better which terms in the combined queries contribute most to performance growth or loss, we evaluated the influence of each category of terms on performance. It was hoped that such analysis would give us some distinct patterns of effect each term category has on performance. The only categories of terms that improve precision in a larger number of cases than degrade, are those with original query terms. Original query terms improve precision in 52% of cases, while degrading it in only 16% of cases. Terms that have a status of both an original query term and a collocate, or an original query term and an Okapi RF term, demonstrate a significantly more positive influence on precision, than just original query terms.

The analysis of degrees of terms' influence on precision change showed more or less similar degrees of positive and negative effects on performance by categories. The majority of terms in the categories including original query terms, that showed more positive effect on performance, turned out to improve precision only marginally.

The results obtained over the course of this set of experiments could not provide strong support for the above hypothesis. Though some tendency towards performance improvement (combined runs of local Z/MI collocates and Okapi RF terms) was observed, no statistically significant improvement was achieved by the experimental runs.

8. Lexical cohesion analysis using local collocations

8.1 Introduction

In the previous experiments we focused mainly on the use of collocates associated with each query term independently of others. In this experimental set we aimed to examine another aspect of a document's textual characteristics, which builds on the relationship between all query terms occurring in it. The object of study in these experiments is lexical cohesion between query terms in a document, estimated through their local collocational environments. Specifically we were interested in examining the hypothesis that document's level of lexical cohesion between query terms is related to its relevance property.

The rationale for this hypothesis is based on the following assumption. A relevant document contains the topic the user is interested in. Query terms in a relevant document are most likely to be used to describe the relevant topic. Words pertinent to the same topic tend to cohere with each other lexically and have some degree of similarity in their local collocational environments. In a non-relevant document query terms are not necessarily bound by the same topic as in relevant documents, therefore can occur in unrelated topics, and hence have no or little lexical cohesion.

We designed experiments to test whether sets of relevant documents have on the whole higher levels of lexical cohesion than sets of non-relevant documents. The experiments, their methodologies and results are presented in section *8.2*.

Following these experiments, we explored another hypothesis – whether lexical cohesion scores estimated for each document can lead to a better document ranking than Okapi document scores.

A set of retrieval experiments was carried out to test this hypothesis. Their methodology and results are presented in section *8.3*.

8.2 Comparison of relevant and non-relevant sets by the level of lexical cohesion

8.2.1 Experimental design

Our method of estimating the level of lexical cohesion between query terms was influenced by Hoey's method [Hoey91] of identifying lexical bonds between sentences (see section *2.3.3* earlier in the thesis). There is, however, a substantial difference between the aims of these two methods. Sentence bonds analysis is aimed at finding semantically related sentences. Our method is aimed at identifying whether query terms occurring in a document are semantically related, and measuring the level of such relatedness.

In both methods the similarity of local context environments is compared: in our method – fixed-size windows around query terms; in Hoey's method – sentences. A further difference is that Hoey's method identifies bonds between specific sentences; whereas the objective of our method is to compare the contextual (collocational) profiles of different query terms in a document. For this reason we combine all windows for one query term, building a merged window for it. Each query term's merged window represents its collocational environment in this document. We then identify the level of lexical cohesion between query terms by comparing their collocational environments. Each document can then be assigned a *lexical cohesion score* (*LCS*), based on the level of lexical cohesion between query terms occurring in it.

In more detail the algorithm for building merged windows for a query term is as follows:

Fixed-size windows are identified around every instance of a query term in a document. In the windowing technique we can encounter a situation, described on page 105 (section *7.3.1*), namely – overlapping windows of two different query terms. This, as already explained, does not matter when we extract collocates for each query term independently of other query terms. Here, we run into the following problem: let us assume that query terms **x** and **y** have overlapping windows and, hence, both are considered to collocate with term *a* (see figure 7.1 on page 105). If we were using our usual windowing technique we would add this instance of term *a* into the merged windows of both **x** and **y**. Next, when we compare these two merged windows, we would count this instance of *a* as a similar term between them. This would be wrong, since we refer to the same instance of *a*, as opposed to a genuine similarity link by two different instances of *a*. Our solution to this problem was to tag each term in a document with a unique number. When we extract collocates of query terms, we record them in the following format:

E.g.

environment	15	forest
protection	15	forest

node term — unique number assigned to the collocate — collocate

In most cases it is impossible to give preference to either of the two nodes and attribute the collocate to it. As we handle large-span collocates, most of them are related to the nodes topically, it is reasonable to assume that in most cases the collocate is likely to be there because it belongs to this topic, and not because it forms lexical-syntactic relationship with one of the nodes. For this reason, we attribute the collocate to one of the nodes randomly.

Two window sizes were tested: 20 and 40. These window sizes are large enough to capture long-span topically-related collocates. We did not use larger window sizes since it would increase the number of overlapping windows of different query terms. Random attribution of collocates to one of such query terms could be less accurate

with very large window sizes, as such windows have more chance to transcend topic boundaries. Thus, a collocate situated in the topic containing one query term, can be wrongly attributed to the other query term, situated further away in a different topic.

Comparison of similarity between merged windows

After merged windows for all query terms in a document are built, the next step is to compare their similarity by the collocates they have in common. We do pairwise comparisons between query terms, using two following methods:

- **Method 1:** Comparison by the number of links they have;
- **Method 2:** Comparison be the number of types they have in common.

Method 1

The first method is similar to Hoey's technique used in identifying bonded sentences (see section *2.3.3*). This method takes into account how many instances of common collocates each query term has. In figure 8.1 the first column contains collocates in the merged window of query term **x**, the second column contains collocates in the merged window of query term **y**. Lines between instances of the same collocate represent links.

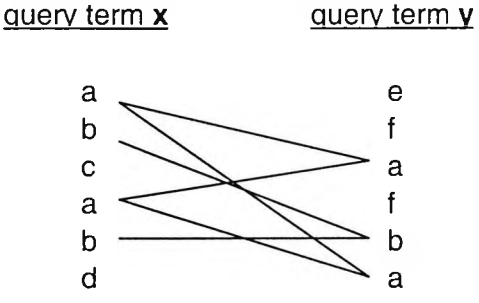


Figure 8.1. Links between instances of common collocates in merged windows of query terms **x** and **y**

In this example there are altogether 6 links. If there are more than 2 query terms in a document, a comparison of each pair of query terms is done, with the total number of links being recorded for the document.

A document's lexical cohesion score calculated using method 1 will be referred to as LCS_{links}. To calculate this score we need to normalise the number of links in a document by the total size of all merged windows in a document. The normalised LCS_{links} score is:

$$LCS_{links} = \frac{L}{V}$$

where:

L – the number of links in a document;
V – the size of all merged windows in a document.

Method 2

In method 2 no account is taken of the number of common collocate instances each query term co-occurs with. Instead only the number of common types between each pair of merged windows is counted.

Comparison of merged windows in figure 8.1 will return 2 types they have in common: *a* and *b*. Again, if there are more than 2 query terms, a pairwise comparison is done. For each document we record a number of types common between at least 2 merged windows.

A document's lexical cohesion score estimated using this method is LCS_{types}. It is calculated by normalising the number of common types by the total size of all merged windows in a document:

$$LCS_{types} = \frac{T}{V}$$

where:

T – the number of types common to 2/more merged windows in a document;
V – the size of all merged windows in a document.

In this formula the size of merged windows in a document – V is used as a simple normalisation factor on the analogy with LCS_{links}.

Construction of sets of relevant and non-relevant documents

To test the hypothesis that lexical cohesion between query terms in a document is related to a document's property of relevance to the query, we estimated average lexical cohesion scores for sets of relevant and non-relevant documents.

Initially we built two sets by taking top 10 relevant and non-relevant documents from the top 100 Okapi-ranked documents retrieved by the initial queries for 50 topics using bm2500. Relevance of documents was established using TREC relevance judgements file. Each set contains 187 documents.

Building relevant and non-relevant sets using the above method means that relevant and non-relevant documents for one topic are taken from different parts of the Okapi-

ranked list of documents. For example relevant documents could be spread throughout the list, in which case non-relevant documents, taken from the top of the list, will have higher average Okapi score than relevant documents found further down in the list. Also the standard deviation of Okapi scores will vary significantly between the sets. To ascertain that the difference between average cohesion scores in the relevant and non-relevant sets is not affected by the difference between average Okapi scores in the two sets, we should compose the sets where documents for each topic have similar mean and standard deviation of Okapi scores.

The procedure for selecting documents with similar Okapi scores was as follows:

First all documents in Okapi-ranked sets are marked as relevant and non-relevant using TREC relevance judgements file. Then each time a relevant document is found it is added to the relevant set and the nearest scoring non-relevant document is added to the non-relevant set until we select 10 documents per topic for each set. If relevant documents are clustered at the top of the list and the extraction of non-relevant documents with similar Okapi scores is impossible, we ignore these relevant documents and extract others further down in the list which have non-relevant documents around them.

After the sets are composed, the mean and standard deviation of Okapi scores are calculated for each topic in the relevant and non-relevant sets. If there is significant difference between the mean and standard deviation in the two sets for a particular topic, then the sets are edited by changing some documents until the difference is minimal.

We created two pairs of sets using this method: from the top 100 Okapi-ranked documents and from the top 1000 Okapi-ranked documents. In the first case each set comprised 178 documents, in the second - 268.

Relevant and non-relevant sets created by the first method will be referred to as *non-aligned*, while sets created by the second method will be referred to as *aligned.*

Comparison between corresponding relevant and non-relevant sets was done by average lexical cohesion score, which was calculated as:

$$Average\ LCS = \frac{\sum_{i=1}^{S} LCS_i}{S}$$

where:

LCS_i – lexical cohesion score of *i*th document in the set.

S – number of documents in the set size.

8.2.2 Analysis of results

Comparison of non-aligned relevant and non-relevant sets using the two methods described in the previous section showed significant differences between their average lexical cohesion scores (Table 8.1).

	Relevant set	Non-relevant set	Difference between sets
Method 1 (links)			
Window size 20	0.128	0.105	**18%**
Window size 40	0.124	0.095	**23.4%**
Method 2 (types)			
Window size 20	0.033	0.028	**15.1%**
Window size 40	0.028	0.023	**17.8%**

Table 8.1. Difference between the non-aligned relevant and non-relevant sets (documents taken from the top 100 Okapi documents)

Following the results of comparison of non-aligned sets, we decided to ascertain that they are not affected by differences in Okapi scores. Comparison of two pairs of relevant and non-relevant aligned sets – derived from 100 and 1000 Okapi ranked documents, also showed large difference between the sets (Tables 8.2 and 8.3). This proves that the difference between lexical cohesion scores of relevant and non-relevant documents is genuine, and not due to the differences in the documents' ranking positions in Okapi ranked output.

	Relevant set	Non-relevant set	Difference between sets
Method 1 (links)			
Window size 20	0.111	0.080	**27.9%**
Window size 40	0.099	0.071	**28.3%**
Method 2 (types)			
Window size 20	0.029	0.023	**20.7%**
Window size 40	0.023	0.018	**21.7%**

Table 8.2. Difference between the aligned relevant and nonrelevant sets (documents selected from the top 100 Okapi documents)

	Relevant set	Non-relevant set	Difference between sets
Method 1 (links)			
Window size 20	0.114	0.093	**18.4%**
Window size 40	0.110	0.094	**14.5%**
Method 2 (types)			
Window size 20	0.030	0.026	**13.3%**
Window size 40	0.026	0.022	**15.4%**

Table 8.3. Difference between the aligned relevant and non-relevant sets (documents selected from the top 1000 Okapi documents)

The first method of comparison by counting the number of links between merged windows appeared to be better than the second method of comparison by types. This suggests that the density of repetition of common collocates in the contextual environments of query terms offers some extra relevance discriminating information.

Interestingly, aligned sets of documents selected from top 100 Okapi ranked documents have greater difference in LCS than aligned sets selected from top1000 Okapi documents. One possible explanation to this is the fact that further down in the Okapi ranked list, documents contain fewer instances of query terms. In this case it is more difficult to establish similarity between collocation environments of query terms by using simple lexical repetition alone. We might need to take into account other types of repetition (see section *2.3.1*) or even count links between semantically related words, establishing their relatedness through a thesaurus (e.g., technique developed by [Morris91] described in section *2.3.2*). But these methods would require either more complex NLP processing, or the use of engineered term networks.

Another important point is the distribution of LCS scores in the sets (Appendix C.2). Although, the mean LCS of relevant documents is higher than the non-relevant, individual documents in both relevant and non-relevant sets have rather variable scores. For example mean LCS_{links} (window 20) in the relevant aligned set selected from top 100 Okapi documents is 0.111, and the score of the corresponding non-relevant set is 0.08 (see table 8.2). However, scores in both sets have similar standard deviation: 0.164 – in the relevant set and 0.15 – in the non-relevant set. This means that in both sets there is a large proportion of documents with the same scores. This can be seen from charts in Appendix C.2.

8.3 Re-ranking of document sets by lexical cohesion scores

8.3.1 Experimental design

Statistically significant difference in the average lexical cohesion scores between relevant and non-relevant sets, discovered in the previous experiments, prompted us to evaluate LCS as a document ranking function.

It was decided to conduct experiments on re-ranking the set of top 1000 Okapi-retrieved documents by their LCS scores.

Okapi sets were formed by using weighted search with the initial queries for 50 topics. Bm25000 weighting function without relevance information, instantiating Combined Weight (CW) was used for searching. Tuning constant k_1 (controlling the effect of within-document frequency) was set to 1.2 and b (controlling document length normalisation) was set to 0.75.

Okapi bm2500 function outputs each document in the ranked set with its matching score – MS. We decided to test re-ranking with a linear combination function (*COMB-LCS*) of MS and LCS. Tuning constant x was introduced into the function to regulate the effect of LCS:

$$COMB\text{-}LCS = MS + x * LCS$$

The following values of x were tried: 0.25, 0.5, 0.75, 1, 1.5, 3 and 30.

We conducted experiments with two types of lexical cohesion scores:

LCS_{links} – calculated using method 1 of comparing query terms' collocation environments (p. 119);
LCS_{types} – calculated using method 2 of comparing query terms' collocation environments (p. 119).

The window sizes tested were 40 and 20.

8.3.2 Analysis of results

Average precision results of re-ranking with the combined linear function of MS and LCS with different values for the tuning constant x are presented in table 8.4. Trec_eval summaries are given in Appendix C.3.

Runs with different *x* values	Window size 40	Window size 20
Okapi	**0.1310**	
Method 1 (links)		
0.25	**0.1332**	**0.1348**
0.5	**0.1339**	**0.1348**
0.75	**0.1348**	**0.1336**
1	**0.1341**	**0.1337**
1.5	**0.1327**	**0.1335**
3	**0.1278**	**0.1320**
30	**0.0809**	**0.0924**
Method 2 (types)		
0.25	**0.1310**	**0.1312**
0.5	**0.1309**	**0.1313**
0.75	**0.1311**	**0.1313**
1	**0.1320**	**0.1318**
1.5	**0.1314**	**0.1318**
3	**0.1308**	**0.1316**
30	**0.1175**	**0.1241**

Table 8.4. Results (in average precision) of re-ranking Okapi document sets by *COMB-LCS*

The results show that there is no or negligible gain from using LCS in ranking over Okapi. One possible reason for this could be sought in the fact that LCS scores in the relevant/non-relevant sets, compared in the previous experiment, were highly variable (see Appendix C.2). As mentioned in section *8.2.2*, a high proportion of documents in both relevant and non-relevant sets have similarly high or low LCS scores. Another possible reason for no or little gain from LCS is the fact that for a large number of relevant documents no links or common types could be identified (see Appendix C.2). We assume that with our method of comparing lexical environments of query terms

through repetition of their collocates, only a certain proportion of lexical links is determined. For a fuller analysis other phenomena forming lexical cohesion could be considered.

8.4 Concluding remarks

In this set of experiments we explored the property of lexical cohesion between query terms in documents, if it is related to relevance, and whether it can be used to predict relevance in document ranking. Two hypotheses were put forward. The first hypothesis we studied was:

Hypothesis 3: There exists statistically significant association between the level of lexical cohesion of the query terms in documents and relevance.

We conducted experiments (section *8.2*) by building sets of relevant and non-relevant documents, calculating their lexical cohesion scores and comparing the averages of these scores. The experiments showed that there exists a statistically significant difference between average LCS of relevant and non-relevant documents. They also proved that this difference is genuine, and not due to the difference in the documents' positions in Okapi-ranked lists.

The experimental results provided support for hypothesis 3, giving evidence that there exists a statistically significant relation between the probability of relevance and the level of lexical cohesion between query terms.

Following these experiments, we explored another hypothesis:

Hypothesis 4: Re-ranking of Okapi document sets by lexical cohesion scores results in significant performance improvement over initial Okapi ranking.

We conducted experiments on re-ranking Okapi document sets with a linear combination function of Okapi matching score and lexical cohesion score. Different values of a tuning constant x, regulating the effect of LCS were tried. The results suggested no significant improvement over Okapi ranking, thus providing no support for hypothesis 4.

Results achieved in the first half of this set of experiments – i.e., difference between relevant and non-relevant documents by their average lexical cohesion scores – are considered to be rather optimistic. Although, our approach to using LCS in document ranking tried in the second half of the experiments did not prove useful, the first half of experiments suggested that the concept of lexical cohesion is linked to relevance. To achieve practical benefit from lexical cohesion in document relevance discrimination, more experimentation is needed. Lexical cohesion, as a text property, is formed not only through word repetition, but other more complex lexical relations (see section *2.3.1*). Limited by the scope of this project, we looked into lexical

cohesion between query terms achieved only through repetition of their collocates. Other phenomena forming lexical cohesion, which are out of the scope of this project, could also be taken into account in identifying lexical cohesive links between environments of query terms. A more complete analysis of lexical environments of query terms is expected to provide more support to what has been suggested by the results of the first half of our experiments. As mentioned in section *2.3.2*, the analysis of lexical link distribution by [Ellman2000], showed that the most common link type – between repetitions of the same word, is closely followed by the link between words belonging to the same thesaurus category. A possible future development of our method could, thus, consist in defining links on the basis of repeated words and words related through an engineered term structure like a thesaurus or a lexical net.

9 Conclusions and recommendations

In this project we have undertaken an experimental study of long-span collocation in its application to probabilistic information retrieval. The research question investigated in the project was whether the use of long-span collocates can improve performance of probabilistic IR. To examine this research question, three different methods of integrating collocation information into a probabilistic model were developed and evaluated.

The type of collocation, we aimed to explore and apply for the use in IR, is long-span collocation, motivated by the lexical-semantic relations. Long-span collocates occur together within large contextual environments because they are bound by some lexical-semantic relationship. They belong together to the textual topic they co-occur in. Long-span collocation is different, on the one hand – from short-span collocation, motivated by lexical-grammatical or habitual relations, and on the other hand – from document-wide co-occurrence.

Four hypotheses were examined over the course of the project. The studies exploring these hypotheses formed three groups:

Global collocation analysis

This group of study was aimed at exploring the first hypothesis:

Hypothesis 1: Expansion of the initial query with statistically significant global collocates of query terms results in significant performance improvement over the initial query evaluated under the same conditions.

Performance of the queries expanded with global collocates of query terms, ranked either by MI, or Z did not prove to be better than performance of the original queries.

The main finding of this study was that information gathered in the form of single words occurring in the environments of all instances of a query term in the corpus, does not have substantial relevance-discriminating power, even though the terms are ranked by significance of their co-occurrence with the query terms. The main reason for this is thought to be the fact that many query terms are words from the general lexicon, that can occur in a broad range of contexts. Even occurrences of the same sense of a word can be used in a wide range of topics. As the number of relevant documents in which a query term occurs is usually much smaller than the number of non-relevant documents with this term, only a small proportion of collocates come from contexts which have any relatedness to the topic of user's interest. It was therefore considered that using collocates from the contexts of query terms in documents for which there exists some evidence of relevance to the user's need, might result in a better performance. This led us to the second hypothesis, explored in the next group of studies.

Local collocation analysis

The hypothesis examined in this study group was:

Hypothesis 2: Expansion of the initial query with statistically significant local collocates following relevance feedback results in significant performance improvement over Okapi relevance feedback under the same conditions.

The experimental results confirmed our initial assumption that using collocates from the documents, for which some relevance information is known, gives significantly better results than using collocates from any contexts of query terms in the collection indiscriminately. As for the improvement over Okapi relevance feedback some tendency towards performance gain was observed, when using both collocates and Okapi RF terms. However it was not significant enough to conclude that local collocation analysis method is superior to the existing Okapi relevance feedback.

Some findings of this study group are summarised below:

1. There is no substantial performance difference between the use of MI and Z ranked collocates, though there is slightly more evidence in favour of using Z statistic.
2. On the whole local variants of MI and Z statistics perform better than global variants of these statistics.
3. Top 8/16 MI- or Z-ranked collocates extracted from smaller window sizes tend to have higher relevance-predicting value than top 8/16 MI- or Z-ranked collocates extracted from larger window sizes. Further down in the ranked lists collocates from different window sizes show rather similar relevance-predicting values.
4. Adding more Z/MI ranked collocates from larger window sizes improves performance noticeably, while adding more collocates from smaller window sizes does not have a significant effect on it.
5. Expanding queries with both collocates and Okapi RF terms on the whole gives better results than using collocates alone. Some runs are also slightly better than the best Okapi RF runs.
6. Query expansion with collocates ranked by Collocation Relevance Weight (CRW) measure is not better than expansion with MI or Z collocates.
7. Analysis of terms' influence on precision by categories showed that original query terms, particularly if they are also either collocates, or Okapi RF terms, improve performance significantly more often then they degrade it. However, the majority of them improve it only marginally.

The conclusion prompted by this study is that no or very little advantage is gained from using long-span collocates, over the use of document-wide co-occurrence, present in the existing Okapi relevance-feedback technique. Although individual results were suggesting some improvement over Okapi RF, they were not conclusive.

It can be inferred from the experiments that the key variables we were focusing on – window size and collocation ranking measures – are not the most critical factors affecting performance (evident from rather similar results of MI/Z collocates and non-consistent effect of window length). Considering, for example, certain composite lexical units as candidates for significant collocates instead of single words could be a more decisive factor. There has been some evidence in past research that selecting collocates as noun groups can lead to improved performance (Xu and Croft's Local Context Analysis [Xu96]).

Lexical cohesion analysis using local collocations

This study group was aimed at analysing documents' property of lexical cohesion between query terms, estimated through comparison of their collocation environments. Specifically we aimed to understand how lexical cohesion is related to relevance, and whether it can be useful for relevance prediction. Two hypotheses were examined in this study. The first hypothesis was:

Hypothesis 3: There exists statistically significant association between the level of lexical cohesion of the query terms in documents and relevance.

The experimental results proved that there is statistically significant difference between the average level of lexical cohesion in relevant and non-relevant document sets. Moreover, it was ascertained that this difference is genuine and not due to the difference in documents' Okapi scores. Difference between sets created from top 100 Okapi-ranked documents was larger than between sets created from top 1000 Okapi-ranked documents.

Our next research direction was targeted at using the relevance-predicting quality of lexical cohesion to improve Okapi document ranking. The hypothesis investigated was:

Hypothesis 4: Re-ranking of Okapi document sets by lexical cohesion scores results in significant performance improvement over initial Okapi ranking.

The results did not prove that re-ranking documents by their lexical cohesion scores is superior to the original Okapi ranking. One reason for this could be that a number of relevant documents does not have any common types or links. This is evident from the results of the analysis of relevant sets in the first half of the experiment. Limited by the scope of this project, we were counting cases of word repetition alone in the form of common collocates. Lexical links, formed by other textual devices, were left out. Taking account of a larger proportion of lexical links could possibly lead to bigger performance gains.

In the recent years there have been undertaken a number of research activities which focus on the related aspects of the IR ad hoc task as this project. Our first two techniques described in chapters 6 and 7, Global collocation analysis and Local collocation analysis, are methods of automatic query expansion. Query expansion is a very widely used technique of incrementing relevance-discriminating potential of the query. Over the years there have been developed, with varying degree of success, numerous variations of statistically-based query expansion. Although our specific techniques of implementing query expansion are novel, there have been and are being undertaken experiments on query expansion comparable to ours, i.e. global query expansion techniques, which analyse the entire collection, and local techniques – relevance and pseudo-relevance feedback. Therefore it is useful to compare our results with the results achieved in related work and see if there are any common patterns of success or failure from using specific techniques. We will attempt to outline what lessons can be learned from our and related research and which techniques appear to work better.

In both of our query expansion approaches – global and local – we selected collocates on the basis of the strength of their association with each query term independently, i.e. without considering the level of association with other query terms. There is an experimental evidence [Qiu93, Xu96] that both global and local query expansion techniques could benefit from taking into account similarity of terms to the whole query.

Qiu and Frei [Qiu93] analysed previous, mostly unsuccessful, attempts at automatic collection-wide query expansion and came to the conclusion that the main unsolved problems are:

> " 1) the selection of suitable terms;
> 2) the weighting of the selected additional search terms. " [Qiu93, p.162]

Specifically they point out that although many methods of estimating relatedness between terms have been tried, one thing they have in common is that all of them select terms that are strongly related to a single query term. Qiu and Frei believe that it is important to select terms that are strongly related to what they call 'query concept', expressed by all query terms. They developed a global query expansion method which attempted to solve the above two problems. Their method relies on an automatically constructed co-occurrence based term-term similarity thesaurus. The main principle of their method is that query expansion terms are selected from the thesaurus depending on the degree of their similarity to all the terms in the query. The expansion terms are assigned weights which also reflect their similarity to the entire query. Their retrieval experiments demonstrated improvements in average precision from 18% to 29%.

Qiu and Frei also warn against the hazard of using in query expansion terms above rather high thresholds of similarity. They argue that although setting high thresholds allows us to get terms very strongly related to individual query terms, the chances of finding among such terms a sufficient number of those similar to all query terms are very weak.

From this it follows that performance of our global query expansion approach might be improved if, instead of adding collocates with very high degree of association with

a single query term, we applied a secondary term selection to collocation lists of all query terms. The secondary term selection stage, applied at search time, should re-rank collocates by their strength of association with all query terms. This task is, however, not trivial in the framework of our approach: MI and Z scores of collocates of different query terms are not comparable (see p. 80). In order to estimate the degree of relatedness of a collocate to all query terms, we need to normalize its different scores of association (i.e. in different collocation lists).

Another query expansion approach which also uses the principle of term relatedness to the entire query is Xu and Croft's LCA – Local Context Analysis [Xu96] (see section 3.3.2, p. 48). LCA is a type of local feedback. Over time it has shown rather consistent performance improvements over the baseline (INQUERY) on TREC collections. Similarly to our local approach to query expansion Xu and Croft extract candidates for query expansion from fixed-size windows (300 words), which are, however, located differently in text. In our method we locate windows around each occurrence of a query term. In LCA the window is the best passage. The best passage is usually a section of text with a maximum concentration of query terms, therefore query expansion terms from best passages could be argued to have an extra value of occurring close to more than one query term.

Another difference between LCA and our method is that in LCA the query expansion units are noun phrases of the form N, NN, NNN, whereas in our method collocates are always single words. Earlier Jing and Croft [Jing94] conducted a rather extensive research of the effect expansion with different syntactic categories has on performance (see p. 48). Their experiments showed that above types of noun phrases result in better performance than any other types of single or compound terms.

Term selection formula in LCA is based on a widely used *tf*idf* measure, but its main feature is that it promotes terms which are significantly related to all query terms. In our local query expansion method, like in our global method, we select terms most strongly associated with a single query term. We did, however, assess the impact of local collocates of 2 or more query terms on performance (see p. 109); this did not show that such collocates are better query expansion terms than collocates of 1 query term. But it might be partially attributed to the problem pointed out by Qiu and Frei (see earlier), i.e. that we only selected collocates ranked high by their relatedness to individual query terms. It is reasonable to hypothesise that collocates which have weaker relatedness to several query terms (i.e. ranked low in individual collocation lists and ignored by our method) might be better query expansion candidates than collocates related strongly to just one query term.

A major problem that we believe affects all title-only retrieval approaches in TREC experimental environment is the fact that TREC documents are judged relevant on the basis of the complete topics. Titles never reflect all nuances of the information need specified in the description and narrative. If relevance of documents was determined on the basis of titles only, we would in many cases have rather different relevant sets from those we have now. For example the title of the topic 292 – "Worldwide Welfare" – contains no indication of the major relevance criterion for this information need, which we find from the topic description: "Identify social programs for poor people in countries other than U.S.". We can find further restrictive criteria in the

narrative: "...A relevant document should identify the source of the monies used to support such welfare programs". No matter how sophisticated a query expansion algorithm is it cannot infer such restrictions from short underspecified titles.

Strzalkowski et al. (GE research group) from their TREC-8 experiments concluded that type and length of the query are some of the major factors contributing to the performance of IR systems, particularly those using NLP techniques [Strzalkowski2000]. They developed a query expansion method in which a query derived from all fields of the topic is expanded with top ranked paragraphs, creating a meta-document which is used as a query (see also p. 40). Their approach is related to ours in the sense that they also make use of locality in text for query expansion.

One distinctive feature of GE approach is that they link all words in the query, co-occurring in the same sentence less than 3 words apart and use them as phrases in searching (applying INQUERY's #phrase operator, which requires ordered co-occurrence within a limited span if words co-occur frequently in the collection). By this they take into account mainly lexical-grammatical relations, whereas in our experiments we were interested in words co-occurring due to pertinence to the same topic, therefore we did not set any restrictions as to how near collocates (from the expanded query) need to occur in the retrieved documents. Moreover, we did not require both members of the collocation pair to occur in a document. Our idea was to retrieve documents which may not contain query terms, but have their collocates instead. The GE group claims that their method results in quite dramatic improvement in average precision – from 40% to 130%.

