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).

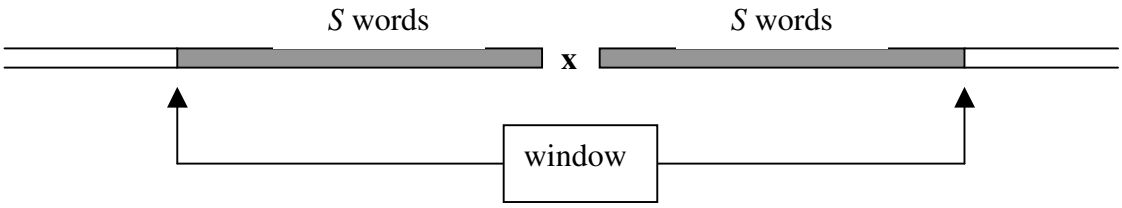

Figure 5.1. Window around node *x*, defined as spans of *S* words to the left/right of *x*

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).

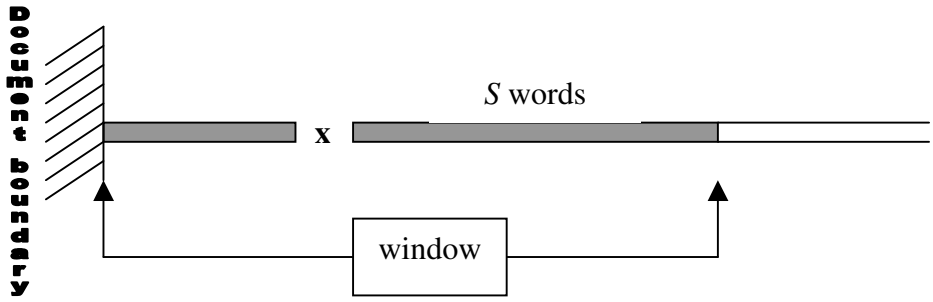

Figure 5.2. Window truncated by hitting the document boundary

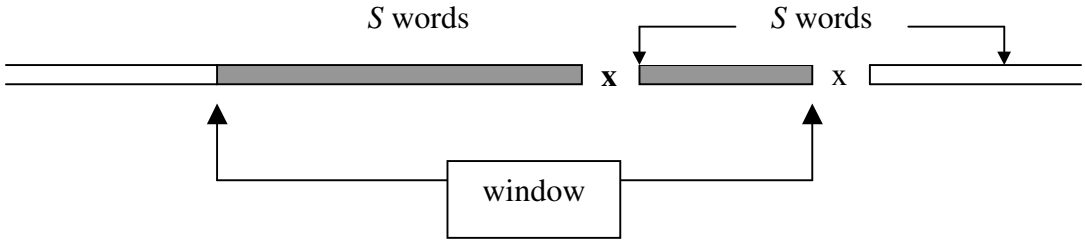

Figure 5.3. Right-hand half of the window truncated by hitting another occurrence of x after the node

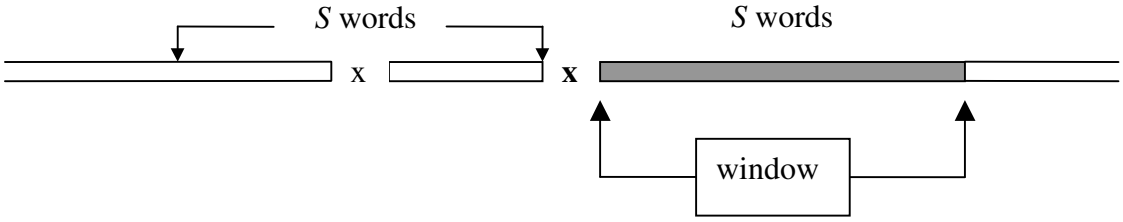

Figure 5.4. Left-hand half of the window ignored when another occurrence of *x* is found before the node

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.