A Graph Based Approach to Estimating Lexical Cohesion

Hayrettin Gürkök, Murat Karamuftuoglu, Markus Schaal
Department of Computer Engineering
Bilkent University
06800, Ankara, Turkey
gurkok, hmk, schaal@cs.bilkent.edu.tr

ABSTRACT
Traditionally, information retrieval systems rank documents according to the query terms they contain. However, even if a document may contain all query terms, this does not guarantee that it is relevant to the query. The query terms can occur together in the same document, but may have been used in different contexts, expressing separate topics. Lexical cohesion is a characteristic of natural language texts, which can be used to determine whether the query terms are used in the same context in the document. In this paper we make use of a graph-based approach to capture term contexts and estimate the level of lexical cohesion in a document. To evaluate the performance of our system, we compare it against two benchmark systems using three TREC document collections.

Categories and Subject Descriptors
H.3.3 [**Information Search and Retrieval**]

General Terms
Algorithms, Experimentation, Measurement, Performance

Keywords
Contextual information retrieval, Lexical cohesion, Term proximity

1. INTRODUCTION
A word's meaning is realized depending on the other words that surround it in text. The lexical-semantic dependencies, or lexical cohesive relationships, between words in a text form a common context that "makes text hang together" [6]. In any well-formed natural language text we expect the sentences and larger units of discourse to be semantically related to each other. This makes lexical cohesion a characteristic of all well-formed texts.

Context-awareness is a crucial concern in information retrieval. A document and a query having matching words does not necessarily imply that the document is relevant to the query. It is possible that the words are in the same document but do not share a common context. As opposed to traditional "bag of words" retrieval methods, lexical cohesion can be used to detect the context of query terms and estimate document relevance based on this information. It is demonstrated in [18] that there exists statistically significant association between lexical cohesion and document relevance.

Various classifications of lexical cohesive relationships were proposed by several authors [6, 7, 8]. An important means of creating lexical cohesion is through *collocation*, which is a relationship between lexical items that occur in the same environment [6]. Collocation can occur due to lexical-grammatical restrictions (e.g., using the adjective beautiful to describe a good-looking woman but the adjective handsome to describe such a man), which occur within short spans (*proximity*) as in noun phrases. In addition, if two words are contextually related in text, they tend to occur in the stretches of text that share large number of same or similar terms. This is known as long-span collocation, whose effects can extend in text up to 300 words [18].

In this paper, we present an approach for calculation of document cohesion with respect to query terms using a graph based approach. Initially, we construct a Lexical Collocation Matrix (LCM) for each document in the set to be re-ranked. The cells of the matrix record the number of times any given two terms co-occur within fixed-sized windows in the document. We then represent the document as an undirected graph whose nodes (vertices) are the terms in the document and the arcs (edges) record the collocation frequency between the nodes.

Performance improvement by ranking a document set using lexical cohesion information has been demonstrated before in [18]. The approach presented in this paper differs from other lexical cohesion based ranking methods reported in the literature in the way the context of the query terms is represented, and consequently the way document cohesion with respect to query terms is calculated. Specifically, instead of simply counting common collocates of query terms, thus, capturing only long-span transitive relationships between query terms [18], we take advantage of the graph representation, and capture both long-span and short-span cohesive relationships. In other words in the approach presented in this paper both proximity and transitive relationships between terms are treated in the same principled way.

We report an evaluation experiment in Section 5, which compares the presented methods to two baseline systems, namely, BM25 [16] and COMB-LCS [18] using TREC collections. The results

IIiX'08, Information Interaction in Context, 2008, London, UK.

suggest that the presented methods yield considerable performance gains against the baseline systems.

An additional advantage of the graph based approach is that, the cohesion graph that captures the contexts of the terms in the document could be used to interactively visualize the content of a document. An example of the way the term contexts could be visualized is given in Figure A1. We briefly discuss the potential of the graph-based approach for interactive document visualization in the "Conclusion and Future Work" section at the end of the paper.

In the following section, we present the previous work on calculating lexical cohesion. This is followed by the details of lexical cohesion based document ranking methods developed. The subsequent section describes the experimental setup used in evaluation of the methods presented in this paper. The results of the evaluation experiments are given and discussed in Section 5. The final section summarizes the experimental results and points to future research directions.