GE group's strategy of using top best paragraphs for expansion is reminiscent of Xu and Croft's use of best paragraphs for LCA. However, unlike stringent selection of expansion terms from best passages in LCA, they simply add best paragraphs to the original query. They argue that passages from related but not necessarily relevant documents are also useful, since their role is to uncover different aspects of the initial query [Strzalkowski2000].

Their query expansion terms, selected from best passages on the basis of maximum concentration of query terms (like in LCA), have a characteristic of occurring close to many query terms. In our approach we defined windows for collocate extraction independently around each query term occurrence, based on the idea that topic does not necessarily manifest itself in text as a single uninterrupted piece of text, but can re-appear in different parts of it. Therefore we decided not to select collocates only from those documents which have query terms near each other. However, it could be argued that topics characterised by the concentration of query terms may be a better source of expansion terms than parts of a topic scattered across the document. The success of results gained by some groups may be partly attributed to their passage-based method of defining text areas for the extraction of query expansion terms. Best passages might be a better source of expansion terms than windows around query terms. Examples of successful passage-based approaches, which could support this argument, include those by the above mentioned GE [Strzalkowski2000] and UMass [Xu96] groups, as well as MultiText team [Cormack2000]. MultiText team's approach consists in selecting only those passages during the first run in which all query terms occur. By this they claim to improve precision, as well as efficiency of the system.

AT&T group [Singhal2000] also make use of the idea of concentration of query terms in a limited span of text. Their document re-ranking technique rewards those documents which contain query terms in the same sentence or adjoining sentences. However, they did not obtain any evidence that this method improves performance.

Cornell group [Buckley97, Buckley2000] too used proximity information of the query terms to re-rank the initial set retrieved by SMART. In addition they also used co-occurrence information between query terms and other terms in query expansion. Following the initial run, they calculate correlation between all query terms in the top 1000 documents, i.e. they determine how frequently each query term co-occurs in the same document with any other query term. Then, taking the top 50 documents, and breaking them into overlapping fixed-size windows, they compare each window to the query. Each query term is assigned a weight which is inversely proportional to the maximum correlation the term has with any previous occurrence of any query term in the same window. The motivation behind this is that if two terms are known to correlate strongly, the occurrence of the second term is not going to contribute much information about the relevance of a window to the query. They identify top 20 documents with the highest scoring windows and use them for blind feedback.

The Cornell's approach to downweighting the terms highly correlated with previously occurring terms is opposite to Xu and Croft's Local Context Analysis, which on the contrary rewards query terms highly correlated with each other. TREC 8 performance results of the two systems – Cornell's SMART [Buckley2000] and INQUERY (implementing LCA) [Allan2000] - were very close. Therefore it is difficult to say which of these two approaches to weighting correlated terms is better.

In our approach we, similarly to LCA, rewarded those documents which contain more query terms or their highly associated collocates, relying on the standard Okapi term weighting and document scoring functions. It is difficult to say at this stage whether our method can benefit from a different weighting technique for queries expanded with collocates significantly co-occurring with query terms.

Kwok et al. [Kwok98, Kwok2000], participating in TREC with their PIRCS system, developed a blind feedback technique somewhat related to our local query expansion approach. From top-ranked documents they select terms by the significance of their document-level co-occurrence with query terms. They use expected Mutual Information to estimate the strength of association between each query term and each term occurring in a given document. They then calculate average MI for each term from the MI values of its association with every query term. By doing this they take into account relatedness of a term to the entire query. Another important aspect of their technique is that they filter out terms with too high or too low collection frequencies. They have obtained experimental evidence that setting certain frequency thresholds on expansion terms has positive effect on performance. The number of terms that they add to the original query is rather small, and is dependant on the query size, for example a 3-word query is expanded with only 1 additional term, and 12-word query is expanded with 6 additional terms.

Our local method is different from Kwok's approach by the following features:

- The collocates in our method are selected from the limited areas around query terms, which we believe is a better technique in long multi-topic documents;
- The association is estimated in relation to each query term independently (see discussion above);
- High- and low-frequency terms are not filtered out; however we apply Okapi term weights to all expansion terms which partly depend on the term's collection frequency.
- We experimented with larger numbers of expansion terms.

Kwok et al. use a range of other techniques alongside co-occurrence based query expansion. They analysed the contribution of each technique to performance and came to the conclusion that their co-occurrence based query expansion contributes to performance only marginally [Kwok98, p. 253].

A somewhat different research direction that could be explored in relation to collocation techniques is their use in interactive IR. It is interesting to explore the effectiveness of automatic term suggestion to users either using global collocation analysis or local collocation analysis following blind feedback.

The Rutgers team [Belkin2000] conducted a study within TREC 8 Interactive Track framework, comparing the performance of user-controlled term suggestion – RF, and system-controlled term suggestion – LCA. Initially, based on earlier studies [Koenemann 1996 and Park 1999 cited in Belkin2000] which indicated that users prefer having more control over the system, they hypothesised that user-controlled term suggestion will be preferred by the users. Their experiments however indicated the opposite. The reason for this is believed to be the fact that the complexity of the system, requiring the users to make relevance judgements, is much higher than that of the system suggesting terms automatically. The higher degree of control over the system was simply not worth the increased task complexity. The performance of two systems was rather similar, however the number of terms that users selected among LCA-suggested terms was higher than the number of terms selected among RF-suggested terms. In the RF system users also spent more time defining their own terms than in the LCA system.

The Rutgers team's study provides positive evidence that terms extracted using their co-occurrence statistics and suggested automatically to users in interactive IR can be useful.

McDonald et al. [McDonald97] (see p. 46) also argue that automatically constructed co-occurrence based term networks can be used in interactive IR as a less effort-demanding alternative to RF. Co-occurrence term networks in their view have an advantage of giving the user a broader perspective of the database coverage, which is more difficult to achieve by examining full texts of documents.

Analysing the results of our query expansion experiments, aimed at studying whether information about words' patterns of collocation in text can give us more relevance-discriminating evidence, we came to the conclusion that the reality of language-behaviour and the nature of the IR task itself are characterised by too many degrees of

uncertainty, making this task not trivial. First, uncertainty is already present to a greater/less extent in the short queries that we used. Secondly, it is often amplified by an even greater uncertainty, when we take into account all the variety of a word's instances in a large thematically heterogeneous text collection. As discussed earlier one way to reduce the effect of the uncertainty is to consider only those features in the environments of query terms' occurrences that are related to the entire query, i.e. that are associated with all query terms.

Local analysis of environments of each query term in a limited set of documents (in our experiments in the known relevant documents) worked better than analysis of the entire collection using the same approach, because here we deal with a much more homogeneous set of contexts which are likely/known to be related to the user's topic. However, here as well we might benefit if we select the candidate expansion features by their relation to the entire query.

There is, however, a possibly negative side to local approaches: if we consider only few documents from the top of the retrieved set, which are likely to be biased towards the initial query, is the information we get from them always enough to retrieve other documents which might cover different aspects of the topics relevant to the user's need? In other words, do we get enough information to diversify the query formulation if we consider only a limited number of documents biased towards the underspecified short query formulation? It might be that global techniques, provided that they can handle the task of selecting features related to the entire query, could give us a richer query expansion material than local techniques. For example, the above mentioned global technique by [Qiu93] or Phrasefinder [Jing94] (see also p. 47) showed improvements over initial queries. Although Phrasefinder performed worse than LCA, Xu and Croft point out that one of the main reasons is that Phrasefinder did not require terms to be related to the entire query [Xu96]. Although at present much of the research effort is directed towards improvement of local feedback techniques, we believe that global techniques also need to be studied further.

One other role of collocates, that we believe is important, is their use in determining contextual relatedness – cohesion – of query terms in documents. There has been very little research of lexical cohesion in IR. Previous studies of cohesion in IR focused either on document indexing [Stairmand97], or text summarisation/passage retrieval [Manabu2000, Hearst93, Hearst94]. Our idea of identifying the degree of relatedness of query terms in text through their lexical cohesion is novel. Cohesion between query terms, determined through their collocates, has been shown by our experiments to be associated with relevance. So far we have not achieved practical benefit from it in IR. But the main objective of this experiment was to ascertain if there is a genuine relation between relevance and cohesion, unaffected by other factors. To achieve practical benefits from it in IR we might need to combine a number of different techniques. For example, our method was specifically developed to determine if query terms occurring in different parts of text are topically related (since the same topic can resurface in different parts of text). This method does not address relatedness of query terms occurring close to each other in text, since their relatedness can already be assumed from the proximity of their occurrence. We need to combine a method which rewards documents where query terms co-occur in the same part of text with the

method rewarding documents with coherent query terms occurring in different parts of text.

The technique of estimating lexical cohesion might also be refined. First, collocates used to compare the similarity between lexical environments of query terms, could be weighted to reflect their collection frequency. Words in the environments of query terms include a large share of topic-independent words, e.g., very common and discourse organising words. Occurrences of such words in the environments of query terms give us less certain indication that these environments are topically related, than occurrences of less frequent terms with higher information value. Therefore we may improve the accuracy of the estimation of cohesion levels between terms if we use term weighting.

It is also worth experimenting with other types of cohesive links between lexical environments, for example considering synonyms and other paradigmatically related words, identified through manually constructed resources.

Taking a broader perspective on collocation research, we can say that this project represents a specific direction in the study of collocation in IR, and it certainly draws no final line under collocation research in IR. One dimension within which we delineated the scope of this project is the approach to text analysis. Our approach is largely statistical. It is agreed that statistical methods are crude and that they treat language elements simply as tokens, ultimately capturing only what lies on the surface of language-expression. However, they have one big plus – namely, not being affected by subjectivity. They rely only on the first-order resources – texts. When we move away from the domain of purely statistical corpus analysis, to the domains of NLP, knowledge-based approaches or AI, we start to rely on second-order human-engineered resources, like dictionaries, rules and knowledge bases; and hence on the degree of their representativeness, generality and objectivity. Such approaches may bring certain performance improvements, though often at a cost of reduced robustness, universality and self-sufficiency – strong points of statistical methods.

Another dimension, in which we delimited the scope of this project, is the IR model. We evaluated our methods within the context of probabilistic retrieval. We believe that collocates do indeed represent useful information, however, first, this information may already be present to some extent in the probabilistic model, and secondly, the relevance-discriminating potential of collocation may be realised differently with other models, for example using different term weighting approaches. It is, therefore, believed that before the final word can be said about the usefulness of collocation in IR, it should be studied in the context of other retrieval models too.

Appendices

Appendix A

Global collocation analysis

A.1 Selected lists of top global collocates ranked by MI and Z

Tables*

* Synonym sets represented in the tables by GSL codes are listed in Appendix D.3

Table A.1. Top 50 MI and Z ranked collocates of the stem *export* (topic 251)

MI list		Z list	
ulc;	4.07097	qtr;	190.15
intmd;	3.82043	import;	158.985
housg;	3.82043	trade;	145.509
cnsmer;	3.82043	tonn;	127.831
ploy;	3.81024	output;	113.17
cofac;	3.80356	product;	104.248
sdrbase;	3.80228	@0192;	99.9556
cocom;	3.80132	@0135;	99.9199
ecgd;	3.7894	@0007;	97.1803
beston;	3.78801	@0290;	96.5365
sace;	3.74935	countri;	95.7611
frigg;	3.74243	@0180;	91.3817
pigmeat;	3.71352	@0099;	89.9242
atpc;	3.7094	produc;	89.2063
trefgarn;	3.70854	surplus;	88.8642
mfg;	3.70339	@0136;	88.7311
unem;	3.69574	manufactur;	83.1684
eep;	3.67957	economi;	82.4373
countertrad;	3.66289	growth;	81.1221
westar;	3.6328	tariff;	80.2385
taa;	3.62451	deficit;	79.543
vac;	3.60526	quota;	79.0593
ncm;	3.60072	domest;	77.3983
tequila;	3.59804	oil;	75.6102
durum;	3.56366	wheat;	75.1104
poundsm;	3.5284	balenc;	74.6966
dollarsbn;	3.51058	@0060;	72.2969
presilei;	3.50632	@0176;	72.1151
natuna;	3.48163	textil;	71.6311
glazebrook;	3.47648	coffe;	71.5933
bcm;	3.47648	foreign;	71.1713
ouzo;	3.43856	econom;	68.8515
gsp;	3.41604	farmer;	68.8443
mfa;	3.37565	farm;	68.2448
eximbank;	3.37027	@0330;	67.1462
vra;	3.36587	@0400;	67.0024
sd;	3.36207	ecgd;	66.1734
yrend;	3.34399	year;	66.1065
paranapanema;	3.335	yen;	65.5251
exim;	3.335	crop;	65.0673
forgemast;	3.32567	banana;	63.4647
cashew;	3.31142	bn;	60.5497
wobb;	3.30106	@0399;	60.1472
kembl;	3.2996	@0186;	60.1427
gutteridg;	3.2996	consumpt;	60.0868
reexport;	3.29601	dump;	59.3323
baxendal;	3.28992	cnsmer;	59.0653
gluten;	3.28609	devalu;	58.5056

Table A.2. Top 50 MI and Z ranked collocates of the synonym group @0180 [*industrial, industry*] (topic 251)

MI list		Z list	
opaqu;	7.64165	manufactur;	121.525
anz;	7.64165	product;	102.841
vondran;	2.09382	coal;	86.0634
fls;	2.06849	steel;	84.156
groupfiat;	2.04023	@0192;	82.7379
hurn;	2.03648	export;	81.4553
mhi;	1.99841	@0068;	80.0811
kukj;	1.99112	sector;	79.8543
ysx;	1.98134	competit;	77.065
wobb;	1.97654	@0134;	70.0765
jandec;	1.97654	output;	68.6317
hhi;	1.97369	electr;	68.4795
iai;	1.95048	technologi;	67.8816
standardpoor;	1.94843	engin;	63.0258
guarino;	1.94572	heseltin;	62.9607
sematech;	1.92101	@0290;	61.3095
vda;	1.91095	regul;	59.4547
nonaid;	1.91095	@0104;	59.2109
intmd;	1.91095	plant;	58.3445
interoper;	1.91095	aerospac;	58.1132
hsct;	1.91095	car;	57.3793
housg;	1.91095	energi;	57.3755
cefic;	1.91095	cbi;	56.1282
cecimo;	1.91095	produc;	56.0842
ploy;	1.90076	economi;	56.0096
cenelec;	1.87992	capac;	54.9737
iro;	1.87853	confeder;	54.281
bitterfeld;	1.87443	develop;	53.5562
baoshan;	1.87443	dti;	53.4666
unem;	1.87372	consum;	52.9895
endperiod;	1.87248	trade;	52.7232
ifpi;	1.86913	busi;	52.2626
allgold;	1.86723	employ;	52.2346
lpf;	1.86656	import;	50.136
fonograf;	1.86515	econom;	49.7428
gerl;	1.86365	chemic;	49.1703
spector;	1.86286	recess;	48.5055
mitis;	1.86033	supplier;	47.7412
savona;	1.85751	dow;	47.596
incl;	1.8565	invest;	46.876
cen;	1.85437	textil;	46.7642
sabic;	1.84955	year;	46.5295
hatakeyama;	1.8382	privat;	46.0643
bdi;	1.8382	growth;	45.7196
vdma;	1.83695	motor;	45.5842
mercosul;	1.83695	job;	44.9389
epb;	1.83295	automot;	44.4913
unemp;	1.83003	@0013;	44.226

Table A.3. Top 50 MI and Z ranked collocates of the stem *alien* (topic 252)

MI list		Z list	
narva;	6.67601	narva;	43.6465
seti;	6.45783	curzon;	39.3772
curzon;	6.00015	seti;	33.4198
@0111;	5.95533	estonia;	30.8962
kashmiris;	5.8298	estonian;	30.5858
thian;	5.68916	immigr;	28.7626
corang;	5.68023	parti;	27.7749
zoe;	5.56167	hindus;	24.8815
kelman;	5.49848	camacho;	24.8815
blasfemi;	5.42809	kashmiris;	24.5734
channon;	5.41053	voter;	23.8587
daim;	5.37983	zoe;	23.303
celin;	5.37983	karajan;	23.1119
karajan;	5.32659	baird;	22.8774
boarder;	5.28791	haider;	22.7896
siegmund;	5.28198	@0111;	21.9208
nostrik;	5.27726	poe;	21.4189
poe;	5.21967	thian;	21.1312
lyceum;	5.20629	boarder;	21.0979
hindus;	5.11081	cultur;	20.5486
beater;	5.09526	polit;	20.3707
bobsleigh;	5.0886	kashmir;	19.9378
meri;	5.07317	pkk;	19.6618
wicked;	5.0428	underclass;	19.4798
todo;	5.0428	film;	19.3725
abram;	5.02785	abram;	19.1794
stallon;	5.00717	democrat;	18.7775
salaryman;	4.9984	corang;	18.5761
csurka;	4.99443	siegmund;	18.2318
bradman;	4.98968	csurka;	18.138
xerx;	4.97241	blasfemi;	18.1279
sulli;	4.96479	democraci;	17.6826
yamahana;	4.95533	abduct;	17.4898
ajax;	4.95533	mortier;	17.4347
bischof;	4.93846	bjp;	17.2956
lehel;	4.91351	aspen;	17.2692
ardzinba;	4.91351	bradman;	17.2637
bitc;	4.90894	opus;	16.8878
armscor;	4.90529	illeg;	16.6208
mortier;	4.88752	skier;	16.4487
lrt;	4.87287	citisenship;	16.3073
haider;	4.87287	sulli;	16.2279
underclass;	4.86147	@0195;	16.1748
camacho;	4.85211	kelman;	16.1051
glaswegian;	4.84116	@0330;	15.9246
tasso;	4.83334	bitc;	15.896
d'amato;	4.80576	republican;	15.8795
sitter;	4.79487	todo;	15.7464

Table A.4. Top 50 MI and Z ranked collocates of the synonym group @0104 [*environment, environmental*] (topic 255)

MI list		Z list	
tribal;	5.98753	wast;	152.124
gef;	4.34336	pollution;	150.426
ecolabel;	4.2203	emiss;	146.35
lalond;	4.1759	recycl;	131.157
voc;	4.15081	energi;	101.456
meana;	4.1442	carbon;	100.399
topfer;	4.13526	water;	95.4962
cpre;	4.13408	pollut;	93.3731
ripa;	4.11802	dioxid;	92.2648
dsd;	4.03618	gummer;	86.2703
unep;	4.01085	forest;	86.1083
deforest;	4.00715	greenpeac;	84.2011
lrb;	3.98326	@0064;	79.4353
paleokrassa;	3.97309	nuclear;	77.4004
chisso;	3.97237	rio;	76.2514
tcf;	3.96018	fuel;	73.6179
minamata;	3.95351	green;	73.2138
benzen;	3.93946	earth;	70.7988
groundwat;	3.92593	ozon;	68.4786
greenpeac;	3.92232	develop;	64.3111
tace;	3.91626	landfil;	61.2398
superfund;	3.91604	ecolog;	60.886
gorleben;	3.91287	toxic;	60.463
scrubber;	3.8896	nafta;	60.3932
narmada;	3.8896	contamin;	60.3529
biodivers;	3.88293	thorp;	59.0407
baldri;	3.8701	@0180;	58.5672
nox;	3.8647	sulfur;	58.0801
chlorofluorocarbon;	3.84874	inciner;	56.9197
oxlea;	3.83487	cfc;	56.4844
effluent;	3.83322	local;	55.382
wallach;	3.80667	meana;	53.2251
hsct;	3.80667	council;	52.5427
emiss;	3.77561	reprocess;	51.9274
cfc;	3.77151	plant;	51.5546
inciner;	3.76346	ripa;	51.3526
dioxid;	3.75743	timber;	51.3039
mto;	3.75533	project;	50.7709
audubon;	3.74706	whale;	50.4413
greenest;	3.74527	greenhous;	50.2784
nitrous;	3.73533	site;	50.0514
maxus;	3.73533	@0380;	49.9454
pollut;	3.73246	depart;	49.4913
wetland;	3.72984	radioact;	47.8375
reforest;	3.72524	bnf;	47.3958
landfil;	3.71726	urban;	47.2251
ozon;	3.71024	area;	46.8256
chlorin;	3.70599	speci;	46.8011

Table A.5. Top 50 MI and Z ranked collocates of the stem *protect* (topic 255)

MI list		Z list	
reportedli;	8.72129	@0104;	96.4111
conservatori;	8.72129	regul;	78.4392
cpre;	4.30021	law;	70.6222
unprofor;	3.8	tariff;	70.396
radon;	3.75964	patent;	68.9456
tupe;	3.75551	bankruptci;	61.6478
riina;	3.71524	creditor;	60.1808
zale;	3.68925	un;	59.8219
kws;	3.67217	@0135;	56.6553
mto;	3.61215	@0099;	54.0938
westar;	3.58651	oy;	52.9361
timeshar;	3.57874	pia;	52.8811
switchov;	3.54532	copyright;	51.8677
baranja;	3.48307	tupe;	51.1987
leakei;	3.45968	@0035;	48.9821
spc;	3.45775	privaci;	45.1032
abta;	3.45775	legisl;	43.3575
afta;	3.43347	serb;	43.1304
armscor;	3.42574	oys;	42.3228
assigne;	3.3975	file;	41.578
pera;	3.38767	troop;	41.3169
demurrag;	3.38267	court;	41.0003
novofarm;	3.37821	banana;	40.2401
maci;	3.36161	enforc;	40.227
nikkatsu;	3.34601	sib;	39.8024
pbgc;	3.32822	maci;	39.6025
carena;	3.2529	pension;	39.2045
oflif;	3.23608	insur;	39.0993
gmp;	3.23608	rule;	38.9597
vondran;	3.17822	chapter;	38.1566
oys;	3.17418	pollution;	37.8531
pia;	3.15654	forc;	37.8506
transgen;	3.1482	depositor;	37.7729
maxus;	3.13974	right;	37.3493
nuswift;	3.13331	unprofor;	37.1883
curragh;	3.13028	abta;	36.9949
morillon;	3.12388	timeshar;	36.9545
biodivers;	3.10867	@0399;	36.6657
amf;	3.10756	act;	36.2749
privaci;	3.10725	quota;	35.3157
whistleblow;	3.06317	restrict;	35.2641
borsellino;	3.06317	twa;	35.0949
azucarera;	3.06317	under;	34.9311
fsa;	3.05228	legal;	34.8502
walbrook;	3.04495	import;	34.0521
nostra;	3.04495	liabil;	33.4405
audiovisu;	3.03997	countri;	33.4311
muramoto;	3.03717	safeguard;	33.0968

Table A.6. Top 50 MI and Z ranked collocates of the stem *theory* (topic 259)

MI list		Z list	
quark;	6.02906	quark;	52.7823
cosmolog;	6.0102	particl;	50.4976
cobe;	6.00042	scienc;	46.561
endogen;	5.91709	cosmolog;	44.7105
o'higgin;	5.80462	o'higgin;	44.6625
deme;	5.72445	cobe;	41.6769
krugman;	5.29063	galaxi;	39.6123
selborn;	5.20987	@0233;	39.0239
popper;	5.1279	endogen;	38.2238
psychoanalysis;	5.05856	practic;	35.7352
penros;	5.00659	deme;	35.6691
neanderth;	4.98748	@0295;	34.2128
aburto;	4.98748	krugman;	33.9437
galaxi;	4.98069	theorist;	33.1061
baudelair;	4.94309	einstein;	32.0969
watl;	4.90002	popper;	31.9819
neutron;	4.90002	@0129;	31.6014
einstein;	4.89773	cosmic;	31.1993
baldrig;	4.88795	univers;	29.0646
cosmic;	4.8582	academ;	27.5001
particl;	4.76611	@0199;	27.4946
cjd;	4.72445	fysic;	27.2776
theorist;	4.55138	theoret;	27.2119
sperm;	4.52805	hypothesis;	26.8816
ormerod;	4.52805	conspiraci;	26.7435
hypothesis;	4.49838	econom;	26.7365
massieu;	4.47178	astronom;	26.066
meffsa;	4.44832	book;	25.2489
hesss;	4.44832	scientist;	24.9229
wittgenstein;	4.40252	galileo;	24.6309
hypothes;	4.40252	penros;	23.9452
electromagnet;	4.38146	idea;	23.3809
kung;	4.37277	professor;	23.3789
samuelson;	4.34362	scientif;	23.2546
galileo;	4.23259	differ;	22.6054
kingdon;	4.19306	bang;	22.1813
coulter;	4.19306	selborn;	22.1479
edelman;	4.16151	baudelair;	21.4643
coppola;	4.15459	@0232;	21.4007
abstrus;	4.15459	@0322;	21.3965
astronom;	4.1493	human;	21.3798
theoris;	4.11712	neanderth;	21.1264
hazlitt;	4.11301	aburto;	21.1264
mise;	4.10135	argum;	21.1072
hrm;	4.08059	model;	20.2323
@0129;	4.05535	psychoanalysis;	20.1897
nader;	4.05202	brain;	19.9105
hopkirk;	4.03995	baldrig;	19.6721

Table A.7. Top 50 MI and Z ranked collocates of the stem *scuba* (topics 266, 295)

MI list		Z list	
amwai;	10.275	dive;	142.061
antigua;	9.4152	amwai;	99.481
allrisk;	9.35749	swatch;	76.2636
whaler;	9.27502	diver;	64.5278
chronograf;	9.27502	antigua;	63.9099
diver;	8.85999	whaler;	55.5666
painkil;	8.82046	coral;	53.2962
calypso;	8.82046	saul;	51.6194
swatch;	8.70541	allrisk;	51.1458
dive;	8.63557	chronograf;	49.7003
saul;	8.57988	reef;	46.644
lam;	8.43	lam;	45.3582
veranda;	8.35749	shark;	43.6185
jalousi;	8.31567	isabel;	42.5771
caico;	8.31567	versac;	37.8431
coral;	8.31109	reebok;	36.9754
bermudian;	8.30199	beach;	36.7623
isabel;	8.24855	painkil;	36.746
aerob;	8.24855	calypso;	36.746
spratli;	8.23549	spratli;	34.6061
whitelei;	8.15955	bermudian;	30.673
belisean;	8.15955	dodd;	30.5138
trapez;	8.0874	aerob;	30.1065
atol;	8.05263	smh;	29.87
versac;	7.91097	f;	29.5871
shark;	7.90579	caribbean;	27.9983
dodd;	7.87509	twobedroom;	27.2342
sunken;	7.85999	torist;	25.5711
duckworth;	7.74451	veranda;	25.5338
reef;	7.64229	jalousi;	25.1641
accra;	7.61206	caico;	25.1641
waterproof;	7.55418	island;	24.813
twobedroom;	7.55013	whitelei;	23.8302
duisburg;	7.53806	belisean;	23.8302
sybarit;	7.49075	waterproof;	23.6189
alvin;	7.49075	trapez;	23.2375
@0049;	7.46006	bahama;	23.0805
quiszic;	7.44495	atol;	22.9571
menial;	7.44495	@0049;	22.8527
gnomic;	7.44495	asean;	21.8313
farawai;	7.44495	sunken;	21.4628
reebok;	7.43372	belis;	20.9005
flamenco;	7.42258	duckworth;	20.6134
wisp;	7.40055	brunei;	19.7113
timepiec;	7.40055	accra;	19.6797
parenthood;	7.40055	zoom;	19.6486
zoolog;	7.35749	underwat;	19.6486
unborn;	7.35749	travel;	19.2732

Table A.8. Top 50 MI and Z ranked collocates of the stem *nation* (topic 268)

MI list		Z list	
yly;	2.79892	united;	122.062
amelio;	2.63746	parti;	106.8
jagan;	2.51094	un;	91.3438
united;	2.47808	anc;	85.4333
seselj;	2.47699	@0147;	80.7355
mmp;	2.47699	lotteri;	78.6053
macavoi;	2.46319	@0035;	77.7356
csurka;	2.44493	countri;	73.501
andri;	2.4326	elect;	73.2048
ramafosa;	2.43232	union;	68.346
vojislav;	2.41426	polit;	66.1742
nsf;	2.41426	state;	66.0428
rnt;	2.40299	presid;	65.0699
rdp;	2.40299	leader;	63.3824
pairin;	2.39899	abbei;	62.708
slovo;	2.38507	power;	61.7502
polsat;	2.38388	serb;	61.5737
potchefstroom;	2.35146	constitut;	58.3216
nvq;	2.35019	na;	57.3725
mbeki;	2.34489	vote;	56.7183
umkhonto;	2.34373	democrat;	56.4171
snb;	2.33015	mandela;	55.8808
scowcroft;	2.32889	mr;	54.2331
gnvq;	2.32499	klerk;	53.7606
fini;	2.318	council;	53.2645
drax;	2.31653	@0339;	53.053
pujol;	2.31571	minist;	52.5775
nasuwt;	2.29539	heritag;	52.2268
ciu;	2.29328	@0099;	52.1554
shaka;	2.29257	social;	51.7819
hendron;	2.28435	westminst;	51.7588
gruchi;	2.27845	commun;	51.6451
atl;	2.27418	labor;	51.4491
nhl;	2.27298	educ;	50.9693
pnb;	2.26678	@0080;	50.6064
ifad;	2.26126	@0431;	50.1465
zwelithini;	2.2546	militari;	49.2504
npc;	2.25143	member;	49.1629
bossis;	2.25072	parliam;	48.8495
codesa;	2.24982	peac;	47.5448
multiraci;	2.24332	inkatha;	47.4529
energis;	2.24069	region;	47.2701
pnv;	2.23598	@0352;	46.6831
sillar;	2.22907	teacher;	46.6069
relf;	2.22907	secretari;	45.2715
nssr;	2.22324	powergen;	45.2109
lotteri;	2.21922	war;	44.2698
rha;	2.21396	boutro;	44.108

