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.

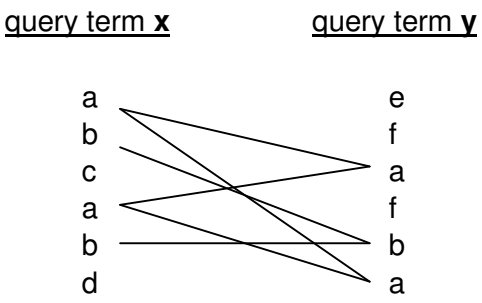

Figure 8.1. Links between instances of common collocates in merged windows of query terms **x** and **y**

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.