2. RELATED WORK

The concept of lexical cohesion is introduced for the first time by Halliday and Hasan [6], and later elaborated by Hoey [8]. Halliday and Hasan demonstrated that one of the basic means of achieving cohesion in natural language texts is lexical reiteration. In this case, two text segments (e.g., sentence, paragraph, etc.) are semantically connected by means of repeating lexical items. There can be different kinds of reiteration between two lexical items: the second lexical item can be an exact repetition of the first one, its synonym or near-synonym, hyponym or hypernym. Previous experiments that involved an ad hoc document retrieval task showed that exact repetition gives comparable results to other types of reiteration [18]. A single instance of a lexical reiteration is referred to as a *lexical link* [8]. Morris and Hirst [14] pointed that in well-formed texts, cohesive relationships extend over the entire text by means of lexically linked segments of text. These sequences of related words are called *lexical chains*. Hoey [8] stated that the text cohesion is realized not only by lexical links between arbitrary text snippets, but also by lexical relationships between sentences - *lexical bonds*. A lexical bond exists between two sentences when they contain a certain number of lexical links.

Lexical chains were used in text retrieval by several researchers. Stairmand [17] used WordNet database [13] in mapping the contents of documents into WordNet synsets, identifying in each document lexical chains. At search time, each query term, mapped into a WordNet synset, is matched against the weighted synsets representing the documents. Ellman and Tait [5] used Roget's thesaurus to identify lexical chains in text and computed the similarity between a Web page retrieved and an exemplar text (i.e. query). They reported inconsistent results for a limited query set.

Lexical bonds were also used in text retrieval. Vechtomova et al. [18] estimated the cohesiveness of a document with respect to a query by counting the number of lexical links between distinct query terms' contexts. The context of a query term is constructed by recording all collocates of the query term in fixed-size windows around each occurrence of it in the document. All collocates of a given query term are then merged to determine the context of it in the document. The number of lexical links between a pair of query terms is computed by counting the number of common collocates in the context lists of the query terms. This method captures only long-span cohesive relationships. They evaluated their system on TREC collections and reported improvements over a baseline system, Okapi BM25. The evaluation results establish a benchmark for us to compare our approach against, and discussed in Section 5. In this paper, in contrast to simply counting common collocates between a pair of query terms, we take advantage of graph representation and capture both long-span and short-span cohesive relationships.

Graphs have been used by some standard ranking algorithms like HITS [9] or PageRank [4]. Such algorithms represent web pages as nodes in a graph and use the connections between them to deduce information on the importance of a node (i.e. web page). This procedure was applied to the task of keyword/sentence extraction from documents by Mihalcea [12]. Mihalcea and Tarau generalized this approach, and applied it to other text processing tasks [11], such as, word sense disambiguation and text summarization. Our approach is similar to this work but different in purpose. Instead of ranking the nodes of the graph and using it for word sense disambiguation and similar tasks, we obtain a total cohesion score for the whole document by representing it as a graph and use it to re-rank documents.

3. SYSTEM DESCRIPTION

We interpret a document as a collocation graph consisting of nodes representing the terms and weighted arcs representing the frequency of collocation between terms. By exploring the paths between query term pairs, we aim to deduce the level of lexical cohesion between query terms, and use this information in estimating relevance of a document to a query. The steps of the process are described below, and illustrated by means of an example in Figures A2 and A3.

3.1 Document Pre-Processing

Before we start constructing the collocation graph, we pre-process the document to eliminate the *stopwords*. Stopwords are common functional words that do not carry content information on their own. In addition, we stem the words to eliminate common morphological and inflexional variations so that we can keep together the words that mean roughly the same thing.

Apart from the stopwords, we also eliminated terms which are non-stopwords but still so common as to form high number of unwanted lexical links with other terms. To do this, we further reduce the document in order to include in our calculations only the most significant F terms determined using the *tf-idf* weighting scheme [15]. By this way, we hope to keep only the significant terms, which contribute to the actual meaning of the document.

3.2 Creation of Lexical Collocation Matrix

The calculation of the Lexical Collocation Matrix (LCM) is done by processing the reduced document, which contains only the most significant stemmed non-stopword terms. We identify fixed-sized windows around every instance of every term in the document. A window is defined as S number of stemmed, non-stopwords to the left and right of a term. We refer to all stemmed, non-stopwords extracted from each window surrounding a term as its *collocates*.

By using the windows identified around each term, we create the LCM for the document. LCM = $[m_{ij}]$ is an LxL symmetric matrix where L is the number of distinct terms (i.e. term types, not tokens) in the reduced document, and each element m_{ij} represents

how many times any instance of $term_i$ occurs in the same window (i.e., collocates) with any instance of $term_j$. First two steps of creating the LCM are illustrated for an example document in Figure A2.

3.3 Conversion of LCM into Lexical Cohesion Graph

In order to estimate the lexical cohesion between query terms, we make use of graphs. We construct a weighted, undirected Lexical Cohesion Graph, LCG = (V, A) such that;

V = {distinct terms in the document}, and

A = {(i, j): w_{ij} = collocation strength between $term_i$ and $term_j$}.