Table A.9. Top 50 MI and Z ranked collocates of the stem *control* (topic 270)

MI list		Z list	
unchart;	3.81437	stake;	73.5086
hector;	3.81437	state;	61.0369
alassad;	3.81437	sharehold;	59.0246
yurko;	3.01126	serb;	57.3605
edper;	2.90108	mr;	54.2852
nazmu;	2.89879	system;	53.0294
gaic;	2.87696	@0077;	48.0472
dno;	2.86316	group;	46.209
misrahi;	2.85926	@0147;	45.6286
tace;	2.83069	privat;	45.5012
motlana;	2.83069	@0035;	43.2826
yentob;	2.82282	@0068;	41.8634
quek;	2.81195	perrier;	40.0075
kolodziejczyk;	2.81195	bank;	38.8434
maserati;	2.80481	croat;	37.4617
cnc;	2.78114	hold;	37.4154
tajudin;	2.77993	@0339;	36.9634
deme;	2.7768	under;	36.9152
micromachin;	2.76615	bronfman;	36.466
cemig;	2.76128	own;	36.3585
vvl;	2.7575	manag;	35.7183
ashman;	2.75492	author;	35.4028
pargesa;	2.74823	bid;	34.1119
elsag;	2.74823	@0190;	34.0719
schengen;	2.74432	iri;	33.4942
cocom;	2.70585	power;	33.0333
bronfman;	2.69946	@0330;	32.7875
bramalea;	2.67765	uap;	32.6033
barilla;	2.67246	ownership;	32.4002
bnc;	2.67022	@0373;	31.3897
italtel;	2.65896	agnellis;	30.99
exor;	2.65851	exor;	30.8638
innocenti;	2.65617	forc;	30.6076
agnellis;	2.65241	cdollar;	30.1264
colonia;	2.65237	hachett;	29.6174
foxboro;	2.65172	reform;	29.5266
virani;	2.65016	ferruzzi;	29.4452
gardinis;	2.64353	central;	29.242
saffa;	2.64089	famili;	29.2121
ifint;	2.63575	finmeccanica;	28.4624
gge;	2.63275	ciga;	28.2497
gmf;	2.6266	skoda;	28.1374
slavonia;	2.62424	lyonnais;	28.0632
cantv;	2.62424	@0121;	27.9264
desmarais;	2.62047	local;	27.8885
kelantan;	2.61698	regul;	27.4602
suntim;	2.59622	chebol;	27.3881
pbc;	2.59622	@0060;	27.384

Table A.10. Top 50 MI and Z ranked collocates of the stem *solar* (topic 271)

MI list		Z list	
seti;	8.71912	energi;	111.796
ulyss;	8.58816	ulyss;	80.6755
biomass;	8.41798	biomass;	80.378
geotherm;	8.19931	@0256;	68.4866
twh;	8.13005	seti;	67.924
hubbl;	7.9236	geotherm;	59.1876
spacecraft;	7.84465	hubbl;	55.95
geostationari;	7.77955	orbit;	48.9847
@0111;	7.77955	leo;	43.8943
audubon;	7.73773	renew;	43.6682
maya;	7.57108	fossil;	42.5949
geo;	7.54509	galaxi;	41.8607
lunar;	7.4698	kwh;	41.2001
kwh;	7.42407	twh;	40.8524
antipolis;	7.17585	heat;	40.8404
nino;	7.16262	earth;	39.3064
constel;	7.15277	spacecraft;	36.976
milki;	7.1391	planet;	36.4552
wenger;	7.13005	maya;	36.2928
equinox;	7.13005	lunar;	35.0275
@0256;	7.12873	constel;	33.5028
deg;	7.11213	sun;	33.1233
galaxi;	7.09578	ozon;	33.0823
nonpollut;	7.00077	galileo;	33.0771
munic;	6.88212	geostationari;	32.9946
eze;	6.73773	@0111;	32.9946
coppic;	6.68711	wind;	32.9339
grimshaw;	6.67062	audubon;	32.5155
galileo;	6.66467	astronom;	32.0261
nonfossil;	6.60649	iridium;	31.9866
nozzl;	6.59089	nino;	31.4477
castillo;	6.59089	sunlight;	31.4343
iridium;	6.56989	pavilion;	30.6845
aquascutum;	6.5602	geo;	30.395
pye;	6.54509	cfc;	30.0678
mbb;	6.54509	telescop;	30.0401
harwel;	6.54509	space;	29.2193
telescop;	6.52717	radiat;	28.4929
observatori;	6.52272	electr;	28.1797
astrolog;	6.52272	odyssei;	27.9926
sunlight;	6.5207	power;	27.6715
orbit;	6.50556	dioxid;	27.5746
@0404;	6.50069	satellit;	27.2951
leo;	6.48488	nuclear;	26.6888
odyssei;	6.47661	@0404;	26.6179
fossil;	6.46573	milki;	26.3593
astronom;	6.45057	deg;	26.1106
pane;	6.42961	cult;	26.0892

Table A.11. Top 50 MI and Z ranked collocates of the stem *power* (topic 271)

MI list		Z list	
troup;	6.40752	electr;	159.753
gigawatt;	3.22857	nuclear;	142.951
igcc;	3.22338	parti;	102.087
kwu;	3.11701	coal;	100.753
gascool;	3.10558	station;	95.4629
schkopau;	3.10053	polit;	83.2746
kozlodui;	3.09612	powergen;	81.8653
tepco;	3.09277	energi;	79.8404
cepa;	3.08857	constitut;	75.0287
sisewel;	3.08174	turbin;	74.1796
pwr;	3.0681	@0147;	71.0559
napocor;	3.0681	parliam;	69.0011
keadbi;	3.0681	reactor;	68.6938
megawatt;	3.04257	elect;	67.5711
egat;	3.03568	@0330;	60.2462
paiton;	3.02504	yeltsin;	58.6367
rokkasho;	3.0208	presid;	58.6345
@0131;	3.01433	@0134;	58.4815
@0067;	3.01261	reform;	56.2424
huaneng;	3.01152	gener;	54.9095
gasfir;	3.00934	nation;	54.7865
scrubber;	3.00921	anc;	54.6106
ccgt;	3.00798	littlechild;	54.0736
orimuls;	3.0023	gasfir;	53.1272
magnox;	2.9908	mr;	52.829
drax;	2.9901	ldp;	52.2448
kilowatt;	2.98642	fuel;	51.3928
fgd;	2.93732	democrat;	51.1308
ipp;	2.93686	abb;	48.6415
chubu;	2.92609	util;	48.4406
oilfir;	2.92126	hydro;	48.0301
connah;	2.91283	democraci;	47.3177
pln;	2.88524	state;	47.0587
gasif;	2.88524	coalition;	47.0171
egco;	2.88169	opposit;	45.8643
rbmk;	2.87907	leader;	45.8563
kwh;	2.87309	sisewel;	45.8306
preussenelektra;	2.86299	social;	44.2793
powergen;	2.86227	minist;	42.8155
flue;	2.84304	militari;	42.636
sydkraft;	2.83984	politician;	39.4591
cegb;	2.83678	engin;	39.2409
turbin;	2.83295	vote;	39.1295
desmarais;	2.82932	plant;	38.9279
codecis;	2.7968	diesel;	38.9168
sevan;	2.788	batteri;	38.6669
omo;	2.78517	ukrain;	38.3858
desulfur;	2.7786	grid;	38.0522

Table A.12. Top 50 MI and Z ranked collocates of the stem *volcan* (topic 273)

MI list		Z list	
lava;	9.34627	volcano;	159.008
volcano;	9.17149	crater;	119.379
crater;	8.99712	lava;	111.035
jalousi;	8.80195	erupt;	100.785
geotherm;	8.17042	island;	75.4608
westerli;	7.64583	jalousi;	59.6206
bariloch;	7.64583	geotherm;	50.7437
hierro;	7.6093	ozon;	48.9959
tenerif;	7.58073	ski;	41.7053
muirfield;	7.4718	palma;	37.3397
cove;	7.4348	cone;	36.9855
ashland;	7.38691	mountain;	36.8281
tortois;	7.34627	cove;	34.6004
waffl;	7.30674	torist;	34.2705
palma;	7.2938	tortois;	31.0543
erupt;	7.22117	tenerif;	30.7769
fuego;	7.19426	subic;	30.399
soleil;	7.08993	mile;	30.1965
rancher;	7.08993	ash;	30.0796
subic;	7.06616	helena;	29.592
skog;	7.04371	diesel;	28.2131
mountainsid;	7.00049	westerli;	28.1623
cone;	6.98127	bariloch;	28.1623
safaris;	6.97703	hierro;	27.8044
overspil;	6.97703	resort;	27.7156
iguana;	6.97703	@0139;	27.3599
gran;	6.96776	mud;	26.6394
sluic;	6.93123	lake;	26.1858
flore;	6.93123	waugh;	25.5053
ditto;	6.88684	skog;	25.4897
kinship;	6.84377	gran;	24.8172
headmistress;	6.84377	bachelor;	23.8541
antipodean;	6.84377	snow;	23.135
stratosfer;	6.80195	muirfield;	22.9469
southerli;	6.80195	mountainsid;	22.4545
millgat;	6.80195	ashland;	22.274
inferno;	6.76131	safaris;	22.2698
frigid;	6.76131	las;	22.128
conquistador;	6.76131	earthquak;	21.97
stanton;	6.70884	flore;	21.9134
gorilla;	6.6833	waffl;	21.6562
snowboard;	6.65821	filippin;	21.641
torqu;	6.63355	ditto;	21.5733
helena;	6.63355	antarct;	21.0843
spew;	6.6093	archipelago;	21.0252
fawn;	6.6093	stratosfer;	20.937
bachelor;	6.59733	fuego;	20.8175
waugh;	6.56866	debris;	20.4137

Table A.13. Top 50 MI and Z ranked collocates of the synonym group @0375 [*tax, taxation, taxable, taxability*] (topic 291)

MI list		Z list	
privi;	8.80379	incom;	179.093
yly;	3.53112	budget;	174.605
iht;	3.36403	revenu;	139.268
shochu;	3.34349	vat;	120.849
prt;	3.32169	spend;	109.071
gst;	3.31964	relief;	108.797
furb;	3.31178	@0290;	104.876
youearn;	3.30873	pep;	104.472
settlor;	3.30042	inland;	99.4814
chargeabl;	3.29604	chancellor;	91.9547
fid;	3.27454	deficit;	90.992
vct;	3.26406	increas;	89.4839
taxplan;	3.26103	taxpay;	89.0197
lowtax;	3.24833	fiscal;	88.2811
nontaxpay;	3.2316	cgt;	84.0386
cgt;	3.22509	pound;	79.7894
misdeclar;	3.21796	labor;	78.4059
ezt;	3.20856	pension;	78.023
untax;	3.19149	rate;	77.9838
labuan;	3.18122	exempt;	77.5642
taxman;	3.17357	adollar;	77.5164
nontax;	3.17357	benefit;	76.0686
allenbridg;	3.17016	bes;	74.9358
lpi;	3.15085	@0147;	72.511
bes;	3.11268	year;	70.5511
thorsen;	3.10645	invest;	70.4605
evasion;	3.09609	profit;	70.3891
darman;	3.075	taxfre;	69.3341
allcock;	3.06477	cut;	66.9847
taxdeduct;	3.05719	deduct;	66.086
hypothec;	3.05071	earn;	65.2715
taxexempt;	3.02041	charg;	65.1868
mavrodi;	3.01092	dividend;	64.4897
garnham;	3.00646	pay;	64.0982
ifsc;	3.00532	scheme;	63.5085
taxeffici;	2.98883	tax;	63.2447
agerel;	2.96162	@0314;	61.5526
avc;	2.95975	inflat;	61.5507
nic;	2.95392	lamont;	60.2571
taxfre;	2.94026	save;	60.0236
stillerman;	2.92166	allow;	59.1358
mortgagor;	2.91286	excis;	58.5601
taxcut;	2.88538	@0069;	58.4195
ucit;	2.88406	@0136;	58.3346
ifs;	2.88286	incent;	58.3129
afgh;	2.84873	gain;	57.1742
baronworth;	2.84099	higher;	56.5863
boren;	2.83653	measur;	54.3491

Table A.14. Top 50 MI and Z ranked collocates of the stem *air* (topic 300)

MI list		Z list	
biggest;	9.02791	@0008;	299.791
jeanniot;	4.73969	@0003;	176.804
bueno;	4.64085	carrier;	159.944
iata;	4.64025	bueno;	155.436
bosson;	4.61351	flight;	136.248
lauda;	4.59669	@0009;	129.084
orli;	4.56576	@0020;	114.725
airto;	4.44646	@0035;	112.721
aerolinea;	4.4415	passeng;	105.327
caac;	4.42982	travel;	102.52
metroga;	4.42775	serb;	102.252
sabena;	4.39419	aviat;	99.8414
tella;	4.38929	traffic;	97.6377
ffp;	4.30575	@0257;	90.0618
jas;	4.27308	ba;	88.2677
helium;	4.2702	un;	83.9231
pwa;	4.17991	airwai;	80.8904
atc;	4.15473	menem;	78.8667
meridien;	4.15165	argentin;	78.8446
peron;	4.15149	orli;	77.2939
fuego;	4.14265	lufthansa;	75.6225
aeroflot;	4.13675	boe;	74.499
merval;	4.12671	militari;	70.8615
radon;	4.11666	fly;	70.6863
gripen;	4.10974	jet;	69.9276
mig;	4.07968	rout;	69.2206
tierra;	4.07226	sabena;	69.056
ypf;	4.06737	heathrow;	65.9443
airfreight;	4.05999	sarajevo;	65.5815
rafal;	4.05859	fare;	65.259
farnborough;	4.03832	airbus;	62.419
alfonsn;	4.02336	iata;	62.3157
tupolev;	4.01189	fighter;	61.486
bisignani;	4.00012	pollution;	61.1125
igman;	3.99826	forc;	59.6625
letterbox;	3.97916	transport;	58.4025
menem;	3.94743	pwa;	57.4522
changi;	3.939	@0186;	56.6664
lnr;	3.92026	strike;	56.6491
airspac;	3.92026	meridien;	55.2733
csa;	3.91379	cavallo;	54.1811
juven;	3.90515	ypf;	53.9037
voc;	3.88671	missil;	52.3627
rudder;	3.88464	@0128;	52.1782
advanta;	3.86873	emiss;	51.4319
sofitel;	3.86076	@0114;	50.7927
plasser;	3.85753	runwai;	49.329
malev;	3.85467	usair;	49.1948

Table A.15. Top 50 MI and Z ranked collocates of the stem *traffic* (topic 300)

MI list		Z list	
trafficmast;	5.88823	road;	165.593
jeanniot;	5.84161	passeng;	160.179
becl;	5.69913	@0009;	160.052
iata;	5.61903	@0008;	160.007
hartsfield;	5.46341	congest;	138.914
atc;	5.44306	transport;	117.282
changi;	5.30556	air;	115.221
prometheus;	5.27591	motorwai;	113.523
intermod;	5.25167	freight;	92.9153
freiburg;	5.25006	rout;	91.7011
congest;	5.20184	carrier;	91.6078
incar;	5.17419	jam;	91.5919
worldcom;	5.12999	rail;	85.5748
bosporus;	5.12113	toll;	81.046
flyover;	5.11416	network;	77.5176
paramed;	5.02379	heathrow;	77.012
teleglob;	4.97677	iata;	76.7848
jam;	4.8901	baa;	73.8284
bisignani;	4.88347	citi;	70.0296
holyhead;	4.8487	driver;	67.2903
skytrain;	4.79099	flight;	67.0757
larn;	4.7873	@0319;	65.7958
mitt;	4.77831	@0003;	64.0029
seacat;	4.76853	runwai;	63.5741
freewai;	4.74917	aviat;	63.5286
mfs;	4.72317	transit;	62.0334
tanayong;	4.7206	port;	61.312
voltri;	4.70186	termin;	60.5298
gotthard;	4.68777	highwai;	59.2212
huangpu;	4.68407	travel;	58.8958
narita;	4.66301	lorri;	58.3201
expresswai;	4.65606	hub;	57.0949
carriagewai;	4.62939	tunnel;	56.7785
roadwork;	4.62631	cargo;	54.3405
warden;	4.60715	distenc;	54.1866
telemat;	4.60161	fare;	50.4549
farnsworth;	4.57633	servic;	50.3135
peek;	4.57152	telecom;	50.1608
vpn;	4.57102	speed;	49.8485
trucker;	4.56686	bridg;	48.9271
teus;	4.55172	atc;	47.7909
streetcar;	4.53053	gatwick;	47.5496
transhipl;	4.52361	ba;	47.0051
filton;	4.52361	bangkok;	46.988
runwai;	4.5035	trafficmast;	46.0228
autobahn;	4.49852	ferri;	45.406
asynchron;	4.48613	@0045;	44.8337
airspac;	4.48613	@0296;	44.4583

A.2 Trec_eval performance results of global runs

Table A.2.16. Performance results of global runs

Retrieved: ≈44000
Relevant: 1583

Measure	OK UNEXPANDED	TOP 8 MI	TOP 16 MI	TOP 8 Z	TOP 16 Z
Relevant retrieved	632	573	526	520	504
Interpolated recall – precision averages					
at 0.00	0.3992	0.1385	0.1197	0.1828	0.1430
at 0.10	0.3013	0.1022	0.0820	0.0999	0.0873
at 0.20	0.2049	0.0759	0.0618	0.0763	0.0643
Average precision	0.1310	0.0432	0.0344	0.0375	0.0340
Precision:					
At 5 docs	0.2136	0.0682	0.0636	0.0682	0.0773
At 10 docs	0.1523	0.0568	0.0477	0.0636	0.0614
At 15 docs	0.1394	0.0561	0.0515	0.0606	0.0530
At 20 docs	0.1295	0.0591	0.0477	0.0557	0.0523
R-Precision	0.1497	0.0437	0.0355	0.0422	0.0338

Measure	PREDICTED OW	PREDICTED RW
Relevant retrieved	554	617
Interpolated recall – precision averages		
at 0.00	0.1669	0.0798
at 0.10	0.0899	0.0652
at 0.20	0.0741	0.0500
Average precision	0.0364	0.0296
Precision:		
At 5 docs	0.0682	0.0318
At 10 docs	0.0523	0.0250
At 15 docs	0.0515	0.0273
At 20 docs	0.0523	0.0227
R-Precision	0.0419	0.0269

A.3 Regression analysis results

Tables

Table A.3.1. Regression (all significant collocates);
dependent variable: OW; independent variables: Z, MI, JF, NOPOS

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.211(a)	.045	.045	7.9502

a Predictors: (Constant), NOPOS, Z, MI, JF

ANOVA(b)

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	1915569.482	4	478892.370	7576.645	.000(a)
	Residual	41011024.834	648843	63.206		
	Total	42926594.316	648847			

a Predictors: (Constant), NOPOS, Z, MI, JF

b Dependent Variable: OW

Coefficients(a)

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	.181	.018		10.153	.000
	MI	-.309	.010	-.041	-30.805	.000
	Z	.152	.001	.158	115.846	.000
	JF	1.787E-04	.000	.041	26.518	.000
	NOPOS	1.090E-04	.000	.077	49.193	.000

a Dependent Variable: OW

Table A.3.2 Regression (all significant collocates);
dependent variable: OW; independent variables: Z, JF, NOPOS

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.208(a)	.043	.043	7.9847

a Predictors: (Constant), NOPOS, Z, JF

ANOVA(b)

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	1850071.218	3	616690.406	9672.656	.000(a)
	Residual	41061771.757	644045	63.756		
	Total	42911842.974	644048			

a Predictors: (Constant), NOPOS, Z, JF

b Dependent Variable: OW

Coefficients(a)

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	-.217	.012		-17.526	.000
	Z	.137	.001	.142	111.734	.000
	JF	1.937E-04	.000	.045	28.677	.000
	NOPOS	1.247E-04	.000	.087	57.507	.000

a Dependent Variable: OW

Table A.3.3. Regression (all significant collocates);
dependent variable: OW; independent variables: Z, JF

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.195(a)	.038	.038	8.0052
a Predictors: (Constant), JF, Z				

ANOVA(b)

Model		Sum of Squares	Df	Mean Square	F	Sig.
1	Regression	1639225.417	2	819612.709	12789.794	.000(a)
	Residual	41272617.557	644046	64.083		
	Total	42911842.974	644048			
a Predictors: (Constant), JF, Z						
b Dependent Variable: OW						

Coefficients(a)

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	-7.814E-02	.012		-6.431	.000
	Z	.139	.001	.144	113.071	.000
	JF	4.177E-04	.000	.096	75.501	.000
a Dependent Variable: OW						

Table A.3.4. Regression (all significant collocates);
dependent variable: OW; independent variables: Z, NOPOS

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.205(a)	.042	.042	7.9629

a Predictors: (Constant), NOPOS, Z

ANOVA(b)

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	1799377.958	2	899688.979	14188.939	.000(a)
	Residual	41119495.396	648493	63.408		
	Total	42918873.354	648495			

a Predictors: (Constant), NOPOS, Z

b Dependent Variable: OW

Coefficients(a)

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	-.269	.012		-22.182	.000
	Z	.145	.001	.150	121.423	.000
	NOPOS	1.605E-04	.000	.113	90.930	.000

a Dependent Variable: OW

Table A.3.5. Regression (all significant collocates); dependent variable: OW; independent variables: Z, MI

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.187(a)	.035	.035	7.9896
a Predictors: (Constant), Z, MI				

ANOVA(b)

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	1508723.124	2	754361.562	11817.694	.000(a)
	Residual	41417871.191	648845	63.833		
	Total	42926594.316	648847			
a Predictors: (Constant), Z, MI						
b Dependent Variable: OW						

Coefficients(a)

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	.595	.017		35.148	.000
	MI	-.573	.010	-.077	-60.180	.000
	Z	.188	.001	.195	152.923	.000
a Dependent Variable: OW						

Table A.3.6. Regression (all significant collocates); dependent variable: OW; independent variables: Z

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.172(a)	.030	.030	8.0135
a Predictors: (Constant), Z				

ANOVA(b)

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	1275105.012	1	1275105.012	19856.463	.000(a)
	Residual	41643768.342	648494	64.216		
	Total	42918873.354	648495			
a Predictors: (Constant), Z						
b Dependent Variable: OW						

Coefficients(a)

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	-.119	.012		-9.855	.000
	Z	.166	.001	.172	140.913	.000
a Dependent Variable: OW						

Table A.3.7. Regression (all significant collocates);
dependent variable: RW; independent variables: MI, Z, JF, NOPOS, QT

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.538(a)	.289	.289	1.7073

a Predictors: (Constant), QT, JF, MI, Z, NOPOS

ANOVA(b)

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	769733.914	5	153946.783	52816.267	.000(a)
	Residual	1891627.256	648982	2.915		
	Total	2661361.170	648987			

a Predictors: (Constant), QT, JF, MI, Z, NOPOS

b Dependent Variable: RW

Coefficients(a)

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	3.401	.004		887.960	.000
	MI	.692	.002	.373	321.302	.000
	Z	-4.860E-02	.000	-.203	-172.724	.000
	JF	1.031E-04	.000	.096	71.239	.000
	NOPOS	-1.202E-04	.000	-.340	-253.101	.000
	QT	2.436	.144	.018	16.867	.000

a Dependent Variable: RW

Table A.3.8. Regression (all significant collocates);
dependent variable: RW; independent variables: Z, JF, NOPOS, QT

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.420(a)	.176	.176	1.8380

a Predictors: (Constant), QT, JF, Z, NOPOS

ANOVA(b)

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	468828.782	4	117207.195	34692.978	.000(a)
	Residual	2192532.388	648983	3.378		
	Total	2661361.170	648987			

a Predictors: (Constant), QT, JF, Z, NOPOS

b Dependent Variable: RW

Coefficients(a)

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	4.295	.003		1515.533	.000
	Z	-1.552E-02	.000	-.065	-55.065	.000
	JF	6.993E-05	.000	.065	45.015	.000
	NOPOS	-1.556E-04	.000	-.440	-312.823	.000
	QT	2.021	.155	.015	12.996	.000

a Dependent Variable: RW

Table A.3.9. Regression (all significant collocates);
dependent variable: RW; independent variables: Z, NOPOS, QT

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.417(a)	.174	.174	1.8409

a Predictors: (Constant), QT, Z, NOPOS

ANOVA(b)

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	461982.960	3	153994.320	45440.047	.000(a)
	Residual	2199378.210	648984	3.389		
	Total	2661361.170	648987			

a Predictors: (Constant), QT, Z, NOPOS

b Dependent Variable: RW

Coefficients(a)

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	4.276	.003		1523.020	.000
	Z	-1.272E-02	.000	-.053	-46.190	.000
	NOPOS	-1.427E-04	.000	-.403	-350.364	.000
	QT	1.865	.156	.014	11.980	.000

a Dependent Variable: RW

Table A.3.10. Regression (all significant collocates); dependent variable: RW; independent variables: Z, JF, QT

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.228(a)	.052	.052	1.9718

a Predictors: (Constant), JF, QT, Z

ANOVA(b)

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	138223.766	3	46074.589	11850.988	.000(a)
	Residual	2523137.403	648984	3.888		
	Total	2661361.170	648987			

a Predictors: (Constant), JF, QT, Z

b Dependent Variable: RW

Coefficients(a)

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	4.123	.003		1382.450	.000
	Z	-1.796E-02	.000	-.075	-59.419	.000
	QT	8.907E-02	.167	.001	.534	.593
	JF	-2.099E-04	.000	-.195	-154.040	.000

a Dependent Variable: RW

Table A.3.11. Regression (collocates with JF≥30);
dependent variable: OW; independent variables: Z, MI, JF, NOPOS

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.210(a)	.044	.044	11.9083

a Predictors: (Constant), NOPOS, Z, JF, MI

ANOVA(b)

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	1724741.138	4	431185.284	3040.660	.000(a)
	Residual	37552487.718	264815	141.806		
	Total	39277228.856	264819			

a Predictors: (Constant), NOPOS, Z, JF, MI

b Dependent Variable: OW

Coefficients(a)

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	.499	.038		12.990	.000
	MI	-1.162	.039	-.079	-29.718	.000
	Z	.199	.003	.196	73.155	.000
	JF	1.027E-04	.000	.024	9.878	.000
	NOPOS	1.096E-04	.000	.076	31.210	.000

a Dependent Variable: OW

Table A.3.12. Regression (collocates with JF≥30);
dependent variable: OW; independent variables: Z, JF, NOPOS

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.202(a)	.041	.041	11.9281

a Predictors: (Constant), NOPOS, Z, JF

ANOVA(b)

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	1599503.338	3	533167.779	3747.343	.000(a)
	Residual	37677725.518	264816	142.279		
	Total	39277228.856	264819			

a Predictors: (Constant), NOPOS, Z, JF

b Dependent Variable: OW

Coefficients(a)

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	-.204	.030		-6.730	.000
	Z	.144	.002	.142	71.874	.000
	JF	1.624E-04	.000	.039	15.897	.000
	NOPOS	1.340E-04	.000	.092	39.191	.000

a Dependent Variable: OW

Table A.3.13. Regression (collocates with JF≥30); dependent variable: OW; independent variables: Z, JF

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.188(a)	.035	.035	11.9626

a Predictors: (Constant), JF, Z

ANOVA(b)

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	1380971.418	2	690485.709	4825.077	.000(a)
	Residual	37896257.438	264817	143.104		
	Total	39277228.856	264819			

a Predictors: (Constant), JF, Z

b Dependent Variable: OW

Coefficients(a)

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	.170	.029		5.871	.000
	Z	.141	.002	.139	70.225	.000
	JF	3.952E-04	.000	.094	47.389	.000

a Dependent Variable: OW

Table A.3.14. Regression (collocates with JF≥30);
dependent variable: OW; independent variables: Z, NOPOS

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.200(a)	.040	.040	11.9337

a Predictors: (Constant), NOPOS, Z

ANOVA(b)

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	1563549.370	2	781774.685	5489.447	.000(a)
	Residual	37713679.486	264817	142.414		
	Total	39277228.856	264819			

a Predictors: (Constant), NOPOS, Z

b Dependent Variable: OW

Coefficients(a)

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	-.281	.030		-9.367	.000
	Z	.152	.002	.150	77.926	.000
	NOPOS	1.657E-04	.000	.114	59.486	.000

a Dependent Variable: OW

Table A.3.15. Regression (collocates with JF≥30); dependent variable: OW; independent variables: Z

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.164(a)	.027	.027	12.0132
a Predictors: (Constant), Z				

ANOVA(b)

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	1059601.907	1	1059601.907	7342.205	.000(a)
	Residual	38217626.949	264818	144.317		
	Total	39277228.856	264819			
a Predictors: (Constant), Z						
b Dependent Variable: OW						

Coefficients(a)

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	.212	.029		7.315	.000
	Z	.167	.002	.164	85.687	.000
a Dependent Variable: OW						

Table A.3.16. Regression (collocates with JF≥30);
dependent variable: OW; independent variables: Z, MI

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.193(a)	.037	.037	11.9502

a Predictors: (Constant), MI, Z

ANOVA(b)

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	1459498.417	2	729749.208	5110.037	.000(a)
	Residual	37817730.439	264817	142.807		
	Total	39277228.856	264819			

a Predictors: (Constant), MI, Z

b Dependent Variable: OW

Coefficients(a)

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	1.219	.035		35.270	.000
	Z	.245	.002	.241	100.578	.000
	MI	-1.872	.035	-.127	-52.917	.000

a Dependent Variable: OW

Table A.3.17. Regression (collocates with JF≥30);
dependent variable: OW; independent variables: JF, NOPOS

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.148(a)	.022	.022	12.0438

a Predictors: (Constant), JF, NOPOS

ANOVA(b)

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	864503.407	2	432251.703	2979.940	.000(a)
	Residual	38412725.450	264817	145.054		
	Total	39277228.856	264819			

a Predictors: (Constant), JF, NOPOS

b Dependent Variable: OW

Coefficients(a)

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	.987	.026		38.440	.000
	NOPOS	1.247E-04	.000	.086	36.122	.000
	JF	3.386E-04	.000	.080	33.804	.000

a Dependent Variable: OW

Appendix B

Local collocation analysis

B.1 Trec_eval performance results of retrospective MI/Z collocate runs

Tables

Values in all tables:

Retrieved: ≈44000
Relevant: 1583

Table B.1.1. Performance of retrospective Okapi runs

Measure	OK UNEXPANDED	RETRO OK 20	RETRO OK 25	RETRO OK 30	RETRO OK 35
Relevant retrieved	632	936	932	926	941
Interpolated recall – precision averages					
at 0.00	0.3992	0.9545	0.9545	0.9545	0.9545
at 0.10	0.3013	0.8141	0.8271	0.8476	0.8512
at 0.20	0.2049	0.7282	0.7356	0.7519	0.7584
Average precision	0.1310	0.4945	0.5096	0.5184	0.5259
Precision:					
At 5 docs	0.2136	0.6500	0.6909	0.7000	0.7227
At 10 docs	0.1523	0.4500	0.4614	0.4727	0.4841
At 15 docs	0.1394	0.3455	0.3576	0.3636	0.3652
At 20 docs	0.1295	0.2864	0.2932	0.3000	0.3057
R-Precision	0.1497	0.4735	0.4957	0.5053	0.5200

Table B.1.2. Performance of retrospective Z runs

Measure	RETRO 8 GLOBAL Z COL	RETRO 8 GLOBAL Z COL + 20 OK	RETRO 8 Z COL (win 200)	RETRO 8 Z COL (win 100)	RETRO 8 Z COL (win 50)	RETRO 8 Z COL (win 30)	RETRO 8 Z COL (win 20)	RETRO 12 Z COL (win 200)	RETRO 16 Z COL (win 200)	RETRO 21 Z COL (win 200)
Relevant retrieved	851	839	847	858	876	852	847	868	886	924
Interpolated recall – precision averages										
at 0.00	0.9003	0.9545	0.9545	0.9545	0.9545	0.9545	0.9545	0.9545	0.9545	0.9545
at 0.10	0.6951	0.7018	0.7554	0.7556	0.8039	0.8180	0.8100	0.7847	0.8011	0.8326
at 0.20	0.5392	0.5549	0.6851	0.6899	0.6984	0.7207	0.7237	0.7005	0.7215	0.7366
Average precision	0.3220	0.3148	0.4758	0.4720	0.4733	0.4857	0.4810	0.4896	0.5034	0.5029
Precision:										
At 5 docs	0.4045	0.4273	0.6455	0.6182	0.6318	0.6636	0.6545	0.6545	0.6818	0.6773
At 10 docs	0.3136	0.3159	0.4182	0.3932	0.4159	0.4250	0.4386	0.4250	0.4409	0.4432
At 15 docs	0.2561	0.2636	0.3091	0.3030	0.3212	0.3333	0.3394	0.3258	0.3318	0.3455
At 20 docs	0.2273	0.2205	0.2568	0.2523	0.2591	0.2807	0.2841	0.2682	0.2818	0.2841
R-Precision	0.3218	0.3169	0.4722	0.4563	0.4495	0.4691	0.4517	0.4783	0.4949	0.5054

Measure	RETRO 8 Z COL + 20 OK (win 200)	RETRO 8 Z COL + 20 OK (win 100)	RETRO 8 Z COL + 20 OK (win 50)	RETRO 8 Z COL + 20 OK (win 30)	RETRO 8 Z COL + 20 OK (win 20)	RETRO 16 Z COL + 10 OK (win 200)	RETRO 16 Z COL + 20 OK (win 200)	RETRO 16 Z COL + 20 OK (win 100)	RETRO 16 Z COL + 20 OK (win 50)	RETRO 16 Z COL + 20 OK (win 30)	RETRO 16 Z COL + 20 OK (win 20)	RETRO 21 Z COL + 20 OK (win 200)
Relevant retrieved	967	955	952	954	963	979	953	965	966	960	959	972
Interpolated recall – precision averages												
at 0.00	0.9545	0.9545	0.9545	0.9545	0.9545	0.9545	0.9318	0.9545	0.9545	0.9545	0.9545	0.9545
at 0.10	0.8376	0.8315	0.8487	0.8348	0.8283	0.8483	0.8261	0.8483	0.8391	0.8476	0.8470	0.8441
at 0.20	0.7409	0.7462	0.7496	0.7495	0.7397	0.7501	0.7324	0.7596	0.7583	0.7763	0.7853	0.7580
Average precision	0.5194	0.5230	0.5245	0.5258	0.5263	0.5257	0.5171	0.5264	0.5271	0.5313	0.5316	0.5219
Precision:												
At 5 docs	0.6955	0.7045	0.7091	0.7091	0.6909	0.6955	0.6773	0.7045	0.7000	0.7227	0.7182	0.7000
At 10 docs	0.4636	0.4636	0.4818	0.4773	0.4659	0.4727	0.4591	0.4636	0.4750	0.4795	0.4841	0.4750
At 15 docs	0.3682	0.3652	0.3712	0.3652	0.3667	0.3591	0.3561	0.3591	0.3712	0.3803	0.3652	0.3591
At 20 docs	0.3034	0.3034	0.3045	0.3034	0.3023	0.3080	0.2955	0.3011	0.3057	0.3148	0.3091	0.2989
R-Precision	0.5129	0.5139	0.5155	0.5145	0.5257	0.5258	0.5115	0.5198	0.5194	0.5243	0.5233	0.5228

Table B.1.3. Performance of retrospective MI runs

Measure	RETRO 8 GLOBAL MI COL	RETRO 8 GLOBAL MI COL + 20 OK	RETRO 8 MI COL (win 200)	RETRO 8 MI COL (win 100)	RETRO 8 MI COL (win 50)	RETRO 8 MI COL (win 30)	RETRO 8 MI COL (win 20)	RETRO 12 MI COL (win 200)	RETRO 16 MI COL (win 200)	RETRO 21 MI COL (win 200)
Relevant retrieved	789	796	794	809	818	852	835	809	848	870
Interpolated recall – precision averages										
at 0.00	0.9410	0.9545	0.9545	0.9545	0.9545	0.9545	0.9545	0.9545	0.9545	0.9545
at 0.10	0.6991	0.7103	0.7699	0.7777	0.7922	0.8086	0.7988	0.7901	0.8020	0.8081
at 0.20	0.5466	0.5527	0.6930	0.6895	0.6670	0.6950	0.7015	0.6993	0.6996	0.7035
Average precision	0.3177	0.3104	0.4458	0.4610	0.4551	0.4690	0.4733	0.4688	0.4877	0.4991
Precision:										
At 5 docs	0.4273	0.4318	0.6273	0.6455	0.6455	0.6545	0.6409	0.6727	0.6864	0.6864
At 10 docs	0.3227	0.3068	0.3909	0.4045	0.4159	0.4364	0.4409	0.4159	0.4455	0.4477
At 15 docs	0.2636	0.2576	0.3000	0.3045	0.3091	0.3348	0.3288	0.3076	0.3258	0.3303
At 20 docs	0.2239	0.2250	0.2500	0.2580	0.2591	0.2784	0.2841	0.2523	0.2614	0.2761
R-Precision	0.3104	0.3095	0.4323	0.4482	0.4321	0.4519	0.4439	0.4634	0.4909	0.4853

Measure	RETRO 8 MI COL + 20 OK (win 200)	RETRO 8 MI COL + 20 OK (win 100)	RETRO 8 MI COL + 20 OK (win 50)	RETRO 8 MI COL + 20 OK (win 30)	RETRO 8 MI COL + 20 OK (win 20)	RETRO 16 MI COL + 10 OK (win 200)	RETRO 16 MI COL + 10 OK (win 100)	RETRO 16 MI COL + 10 OK (win 50)	RETRO 16 MI COL + 10 OK (win 30)	RETRO 16 MI COL + 10 OK (win 20)
Relevant retrieved	956	954	955	955	955	976	971	968	967	973
Interpolated recall – precision averages										
at 0.00	0.9545	0.9545	0.9545	0.9545	0.9545	0.9545	0.9545	0.9545	0.9545	0.9545
at 0.10	0.8302	0.8292	0.8485	0.8422	0.8364	0.8539	0.8406	0.8360	0.8509	0.8495
at 0.20	0.7418	0.7450	0.7500	0.7396	0.7517	0.7497	0.7456	0.7506	0.7654	0.7585
Average precision	0.5227	0.5251	0.5240	0.5249	0.5274	0.5274	0.5260	0.5292	0.5291	0.5270
Precision:										
At 5 docs	0.7045	0.7136	0.7182	0.7045	0.7000	0.7182	0.7136	0.7045	0.7273	0.7182
At 10 docs	0.4659	0.4636	0.4727	0.4795	0.4682	0.4773	0.4750	0.4795	0.4773	0.4795
At 15 docs	0.3652	0.3652	0.3636	0.3636	0.3667	0.3652	0.3561	0.3652	0.3758	0.3697
At 20 docs	0.3023	0.3045	0.3000	0.3023	0.3023	0.3080	0.2977	0.3045	0.3114	0.3136
R-Precision	0.5139	0.5229	0.5136	0.5199	0.5234	0.5256	0.5274	0.5286	0.5228	0.5194

Table B.1.3 (continued)

Measure	RETRO 16 MI COL + 20 OK (win 200)	RETRO 16 MI COL + 20 OK (win 100)	RETRO 16 MI COL + 20 OK (win 50)	RETRO 16 MI COL + 20 OK (win 30)	RETRO 16 MI COL + 20 OK (win 20)	RETRO 21 MI COL + 20 OK (win 200)
Relevant retrieved	970	965	963	959	951	972
Interpolated recall – precision averages						
at 0.00	0.9545	0.9545	0.9545	0.9545	0.9545	0.9545
at 0.10	0.8481	0.8371	0.8335	0.8465	0.8554	0.8446
at 0.20	0.7483	0.7445	0.7571	0.7765	0.7673	0.7510
Average precision	0.5264	0.5267	0.5282	0.5301	0.5290	0.5272
Precision:						
At 5 docs	0.7182	0.7182	0.7136	0.7182	0.7318	0.7273
At 10 docs	0.4705	0.4750	0.4705	0.4818	0.4750	0.4773
At 15 docs	0.3576	0.3561	0.3682	0.3712	0.3682	0.3606
At 20 docs	0.3011	0.2977	0.3045	0.3125	0.3080	0.3023
R-Precision	0.5222	0.5258	0.5226	0.5210	0.5162	0.5301

B.2 Trec_eval performance results of predictive MI/Z collocate runs

Tables

Values in all tables:

Retrieved: ≈44000
Relevant: 1583

Table B.2.1. Performance of predictive Okapi runs

Measure	PRED UNEXPANDED	PRED OK 10	PRED OK 20	PRED OK 25	PRED OK 30	PRED OK 35
Relevant retrieved	417	548	559	567	570	577
Interpolated recall – precision averages						
at 0.00	0.3135	0.5144	0.5105	0.5441	0.5420	0.5602
at 0.10	0.2218	0.4014	0.4123	0.4582	0.4549	0.4470
at 0.20	0.1577	0.2946	0.3288	0.3455	0.3290	0.3547
Average precision	0.0799	0.1343	0.1400	0.1520	0.1483	0.1533
Precision:						
At 5 docs	0.1727	0.2864	0.3318	0.3409	0.3182	0.3409
At 10 docs	0.1227	0.2159	0.2364	0.2432	0.2477	0.2432
At 15 docs	0.1076	0.1848	0.1985	0.1985	0.1955	0.1970
At 20 docs	0.0977	0.1682	0.1705	0.1727	0.1727	0.1739
R-Precision	0.1063	0.1584	0.1663	0.1707	0.1768	0.1691

Table B.2.2. Performance of predictive Z runs

Measure	PRED 8 GLOBAL Z COL	PRED 8 GLOBAL Z COL + 20 OK	PRED 8 Z COL (win 200)	PRED 8 Z COL (win 100)	PRED 8 Z COL (win 50)	PRED 8 Z COL (win 30)	PRED 8 Z COL (win 20)	PRED 12 Z COL (win 200)	PRED 12 Z COL (win 100)	PRED 12 Z COL (win 50)	PRED 12 Z COL (win 30)	PRED 12 Z COL (win 20)
Relevant retrieved	545	590	539	538	516	531	531	564	563	540	540	529
Interpolated recall – precision averages												
at 0.00	0.4865	0.5491	0.5192	0.5136	0.5160	0.5305	0.5502	0.5352	0.5391	0.5446	0.5294	0.5358
at 0.10	0.2874	0.4330	0.3684	0.3715	0.3781	0.3821	0.4004	0.3859	0.3854	0.4411	0.4058	0.4177
at 0.20	0.1944	0.3084	0.2901	0.3022	0.3032	0.2801	0.3099	0.3018	0.2925	0.3157	0.3049	0.3289
Average precision	0.0974	0.1360	0.1268	0.1294	0.1302	0.1376	0.1433	0.1346	0.1383	0.1459	0.1401	0.1518
Precision:												
At 5 docs	0.2227	0.3182	0.2591	0.2682	0.2864	0.2955	0.2636	0.2818	0.2864	0.2909	0.2636	0.2955
At 10 docs	0.1750	0.2273	0.2045	0.2114	0.2205	0.2386	0.2250	0.2000	0.2159	0.2205	0.2205	0.2273
At 15 docs	0.1530	0.2000	0.1712	0.1788	0.1939	0.1970	0.1894	0.1712	0.1864	0.1894	0.1924	0.1924
At 20 docs	0.1307	0.1716	0.1477	0.1511	0.1670	0.1750	0.1602	0.1545	0.1625	0.1670	0.1614	0.1693
R-Precision	0.1265	0.1581	0.1686	0.1577	0.1584	0.1666	0.1718	0.1768	0.1717	0.1685	0.1727	0.1803

Measure	PRED 16 Z COL (win 200)	PRED 16 Z COL (win 100)	PRED 16 Z COL (win 50)	PRED 16 Z COL (win 30)	PRED 16 Z COL (win 20)	PRED 21 Z COL (win 200)	PRED 21 Z COL (win 100)	PRED 21 Z COL (win 50)	PRED 21 Z COL (win 30)	PRED 21 Z COL (win 20)
Relevant retrieved	574	567	542	533	532	576	574	552	553	531
Interpolated recall – precision averages										
at 0.00	0.5395	0.5365	0.5200	0.5351	0.5568	0.5193	0.5530	0.5230	0.5357	0.5214
at 0.10	0.3953	0.3833	0.4002	0.4117	0.4070	0.3975	0.4280	0.3982	0.4197	0.4144
at 0.20	0.3031	0.3163	0.2845	0.3014	0.3049	0.3163	0.3166	0.2963	0.3030	0.2996
Average precision	0.1386	0.1456	0.1362	0.1367	0.1432	0.1391	0.1480	0.1325	0.1396	0.1384
Precision:										
At 5 docs	0.2864	0.2773	0.2682	0.2909	0.2909	0.2909	0.2955	0.2864	0.2955	0.2909
At 10 docs	0.2114	0.2227	0.2182	0.2295	0.2341	0.2227	0.2159	0.2250	0.2364	0.2273
At 15 docs	0.1818	0.1833	0.1864	0.1909	0.2076	0.1682	0.1818	0.1833	0.2015	0.1924
At 20 docs	0.1693	0.1591	0.1636	0.1659	0.1750	0.1545	0.1591	0.1614	0.1818	0.1716
R-Precision	0.1751	0.1740	0.1662	0.1696	0.1822	0.1717	0.1727	0.1669	0.1729	0.1743

Table B.2.2. (continued)

Measure	PRED 8 Z COL + 20 OK (win 200)	PRED 8 Z COL + 20 OK (win 100)	PRED 8 Z COL + 20 OK (win 50)	PRED 8 Z COL + 20 OK (win 30)	PRED 8 Z COL + 20 OK (win 20)	PRED 16 Z COL + 10 OK (win 200)	PRED 16 Z COL + 20 OK (win 200)	PRED 16 Z COL + 20 OK (win 100)	PRED 16 Z COL + 20 OK (win 50)	PRED 16 Z COL + 20 OK (win 30)	PRED 16 Z COL + 20 OK (win 20)	PRED 21 Z COL + 20 OK (win 200)
Relevant retrieved	579	578	580	581	581	583	586	583	579	572	569	582
Interpolated recall – precision averages												
at 0.00	0.5421	0.5631	0.5422	0.5472	0.5426	0.5782	0.5781	0.5491	0.5417	0.5660	0.5534	0.5642
at 0.10	0.4542	0.4679	0.4441	0.4396	0.4357	0.3855	0.4347	0.4439	0.4353	0.4398	0.4372	0.3978
at 0.20	0.3613	0.3747	0.3724	0.3467	0.3480	0.3089	0.3372	0.3484	0.3370	0.3419	0.3410	0.3206
Average precision	0.1536	0.1602	0.1549	0.1568	0.1553	0.1388	0.1527	0.1561	0.1495	0.1485	0.1477	0.1413
Precision:												
At 5 docs	0.3318	0.3409	0.3273	0.3409	0.3364	0.2909	0.2955	0.3182	0.3000	0.3409	0.3364	0.3182
At 10 docs	0.2477	0.2591	0.2477	0.2500	0.2477	0.2182	0.2341	0.2386	0.2318	0.2477	0.2500	0.2318
At 15 docs	0.2091	0.2076	0.2061	0.2091	0.2030	0.1879	0.2091	0.2015	0.2045	0.2015	0.1955	0.1909
At 20 docs	0.1795	0.1795	0.1784	0.1818	0.1784	0.1682	0.1773	0.1784	0.1807	0.1739	0.1727	0.1648
R-Precision	0.1853	0.1837	0.1797	0.1822	0.1829	0.1760	0.1883	0.1855	0.1764	0.1828	0.1795	0.1763

Table B.2.3. Performance of predictive MI runs

Measure	PRED 8 GLOBAL MI COL	PRED 8 GLOBAL MI COL + 20 OK	PRED 8 MI COL (win 200)	PRED 8 MI COL (win 100)	PRED 8 MI COL (win 50)	PRED 8 MI COL (win 30)	PRED 8 MI COL (win 20)	PRED 12 MI COL (win 200)	PRED 12 MI COL (win 100)	PRED 12 MI COL (win 50)	PRED 12 MI COL (win 30)	PRED 12 MI COL (win 20)
Relevant retrieved	505	503	484	490	488	514	510	521	530	513	523	521
Interpolated recall – precision averages												
at 0.00	0.4745	0.4583	0.4229	0.4564	0.4589	0.5036	0.4994	0.4880	0.4946	0.5086	0.5397	0.5687
at 0.10	0.3251	0.3162	0.3057	0.2953	0.3531	0.3930	0.4002	0.3694	0.3319	0.4007	0.4093	0.4103
at 0.20	0.2485	0.2358	0.2399	0.2159	0.2531	0.2743	0.3067	0.2769	0.2656	0.2996	0.3105	0.3132
Average precision	0.1155	0.1099	0.1006	0.0971	0.1158	0.1399	0.1371	0.1222	0.1202	0.1365	0.1435	0.1524
Precision:												
At 5 docs	0.2682	0.2591	0.2318	0.2273	0.2500	0.2773	0.2591	0.2591	0.2591	0.2682	0.2818	0.3000
At 10 docs	0.1955	0.2023	0.1705	0.1705	0.1886	0.2136	0.2023	0.1955	0.2068	0.2159	0.2182	0.2295
At 15 docs	0.1621	0.1727	0.1409	0.1530	0.1652	0.1803	0.1697	0.1742	0.1712	0.1758	0.1879	0.1985
At 20 docs	0.1375	0.1420	0.1284	0.1318	0.1420	0.1534	0.1443	0.1534	0.1500	0.1511	0.1580	0.1682
R-Precision	0.1396	0.1391	0.1387	0.1256	0.1551	0.1667	0.1607	0.1696	0.1539	0.1627	0.1672	0.1748

Measure	PRED 16 MI COL (win 200)	PRED 16 MI COL (win 100)	PRED 16 MI COL (win 50)	PRED 16 MI COL (win 30)	PRED 16 MI COL (win 20)	PRED 21 MI COL (win 200)	PRED 21 MI COL (win 100)	PRED 21 MI COL (win 50)	PRED 21 MI COL (win 30)	PRED 21 MI COL (win 20)
Relevant retrieved	533	535	522	524	515	541	542	527	532	517
Interpolated recall – precision averages										
at 0.00	0.5215	0.5549	0.5071	0.5325	0.5230	0.5335	0.5329	0.5454	0.5476	0.4866
at 0.10	0.3918	0.4134	0.3874	0.4011	0.3791	0.4053	0.3889	0.3772	0.3926	0.3784
at 0.20	0.3084	0.3132	0.2833	0.3035	0.2905	0.3062	0.3093	0.2778	0.2951	0.2931
Average precision	0.1302	0.1437	0.1341	0.1372	0.1380	0.1390	0.1440	0.1295	0.1355	0.1359
Precision:										
At 5 docs	0.2636	0.2864	0.2682	0.2909	0.2864	0.2909	0.3091	0.2818	0.2864	0.2818
At 10 docs	0.1864	0.2045	0.2114	0.2182	0.2273	0.2023	0.2136	0.2091	0.2227	0.2227
At 15 docs	0.1667	0.1727	0.1803	0.1818	0.1879	0.1742	0.1682	0.1667	0.1788	0.1803
At 20 docs	0.1466	0.1545	0.1568	0.1648	0.1568	0.1523	0.1500	0.1511	0.1636	0.1580
R-Precision	0.1591	0.1677	0.1662	0.1728	0.1796	0.1658	0.1691	0.1615	0.1758	0.1618

Table B.2.3. (continued)

Measure	PRED 8 MI COL + 20 OK (win 200)	PRED 8 MI COL + 20 OK (win 100)	PRED 8 MI COL + 20 OK (win 50)	PRED 8 MI COL + 20 OK (win 30)	PRED 8 MI COL + 20 OK (win 20)	PRED 16 MI COL + 10 OK (win 200)	PRED 16 MI COL + 20 OK (win 200)	PRED 16 MI COL + 20 OK (win 100)	PRED 16 MI COL + 20 OK (win 50)	PRED 16 MI COL + 20 OK (win 30)	PRED 16 MI COL + 20 OK (win 20)	PRED 21 MI COL + 20 OK (win 200)
Relevant retrieved	564	566	581	582	581	582	583	580	575	573	576	578
Interpolated recall – precision averages												
at 0.00	0.5260	0.5426	0.5509	0.5415	0.5339	0.5417	0.5655	0.5665	0.5383	0.5601	0.5556	0.5599
at 0.10	0.4339	0.4259	0.4379	0.4273	0.4202	0.4008	0.4412	0.4433	0.4269	0.4294	0.4241	0.4159
at 0.20	0.3437	0.3300	0.3581	0.3498	0.3472	0.3214	0.3549	0.3619	0.3513	0.3471	0.3310	0.3364
Average precision	0.1455	0.1426	0.1501	0.1565	0.1528	0.1398	0.1515	0.1626	0.1522	0.1486	0.1487	0.1477
Precision:												
At 5 docs	0.3182	0.3273	0.3182	0.3409	0.3136	0.3045	0.3182	0.3273	0.3091	0.3182	0.3273	0.3136
At 10 docs	0.2432	0.2500	0.2409	0.2364	0.2341	0.2273	0.2409	0.2523	0.2409	0.2455	0.2477	0.2273
At 15 docs	0.2000	0.2030	0.2000	0.2000	0.1970	0.1833	0.1970	0.2045	0.2000	0.1985	0.2000	0.1864
At 20 docs	0.1739	0.1773	0.1784	0.1784	0.1761	0.1625	0.1739	0.1830	0.1784	0.1773	0.1727	0.1670
R-Precision	0.1760	0.1710	0.1754	0.1825	0.1809	0.1716	0.1816	0.1880	0.1835	0.1797	0.1800	0.1737

B.3 Trec_eval performance results of CRW runs

Values in all tables:

Retrieved: ≈44000
Relevant: 1583

Table B.3.1 Performance results of CRW runs

Measure	RETRO 20 CRW (200 win)	RETRO 20 CRW (50 win)	RETRO 35 CRW (200 win)	RETRO 35 CRW (50 win)	PRED 20 CRW (200 win)	PRED 20 CRW (50 win)	PRED 35 CRW (200 win)	PRED 35 CRW (50 win)
Relevant retrieved	831	800	773	811	550	538	539	544
Interpolated recall – precision averages								
at 0.00	0.8447	0.8297	0.8298	0.8320	0.4649	0.5524	0.5352	0.5331
at 0.10	0.6903	0.6905	0.7134	0.7302	0.3233	0.3749	0.3174	0.3885
at 0.20	0.6018	0.6017	0.5856	0.6363	0.2543	0.2805	0.2382	0.2799
Average precision	0.3917	0.3845	0.4000	0.4048	0.1082	0.1228	0.1085	0.1221
Precision:								
At 5 docs	0.5273	0.5000	0.5136	0.5409	0.2636	0.2909	0.2591	0.2955
At 10 docs	0.3591	0.3636	0.3523	0.3727	0.1955	0.2114	0.2045	0.2295
At 15 docs	0.2833	0.2894	0.2773	0.2864	0.1545	0.1833	0.1727	0.1985
At 20 docs	0.2477	0.2455	0.2420	0.2580	0.1352	0.1648	0.1534	0.1682
R-Precision	0.4034	0.3829	0.3972	0.4131	0.1549	0.1552	0.1390	0.1588

B.4 Selected expanded queries from the run 'PRED 8 Z COL + 20 OK (100 window size)'

Tables*

* Synonym sets represented in the tables by GSL codes are listed in Appendix D.3

Table B.4.1. Expanded query for topic 252: Combatting alien smuggling

xenofobia	collocate of 1 query term(s);
balk	collocate of 1 query term(s);
immigr	collocate of 3 query term(s); okapi RF term;
migrent	collocate of 1 query term(s);
allai	collocate of 1 query term(s);
brunt	collocate of 1 query term(s);
influx	collocate of 1 query term(s); okapi RF term;
surveil	collocate of 1 query term(s);
understaf	collocate of 2 query term(s);
cozi	collocate of 1 query term(s);
breathtakingli	collocate of 2 query term(s);
omar	collocate of 1 query term(s);
smuggl	collocate of 1 query term(s); okapi RF term; original query term;
consular	collocate of 1 query term(s);
illeg	collocate of 2 query term(s); okapi RF term;
sheng	collocate of 1 query term(s); okapi RF term;
smuggler	collocate of 1 query term(s); okapi RF term;
jianmin	collocate of 1 query term(s);
emigr	collocate of 1 query term(s);
asylum	okapi RF term;
alien	okapi RF term; original query term;
foreign	okapi RF term;
agent	okapi RF term;
ship	okapi RF term;
seeker	okapi RF term;
land	okapi RF term;
vessel	okapi RF term;
dm55m	okapi RF term;
effort	okapi RF term;
turn	okapi RF term;
ring	okapi RF term;
fare	okapi RF term;
crime	okapi RF term;
combat	original query term;

Table B.4.2. Expanded query for topic 255: Environmental protection

kampur	collocate of 2 query term(s);
ganga	collocate of 2 query term(s);
khodabux	collocate of 2 query term(s);
borrego	collocate of 1 query term(s);
macario	collocate of 2 query term(s);
arrabida	collocate of 2 query term(s);
nath	collocate of 2 query term(s);
eloi	collocate of 1 query term(s);
quercus	collocate of 1 query term(s);
tombeau	collocate of 1 query term(s);
pollution	okapi RF term;
pollut	okapi RF term;
protect	okapi RF term; original query term;
@0104	okapi RF term; original query term;
pesticid	okapi RF term;
degrad	okapi RF term;
wast	okapi RF term;
sewag	okapi RF term;
factori	okapi RF term;
impos	okapi RF term;
hazard	okapi RF term;
water	okapi RF term;
untreat	okapi RF term;
group	okapi RF term;
law	okapi RF term;
implement	okapi RF term;
earth	okapi RF term;
standard	okapi RF term;
aim	okapi RF term;
action	okapi RF term;

Table B.4.3. Expanded query for topic 257: Cigarette consumption

jinan	collocate of 2 query term(s);
kremenchuh	collocate of 2 query term(s); okapi RF term;
winn	collocate of 2 query term(s);
zpt	collocate of 1 query term(s);
swaythl	collocate of 1 query term(s); okapi RF term;
radom	collocate of 1 query term(s);
ixel	collocate of 2 query term(s);
nyren	collocate of 2 query term(s);
tabacalera	collocate of 1 query term(s);
shandong	collocate of 1 query term(s);
cigarett	collocate of 1 query term(s); okapi RF term; original query term;
consumpt	okapi RF term; original query term;
tobacco	okapi RF term;
brand	okapi RF term;
smoke	okapi RF term;
market	okapi RF term;
rj	okapi RF term;
@0018	okapi RF term;
smoker	okapi RF term;
@0290	okapi RF term;
rothman	okapi RF term;
factori	okapi RF term;
@0068	okapi RF term;
serhi	okapi RF term;
pta750m	okapi RF term;
boriak	okapi RF term;
reynold	okapi RF term;
product	okapi RF term;