In LCG, a direct path between two nodes implies that the two terms represented by these nodes co-occur in the same window at least once. A multi-hop path implies that the two terms are related transitively by means of some other common term(s). It is assumed that, as these terms co-occur within a common subset of terms, they should also be contextually related.

To find the degree of this relation, we calculate the collocation strength between terms using the LCM created in the previous step. We use the collocation frequencies, i.e. m_{ij} values, from the LCM, as collocation strength. So for an arc (i, j) ε A, $w_{ij} = m_{ij}$.

3.4 Calculation of Lexical Cohesion Graph Score

The Lexical Cohesion Graph Score (LCGS) of query terms for a document is derived from the strength of the paths between query terms. The algorithm to calculate the score of a document {d} for a query term set {*query_term_set*} is as follows:

```
begin
 {query_terms} = {d} ∩ {query_term_set};
 if |query_terms| < 2 then
   return 0;
 else
   for all query term pair {q_i,q_j} where q_i,q_j ϵ {query_terms} do
     construct P, set of paths between q_i and q_j with max length
     of M;
     for all path p_k ϵ P do
         calculate path score PATH_SC(q_i,q_j)_k;
     end for
     calculate pair score PAIR_SC(q_i,q_j) based on
     PATH_SC(q_i,q_j)_k values;
   end for
   calculate document score DOC_SC based on
   PAIR_SC(q_i,q_j) values;
   return DOC_SC;  //i.e. LCGS
 end if
end
```

Calculation of the Path Score. We tested the following methods in computing the path score: *average* of the weights of the arcs in the path (Av), *minimum* weighted arc in the path (Mn), *maximum* weighted arc in the path (Mx). The minimum and maximum values identify the weakest and strongest chains in the path. Averaging assume that the overall path strength lies somewhere between these extreme values. Trivially, any of the path score calculation methods described above reduces to the same value for direct links (i.e. paths without any intermediate node).

Calculation of the Pair Score. Usually there are several paths between query term pairs. Similar to the path score calculation, we calculated the pair scores by taking either the *minimum* (Mn) or the *maximum* (Mx) path score between a query term pair as the overall pair score. In order to investigate the effect of the number of distinct paths between query term pairs we also experimented with taking the *sum* (Sm) and *average* (Av) of path scores.

Calculation of the Document Score (LCGS). To calculate the final score of the document we evaluated three different methods: *summing* all pair scores (Sm), *taking the average* of pair scores (Av), and *multiplying* pair scores (Ml). The last method is particularly useful in penalizing documents where one or more of the query term pairs are weakly linked. A non-existing query term in the document is ignored in the calculations.

The final LCGS for a document is arrived at by following the procedure described above. For each of the three scores there are a number of alternative methods of calculation. These are summarized in Table 1. We evaluated the performance of our system for all possible combinations of each of the alternative methods of calculating the above three types of scores. For example, Ml-Av-Mn means that *multiplication* (Ml), *average* (Av), and *minimum* (Mn) are used in the calculation of document score, pair score and path score, respectively.

The score calculation method as described above enables us to observe the factors that affect the level of lexical cohesion in a document. By analyzing the three different scores that make up the LCGS a document gets, it is possible to deduce certain conclusions about the effect of collocation patterns in text on lexical cohesion (see Section 5.2).

3.5 Re-Ranking of Documents

We re-rank the documents according to their LCGS scores. We also experimented with combining the LCGS with BM25 scores returned by the Okapi retrieval system. For this purpose, we adopted the COMB-LCS method described in [18], and applied it to each of the top T documents retrieved by Okapi as follows:

$$\text{COMB-LCGS} = \text{MS} + x\text{*LCGS} \quad (1)$$

where MS is the matching score returned by Okapi (BM25) and x is a tuning constant to regulate the final score.

4. EXPERIMENTAL DESIGN

We conducted experiments to re-rank the set of top 1000 BM25-retrieved documents by their LCGS scores. In our experiments, we used the weak stemming feature of the Okapi IR system to reduce the document text and the query terms. We experimented with the most significant 50, 100 and 1000 terms selected according to the *tf-idf* weighting scheme, and window sizes of 5, 10 and 15 in constructing the document representations. We permitted maximum of two hops (i.e., one intermediate term) between query term pairs. This is mainly to reduce the processing

Table 1. Alternative methods to calculate path, pair and document scores

Document Score		Pair Score		Path Score	
Method	**Abbreviation**	**Method**	**Abbreviation**	**Method**	**Abbreviation**
Average	Av	Average	Av	Average	Av
Multiplication	Ml	Minimum	Mn	Minimum	Mn
Sum	Sm	Maximum	Mx	Maximum	Mx
		Average	Sm		

time in constructing the collocation graph. For the possible 9 configurations of window size and the number of the most significant terms used in document representations, we ran our experiments on three datasets:

1. TREC 2004 HARD track collection (*HARD04*): 652,710 documents from 8 newswire corpora. Five of the 50 topics had no relevant documents and were excluded from the official HARD 2004 evaluation [2]. This dataset was also used by [18]. We used the same Okapi BM25 parameters as reported in this work to make the results comparable.