Table B.4.4. Expanded query for topic 258: Computer security

urvil	collocate of 1 query term(s); okapi RF term;
sundevil	collocate of 2 query term(s); okapi RF term;
perv	collocate of 1 query term(s); okapi RF term;
nealesbp	collocate of 2 query term(s); okapi RF term;
morrisreo	collocate of 2 query term(s); okapi RF term;
fono	collocate of 1 query term(s); okapi RF term;
hacker	collocate of 2 query term(s); okapi RF term;
firewal	collocate of 2 query term(s); okapi RF term;
stoll	collocate of 1 query term(s);
internet	collocate of 1 query term(s); okapi RF term;
unauthor	collocate of 1 query term(s);
@0071	okapi RF term; original query term;
network	okapi RF term;
penetr	okapi RF term;
goddam	okapi RF term;
delrai	okapi RF term;
alter	okapi RF term;
@0041	okapi RF term;
code	okapi RF term;
sofist	okapi RF term;
suffix	okapi RF term;
breach	okapi RF term;
secur	original query term;

Table B.4.5. Expanded query for topic 261: Threat posed by fissionable material

vasiliyev	collocate of 2 query term(s);
plutonium	collocate of 4 query term(s); okapi RF term;
smuggl	collocate of 1 query term(s);
nuclear	collocate of 2 query term(s); okapi RF term;
reactor	collocate of 2 query term(s); okapi RF term;
grafit	collocate of 2 query term(s);
weapon	collocate of 1 query term(s); okapi RF term;
gram	collocate of 1 query term(s);
tomonitor	collocate of 1 query term(s);
geni	collocate of 1 query term(s);
renegad	collocate of 1 query term(s);
wean	collocate of 1 query term(s);
deton	collocate of 1 query term(s); okapi RF term;
deuterid	collocate of 1 query term(s);
nonweapon	collocate of 1 query term(s);
tritium	collocate of 1 query term(s);
unenrich	collocate of 1 query term(s);
deuterium	collocate of 1 query term(s);
thermonuclear	collocate of 1 query term(s);
remix	collocate of 1 query term(s);
fissil	collocate of 1 query term(s); okapi RF term;
uranium	collocate of 1 query term(s); okapi RF term;
enrich	collocate of 1 query term(s); okapi RF term;
novikov	collocate of 1 query term(s);
kruchenkov	collocate of 1 query term(s);
bomb	okapi RF term;
radioact	okapi RF term;
fission	okapi RF term; original query term;
civil	okapi RF term;
rod	okapi RF term;
stockpil	okapi RF term;
prevent	okapi RF term;
atom	okapi RF term;
@0330	okapi RF term;
dismantl	okapi RF term;
storag	okapi RF term;
grade	okapi RF term;
threat	original query term;
pose	original query term;
materi	original query term;

Table B.4.6. Expanded query for topic 263: Algae as food supplement

spirulina	collocate of 1 query term(s); okapi RF term;
platensis	collocate of 1 query term(s); okapi RF term;
knoydart	collocate of 1 query term(s); okapi RF term;
titaghur	collocate of 1 query term(s); okapi RF term;
seaweed	collocate of 1 query term(s); okapi RF term;
spillag	collocate of 1 query term(s); okapi RF term;
jute	collocate of 1 query term(s); okapi RF term;
kilo	collocate of 1 query term(s); okapi RF term;
brealei	okapi RF term;
alga	okapi RF term; original query term;
chare	okapi RF term;
186	okapi RF term;
puriti	okapi RF term;
jar	okapi RF term;
protein	okapi RF term;
slick	okapi RF term;
pond	okapi RF term;
calcutta	okapi RF term;
reg	okapi RF term;
61m	okapi RF term;
food	original query term;
supplem	original query term;

Table B.4.7. Expanded query for topic 266: Professional scuba diving

woodwood	collocate of 2 query term(s); okapi RF term;
mensun	collocate of 2 query term(s); okapi RF term;
magnotomet	collocate of 2 query term(s); okapi RF term;
bowyer	collocate of 2 query term(s);
cosheril	collocate of 1 query term(s); okapi RF term;
diver	collocate of 2 query term(s); okapi RF term;
aldernei	collocate of 1 query term(s);
codirect	collocate of 1 query term(s);
divetrack	collocate of 1 query term(s); okapi RF term;
matchlock	collocate of 1 query term(s); okapi RF term;
flagg	collocate of 1 query term(s); okapi RF term;
1600	okapi RF term;
dive	okapi RF term; original query term;
@0043	okapi RF term;
rs232c	okapi RF term;
16kei	okapi RF term;
surfac	okapi RF term;
exactli	okapi RF term;
licence	okapi RF term;
command	okapi RF term;
ship	okapi RF term;
kilobyt	okapi RF term;
scabbard	okapi RF term;
profession	original query term;
scuba	original query term;

Table B.4.8. Expanded query for topic 271: Solar power

flagsol	collocate of 2 query term(s); okapi RF term;
benemann	collocate of 2 query term(s); okapi RF term;
telaga	collocate of 2 query term(s); okapi RF term;
otten	collocate of 1 query term(s);
voltaic	collocate of 1 query term(s); okapi RF term;
rheinelb	collocate of 2 query term(s); okapi RF term;
etsu	collocate of 2 query term(s);
energiesystem	collocate of 2 query term(s); okapi RF term;
solar	collocate of 1 query term(s); okapi RF term; original query term;
windpow	collocate of 1 query term(s);
technologi	okapi RF term;
fotovolta	okapi RF term;
power	okapi RF term; original query term;
develop	okapi RF term;
cell	okapi RF term;
energi	okapi RF term;
research	okapi RF term;
unit	okapi RF term;
sun	okapi RF term;
fossil	okapi RF term;
17000	okapi RF term;
4588	okapi RF term;
30panel	okapi RF term;

Table B.4.9. Expanded query for topic 273: Volcanic and seismic activity levels

gomera	collocate of 1 query term(s);
tarawera	collocate of 3 query term(s); okapi RF term;
rotorua	collocate of 2 query term(s);
vulcanolog	collocate of 1 query term(s); okapi RF term;
teneguia	collocate of 2 query term(s); okapi RF term;
tecina	collocate of 2 query term(s); okapi RF term;
taburient	collocate of 2 query term(s);
muchacho	collocate of 2 query term(s);
lowinterest	collocate of 1 query term(s);
monica	collocate of 1 query term(s);
fernando	collocate of 1 query term(s);
unscath	collocate of 1 query term(s);
rilei	collocate of 1 query term(s);
uneven	collocate of 1 query term(s);
vallei	collocate of 1 query term(s); okapi RF term;
earthquak	collocate of 1 query term(s); okapi RF term;
letrero	collocate of 1 query term(s);
rotomahana	collocate of 2 query term(s);
silbo	collocate of 1 query term(s);
waimang	collocate of 1 query term(s); okapi RF term;
hydrotherm	collocate of 1 query term(s);
inferno	collocate of 1 query term(s);
crater	collocate of 1 query term(s); okapi RF term;
scree	collocate of 1 query term(s);
volcan	okapi RF term; original query term;
erupt	okapi RF term;
volcano	okapi RF term;
san	okapi RF term;
aftershock	okapi RF term;
mountain	okapi RF term;
quak	okapi RF term;
disast	okapi RF term;
mud	okapi RF term;
santa	okapi RF term;
wairoa	okapi RF term;
uraba	okapi RF term;
seismic	original query term;
activ	original query term;
level	original query term;

Table B.4.10. Expanded query for topic 290: Foreign automobile manufacturers in U.S.

schriner	collocate of 2 query term(s);
keiretsus	collocate of 4 query term(s);
ussorc	collocate of 2 query term(s);
phh	collocate of 1 query term(s);
ofliv	collocate of 1 query term(s);
greenfield	collocate of 1 query term(s);
vehicl	collocate of 1 query term(s); okapi RF term;
coolli	collocate of 1 query term(s);
transplent	collocate of 2 query term(s);
huf	collocate of 1 query term(s);
sarcast	collocate of 1 query term(s);
gefardt	collocate of 1 query term(s);
honda	collocate of 3 query term(s); okapi RF term;
reilli	collocate of 1 query term(s);
feroc	collocate of 1 query term(s);
tmm	collocate of 2 query term(s);
acura	collocate of 1 query term(s);
merced	collocate of 1 query term(s);
kinzer	collocate of 1 query term(s);
alamaba	collocate of 1 query term(s);
marysvil	collocate of 1 query term(s);
alliston	collocate of 1 query term(s);
blueberri	collocate of 1 query term(s);
car	okapi RF term;
engin	okapi RF term;
toyota	okapi RF term;
export	okapi RF term;
@0192	okapi RF term;
@0014	okapi RF term;
manufactur	okapi RF term; original query term;
largest	okapi RF term;
import	okapi RF term;
announc	okapi RF term;
detroit	okapi RF term;
product	okapi RF term;
carolina	okapi RF term;
533000	okapi RF term;
world	okapi RF term;
chrysler	okapi RF term;
bmw	okapi RF term;
freightlin	okapi RF term;
foreign	original query term;
automobil	original query term;
@0400	original query term;

B.5 Evaluation of performance by categories of terms in the expanded queries of the run 'PRED 16 MI COL + 20 OK (100 window size)'

Tables

Figures

Table B.5.1. Influence of categories of terms in the expanded queries 'PRED 16 MI COL + 20 OK (100 window size)' on average precision

Category	Improve	Indifferent	Degrade	Total
1. Collocate	150	461	300	911
2. Collocate of 2 or more query terms	45	176	93	314
3. Collocate of 1 query term	105	285	207	597
4. Okapi RF term	179	201	279	659
5. Original query term	73	57	30	160
6. Collocate of 2 or more query terms and an Okapi RF term	20	43	25	88
7. Collocate of 1 query term and an Okapi RF term	27	39	57	123
8. Collocate and an Okapi RF term	47	82	82	211
9. Collocate and an original term	8	1	3	12
10. Okapi RF term and an original query term	34	6	10	50

Table B.5.2. Influence of categories of terms in the expanded queries 'PRED 16 MI COL + 20 OK (100 window size)' on average precision (in percentage)

Term category	Improve	Indifferent	Degrade
1. Collocate	17%	50%	33%
2. Collocate of 2 or more query terms	14%	56%	30%
3. Collocate of 1 query term	18%	48%	34%
4. Okapi RF term	27%	31%	42%
5. Original query term	45%	36%	19%
6. Collocate of 2 or more query terms and an Okapi RF term	23%	49%	28%
7. Collocate of 1 query term and an Okapi RF term	22%	32%	46%
8. Collocate and an Okapi RF term	22%	39%	39%
9. Collocate and an original term	67%	8%	25%
10. Okapi RF term and an original query term	68%	12%	20%

Figure B.5.1. Influence of categories of terms in the expanded queries 'PRED 16 MI COL + 20 OK (100 window size)' on average precision

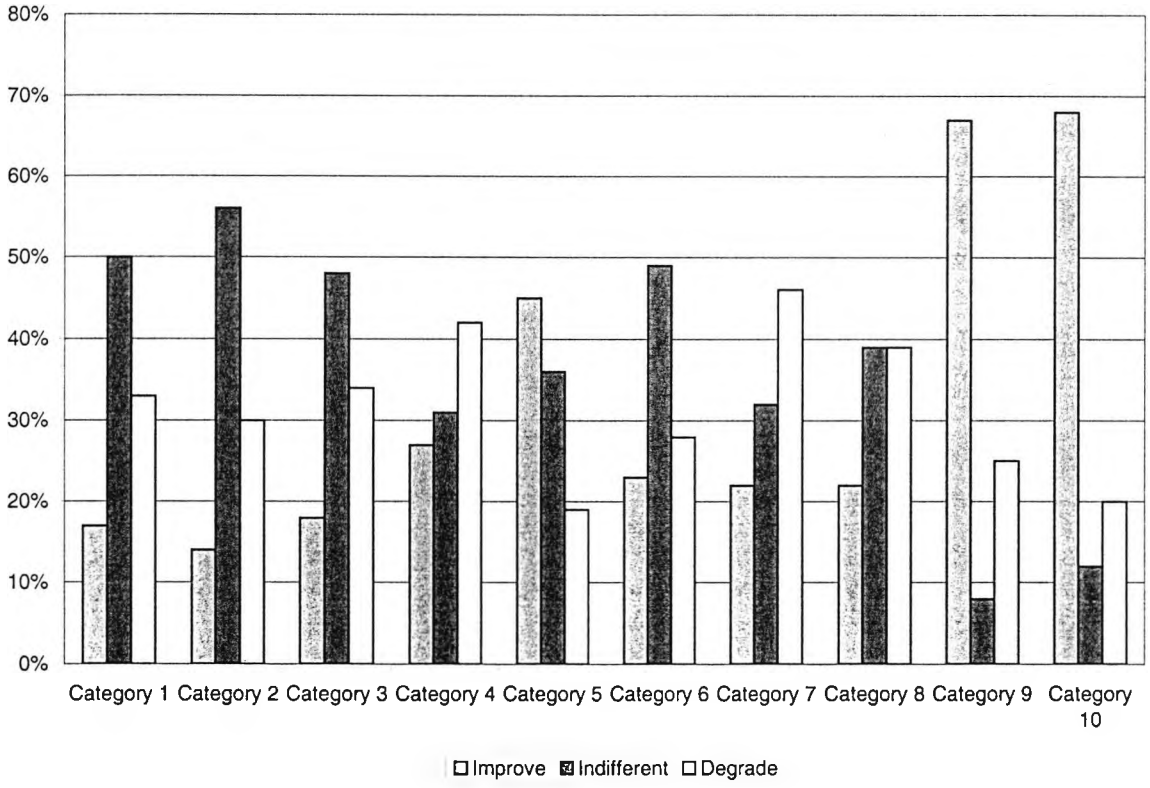


Figure B.5.2. Distribution of precision differences for category 1: *Collocate*

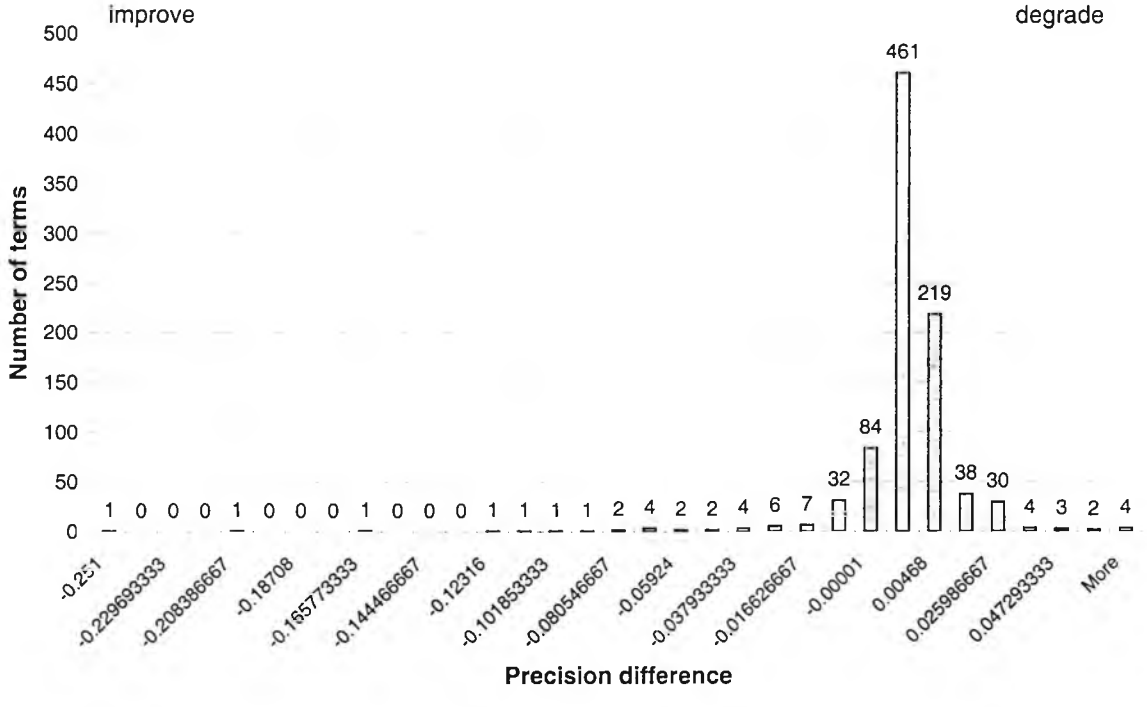


Figure B.5.3. Distribution of precision differences for category 2:
Collocate of 2 or more query terms

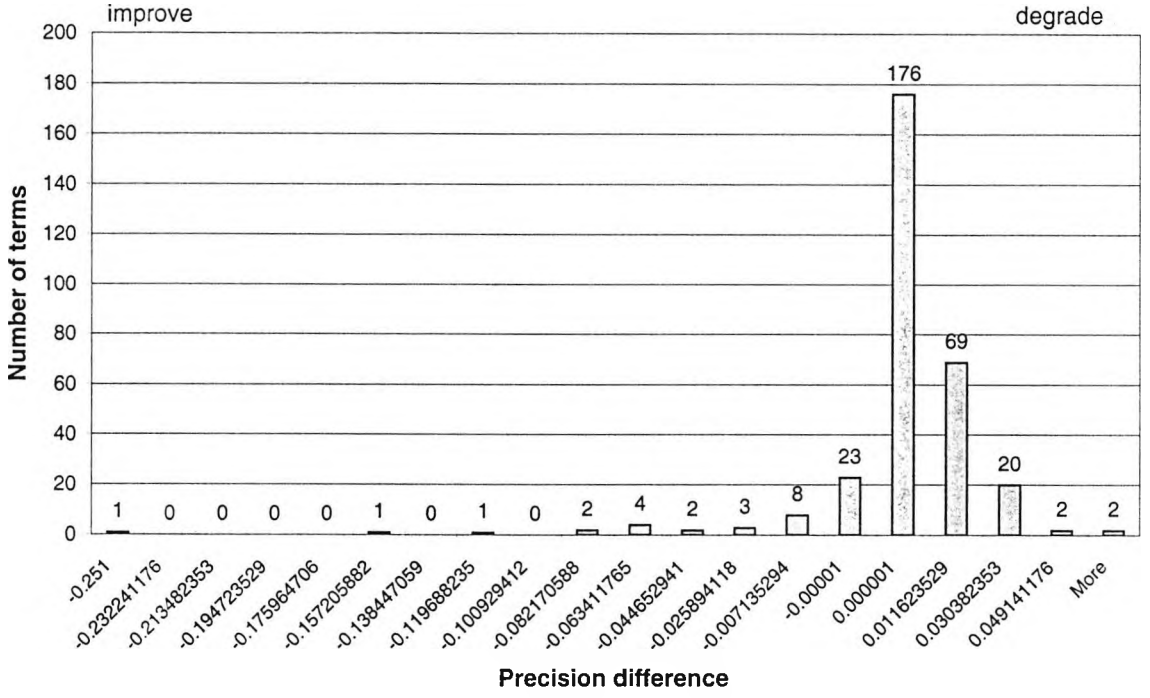


Figure B.5.4. Distribution of precision differences for category 3:
Collocate of 1 query term

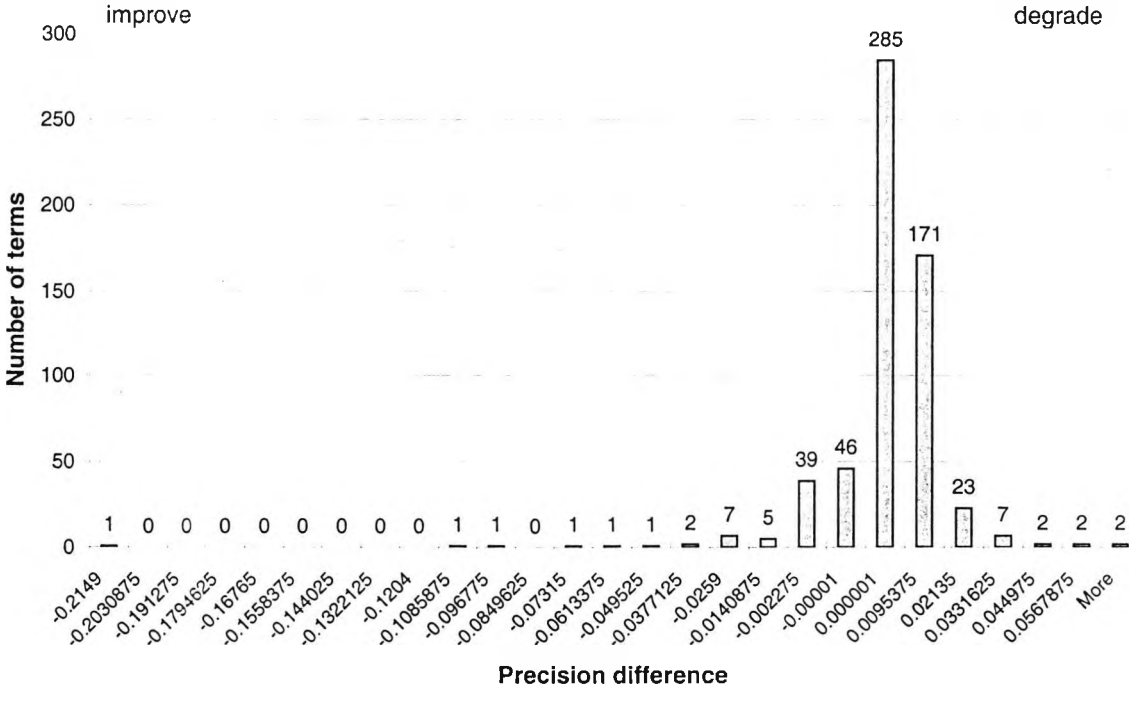


Figure B.5.5. Distribution of precision differences for category 4: *Okapi RF term*

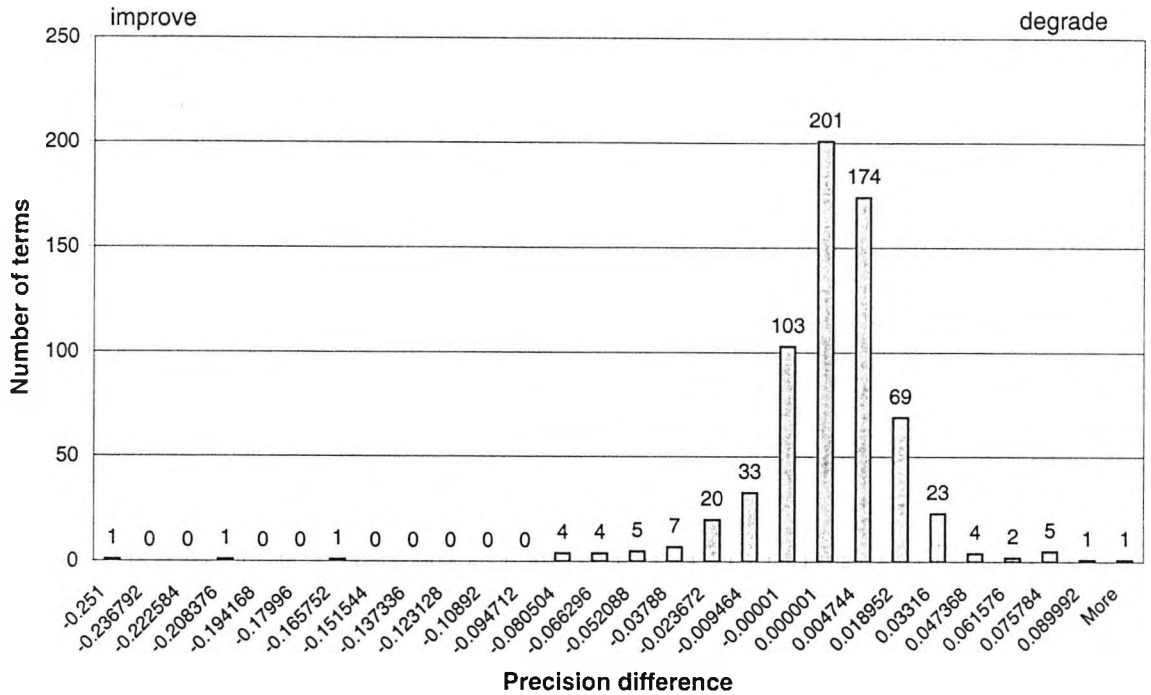


Figure B.5.6. Distribution of precision differences for category 5: *Original query term*

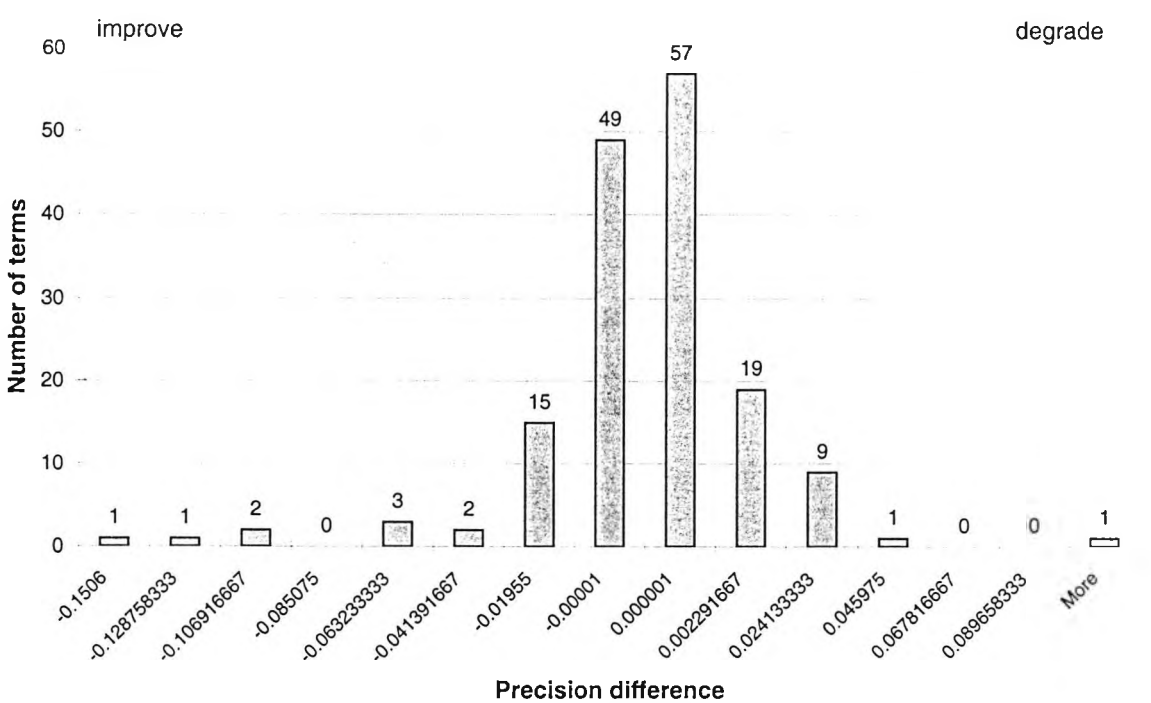


Figure B.5.7. Distribution of precision differences for category 9: *Collocate and an original query term*

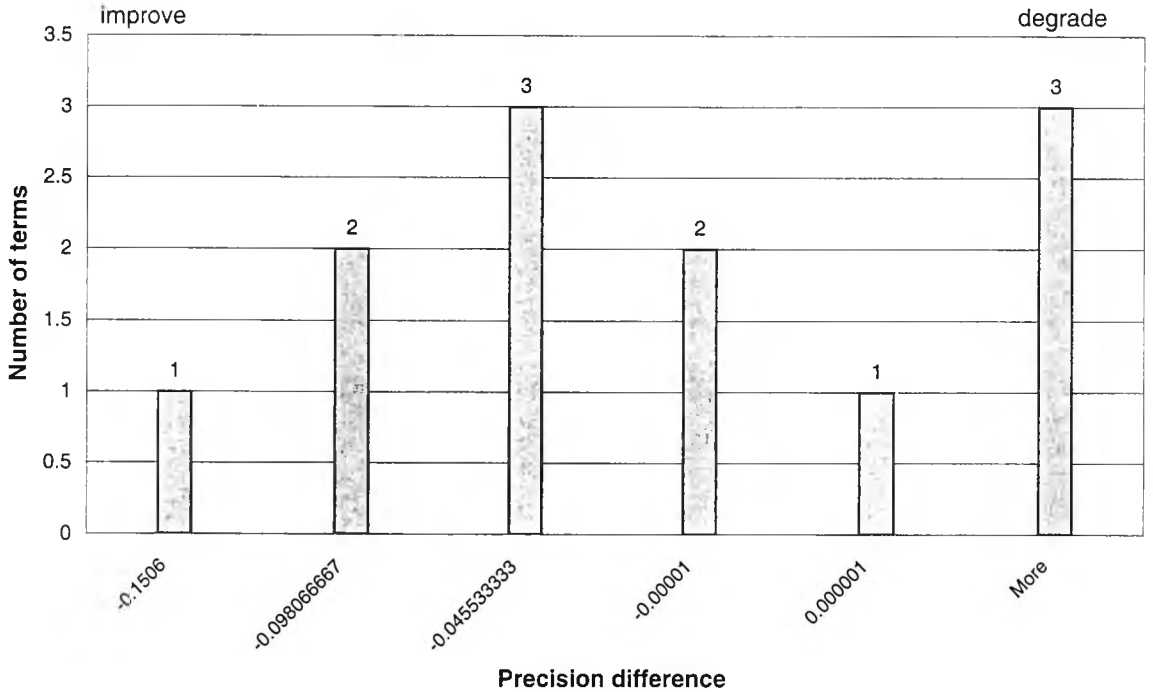
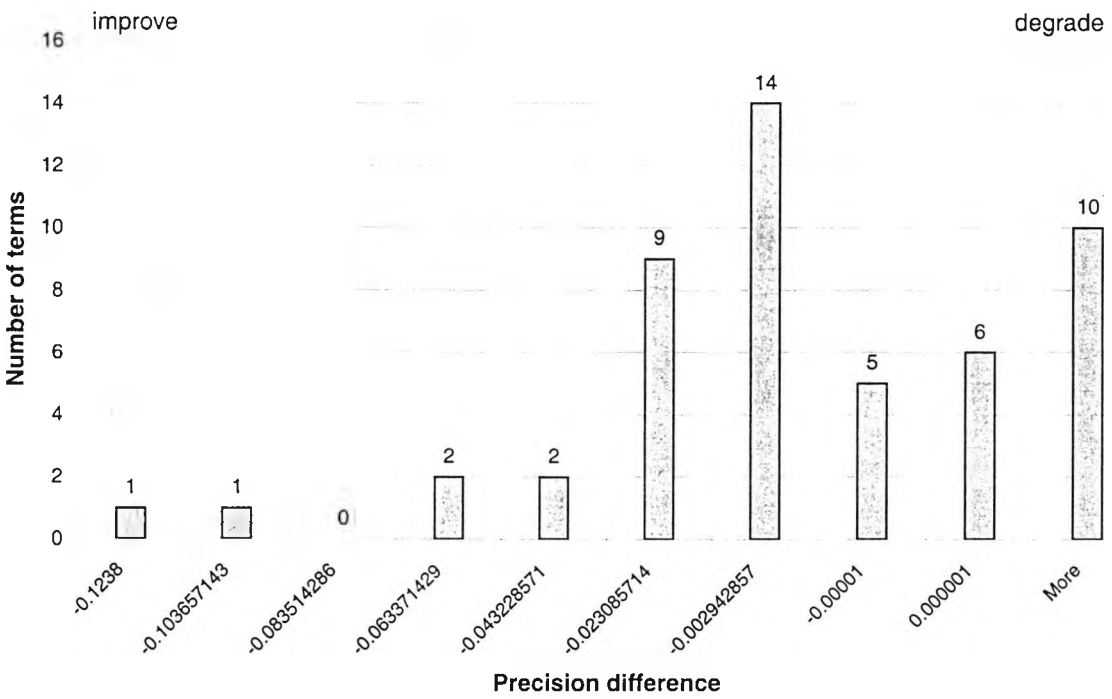