2. TREC 2003 HARD track collection (*HARD03*): 372,219 documents from 3 newswire corpora and U.S. government documents. Two of the 50 topics had no relevant documents and were excluded from the official HARD 2003 evaluation [1].

3. TREC 2005 HARD track collection (*HARD05*): 1,033,461 documents from 3 newswire corpora [3].

Short queries were created from all non-stopword terms in the "Title" fields of TREC topics. The queries were run in the Okapi using the BM25 document ranking function to retrieve top 1000 documents. We used the default Okapi values k_1=1.2 and b=0.75 in the BM25 method [16]. In CMB- LCGS calculations (Eqn. 1), we tried values between 0.005 and 2 for the tuning constant.

Instead of separating the collections for testing and training, we present the best run in one collection in the other two as well (Tables 3 and 4). In this way it is possible to cross-validate the evaluation results.

5. EVALUATION AND RESULTS

5.1 Comparison with Benchmarks

We compared the retrieval effectiveness of BM25, COMB-LCGS, and COMB-LCS [18] by their MAP, P10 and R-PREC performances. In assessment of performance we used TREC-EVAL, a program to evaluate TREC results using the standard NIST evaluation procedures (http://trec.nist.gov/trec_eval).

As Table 2 shows, on HARD04, COMB-LCGS performed better than COMB-LCS and BM25 in all measures of performance. COMB-LCGS outperformed BM25 also on HARD03 and HARD05.

5.2 Performance analysis of LCGS

To analyze the effect of the various parameters used in calculating LCGS on the retrieval performance, we present performance of LCGS on its own (not combined with BM25 scores) in Table 3. This makes it possible to see what combinations of LCGS parameters yield the best performance scores. The following combination of parameters are isolated in Table 3: window size (S), number of terms (F) used in document representations, and the methods used in calculating path, pair and document scores (Av, Ml, Mn, Mx, Sm). The best performing combinations of these parameters are given for all three collections in Table 3 (the highest scores for a given collection-evaluation measure combination are typed in bold).

We observe in Table 3 that S=15 is the best performing window size in all metrics and collections except for P10 on HARD03.

Table 2. The highest performance scores of BM25, COMB-LCS and COMB-LCGS (* statistically significant at 0.05, two-tailed paired t-test)

METHOD	HARD04			HARD03			HARD05		
	MAP	**P10**	**R-PREC**	**MAP**	**P10**	**R-PREC**	**MAP**	**P10**	**R-PREC**
BM25	0.2196	0.3089	0.2499	0.3249	0.5375	0.3480	0.1674	0.3640	0.2277
COMB-LCS	0.2322	0.3556	0.2644	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available
COMB-LCGS	0.2413	0.3889	0.2869	0.3264	0.5562	0.3559	0.1930	0.4460	0.2546
Improvement over BM25	9.9% *	25.9% *	14.8% *	0.5%	3.5 %	2.3%	15.3% *	22.5% *	11.8% *

Table 3. Best Performing LCGS Runs

Best combinations found			Sets and metrics tested on								
			HARD04			HARD03			HARD05		
F	*S*	Method	MAP	P10	R-PREC	MAP	P10	R-PREC	MAP	P10	R-PREC
1000	**15**	**Ml-Sm-Av**	**0.1764**	0.3022	**0.2292**	0.2418	0.3958	0.2779	0.1685	0.4200	0.2292
100	**15**	**Ml-Sm-Av**	0.1682	**0.3200**	0.2063	0.2305	0.4250	0.2588	0.1581	**0.4400**	0.2097
1000	**15**	**Ml-Sm-Mn**	0.1681	0.3022	0.2158	**0.2453**	0.4104	**0.2783**	**0.1712**	0.4220	**0.2307**
100	**5**	**Ml-Sm-Mx**	0.1644	0.2933	0.2084	0.2230	**0.4271**	0.2551	0.1574	0.4160	0.2092

F=1000 yields the best results in MAP and R-PREC, while F=100 gives the best result in P10 on all collections. It could be concluded from these observations that for high precision it is best to represent the documents with fewer terms (*F*).

In calculating the document score, multiplying (Ml) the pair scores performs better than summing (Sm), or averaging (Av) them over the number of query pairs. The superiority of the multiplication over summing and averaging is likely due to the fact that the greater the number of lexically related query term pairs in a document the more cohesive the document is with respect to query terms. Averaging the pair scores hide the effect of weakly linked or missing query term pairs to a large extent. The superiority of multiplication over summing suggests that the more the pair scores vary across query term pairs in a document the less the document is cohesive with respect to query terms, hence, the less likely that the document is relevant. Thus, we can conclude that in relevant documents there are higher number of query term pairs that are lexically connected, and the strength of this connection tends to be uniform among all query term pairs.

It can also be observed from the results that summing path scores to arrive pair scores is always better than taking the minimum (Mn), maximum (Mx), or average (Av) of the path scores. This result indicates that the higher the number of distinct paths between a query term pair the more likely the document is relevant. Thus, in relevant documents query terms tend to have more common collocates than in non-relevant documents. In obtaining the path scores different methods yield the best performance: averaging the weights of the arcs, taking the minimum or maximum of arc weights.

As described previously, LCGS is calculated using solely intra-document relationships between terms. Therefore, it does not contain any collection-wide term information. This is probably why LCGS on its own does not always produce results as good as the baseline Okapi BM25 system. However, when the scores of the both systems are fused (Equation 1), the results are better than the either system on its own, suggesting that BM25 and LCGS captures complementary relevance information.

5.3 Performance analysis of COMB-LCGS

We investigated how retrieval performance changes with respect to different combinations of parameters (*F*, *S*, *x*) and score calculation methods (Av, Ml, Mx, Mn Sm) in COMB-LCGS runs. The best performing runs are given in Table 4 for three

Table 4. Best Performing COMB-LCGS Runs

Best combinations found				Sets and metrics tested on								
				HARD04			HARD03			HARD05		
F	*S*	*x*	Method	MAP	P10	R-PREC	MAP	P10	R-PREC	MAP	P10	R-PREC
100	**15**	**0.25**	**Av-Mx-Av**	**0.2413**	0.3778	0.2822	0.2856	0.4396	0.3263	0.1660	0.3740	0.2302
50	**15**	**0.25**	**Sm-Mx-Av**	0.2385	**0.3889**	0.2749	0.2902	0.4667	0.3312	0.1677	0.3780	0.2232
1000	**15**	**0.125**	**Ml-Sm-Mx**	0.2296	0.3667	**0.2869**	0.2852	0.4313	0.3238	0.1765	0.4220	0.2400
50	**10**	**0.008**	**Ml-Av-Mn**	0.2160	0.3267	0.2491	**0.3264**	0.5479	0.3496	0.1675	0.3660	0.2277
100	**5**	**0.008**	**Ml-Mx-Mn**	0.2186	0.3311	0.2601	0.3228	**0.5562**	0.3508	0.1703	0.3940	0.2298
1000	**5**	**0.25**	**Sm-Mn-Mn**	0.2301	0.3200	0.2619	0.3237	0.5271	**0.3559**	0.1677	0.3640	0.2286
1000	**5**	**0.25**	**Sm-Sm-Mn**	0.2332	0.3644	0.2657	0.2773	0.4271	0.3239	**0.1930**	0.4260	0.2532
100	**10**	**0.25**	**Ml-Sm-Mn**	0.2234	0.3756	0.2640	0.2972	0.4771	0.3345	0.1784	**0.4460**	0.2336
1000	**5**	**0.25**	**Sm-Sm-Av**	0.2331	0.3644	0.2689	0.2645	0.3917	0.3174	0.1872	0.4000	**0.2546**

collections (the highest scores for a given collection-evaluation measure combination are typed in bold).

Table 4 shows that there is no unique combination of parameters that yields the highest score in all measures in a collection or for a given evaluation metric on three collections. In document score calculations, Ml and Sm seem to be the most popular methods. There is no regular pattern with respect to the calculation of the other two types of scores (pair and path). Comparison of Table 3 and Table 4 suggests that the selection of parameters and methods depends on the document collection more in COMB-LCGS runs than in LCGS runs.

6. CONCLUSION AND FUTURE WORK

We have investigated different methods for document ranking based on lexical cohesion among query terms in a document. To compute the degree of cohesion in a document with respect to a query we interpreted a document as a graph whose nodes are the terms in the document, and arcs representing the strength of association between the terms connected by it. In this way, we capture, in contrast to other similar methods reported in the literature, not only long-span (transitive) but also short span (proximity) lexical cohesion relationships between query terms in a principled way in our document representations.

The associations a term has with other terms in the cohesion graph constitute its context in the document. The overall strength of the cohesive relationships between all query terms in a document is indicator of a common context that makes the document relevant to a given query.