Figure B.5.8. Distribution of precision differences for category 10: *Okapi RF term and an original query term*



Appendix C

Lexical cohesion analysis using local collocations

C.1 Mean and standard deviation of Okapi scores in the aligned sets of relevant and non-relevant documents

Tables

Table C.1.1. Mean and standard deviation of Okapi scores in the aligned sets of relevant documents selected from the top 100 Okapi-ranked documents

Topic	Mean Okapi score	Standard deviation
251	8.1676	0.11853607
252	14.678	3.298275812
253	13.256	0
254	10.7855	2.131926945
255	11.016375	0.388297727
257	15.2012	1.442859722
258	11.0708	1.130715128
259	22.36266667	7.316521737
260	11.28733333	1.652631941
261	13.51733333	2.29944573
263	14.70266667	6.624058383
264	14.019	2.760295697
266	12.532	1.872610744
268	9.78	0
271	12.4041	1.970015931
272	10.5435	0.161927453
273	12.0945	2.902460525
274	12.098	1.685742566
275	26.746	0
277	18.317	2.427154095
278	20.4535	9.050259692
280	18.67088889	5.128866772
282	14.886	2.476287948
283	10.9035	0.185957791
284	13.429	0.212132034
285	14.8409	0.938582673
286	9.3501	0.404318205
287	10.12175	1.008600144
288	14.10575	2.232416684
289	11.95775	4.398466478
290	13.274	0
291	9.4157	0.680563174
292	9.986125	0.789788478
295	13.809	5.00273845
298	9.821222222	5.555214842
299	11.005	0.938172159

Table C.1.2. Mean and standard deviation of Okapi scores in the aligned sets of non-relevant documents selected from the top 100 Okapi-ranked documents

Topic	Mean Okapi score	Standard deviation
251	8.1662	0.113224114
252	14.53425	3.184937087
253	13.268	0
254	10.781	2.158089896
255	11.021	0.382975568
257	15.2591	1.484438798
258	11.115	1.160685358
259	21.34033333	4.373964144
260	11.29533333	1.666213172
261	13.353	2.005139895
263	14.5775	5.104743431
264	13.96957143	2.727472142
266	12.678	1.976198624
268	9.798	0
271	12.3896	1.944278112
272	10.5435	0.183140656
273	12.06833333	2.794842226
274	11.8755	1.342795777
275	26.579	0
277	17.741	1.204368299
278	19.2815	7.343303923
280	18.82255556	5.441571145
282	14.919	2.423962046
283	10.90433333	0.190281546
284	13.43	0.222031529
285	14.8609	1.026696585
286	9.3393	0.380719217
287	10.1215	1.005932238
288	14.332	2.672516791
289	11.10025	1.96064645
290	13.206	0
291	9.4174	0.682510757
292	10.013375	0.808739573
295	13.55533333	4.429382162
298	9.588111111	5.216169702
299	11.01766667	0.980854899

Table C.1.3. Mean and standard deviation of Okapi scores in the aligned sets of relevant documents selected from the top 1000 Okapi-ranked documents

Topic	Mean Okapi score	Standard deviation
251	7.8349	0.382867964
252	11.61025	4.002150699
253	10.814	3.453509519
254	8.508714286	1.875768439
255	10.7034	0.743391059
257	15.2012	1.442859722
258	10.43066667	1.865861052
259	19.2905	8.569742839
260	11.28733333	1.652631941
261	11.4108	3.281965258
263	14.70266667	6.624058383
264	13.0302	2.764395887
265	7.841	0
266	11.2015	3.068990333
268	8.12275	1.235707995
269	5.7587	0.348441945
271	12.4041	1.970015931
272	10.5435	0.161927453
273	10.646	3.640962196
274	8.3182	2.100927774
275	19.7575	9.883231481
277	13.66	4.192800769
278	14.17225	8.942022306
280	18.67088889	5.128866772
282	12.26766667	4.861379365
283	10.7355	0.258911933
284	13.429	0.212132034
285	14.8409	0.938582673
286	9.3501	0.404318205
287	9.279666667	1.636980106
288	14.10575	2.232416684
289	10.0958	0.632538062
290	8.9817	1.646519295
291	9.4157	0.680563174
292	9.7095	0.908787257
293	10.22533333	1.262232282
294	11.223	0
295	11.96075	5.508996302
297	7.716	0
298	9.821222222	5.555214842
299	10.2825	1.008806572
300	14.251	0

Table C.1.4. Mean and standard deviation of Okapi scores in the aligned sets of non-relevant documents selected from the top 1000 Okapi-ranked documents

Topic	Mean Okapi score	Standard deviation
251	7.8344	0.381333683
252	11.540375	3.897662447
253	10.8225	3.458459267
254	8.508428571	1.878684722
255	10.7078	0.741658172
257	15.2591	1.484438798
258	10.46733333	1.895939099
259	18.5225	6.671964753
260	11.29533333	1.666213172
261	11.3114	3.100665993
263	14.5775	5.104743431
264	12.9976	2.726821071
265	7.841	0
266	11.31075	3.175069172
268	8.12675	1.244162201
269	5.7589	0.348893314
271	12.3896	1.944278112
272	10.5435	0.183140656
273	10.626375	3.57128164
274	8.273	1.983113097
275	19.674	9.765144648
277	13.46275	3.744605215
278	13.58925	7.825008301
280	18.82255556	5.441571145
282	12.29	4.865461849
283	10.7361	0.260921636
284	13.43	0.222031529
285	14.8609	1.026696585
286	9.3393	0.380719217
287	9.283666667	1.633704829
288	14.332	2.672516791
289	10.1016	0.636227458
290	8.9788	1.625926457
291	9.4174	0.682510757
292	9.7266	0.935044527
293	10.22533333	1.26078005
294	11.233	0
295	11.7695	5.082953898
297	7.721	0
298	9.588111111	5.216169702
299	10.28816667	1.030470071
300	14.25	0

C.2 Distribution of lexical cohesion scores in the relevant and non-relevant sets

Figures

Figure C.2.1 Distribution of LCS_{links} scores (window size 20) in the aligned relevant and non-relevant sets, created from the top 100 Okapi documents

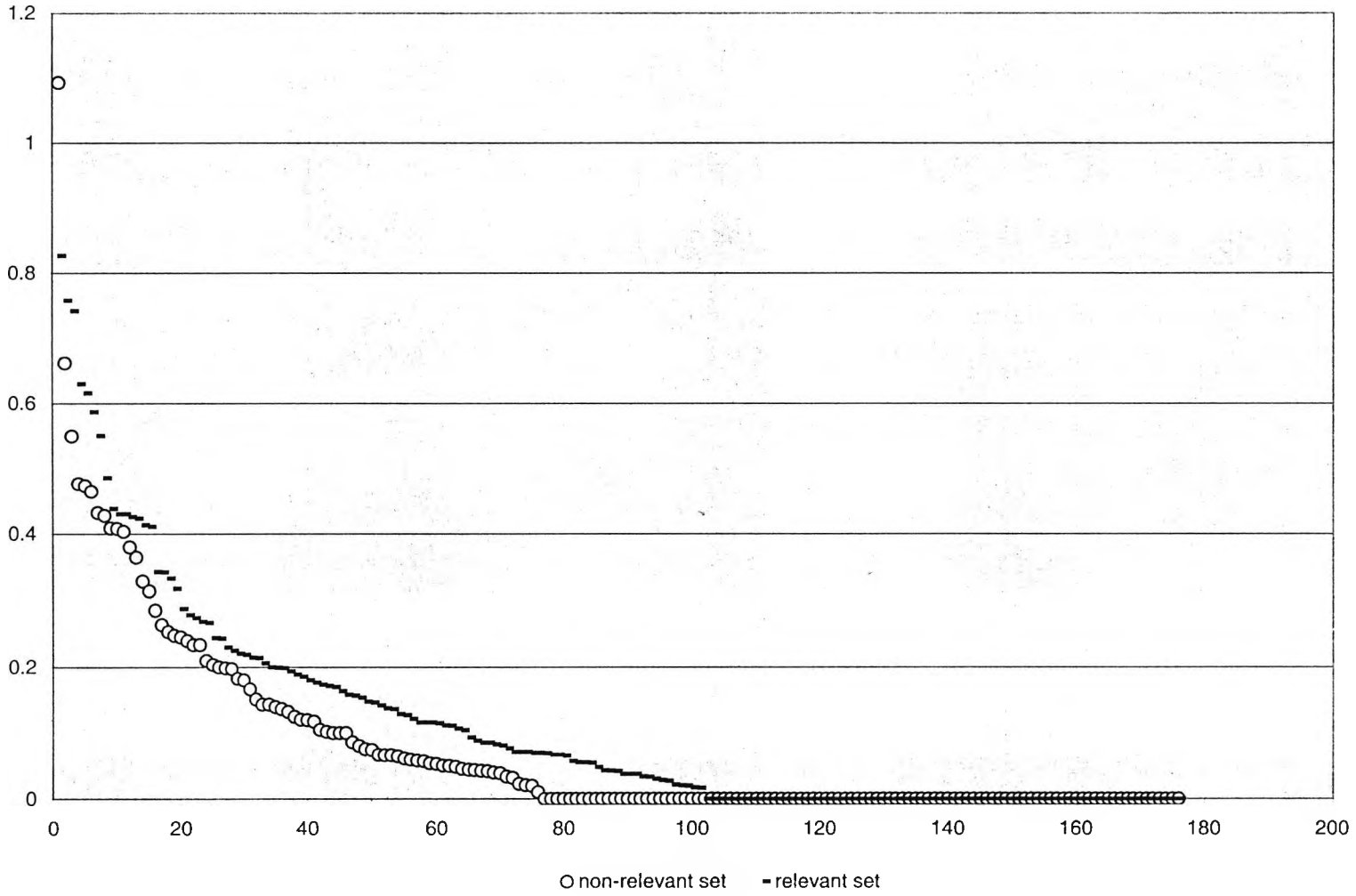


Figure C.2.2 Distribution of LCS_{links} scores (window size 40) in the aligned relevant and non-relevant sets, created from the top 100 Okapi documents

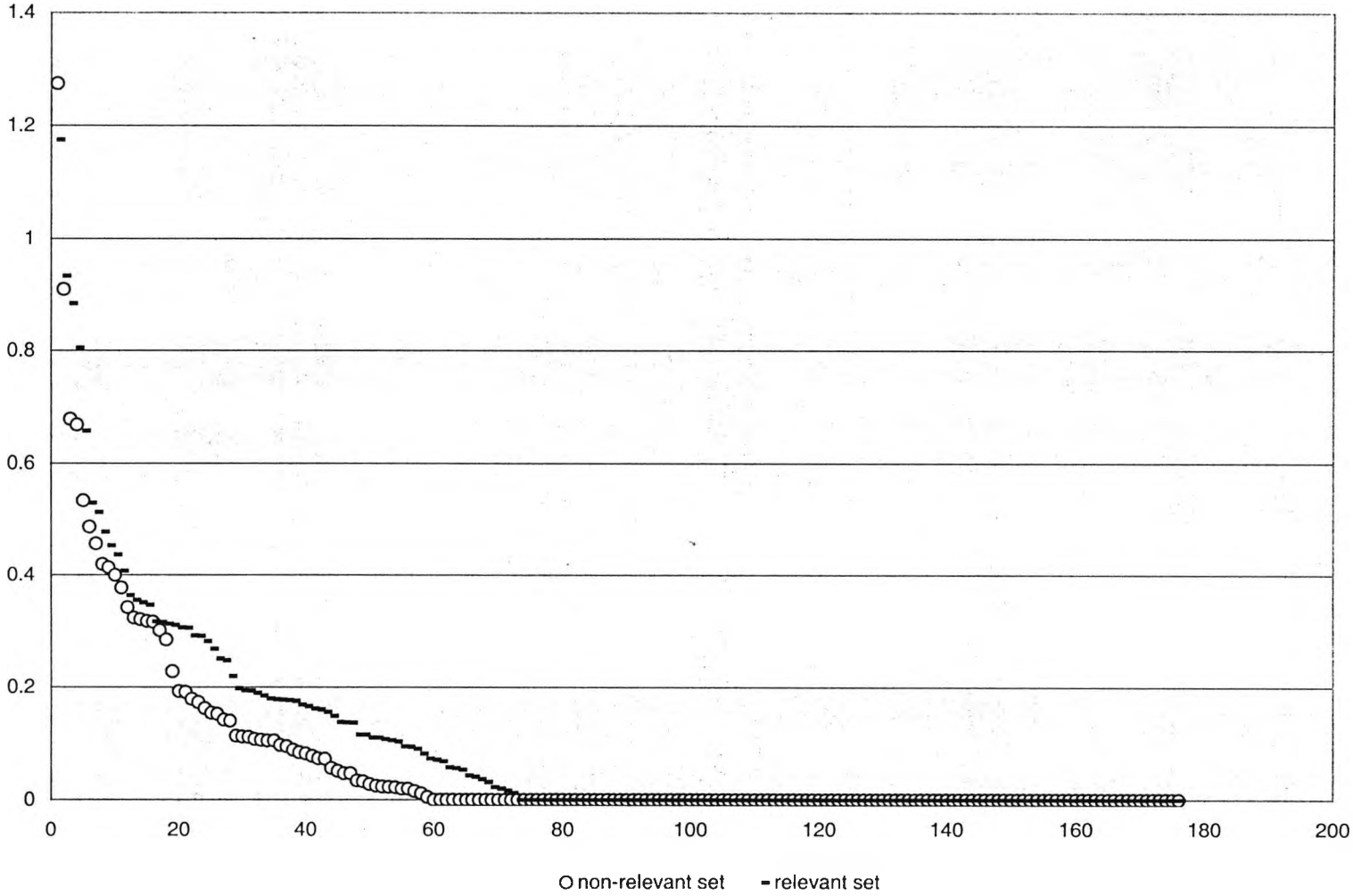


Figure C.2.3 Distribution of LCS_{types} scores (window size 20) in the aligned relevant and non-relevant sets, created from the top 100 Okapi documents

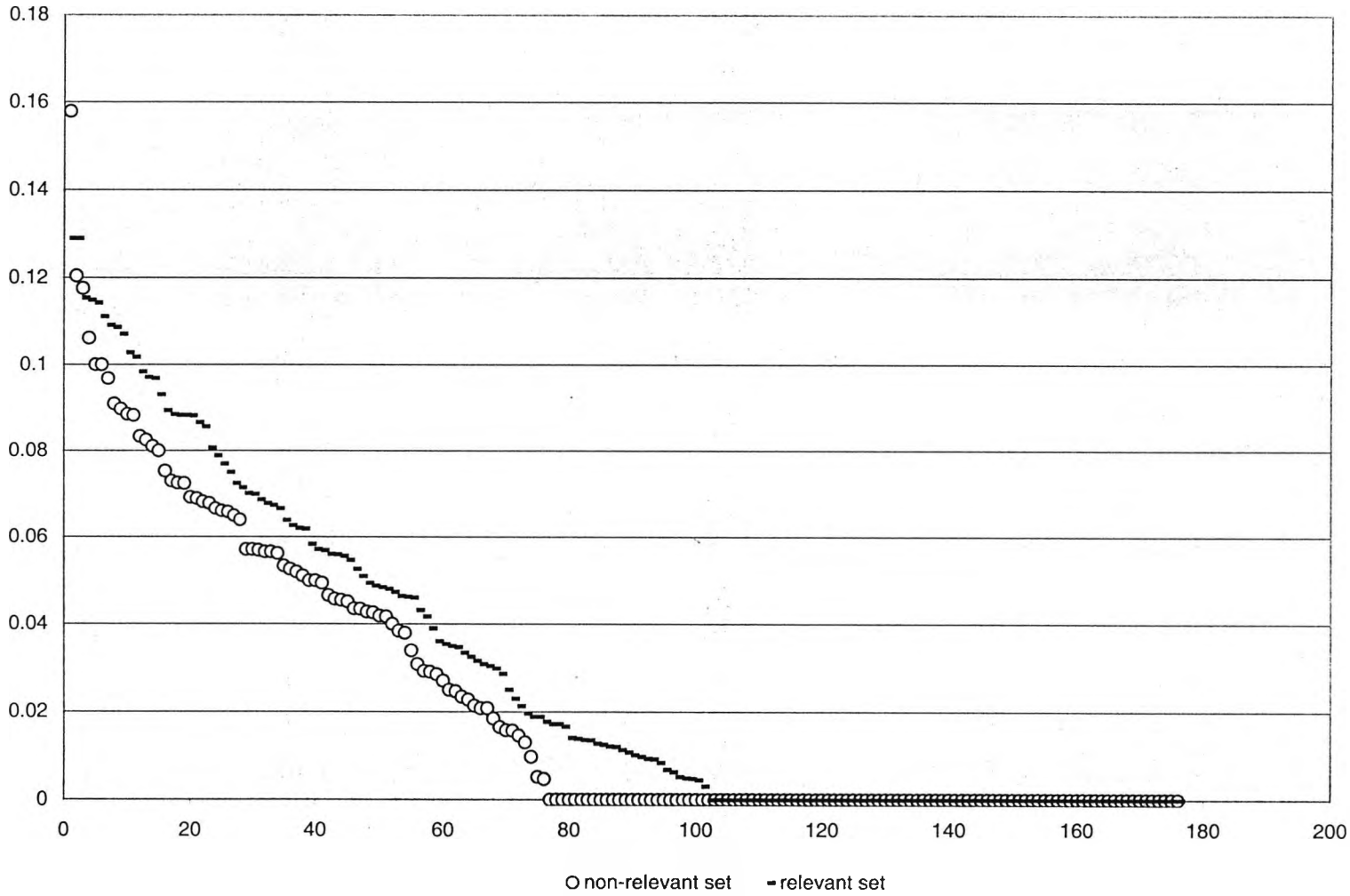


Figure C.2.4 Distribution of LCS_{types} scores (window size 40) in the aligned relevant and non-relevant sets, created from the top 100 Okapi documents

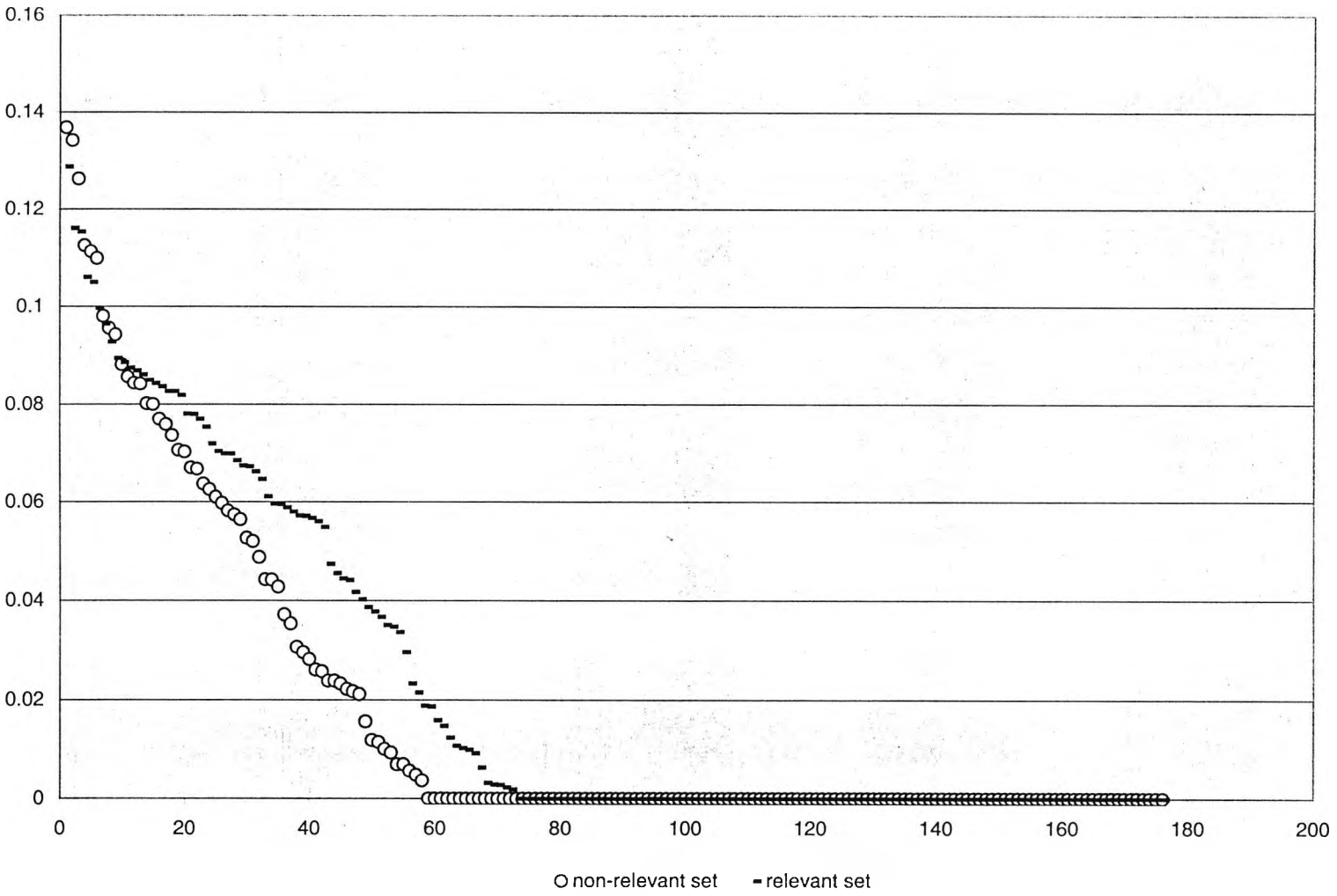


Figure C.2.5 Distribution of LCS_{links} scores (window size 20) in the aligned relevant and non-relevant sets, created from the top 1000 Okapi documents

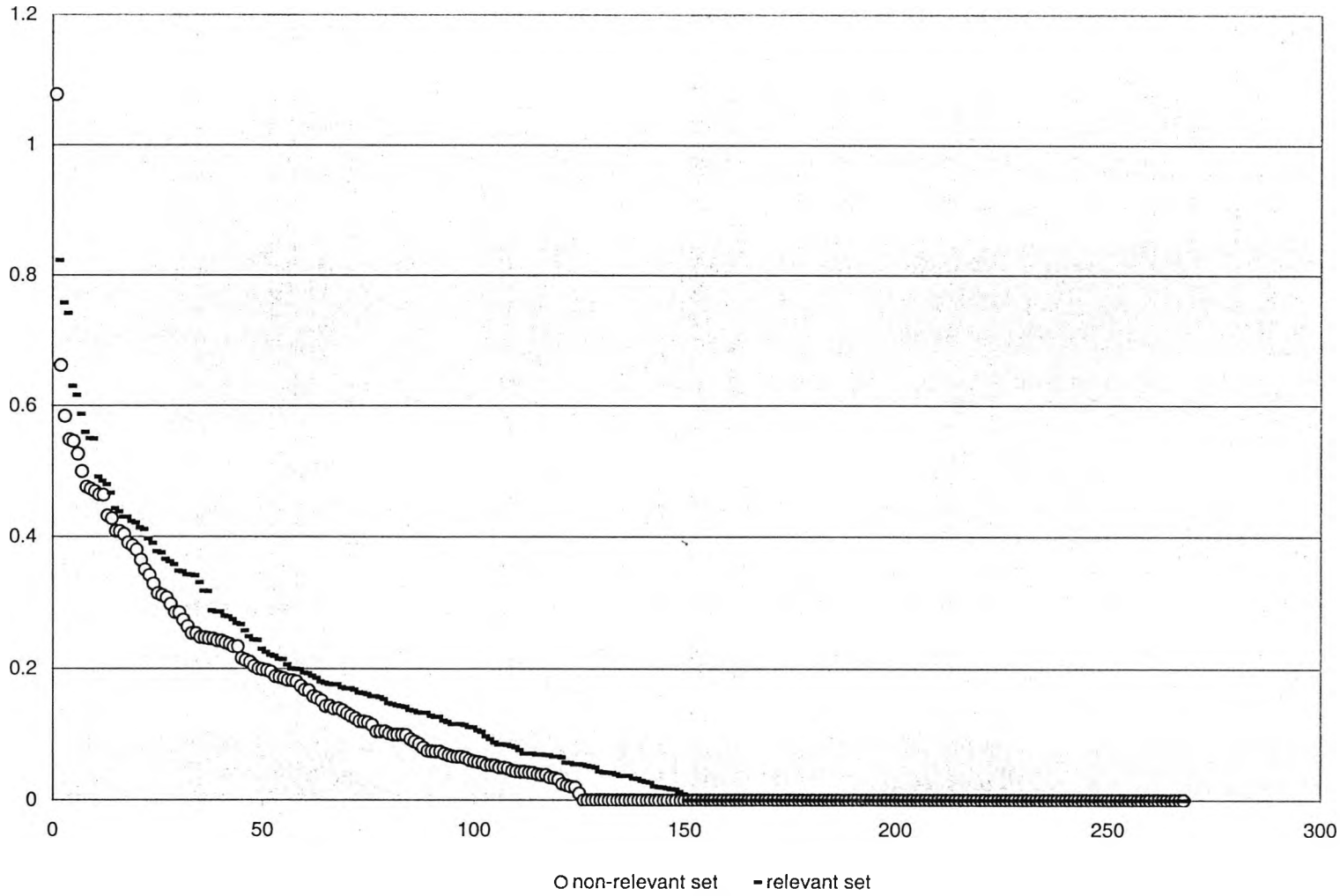


Figure C.2.6 Distribution of LCS_{links} scores (window size 40) in the aligned relevant and non-relevant sets, created from the top 1000 Okapi documents

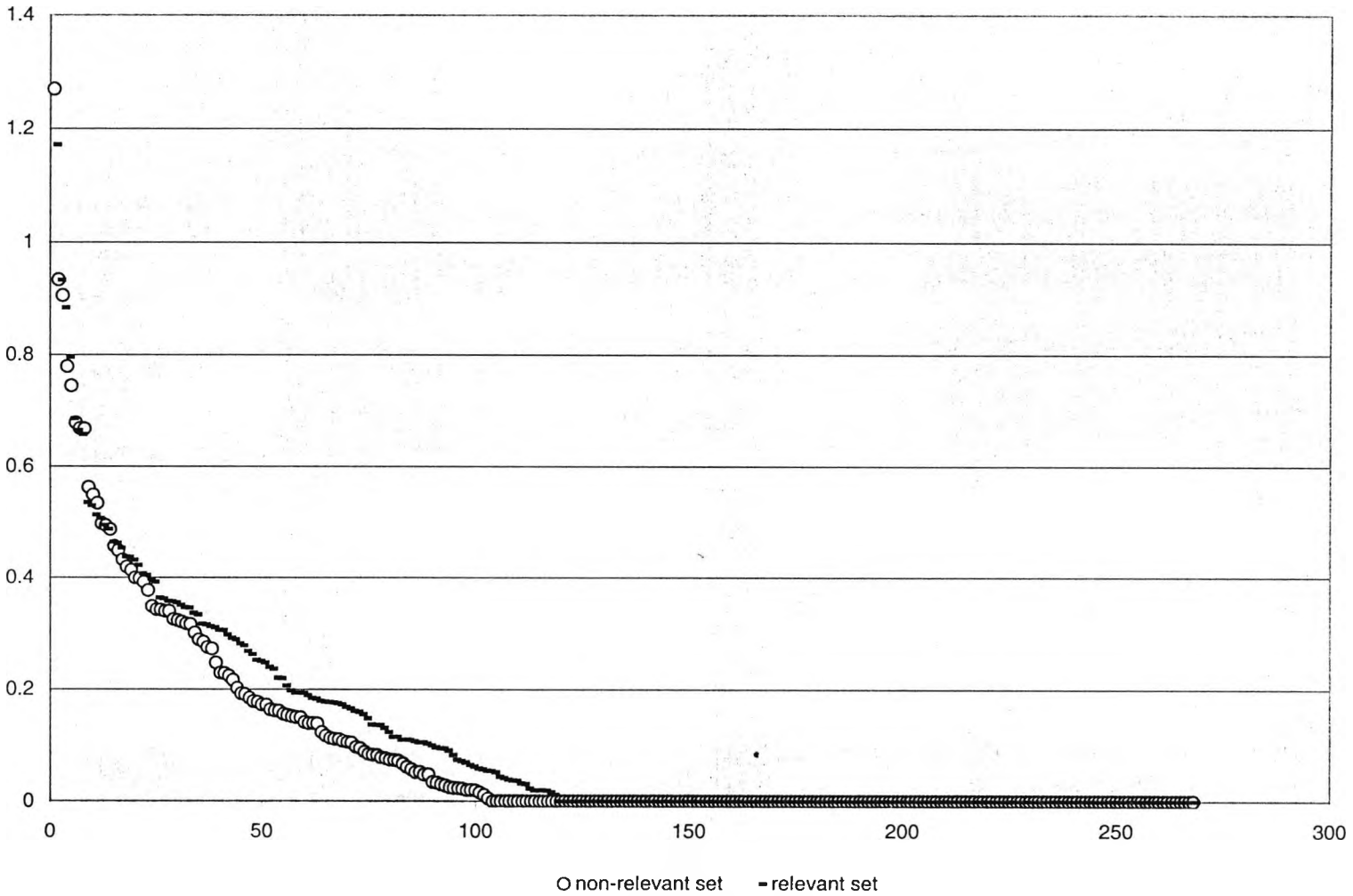


Figure C.2.7 Distribution of LCS_{types} scores (window size 20) in the aligned relevant and non-relevant sets, created from the top 1000 Okapi documents