The results of the experiments conducted on three TREC collections demonstrate that the proposed methods yield performance improvements over the benchmark BM25 document ranking function and also the previous work by [18]. We also found that the retrieval effectiveness of different parameters and methods used in COMB-LCGS runs depends on the document collection.

In our future work, we would like to extend the calculation of the lexical cohesion score such that the collection distributions of the terms are directly incorporated in the document scores. This would make the need to combine two complementary document ranking functions unnecessary.

Another future research direction is to extend the use of the graph-based method presented in this paper to the visualization of document contents. In the Appendix, cohesion graphs for a relevant (Fig. A1-a) and a non-relevant document (Fig. A1-b) are given. The cohesion graph for a document captures the contexts of the terms in the document, and therefore, could be used to explore the relationships between them in an interactive search scenario. Fig. A1-a suggests that, query terms in relevant documents are strongly connected with each other. In contrast, in non-relevant documents they are not directly linked with each other (Fig. A1-b). This suggests that by examining the cohesion graph for a document, users may be able to infer the context in which query terms are used in a document, and make preliminary relevance judgment without having to read the whole document.

7. REFERENCES

[1] J. Allan. Hard track overview in TREC 2003: High accuracy retrieval from documents. In *Proceedings of TREC 2003*, pages 24–37, 2004.

[2] J. Allan. Hard track overview in TREC 2004: High accuracy retrieval from documents. In *Proceedings of TREC 2004*, pages 25–35, 2005.

[3] J. Allan. Hard track overview in TREC 2005: High accuracy retrieval from documents. In *Proceedings of TREC 2005*, pages 52–68, 2006.

[4] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117, 1998.

[5] J. Ellman and J. Tait. Meta searching the web using exemplar texts: Initial results. In *Proceedings of the 20th Annual Colloquium on IR Research*, 1998.

[6] M. A. K. Halliday and R. Hasan. *Cohesion in English*. Longman, London, UK, 1976.

[7] M. Hoey. *Patterns of Lexis in Text*. Oxford University Press, Oxford, UK, 1991.

[8] M. Hoey. *Lexical Priming: A new theory of words and language*. Routledge, Abingdon, UK, 2005.

[9] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[10] C. Kucukkececi, U. Dogrusoz, M. E. Belviranli, A. Dilek, and E. Giral. Chisio: Compund or hierarchical graph visualization tool. Online at: http://www.cs.bilkent.edu.tr/~ivis/chisio.html.

[11] R. Mihalcea. Random walks on text structures. In *CICLing*, pages 249-262, 2006.

[12] R. Mihalcea and P. Tarau. TextRank: Bringing order into texts. In *Proceedings of EMNLP-04*, pages 404-411, 2004.

[13] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller. Introduction to wordnet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235-244, 1990.

[14] J. Morris and G. Hirst. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17(1):21-48, 1991.

[15] G. Salton, E. A. Fox, and H. Wu. Extended Boolean information retrieval. *CACM*, 26(12):1022-1036, 1983.

[16] K. Spärk Jones, S. Walker, and S. E. Robertson. A probabilistic model of information retrieval: development and status. Technical report, Computer Laboratory, University of Cambridge, August 1998.

[17] M. A. Stairmand. Textual context analysis for information retrieval. In *Proceedings of SIGIR'97*, pages 140-147. ACM, 1997.

[18] O. Vechtomova, M. Karamuftuoglu, and S. E. Robertson. On document relevance and lexical cohesion between query terms. *Information Processing and Management*, 42(5):1230-1247, 2006.