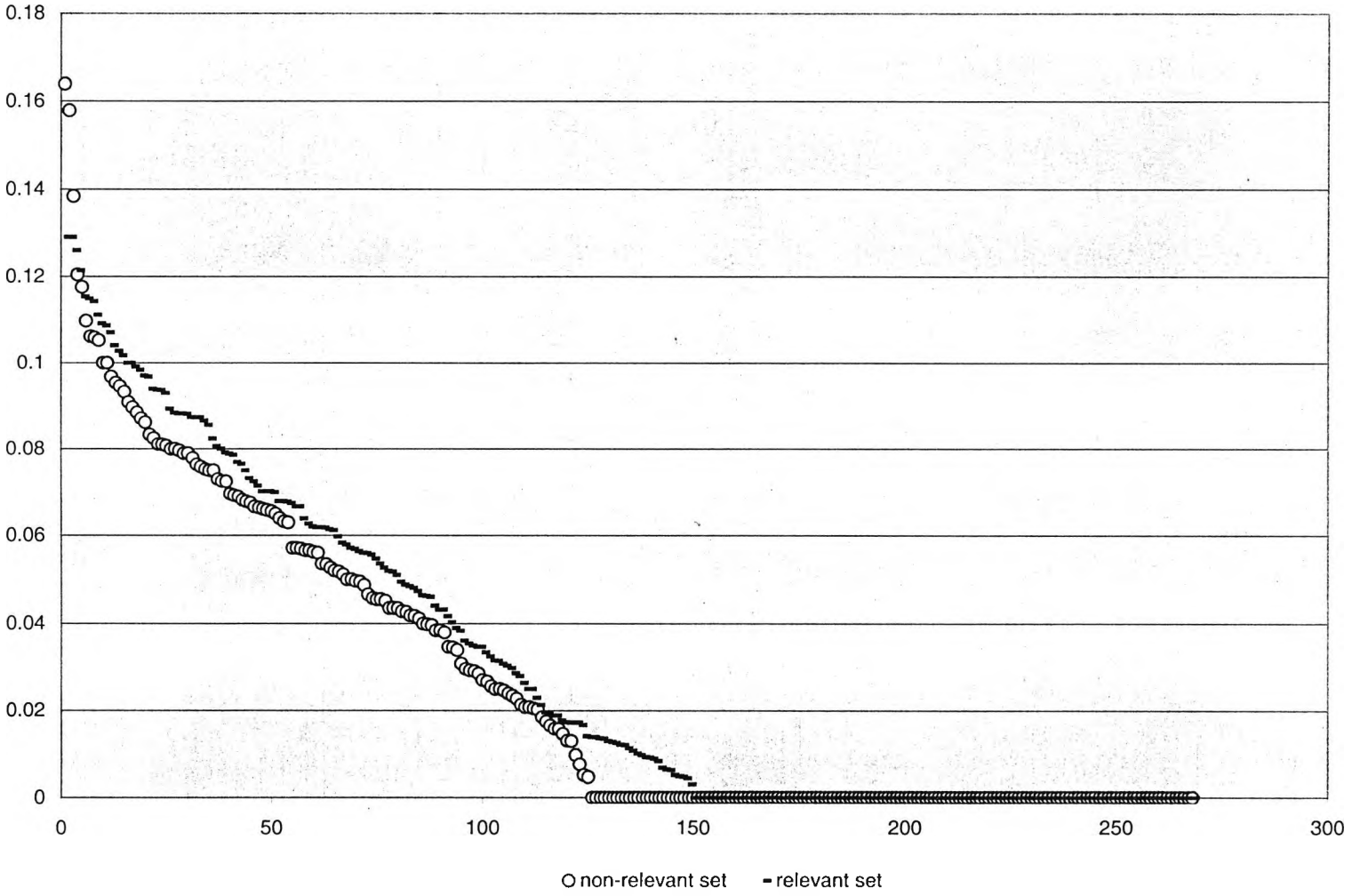
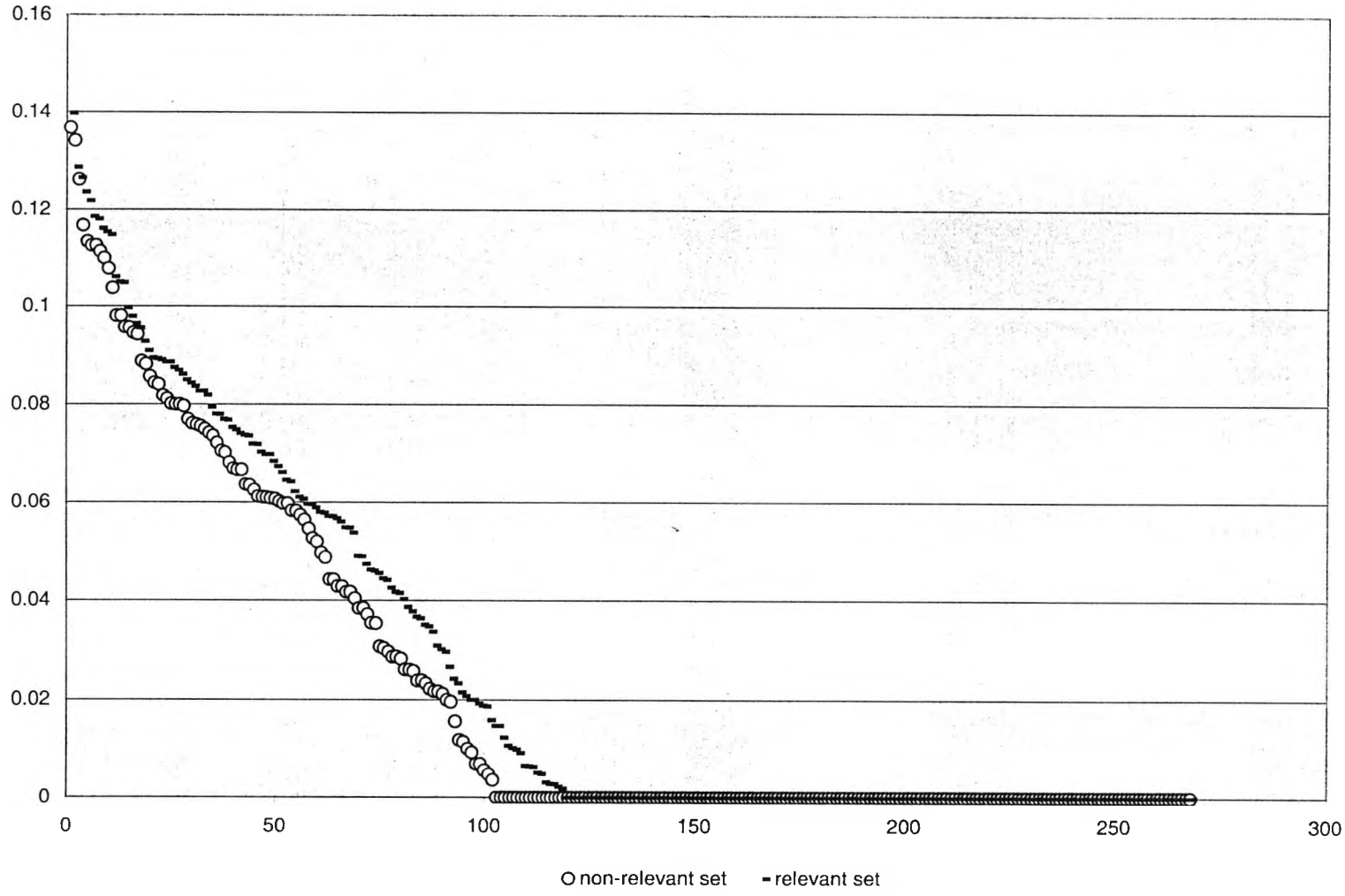


Figure C.2.8 Distribution of LCS_{types} scores (window size 40) in the aligned relevant and non-relevant sets, created from the top 1000 Okapi documents



C.3 Trec_eval performance results of re-ranking Okapi sets by *COMB-LCS*

Tables

Values in all tables:

Retrieved:	42686
Relevant:	1583
Relevant retrieved:	632

Table C.3.1. Performance of re-ranking Okapi sets by *COMB-LCS* (method 1 – links)

Measure	LINKS x=0.25 win 20	LINKS x=0.25 win 40	LINKS x=0.5 win 20	LINKS x=0.5 win 40	LINKS x=0.75 win 20	LINKS x=0.75 win 40	LINKS x=1 win 20	LINKS x=1 win 40	LINKS x=1.5 win 20	LINKS x=1.5 win 40
Interpolated recall – precision averages										
at 0.00	0.4173	0.4131	0.4219	0.4153	0.4236	0.4235	0.4241	0.4197	0.4251	0.4248
at 0.10	0.3130	0.3133	0.3126	0.3132	0.3058	0.3140	0.3063	0.3136	0.3064	0.3146
at 0.20	0.2071	0.2049	0.2082	0.2079	0.2091	0.2096	0.2098	0.2074	0.2083	0.2031
Average precision	0.1348	0.1332	0.1348	0.1339	0.1336	0.1348	0.1337	0.1341	0.1335	0.1327
Precision:										
At 5 docs	0.2136	0.2000	0.2136	0.2091	0.2091	0.2091	0.2091	0.2136	0.2091	0.2045
At 10 docs	0.1591	0.1545	0.1614	0.1591	0.1614	0.1568	0.1591	0.1568	0.1614	0.1614
At 15 docs	0.1409	0.1424	0.1439	0.1409	0.1439	0.1409	0.1439	0.1409	0.1455	0.1439
At 20 docs	0.1284	0.1273	0.1295	0.1273	0.1295	0.1261	0.1295	0.1261	0.1284	0.1284
R-Precision	0.1499	0.1497	0.1494	0.1501	0.1502	0.1561	0.1502	0.1558	0.1502	0.1498

Measure	LINKS x=3 win 20	LINKS x=3 win 40	LINKS x=30 win 20	LINKS x=30 win 40
Relevant retrieved				
Interpolated recall – precision averages				
at 0.00	0.4384	0.4092	0.3456	0.2982
at 0.10	0.3054	0.3003	0.2198	0.1922
at 0.20	0.2010	0.1978	0.1475	0.1360
Average precision	0.1320	0.1278	0.0924	0.0809
Precision:				
At 5 docs	0.2136	0.2182	0.1591	0.1318
At 10 docs	0.1636	0.1568	0.1295	0.1114
At 15 docs	0.1470	0.1409	0.1197	0.1000
At 20 docs	0.1330	0.1273	0.1136	0.0943
R-Precision	0.1512	0.1435	0.0994	0.0912

Table C.3.2. Performance of re-ranking Okapi sets by *COMB-LCS* (method 2 – types)

Measure	TYPES x=0.25 win 20	TYPES x=0.25 win 40	TYPES x=0.5 win 20	TYPES x=0.5 Win 40	TYPES x=0.75 win 20	TYPES x=0.75 win 40	TYPES x=1 win 20	TYPES x=1 win 40	TYPES x=1.5 win 20	TYPES x=1.5 win 40
Interpolated recall – precision averages										
at 0.00	0.3992	0.3991	0.3994	0.3991	0.4002	0.3990	0.4001	0.4013	0.4013	0.4013
at 0.10	0.3013	0.3013	0.3009	0.3012	0.3009	0.3011	0.3009	0.3036	0.3002	0.3036
at 0.20	0.2049	0.2049	0.2052	0.2052	0.2056	0.2052	0.2055	0.2049	0.2055	0.2050
Average precision	0.1312	0.1311	0.1313	0.1312	0.1313	0.1312	0.1318	0.1320	0.1318	0.1314
Precision:										
At 5 docs	0.2136	0.2136	0.2136	0.2136	0.2136	0.2136	0.2136	0.2091	0.2136	0.2000
At 10 docs	0.1568	0.1523	0.1568	0.1523	0.1614	0.1545	0.1636	0.1545	0.1636	0.1545
At 15 docs	0.1394	0.1394	0.1394	0.1379	0.1394	0.1379	0.1394	0.1379	0.1394	0.1379
At 20 docs	0.1284	0.1284	0.1273	0.1284	0.1273	0.1284	0.1273	0.1284	0.1273	0.1284
R-Precision	0.0144	0.1497	0.1497	0.1497	0.1497	0.1497	0.1497	0.1495	0.1490	0.1495

Measure	TYPES x=3 win 20	TYPES x=3 win 40	TYPES x=30 win 20	TYPES x=30 win 40
Interpolated recall – precision averages				
at 0.00	0.4037	0.4015	0.4209	0.3756
at 0.10	0.2953	0.3039	0.2823	0.2648
at 0.20	0.2090	0.2043	0.1926	0.1875
Average precision	0.1316	0.1308	0.1241	0.1175
Precision:				
At 5 docs	0.2136	0.2000	0.1909	0.1909
At 10 docs	0.1614	0.1568	0.1545	0.1500
At 15 docs	0.1394	0.1364	0.1242	0.1318
At 20 docs	0.1250	0.1284	0.1182	0.1216
R-Precision	0.1500	0.1456	0.1421	0.1411

Appendix D

Programs

D.1 Selected Perl scripts for global collocation analysis

D.1.1 Script for extraction of global collocates from ft_96 corpus

This script extracts all collocates for each query term in the corpus using the windowing technique described in section *5.3*. The corpus is scanned separately for each query term. Each time the next occurrence of the query term in question is located, its preceding and following collocates are identified and written out as a collocation pair [*node term, collocate*]. The script also writes out the total size of all windows around a query term in each document it occurs. This is necessary for subsequent calculation of average window sizes used in our MI and Z formulae.

```
#! /bin/perl


$\ = "\n";
$| = 1;

$span = 101;

open (KEYS, "/home/bc560/PAIRS/keys.txt");
open (OUT, ">/home/bc560/PAIRS/COLS/collocates.col");
open (WIN, ">/home/bc560/PAIRS/WIN/windows.col");

while ($keyword = <KEYS>){
chop $keyword;
@list = ();

open (IN, "/home/bc560/PAIRS/parseall.col");

while ($line = <IN>){
chop $line;

if ($line =~ /<DOC>/){

##############################################
for ($k=0; $k<@list; $k++) {

if ($list[$k] eq $keyword) {

######### Extract preceding collocates ##########

@backwindow = ();
@forwindow = ();
$winkount = 0;

$block = 0;
for ($h=1; $h<$span; $h++) {
$addr = $k - $h;
if ($addr >= 0) {
  push @backwindow, $list[$k - $h];
```

```
 };
};

for ($d=0; $d<@backwindow; $d++) {
 if ($backwindow[$d] eq $list[$k]) {
      $block = 1;
 };
};

if ($block == 0) {
for ($d=0; $d<@backwindow; $d++) {
print OUT "$list[$k], $backwindow[$d]";
$winkount = $winkount + 1;
};
};

######### Extract following collocates ##########

$stop = 0;
 for ($i=1; $i<$span; $i++) {
  if ($list[$k + $i] ne "") {
   push @forwindow, $list[$k + $i];
  };
 };

for ($d=0; $d<@forwindow; $d++) {
 if($forwindow[$d] eq $list[$k]) {
 $stop = 1;
 } else {
  if($stop != 1) {
print OUT "$list[$k], $forwindow[$d]";
$winkount = $winkount + 1;
  };
 };
};
print WIN "$winkount $list[$k]";
 };
};
##########################################

@list = ();
}else{
push @list, $line;
};

};
close(IN);
};

close(KEYS);
close(OUT);
close(WIN);
```

D.1.2 Script for expanding queries with global collocates and searching ft_96 collection

This script expands initial queries with global collocates stored in Okapi collocation database. First, each initial query term is searched against the collocation database (here top 8 mi) using unweighted search. Then collocates, returned in the result of this search, are searched against ft_96 collection. Because in Perl during a single session with BSS, the write filehandle must be closed before the read filehandle is opened, it was not possible to get numbers of postings, calculate term weights and submit final weighted query all in one go. For this reason a separate BSS session had to be initiated, first, to get terms' numbers of postings, secondly, to get terms' weights, and finally, to submit the weighted query. The script outputs ranked document sets for all topics, which are then formatted using another script for input to trec_eval program.

```
#! /usr/bin/perl


$\ = "\n";
$| = 1;

open (IN, "parsed_titles.251-300");
open (OUT, ">mi-8");


while ($qline=<IN>){
chop $qline;
if ($qline !~ /<\d+>/){
push @qterms, $qline;
}else{

for ($q=0; $q<@qterms; $q++){


######## Open collocation database ########

$text = "";


open (BSS1, "| i1+ -silent > t1");

print BSS1 "ch top8-mi";
print BSS1 "f t=$qterms[$q]";
print BSS1 "s";

close (BSS1);

open (TEMP1, "t1");

for ($i=0; $i<6; $i++) {
$junk = <TEMP1>;
};

$top = <TEMP1>;
chop $top;
$top =~ s/^\s+3:\s+(.+)$/$1/;
$text = $top;

while ($in = <TEMP1>) {
```

```
chop $in;
$text = $text . " " . $in;
};

close(TEMP1);

push @colset, split / /, $text;
if ($qterms[$q] =~ /\d+/){
$qterm = "\@" . $qterms[$q];
push @colset, $qterm;
}else{
push @colset, $qterms[$q];
};
};

##### Open ft_96 database I ####################
##### to get terms' numbers of postings ########

open (BSS2, "| i1+ -silent > t2");


print BSS2 "ch ft_96";
for ($j=0; $j<@colset; $j++){
print BSS2 "find t=$colset[$j]";
};

close (BSS2);

open (TEMP2, "t2");
while ($in=<TEMP2>){
chop $in;
$in =~ s/^S\d+\snp=(\d+).+$/$1/;
push @nopos, $in;
};

close(TEMP2);

##### Open ft_96 database II ###################
##### to get terms' weights ####################

open (BSS3, "| i1+ -silent > t3");

print BSS3 "ch ft_96";

for ($k=0; $k<@nopos; $k++){
print BSS3 "w fn=0 n=$nopos[$k]";
};

close(BSS3);

open (TEMP3, "t3");
while ($in=<TEMP3>){
chop $in;
push @weights, $in;
};

close(TEMP3);

##### Open ft_96 database III ##################
##### to submit the final weighted query #########
```

```
open (BSS4, "| i1+ -silent > t4");

print BSS4 "ch ft_96";

for ($1=0; $1<@weights; $1++){
$query = $query . " s=" . $1 . " w=" . $weights[$1];
print BSS4 "f t=$colset[$1]";
};
print BSS4 "f $query op=bm2500 k1=1.2 bm25b=0.75";
print BSS4 "s f=197 n=1000";
close(BSS4);

open (TEMP4, "t4");
while ($in=<TEMP4>){
chop $in;
print OUT "<" . $qno . ">" . $in;
};

close(TEMP4);

$query = "";
@qterms = ();
@colset = ();
@nopos = ();
@weights = ();
$qno = $qline;
$qno =~ s/<(\d+)>/$1/;
 };
};
```

D.2 Selected Perl scripts for local collocation analysis

These scripts were written for the combined runs – the runs where the initial queries are expanded with top *N* local Z-ranked collocates of query terms and Okapi RF terms from the relevant documents. There are five scripts run consecutively:

merge-101.pl – extracts collocates of each query term from the windows of size 200 around their occurrences in the relevant documents; gets top 20 Okapi RF terms, ranked by Offer Weight;
merge-102.pl – calculates local Z score for each found collocate;
merge-103.pl – ranks collocates of each query term by local Z and gets 8 top-ranked collocates per query term;
merge-104.pl – merges into the final expanded query top 8 collocates per query term, 20 Okapi terms and the original query terms;
merge-105.pl – searches ft 96 database with the expanded queries using bm2500 weighted function with relevance information (the relevant documents used for query expansion). This script writes out a file with the document numbers of 1000 ranked documents per query, which are then submitted to trec_eval for evaluation.

Script **merge-101.pl**

```
#! /usr/bin/perl


$\ = "\n";
$| = 1;

$prevtopic = 251;
$bigr = 5;

open (IN, "/export/IS-F/Okapi/olga/LEXIC/5-ft-rel");

while ($line = <IN>){
chop $line;

($topic, $irn, $nbigr) = split / /, $line;


if ($topic == $prevtopic){

push @irnums, $irn;

}else{

### Open ft_96 database I

open (BSS1, "| i1+ -silent > t1");
print BSS1 "ch ft_96";

for ($a=0; $a<@irnums; $a++){
print BSS1 "find a=dn t=$irnums[$a]";
print BSS1 "extract set=$a record=0";
```

```
};
close (BSS1);

open (TEMP1, "t1");
$wcount = 0;
while ($term = <TEMP1>){
chop $term;

if ($term =~ /^\d+$/){
close (W);
open (W, "| sort -u > $prevtopic-$wcount");
close (COLW);
open (COLW, ">COL-$wcount");

$wcount++;
}else{

$term =~ s/t=//;
$term =~ s/\sc=/</;
$term =~ s/\ss=/</;

($stem, $gsl, $lexeme) = split /</, $term;

if ($gsl eq "G" || $gsl eq "N" || $gsl eq "S"){
print W $stem;
print COLW $stem;
  };
 };
};
close(W);
close(COLW);
close(TEMP1);
$wcount++;

$merge = `sort -m $prevtopic-* | uniq -c > $prevtopic-all`;

######### Extract collocates ########################################

undef %smlr;

open (WORDS, "$prevtopic-all");

while ($word=<WORDS>){
chop $word;

$word =~ s/^\s*//;

($smr, $trm) = split /\s+/, $word;

$smlr{$trm} = $smr;

};
close(WORDS);

$span = 101;

open (KEYS, "/export/IS-F/Okapi/olga/LEXIC/TOPICS/$prevtopic");
open (OUT, ">>/export/IS-F/Okapi/olga/LEXIC/COLS/$prevtopic");
open (WIN, ">>/export/IS-F/Okapi/olga/LEXIC/WINS/$prevtopic");
open (INDEX, ">>/export/IS-F/Okapi/olga/LEXIC/INDEX/$prevtopic");
```

```
while ($keyword = <KEYS>){
chop $keyword;

for ($b=0; $b<$wcount; $b++){

@list = ();

#-----------------------------------------

open (CIN, "COL-$b");

while ($cin = <CIN>){
chop $cin;
push @list, $cin;
};

close(CIN);

########################
for ($k=0; $k<@list; $k++) {

if ($list[$k] eq $keyword) {
print INDEX $list[$k];
## Extract preceding collocates

@backwindow = ();
@forwindow = ();
$winkount = 0;

$block = 0;
for ($h=1; $h<$span; $h++) {
$addr = $k - $h;
if ($addr >= 0) {
  push @backwindow, $list[$k - $h];
 };
};

for ($d=0; $d<@backwindow; $d++) {
 if ($backwindow[$d] eq $list[$k]) {
        $block = 1;
 };
};

if ($block == 0) {
for ($d=0; $d<@backwindow; $d++) {
print OUT "$list[$k],
$backwindow[$d]<small_r>$smlr{$backwindow[$d]}";

$winkount = $winkount + 1;
 };
};

## Extract following collocates

$stop = 0;
 for ($i=1; $i<$span; $i++) {
  if ($list[$k + $i] ne "") {
   push @forwindow, $list[$k + $i];
  };
 };
```

```
for ($d=0; $d<@forwindow; $d++) {
 if($forwindow[$d] eq $list[$k]) {
 $stop = 1;
 } else {
  if($stop != 1) {
print OUT "$list[$k], $forwindow[$d]<small_r>$smlr{$forwindow[$d]}";

$winkount = $winkount + 1;
  };
 };
};
print WIN "$winkount $list[$k]";
 };
};
####################

@list = ();
 };
};
close(KEYS);
close(OUT);
close(WIN);
close(INDEX);


######### Get Okapi RF terms ###################################

## Get OW of terms

open (OUT, ">$prevtopic-OW");
open (WORDS, "$prevtopic-all");

while ($word=<WORDS>){
chop $word;

$word =~ s/^\s*//;

($smr, $trm) = split /\s+/, $word;

push @smallr, $smr;
push @term, $trm;

};

##open BSS1

open (BSS1, "| i1+ -silent > t1");
print BSS1 "ch ft_96";

for ($u=0; $u<@term; $u++){

print BSS1 "f t=$term[$u]";

};
close (BSS1);

open (TEMP1, "t1");
while ($in=<TEMP1>){
chop $in;
$np = $in;
$np =~ s/^S\d+\snp=(\d+).+$/$1/;
```

```
push @nopos, $np;
};
close (TEMP1);

##open BSS2

open (BSS2, "| i1+ -silent > t2");
print BSS2 "ch ft_96";

for ($y=0; $y<@term; $y++){

print BSS2 "weight n=$nopos[$y] big_r=$bigr r=$smallr[$y]";
};

close (BSS2);

open (TEMP2, "t2");
while ($in=<TEMP2>){
chop $in;

push @weight, $in;
};
close(TEMP2);

for ($i=0; $i<@term; $i++){

$OW = $weight[$i] * $smallr[$i];

print OUT "$OW $term[$i]";

};
close(OUT);
close(WORDS);

$sort = `sort -n -r $prevtopic-OW -o $prevtopic-OW-s`;

open (TOPOUT, ">/export/IS-F/Okapi/olga/LEXIC/OK-TERMS/$prevtopic-
qw");
open (TOPIN, "$prevtopic-OW-s");

for ($q=0; $q<20; $q++){
$topqw = <TOPIN>;
chop $topqw;
($ownew, $tmnew) = split / /, $topqw;

print TOPOUT "$tmnew $smlr{$tmnew}";

};
close(TOPOUT);
close(TOPIN);

$rm = `rm $prevtopic-*`;
$rm = `rm COL-*`;

@term = ();
@smallr = ();
@nopos = ();
@weight = ();
@irnums = ();
push @irnums, $irn;
```

```
};
$prevtopic = $topic;
$bigr = $nbigr;
};
close(IN);
```

Script **merge-102.pl**

```
#! /usr/bin/perl


$\ = "\n";
$| = 1;

#################### Get JF from colloc.titles file

for ($a=251; $a<301; $a++){

$sort = 'sort /export/IS-F/Okapi/olga/LEXIC/COLS/$a | uniq -c >
/export/IS-F/Okapi/olga/LEXIC/COLS/jf-$a';
$uniq = 'sort /export/IS-F/Okapi/olga/LEXIC/INDEX/$a | uniq -c >
/export/IS-F/Okapi/olga/LEXIC/INDEX/u-$a';

######## Calculate average window sizes

open (WIN, "/export/IS-F/Okapi/olga/LEXIC/WINS/$a");
open (WOUT, ">/export/IS-F/Okapi/olga/LEXIC/WINS/avg-$a");

while ($win = <WIN>){
chop $win;
($wsize, $wterm) = split / /, $win;

$winsize{$wterm} = $winsize{$wterm} + $wsize;
$num{$wterm} = $num{$wterm} + 1;
};

foreach $key (sort keys %winsize){
$average = $winsize{$key} / $num{$key};
print WOUT "$key $average";
};

undef %winsize;

close(WIN);
close(WOUT);
};

################### Calculate local Z score ######################

$corpus = 43279064;

open (GLOB, "/export/IS-F/Okapi/olga/LEXIC/quantindex");

while ($glob = <GLOB>){
chop $glob;
$glob =~ s/^\s*(.+)$/$1/;

($globf, $gterm) = split / /, $glob;

$globfreq{$gterm} = $globf;
```

```
};

close(GLOB);

for ($i=251; $i<301; $i++){
open (SCORES, ">/export/IS-F/Okapi/olga/LEXIC/COLS/z-$i");
open (COLLOCS, "/export/IS-F/Okapi/olga/LEXIC/COLS/jf-$i");
open (INDEX, "/export/IS-F/Okapi/olga/LEXIC/INDEX/u-$i");
open (WIN, "/export/IS-F/Okapi/olga/LEXIC/WINS/avg-$i");

while ($ind = <INDEX>){
chop $ind;
$ind =~ s/^\s*(.+)$/$1/;

($indf, $term) = split / /, $ind;

$freq{$term} = $indf;
};

while ($win = <WIN>){
chop $win;
($winterm, $winsize) = split / /, $win;

$window{$winterm} = $winsize;
};

while ($line = <COLLOCS>){
chop $line;

$line =~ s/^\s*(.+)$/$1/;

($lead, $end) = split /, /, $line;
($y, $smallr) = split /<small_r>/, $end;
($joinfreq, $x) = split / /, $lead;

if ($globfreq{$y} >= 1){


############   Calculate local Z score

$localz = ($joinfreq - $globfreq{$y}*$freq{$x}*$window{$x}/
$corpus)/sqrt( $globfreq{$y}*$freq{$x}*$window{$x}/$corpus);

$localz =~ s/^(.......).+$/$1/;

print SCORES "$x $y $localz $smallr";
  };
 };
};
close (SCORES);
close (COLLOCS);
close (INDEX);
close (WIN);
```