APPENDIX

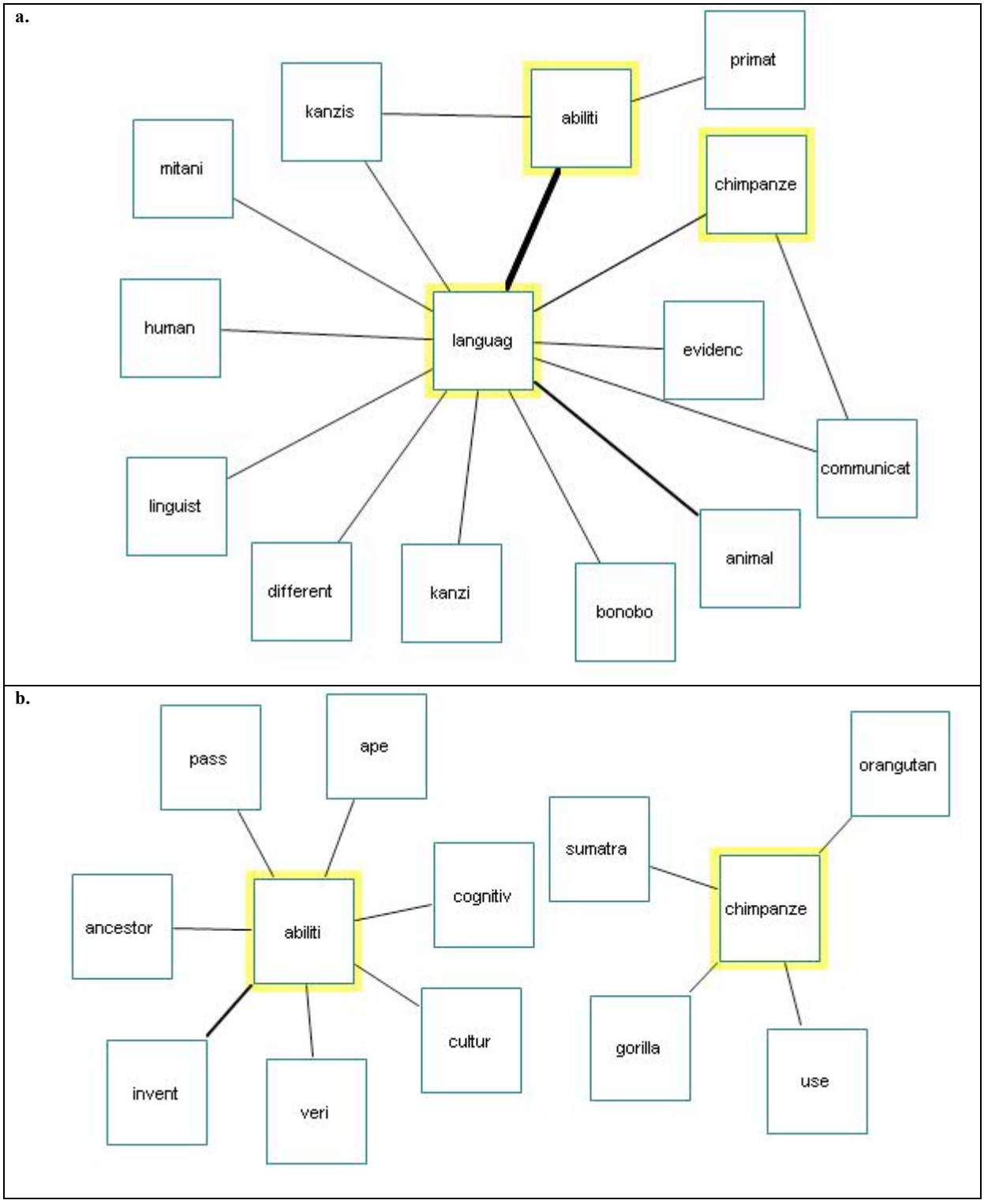

Figure A1 – Visual representation of two documents for the query "Chimpanzee language ability" (HARD-407) drawn using Chisio graph visualization tool [10] using the Lexical Cohesion Graph ($F50S1$). The thickness of arcs represents the strength of association between the nodes (i.e. terms). **a.** A relevant document (NYT20030103.0110) ranked 4th by BM25. **b.** A non-relevant document (APE20030102.0060) ranked 5th by BM25. LCGS demotes document represented in (b), and promotes document represented in (a).

Iteration 1

[**weather** water weather council flood] drought global warm weather disaster cosgrov water council global warm flood flood water climat water global warm council climat drought global global warm cosgrov warm climat warm weather cosgrov flood drought disaster council flood disaster flood council weather disaster council flood flood cosgrov flood drought water drought disaster disaster water climat water council water

	weather	drought	disaster	council	global	water	warm	cosgrov	flood	climat
weather	0	0	0	**1**	0	**1**	0	0	**1**	0
drought	0	0	0	0	0	0	0	0	0	0
disaster	0	0	0	0	0	0	0	0	0	0
council	0	0	0	0	0	0	0	0	0	0
global	0	0	0	0	0	0	0	0	0	0
water	0	0	0	0	0	0	0	0	0	0
warm	0	0	0	0	0	0	0	0	0	0
cosgrov	0	0	0	0	0	0	0	0	0	0
flood	0	0	0	0	0	0	0	0	0	0
climat	0	0	0	0	0	0	0	0	0	0

Iteration 2

[weather **water** weather council flood drought] global warm weather disaster cosgrov water council global warm flood flood water climat water global warm council climat drought global global warm cosgrov warm climat warm weather cosgrov flood drought disaster council flood disaster flood council weather disaster council flood flood cosgrov flood drought water drought disaster disaster water climat water council water

	weather	drought	disaster	council	global	water	warm	cosgrov	flood	climat
weather	0	0	0	1	0	1	0	0	1	0
drought	0	0	0	0	0	0	0	0	0	0
disaster	0	0	0	0	0	0	0	0	0	0
council	0	0	0	0	0	0	0	0	0	0
global	0	0	0	0	0	0	0	0	0	0
water	**2**	**1**	0	**1**	0	0	0	0	**1**	0
warm	0	0	0	0	0	0	0	0	0	0
cosgrov	0	0	0	0	0	0	0	0	0	0
flood	0	0	0	0	0	0	0	0	0	0
climat	0	0	0	0	0	0	0	0	0	0

Figure A2 – First two iterations of building LCM (S=4) out of a reduced document (F=10). The term under consideration is in bold font. Borders of window are marked by "[" and "]". Self-collocations are ignored, so the diagonal is always 0.

a.