Script **merge-103.pl**

```
#! /usr/bin/perl


$\ = "\n";
$| = 1;

$qsize = 8;

$topic = 250;

open (KEYS, "/export/IS-F/Okapi/olga/LEXIC/title_keys");
while ($key=<KEYS>){
chop $key;

if ($key !~ /<topic>/){

push @qterms, $key;

}else{

open (OUT, ">/export/IS-F/Okapi/olga/LEXIC/COLS/$topic-top8");
for ($a=0; $a<@qterms; $a++){

$f = `egrep '^$qterms[$a] ' /export/IS-F/Okapi/olga/LEXIC/COLS/z-
$topic |sort -r -n - k 3 > /export/IS-F/Okapi/olga/LEXIC/TEMP/
$qterms[$a]`;

open (IN, "/export/IS-F/Okapi/olga/LEXIC/TEMP/$qterms[$a]");

for ($b=0; $b<$qsize; $b++){
$in = <IN>;
chop $in;

($x, $y, $localz, $smallr) = split / /, $in;

print OUT "$y $smallr";

};

close(IN);

};
close(OUT);


$topic = $key;
$topic =~ s/^<topic>(\d+).+$/$1/;

@qterms = ();
};
};
close(KEYS);

######################### Remove empty lines

for ($i=251; $i<301; $i++){

open (RIN, "/export/IS-F/Okapi/olga/LEXIC/COLS/$i-top8");
open (ROUT, ">/export/IS-F/Okapi/olga/LEXIC/COLS/s-$i");
```

```
while ($line = <RIN>){
chop $line;
if ($line =~ /\w/){
print ROUT $line;
};
};
close(RIN);
close(ROUT);
};
```

Script **merge-104.pl**

```
#! /usr/bin/perl


$\ = "\n";
$| = 1;

for ($a=251; $a<301; $a++){

open (COLS, "/export/IS-F/Okapi/olga/LEXIC/COLS/$a-top");
open (OK, "/export/IS-F/Okapi/olga/LEXIC/OK/ok-$a");
open (OUT, ">/export/IS-F/Okapi/olga/LEXIC/MERGED-Q/query-$a");
open (TERMS, "/export/IS-F/Okapi/olga/LEXIC/TOPICS-SMALLR/$a");

############ Write out collocates

while ($collocate=<COLS>){
chop $collocate;
print OUT $collocate;
};

############ Write out Okapi RF terms

while ($ok=<OK>){
chop $ok;
print OUT $ok;
};

########### Write out original query terms

while ($term=<TERMS>){
chop $term;
print OUT $term;
};

###########

close(COLS);
close(OK);
close(OUT);
close(TERMS);
};

########## Remove duplicate lines

for ($b=251; $b<301; $b++){
```

```
$sort = 'sort /export/IS-F/Okapi/olga/LEXIC/MERGED-Q/query-$b | uniq
> /export/IS-F/Okapi/olga/LEXIC/MERGED-Q/qr-$b';

$rm = 'rm /export/IS-F/Okapi/olga/LEXIC/MERGED-Q/query-$b';

};
```

Script **merge-105.pl**

```
#! /usr/bin/perl


$\ = "\n";
$| = 1;


open (OUT, ">/export/IS-F/Okapi/olga/LEXIC/RETRO-RESULTS/Z/z+ok20-
w200");

###### Get big_r ######

open (BIGR, "/export/IS-F/Okapi/olga/LEXIC/topics-bigr");
while ($br=<BIGR>){
chop $br;
($top, $big) = split /  /, $br;
$bigr{$top} = $big;
};
close(BIGR);

########################

for ($qno=251; $qno<301; $qno++){
open (IN, "/export/IS-F/Okapi/olga/LEXIC/MERGED-Q/qr-$qno");

@terms = ();
@smallr = ();
@nopos = ();
@weights = ();
$query = "";

while ($line = <IN>){
chop $line;

($te, $sr) = split / /, $line;

push @terms, $te;
push @smallr, $sr;

};

### Open ft_96 database I

open (BSS1, "| i1+ -silent > t1");

print BSS1 "ch ft_96";
for ($j=0; $j<@terms; $j++){
print BSS1 "find t=$terms[$j]";
};

close (BSS1);
```

```
open (TEMP1, "t1");
while ($in=<TEMP1>){
chop $in;
$in =~ s/^S\d+\snp=(\d+).*$/$1/;
push @nopos, $in;
};

close(TEMP1);

### Open ft_96 database II

open (BSS2, "| i1+ -silent > t2");

print BSS2 "ch ft_96";
for ($k=0; $k<@nopos; $k++){
print BSS2 "weight n=$nopos[$k] big_r=$bigr{$qno} r=$smallr[$k]";
};

close(BSS2);

open (TEMP2, "t2");
while ($in=<TEMP2>){
chop $in;
push @weights, $in;
};

### Open ft_96 database III

open (BSS3, "| i1+ -silent > t3");

print BSS3 "ch ft_96";

for ($l=0; $l<@weights; $l++){
$query = $query . " s=" . $l . " w=" . $weights[$l];
print BSS3 "f t=$terms[$l]";
};
print BSS3 "f $query op=bm2500 k1=2 bm25b=0.75";
print BSS3 "s f=197 n=1000";

close(BSS3);

open (TEMP3, "t3");
while ($in=<TEMP3>){
chop $in;
print OUT "<" . $qno . ">" . $in;
};

close(TEMP3);
close(IN);
};
close(OUT);
```

D3 GSL codes

@0003	aeroplane, airplane, aircraft,
@0007	agricultural, agriculture,
@0008	air lines, airlines,
@0009	airport, aerodrome,
@0013	aluminium, aluminum,
@0014	american, america,
@0018	anti smoking, no smoking, no-smoking, nonsmoking, non smoking, antismoking,
@0020	argentina, argentinian,
@0035	bosnian, bosnia,
@0041	bt, british telecom,
@0043	built in, builtin,
@0045	buses, omnibus, bus,
@0049	cal, cai, computer aided instruction, computer aided learning, computer assisted learning,
@0060	china, chinese,
@0064	cleanup, clean up,
@0067	co generation, cogeneration,
@0068	co, company,
@0069	community charge, poll tax,
@0071	computation, computational,
@0077	croatia, croatian,
@0080	curriculum, curricula, curricular,
@0099	eec, european economic community, eu, european union, ec,
@0104	environment, environmental,
@0111	extra-terrestrial, extraterrestrial, extra terrestrial,
@0114	falklands, falkland islands, malvinas,
@0121	financial, finance,
@0128	france, french, francais,
@0129	freudian, freud, sigmund freud,
@0131	fuel-cell, fuel cell,
@0134	gases, gassing, gaseous, gas,
@0135	gatt, general agreement on tariffs and trade,
@0136	gdp, gross domestic product,
@0139	geological, geology,
@0147	gov, govt, governmental, government,
@0176	india, indian,
@0180	industrial, industry,
@0186	iraq, iraqi,
@0190	italy, italian, italia, italiana, italiano,
@0192	japan, japanese,
@0195	jewish, judaism, jew,
@0199	keynes, keynesian,
@0232	marx, marxian, marxist,
@0233	math, maths, mathematics,
@0256	nasa, national aeronautical space agency,
@0257	nato, north atlantic treaty organization,
@0290	per cent, percent,
@0295	philosophical, philosophy,
@0296	phone, telephone,
@0314	psbr, public sector borrowing requirement,
@0319	rail road, railroad, railway,
@0322	religious, religion,
@0330	russia, russian, soviet, soviet union, ussr, union of soviet socialist republics,
@0339	serbia, serbian,
@0352	south african, south africa,
@0373	takeover, take over, take-over,
@0380	third world, 3rd world, under developed countries, underdeveloped areas, underdeveloped countries, under-developed countries, developing countries,
@0399	uruguay, uruguayan,
@0400	usa, us of a, us, united states, united states of america,
@0404	uv, ultra violet, ultraviolet,
@0431	yugoslavia, yugoslav, yugoslavian,

References

Allan2000

Allan J., Callan J., Feng F. and Malin D. INQUERY and TREC-8. *In The Eighth Text REtrieval Conference (TREC-8).* Ed. E.M. Voorhees and D.K. Harman. Gaithersburg, MD: NIST, 2000.

Apresjan74

Apresjan, J.D. Lexicheskaja semantika: synonimicheskie sredstva jazyka. Moskva: Nauka, 1974.

Beeferman97

Beeferman D., Berger A., Lafferty J. A model of lexical attraction and repulsion. *In Proc. ACL-EACL Joint Conference*, Madrid, Spain, 1997.

Belkin2000

Belkin N., Cool C., Head J., Jeng J., Kelly D and Lin S. Relevance Feedback versus Local Context Analysis as Term Suggestion Devices: Rutgers' TREC-8 Interactive Track Experience. *In The Eighth Text REtrieval Conference (TREC-8).* Ed. E.M. Voorhees and D.K. Harman. Gaithersburg, MD: NIST, 2000.

Brown83 G.

Brown, G.Yule *Discourse analysis.* Cambridge University Press, 1983.

Buckley95

Buckley C., Salton G., Allan J., Singhal A. Automatic Query Expansion Using SMART – TREC 3. *In Overview of the Third Text REtrieval Conference (TREC-3).* Ed. D.K. Harman. Gaithersburg, MD: NIST, 1995.

Buckley97

Buckley C., Singhal A. and Mitra M. Using Query Zoning and Correlation Within SMART: TREC 5. *In The Fifth Text REtrieval Conference (TREC-5).* Ed. E.M. Voorhees and D.K. Harman. Gaithersburg, MD: NIST, 1997.

Buckley2000

Buckley C., Waltz J. SMART in TREC 8. *In The Eighth Text REtrieval Conference (TREC-8).* Ed. E.M. Voorhees and D.K. Harman. Gaithersburg, MD: NIST, 2000.

Callan94

Callan J.P. Passage-level evidence in document retrieval. *In Proc. ACM-SIGIR*, 1994, pp.302-309.

Chowdhury99

Chowdhury G.G., Gobinda G. *Introduction to modern information retrieval.* London: Library Association Publishing, 1999.

Church90

Church K., Hanks P. Word association norms, mutual information and lexicography. *In American Journal of Computational Linguistics*, 16(1), 1990, pp. 22-29.

Church91

Church K., Gale W., Hanks P., Hindle D. Using statistics in lexical analysis. *In Lexical Acquisition: Using On-line Resources to Build a Lexicon*, ed. U.Zernik, Englewood Cliffs, NJ: Lawrence Elbraum Associates, 1991, pp. 115-164.

Church94

Church K., Gale W., Hanks P., Hindle D. Lexical substitutability. *In Computational Approaches to the Lexicon* by B.T.S. Atkins and A. Zampoli, Oxford University Press, 1994, pp. 153-177.

Clear99

Clear J. "Re: Corpora: T-score in collocational analysis" *jem@cobuild.collins.co.uk* Corpora mailing list (12 Dec 1999).

CobuildDirect

CobuildDirect, Collins Cobuild, http://titania.cobuild.collins.co.uk/direct_info.html (Accessed: 6 February, 2001).

Cooper90

Cooper W.S. Some inconsistencies and misnomers in probabilistic information retrieval. *In Proc. ACM-SIGIR*, 1991, pp.57-61.

Cormack2000

Cormack G.V., Clarke C.L.A., Palmer C.R. and Kisman D.I.E. Fast Automatic Passage Ranking (MultiText Experiments for TREC-8). *In The Eighth Text REtrieval Conference (TREC-8)*. Ed. E.M. Voorhees and D.K. Harman. Gaithersburg, MD: NIST, 2000.

Croft79

Croft W.B. and Harper D.J. Using probabilistic models of document retrieval without relevance information. *In Journal of Documentation* 35 (4), 1979, pp. 285-295.

Culicover76

P. W. Culicover *Syntax.* Academic Press, 1976.

Dijk77

T.A. Van Dijk *Text and Context.* Longman, 1977.

Edmonds97

Edmonds P. Choosing the word most typical in context using a lexical co-occurrence network. *In Proc. 35th Annual Meeting of ACL*, Madrid, Spain, 1997, pp.507-509.

Ellman2000

Ellman, J. and Tait, J. On the generality of thesaurally derived lexical links. *In the Proceedings of 5th JADT*, 2000, pp.147-154.

Evens88

Evens M.W. (ed.) Relational models of the lexicon: representing knowledge in semantic networks. Cambridge University Press, 1988.

Fagan89

Fagan J.L. The effectiveness of a non-syntactic approach to automatic phrase indexing for document retrieval. *In Journal of the American Society for Information Science*, 40(2), 1989, pp. 115-132.

Fano61

Fano R. *Transmission of information*, Cambridge, Mass., MIT Press, 1961.

Finegan94

E. Finegan *Language. Its structure and use.* Harcourt Brace College Publishers, 1994.

Firth58

Firth L.R. Modes of Meaning in *Papers in Linguistics 1934-51.* London. Oxford University Press, 1958, pp. 190-215.

Foskett96

Foskett A.C. The subject approach to information. Library Association Publishing, London, 1996.

Fuller2000

Fuller M., Kaszkiel M., Kimberley S., Ng C., Wilkinson R., Wu M., Zobel J. The RMIT/CSIRO Ad Hoc, Q&A, Web, Interactive, and Speech Experiments at TREC 8. *In The Eighth Text REtrieval Conference (TREC-8).* Ed. E.M. Voorhees and D.K. Harman. Gaithersburg, MD: NIST, 2000.

Haas94

Haas S. W., Losee R. M. Looking in text windows: their size and composition. *In Information Processing and Management*, Vol. 30, No. 5, 1994, pp. 619-629.

Halliday76

M.A.K.Halliday, R.Hasan *Cohesion in English.* Longman, 1976.

Han93

Han Y.S. and Choi K.S. Lexical concept acquisition from collocation map. In SIGLEX Workshop on Acquisition of Lexical Knowledge from Text, 31st Annual Meeting of the ACL, 1993, pp. 22-31.

Harper78

Harper D. J., Van Rijsbergen C. J. "An evaluation of feedback in document retrieval using co-occurrence data" *In Journal of Documentation*, Vol.34, No. 3, September 1978, pp. 189-216.

Hasan84

R. Hasan Coherence and cohesive harmony. *In Flood, J. (ed.) Understanding Reading Comprehension.* 1984. pp.181-219. Delaware: International Reading Association.

Hawking96

Hawking D. and Thistlewaite P. Relevance weighting using distance between term occurrences. *Technical Report* TR-CS-96-08, 1996, Department of Computer Science, Australian National University. http://cs.anu.edu.au/techreports/1996/index.html (Accessed: Jan 2001).

Hawking98

Hawking D., Thistlewaite P., Craswell N. ANU/ACSys TREC-6 Experiments. *In The Sixth Text REtrieval Conference (TREC-6).* Ed. E.M. Voorhees and D.K. Harman. Gaithersburg, MD: NIST, 1998.

Hearst93

Hearst M., Plaunt C. Subtopic structuring for full-length document access. *In Proc. ACM-SIGIR*, 1993, pp. 59-68.

Hearst94

Hearst, M. Multi-paragraph segmentation of expository text. In the Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics. 1994.

Hirst97

Hirst, G. and St-Onge, D. Lexical chains as representation of context for the detection and correction of malapropisms. *Wordnet. An Electronic Lexical Database.* C.Fellbaum (ed.), MIT Press, 1997, pp.305-332.

Hoey91

M. Hoey *Patterns of Lexis in Text.* Oxford University Press; 1991.

Ishikawa98

Ishikawa K., Satoh K., Okumura A. Query Term Expansion based on Paragraphs of the Relevant Documents. *In The Sixth Text REtrieval Conference (TREC-6).* Ed. E.M. Voorhees and D.K. Harman. Gaithersburg, MD: NIST, 1998.

Jelinek90

Jelinek F. Self-organised language modelling for speech recognition. *In Readings in Speech Recognition*, ed. A. Waibel and K. Lee, San Mateo, California, Morgan Kaufmann Publishers, 1990.

Jing94

Jing Y. and Croft B. An association thesaurus for information retrieval. *In Proc. ACM-SIGIR Conference*, Seattle, ACM Press, 1994, pp.37-50.

Kaszkiel97

Kaszkiel M. and Zobel J. Passage retrieval revisited. *In Proc. ACM-SIGIR*, 1997, pp. 178-185.

Knaus95

Knaus D., Mittendorf E., Schauble P. Improving a basic retrieval method by links and passage level evidence. *In Overview of the Third Text REtrieval Conference (TREC-3).* Ed. D.K. Harman. Gaithersburg, MD: NIST, 1995.

Kucera67

Kucera H., Francis W. *Computational analysis of present-day American English.* Brown University Press, 1967.

Kwok98

Kwok K. and Chan M. Improving Two-Stage Ad-Hoc Retrieval for Short Queries. *In Proc. ACM-SIGIR*, 1998, pp. 250-256.

Kwok2000

Kwok K., Grunfeld L. and Chan M. TREC-8 Ad-Hoc, Query and Filtering Track experiments using PIRCS. *In The Eighth Text REtrieval Conference (TREC-8).* Ed. E.M. Voorhees and D.K. Harman. Gaithersburg, MD: NIST, 2000.

Lancaster79

Lancaster F. W. Information retrieval systems: characteristics, testing and evaluation. Wiley, 1979.

Levkovskaja62

Levkovskaja K.A. Teorija slova, principi eje postrojenija i aspekti izuchenija leksicheskogo materiala. Moskva: Prosveshenije, 1962.

Lewis91

Lewis D.D. Representation and learning in information retrieval. PhD Thesis, Department of Computer and Information Science, University of Massachusetts at Amherst, TR 91-93, 1991.

Lewis96

Lewis D.D., Sparck Jones K. Natural language processing for information retrieval. *In Communications of the ACM, 39*, 1996, pp. 92-101.

Losee94

Losee R. M. Term dependence: truncating the Bahadur Lazarsfeld expansion. *In Information Processing and Management*, Vol. 30, No. 2, 1994, pp. 293-303.

Luhn58

Luhn H. P. The automatic creation of literature abstracts. *In IBM Journal of Research and Development*, 2(2), April 1958, pp. 159-165.

Lyons77

J. Lyons *Semantics.* Cambridge University Press, 1977.

Manabu2000

Manabu O., Hajime M. Query-biased summarization based on lexical chaining. *In Computational Intelligence*, Vol. 16, N 4, 2000, pp. 578-585.

Manning99

Manning C.D., Schuetze H. *Foundations of Statistical Natural Language Processing*, The MIT Press, Cambridge, Massachusetts, 1999.

Maron60

Maron M.E. and Kuhns J.L. On relevance, probabilistic indexing and information retrieval. *In Journal of the ACM*, 7, pp. 216-244.

Mason97

Mason O. The weight of words: an investigation of lexical gravity. *In International Conference on Practical Applications in Language Corpora*, Lodz, Poland, 10-14 April 1997, pp. 361-375.

McDonald97

McDonald J., Ogden W., Foltz P. Interactive information retrieval using term relationship networks. *In the Proc. Sixth Text Retrieval Conference (TREC-6)*, Gaithersburg, MD, USA, 1997.

McEnery96

McEnery T. and Wilson A. *Corpus Linguistics*, Edinbourgh, 1996.

Miller90

Miller G.A. et al. Five Papers on WordNet. Princeton University, Cognitive Science Laboratory Technical Report N 43, July 1990.

Mittendorf2000

Mittendorf E., Mateev B., Schauble P. Using co-occurrence of words for retrieval weighting. *In Information Retrieval*, 3, 2000, pp. 243-251.

Mittendorf94

Mittendorf E. and Schauble P. Document and passage retrieval based on Hidden Markov Models. *In ACM-SIGIR*, 1994, pp. 318-327.

Morris91

Morris, J. and Hirst, G. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics,* 17, 1991, pp. 21-48.

Namba2000

Namba I. and Igata N. Fujitsu Laboratories TREC 8 Report. Ad Hoc, Small Web, and Large Web Track. *In The Eighth Text REtrieval Conference (TREC-8).* Ed. E.M. Voorhees and D.K. Harman. Gaithersburg, MD: NIST, 2000.

Novikov83

Novikov A.I. Semantika teksta i eje formalizacija. Moskva, 1983.

O'Grady97

W. O'Grady, M. Dobrovolsky, F. Katamba *Contemporary Linguistics. An Introduction.* Longman, 1997.

Paijmans93

Paijmans H. Comparing the document representations of two IR-systems: CLARIT and TOPIC. *In Journal of the American Society for Information Science*, 44(7), pp. 383-392, 1993.

Palmer81

Palmer F.R. *Semantics*. Second edition. Cambridge University Press, 1981.

Park96

Park Y.C. and Choi K.S. Automatic thesaurus construction using Bayesian networks. *In Information Processing and Management*, 32(5), pp. 543-554.

Peat91

Peat H.J. and Willet P. The limitations of term cooccurrence data for query expansion in document retrieval systems. *In Journal of the American Society for Information Science*. Vol. 42, No.5, 1991, pp. 378-383.

Porter80

Porter M.F. An algorithm for suffix stripping. *In Program*, 14(3), 1980, pp. 130-137.

Qiu93

Qiu Y. and Frei H.P. Concept based query expansion. *In Proceedings of ACM SIGIR*, 1993, pp. 160-169.

Renouf93

Renouf A. What the linguist has to say to the information scientist. *In Journal of Document and Text Management*, Vol.1, No.2., 1993, pp. 173-190.

Robertson76

Robertson S.E., Sparck Jones K. Relevance Weighting of Search Terms. *In Journal of the American Society for Information Science*, 27, 1976, pp. 129-146.

Robertson77

Robertson S.E. The probability ranking principle in IR. *In Journal of Documentation*, 33, 1977, pp. 294-304.

Robertson82

Robertson S.E., Maron M.E. and Cooper W.S. Probability of relevance: A unification of two competing models for document retrieval. *In Information Technology: Research and Development*, 1, 1982, pp. 1-21.

Robertson90

Robertson S.E. On term selection for query expansion. *In Journal of Documentation*, 46(4), pp. 359-364.

Robertson94

Robertson S.E. and Walker S. Some simple effective approximations to the 2 Poisson model for probabilistic weighted retrieval. *In Proc. ACM-SIGIR*, 1994, pp.232-241.

Robertson95

Robertson S.E., Walker S., Jones S., Hancock-Beaulieu M.M. Okapi at TREC-3. *In Overview of the Third Text REtrieval Conference (TREC-3)*. Ed. D.K. Harman. Gaithersburg, MD: NIST, 1995, pp. 109-126.

Robertson99

Robertson S.E., Walker S., Beaulieu M. Okapi at TREC-7: automatic, ad hoc, filtering, VLC and interactive track. *In The Seventh Text REtrieval Conference (TREC-7).* Ed. E.M. Voorhees and D.K. Harman. Gaithersburg, MD: NIST, 1999, pp. 253-264.

Robins89

R. H. Robins *General Linguistics.* Longman, 1989.

Rocchio65

Rocchio J.J. Relevance feedback in information retrieval. *In Scientific Report ISR-9,* Harvard University, 1965. Reprinted as Chapter 14 in *The SMART retrieval system* (Ed. G.Salton). Englewood Cliffs, NJ: Prentice-Hall, 1971.

Salton93

Salton G., Allan J., Buckley C. Approaches to passage retrieval in full text information systems. *In Proc. ACM-SIGIR,* 1993, pp. 49-58.

Sanfilippo98

Sanfilippo, A. Ranking text units according to textual saliency. *In the Proceedings of COLING-ACL,* 1998, pp.1157-1163.

Saussure16

F. De Saussure *Cours de linguistique generale* (1916). English translation: *A course in general linguistics.* Fontana/Collins, 1978.

Sinclair74

Sinclair J. M., Jones S. "English lexical collocations: A study in computational linguistics.", 1974. Reprinted as chapter 2 of Foley J.A. (ed.), *"J.M. Sinclair on Lexis and Lexicography."* Singapore: UniPress, 1996.

Sinclair91

Sinclair J. M. *Corpus, concordance, collocation.* Oxford University Press, 1991.

Singhal2000

Singhal A., Abney S., Bacchiani M., Collins M., Hindle D., Pereira F. AT&T at TREC-8. *In The Eighth Text REtrieval Conference (TREC-8).* Ed. E.M. Voorhees and D.K. Harman. Gaithersburg, MD: NIST, 2000.

Skorochod'ko72

Skorochod'ko E.F. Adaptive method of automatic abstracting and indexing. *In the Proceedings of the IFIP Congress 71. Information Processing 71,* C.V. Freiman (ed.), North-Holland Publishing Company, 1972, pp. 1179-1182.

Smeaton83

Smeaton A.F. and Van Rijsbergen C.J. The retrieval effects of query expansion on a feedback document retrieval system. *In The Computer Journal,* Vol. 26 No. 3, 1983, pp. 239-246.

Smeaton95

Smeaton A.F. Low level language processing for large scale information retrieval: what techniques actually work. *In Proceedings of a workshop "Terminology, information retrieval and linguistics", organised by CNR*, Rome, 10 Oct. 1995.

Smeaton97

Smeaton A.F. Information retrieval: still butting heads with natural language processing? *In "Information Extraction: a Multidisciplinary Approach to Emerging Information Technology"*. Ed. M.T. Pazienza, Springer-Verlag Lecture Notes in Computer Science N 1299, 1997, pp. 115-138.

Sparck Jones68

Sparck Jones K. and Needham R.M. Automatic term classification and retrieval. *In IP&M*, Vol. 4, 1968, pp. 91-100.

Sparck Jones70

Sparck Jones K. and Jackson D.M. The use of automatically obtained keyword classifications for information retrieval. *In IP&M*, Vol. 5, 1970, pp.175-201.

Sparck Jones71(a)

Sparck Jones K. Automatic keyword classification for information retrieval. London: Butterworths, 1971.

Sparck Jones71(b)

Sparck Jones K. A statistical interpretation of term specificity and its application in retrieval. *In Journal of Documentation*, 28, 1971, pp.11-21.

Sparck Jones73

Sparck Jones K. Does indexing exhaustivity matter? *In Journal of the American Society for Information Science*, 24, 1973, pp. 313-316.

Sparck Jones84

Sparck Jones K. and Tait J.I. Automatic search term variant generation. *In Journal of Documentation*, 40, 1984, pp.50-66.

Sparck Jones98

Sparck Jones K., Walker S., Robertson S. A Probabilistic Model of Information Retrieval: Development and Status. University of Cambridge Computer Laboratory Technical Report N 446, 1998.

Sparck Jones99

Sparck Jones K. What is the role of NLP in text retrieval? *In "Natural language information retrieval" ed. Strzalkowski, T.* Kluwer Academic Publishers, 1999, pp. 1-24.

Stairmand97

Stairmand M.A. Textual context analysis for information retrieval. *In Proc. ACM-SIGIR*, 1997, pp. 140-147.

Strzalkowski99(a)

Strzalkowski T. Perez-Carballo J. Evaluating natural language processing techniques in information retrieval. *In "Natural language information retrieval" ed. Strzalkowski, T.* Kluwer Academic Publishers, 1999, pp. 113-145.

Strzalkowski99(b)

Strzalkowski T. et al. Natural language information retrieval: TREC-7 Report. *In The Seventh Text REtrieval Conference (TREC-7).* Ed. E.M.Voorhees and D.K.Harman. Gaithersburg, MD: NIST, 1999.

Strzalkowski2000

Strzalkowski T., Perez-Caballo J., Karlgren J., Hulth A., Tapanainen P., Lahtinen T. Natural Language Information Retrieval: TREC-8 Report. *In The Eighth Text REtrieval Conference (TREC-8).* Ed. E.M. Voorhees and D.K. Harman. Gaithersburg, MD: NIST, 2000.

Tombros98

Tombros A. and Sanderson M. Advantages of query biased summaries in information retrieval. *In Proc. 21st ACM-SIGIR*, 1998, pp. 2-10.

Van Rijsbergen77

Van Rijsbergen C. J. A theoretical basis for the use of co-occurrence data in information retrieval. *In Journal of Documentation*, Vol. 33, No. 2, June 1977, pp. 106-119.

Van Rijsbergen79

Van Rijsbergen C.J. *Information Retrieval.* 2nd edition, London: Butterworth, 1979.

Vechtomova2000

Vechtomova O. and Robertson S. Integration of collocation statistics into the probabilistic retrieval model. *In Proc. 22nd BCS-IRSG*, 2000, pp. 165-177.

Wilkinson94

Wilkinson R. Effective retrieval of structured documents. *In Proc. ACM-SIGIR*, 1994, pp. 311-317.

Wilson96

Wilson A., McEnery T. *A Corpus Linguistics.* Edinbourgh, 1996.

Xu96

Xu J. and Croft B. Query expansion using local and global document analysis. *In Proc. 19th International Conference on Research and Development in Information Retrieval (SIGIR '96),* Zurich, Switzerland, 1996, pp. 4-11.

Yochum96

Yochum J.A. Reserch in Automatic Profile Generation and Passage-Level Routing with LMDS. *In The Fourth Text REtrieval Conference (TREC-4)*, Ed. D.K. Harman, Gaithersburg, MD: NIST, 1996.