EU stays out of Gibraltar vote dispute between Britain, Spain Wary about getting snared by an Anglo-Spanish dispute, the European Union's head office Wednesday said it will not push for a court ruling on the legality under EU law of a British law that lets people in Gibraltar vote in European Parliament elections. At first glance, the European Commission said there seemed nothing awry with the law. It urged Spain and Britain to pursue ``an amicable solution'' to their latest dispute over the British possession. Britain lets Commonwealth citizens in Gibraltar, even if they are not British nationals, vote in next June's European Parliament elections. Spain says EU law lets only British nationals vote in European elections. It also questions the legality of making Gibraltar part of an existing constituency in England and Wales and wants a legal view from the European Commission. EU officials held a hearing on Oct. 1 with British and Spanish officials. They concluded Spain has no case. Britain ``has organized the extension of voting rights to residents in Gibraltar within the margin of discretion'' allowed to EU governments under EU law, the Commission said Wednesday. It left matters there for now, preferring not to ask the EU high court for a ruling for fear that would only aggravate British-Spanish relations. ``Given the sensitivity of the underlying bilateral issue, the Commission at this stage refrains from (opening legal action) and invites the parties to find an amicable solution.'' The European election dispute is yet another source of friction between London and Madrid over Gibraltar, a British possession since 1713. Britain is prepared to share sovereignty of Gibraltar with Spain, which disputes Britain's possession of ``the Rock'' at the tip of the Iberian peninsula, but only if people in Gibraltar approve that. To date, they have opposed shared sovereignty. In a July 27 complaint, Spain alleged Britain's European Parliament Representation Act violated citizenship and constituency requirements of EU election rules. The law was enacted this year, four years after Britain was condemned by the European Court of Human Rights for never having staged any European parliamentary elections in Gibraltar. The EU treaty sets uniform voting rules in all 15 EU nations but ``does not address the issue of franchise,'' said the Commission. ``Thus, national provisions are applicable,'' it added. ``There is no general principle (in EU law) according to which the electorate in European Parliament elections cannot be extended beyond EU citizens.'' Similarly, it said, EU law does not set rules for forming electoral constituencies ``so it is for the member states to lay down such provisions.''

b.

eu gibraltar britain spain european eu law law gibraltar european election european law spain britain britain gibraltar european election spain eu law european election gibraltar european eu spain britain gibraltar eu eu law eu european election gibraltar britain gibraltar spain britain gibraltar spain britain european eu election law britain european european election gibraltar eu eu eu law european election eu eu law

c.

	election	law	european	britain	gibraltar	spain	eu
election	0	4	10	2	5	1	6
law	4	0	7	2	2	2	11
european	10	7	0	5	6	5	7
britain	2	2	5	0	9	8	4
gibraltar	5	2	6	9	0	5	6
spain	1	2	5	8	5	0	3
eu	6	11	7	4	6	3	0

d.

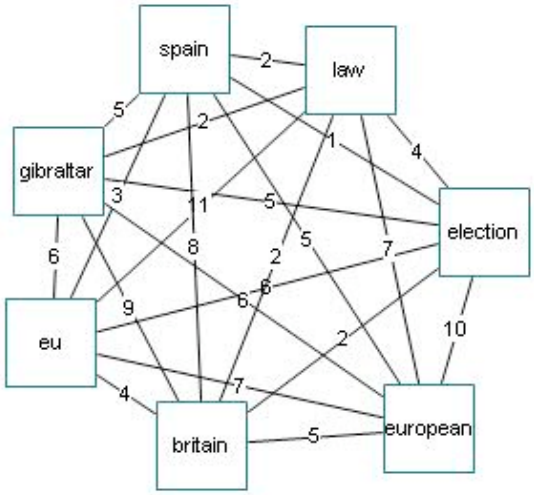

Figure A3 – Transforming a text document into lexical collocation graph. **a.** An example document, APE20031029.0316, relevant to the query "European elections" (HARD-444). **b.** After stemming and reducing for F=7. **c.** LCM for S=2 **d.** LCG derived from LCM (drawn by Chisio [10]).