WORDIJ: A WORD-PAIR APPROACH TO INFORMATION RETRIEVAL

James A. Danowski

University of Illinois at Chicago

CONCEPTUAL MODEL

WORDij is a system based on a linkage or network model for representing textual information. The fundamental unit of analysis is the word pair, or bi-gram phrase, rather than the individual term. WORDij also takes a local approach to term cooccurrence. Systems such as SMART historically used the entire document as the field within which to define term cooccurrence. More recent research has suggested that defining cooccurrence within smaller text units such as paragraphs may be better [Salton & Buckley 91]. WORDij is even more local in focus. It defines cooccurrence of terms within three word positions (after dropping stop words). In addition, WORDij uses direct and indirect pair information to compute shortest paths among words in retrieved documents. This counts both direct and indirect matches between queries and documents.

Consider a query Q containing the phrase {t1, t3} and a document D containing the phrases {t1, t2}, and {t2, t3} but not the phrase {t1, t3}. Existing algorithms [Salton & Buckley 91, Croft, Turtle & Lewis 91, Fagan 89] would not consider the dependency between t1 and t3 as there is no match for the phrase. However, tree-dependency models [van Rijsbergen 77; Yu, Buckley, Lam and Salton 83] recognize such indirect dependencies and produce a formula to compute the degree of dependency between t1 and t3. The WORDij approach considers not only the direct phrases but also indirect phrases.

METHODS

TREC work was begun using a network of Sun workstations in the Database and Information Systems Laboratory in the Electrical Engineering and Computer Science Department at the University of Illinois at Chicago. Because the lead Research Assistant, Nainesh Khimasia, died during the project, software development using C and Unix tools was impeded. Earlier generations of tools had been optimized for an IBM mainframe computer, so work was switched to that platform. The machine used was an IBM 3090/300J platform running VMXA, CMS. A virtual machine CPU size of 16meg was used along with three gigabytes of disk space. The CPU clock speed is rated at 14.5 nanoseconds, or 69 MHz.

We modified earlier generations of WORDij software written in SPITBOL [Danowski 82, Danowski & Andrews 85]. These modifications consisted mainly of replacing some SPITBOL code where possible with CMS PIPELINE code, because it runs approximately one thousand times faster. The *.Z text files were uncompressed using a compress utility on CMS that works with Unix based compressed files. WORDij code was run on each uncompressed text file, generating an inverted file of word pairs by document identification numbers. All word pairs occurring only once in each document were dropped to save disk space.

No spell checking, stemming, morphological analysis, parsing, or tokenizing was done. A stop list of 631 words was used, comprised of the 570 stop words in SMART v.10 and some additional stop words forming the markup format of the raw text. Processing time to create the word pair index averaged three minutes per file.

Ad hoc queries were automatically processed in the same way as raw documents, except that no single pairs were dropped. Query text used to generate word pairs for matching included all text provided, except the factors and definitions, and concepts numbered higher than two. Total CPU seconds to build a query averaged .26 seconds. For the ad hoc queries, nothing further was done to them, either automatically or manually.

For the routing topics, queries were also constructed automatically, but in a different way. The training sets of relevant and irrelevant documents were separately analyzed to identify all word pairs that occurred in the relevant set but not in the irrelevant set. These unique relevant word pairs were used as routing queries.

PIPELINE matching of the query pairs against the pair files for each text file executed in approximately 16 milliseconds per file per 100 sets of query pairs. This meant that to run all 100 queries against the entire collection took approximately five hours of PIPELINE processing on the word pair index files, or three minutes per query.

Time constraints precluded completing a word and word-pair by document count on the entire collection for inverse document frequency or entropy word and word pair weighting. Retrieved documents were ranked from 1 to 200 by counting the number of matching pairs each document had to the query. Frequency of pair occurrence in documents was not used to weight except in breaking ties at the 200 document-rank threshold.

Time limitations also prevented full implementation of the indirect matching process. Only directly matching pairs were used for the main analysis to produce the results. Indirect matching was, however, later tested. This will be described after presentation of the basic results.

RESULTS

WORDij results were greater than or equal to the median levels of performance for seven topics. Our results were within one standard deviation on 55 topics, and within two standard deviations on 82 topics. Performance was significantly lower than the median for 14 topics, as judged by counting topics whose results were greater than two standard deviations below the median. Table 1 lists the topics in two categories, those that were better than or equal to the median, and those that were significantly below the median.

Table 1: Topic Results Ordered by Performance

TOPIC	Difference (median - result)	
Better than or Equal to Median		
66	-.08980	Natural Language Processing
29	-.04540	OS/2 problems
94	-.03180	Computer-aided Crime
95	-.00800	Computer-aided Crime Detection
18	.00000	Global Stock Market Trends
44	.00000	What Makes CASE succeed or fail
88	.00000	Crude Oil Price Trends
100	.00000	Controlling High Tech Transfer
50	.00250	Virtual Reality Military Apps.
Significantly Below Median (Failures)		
22	.19590	Legal Repercus.-Agrochemicals
58	.20740	Rail Strikes
37	.21290	Role of Minis and Mainframes
20	.21770	Superconductors
77	.23290	Poaching
17	.24350	Japanese Stock Market Trends
93	.24560	What Backing Does the NRA Have
13	.24780	Drug Approval
54	.26840	Satellite Launch Contracts
51	.29490	Airbus Subsidies
10	.33340	Space Program
70	.35440	Surrogate Motherhood
78	.38240	Greenpeace
21	.48710	Counternarcotics

Failure Analysis

Query Style.

Several kinds of failure analysis were performed. To investigate whether stylistic features of queries were associated with performance, we computed the following variables for each query using the shareware program, PC-STYLE:

- Number of Sentences
- Number of Words
- Words per sentence
- Percentage of long words
- Percentage of personal words
- Percentage of action verbs
- Average number of syllables per word
- Reading grade level

These variables were correlated with a criterion variable, which was the difference between the median and our result. We subtracted for each query our obtained result from the median result on the 11-point averages of recall-precision contained in the official results across systems for the test queries 51-100. Table 2 displays these correlations. None of them are statistically significant at the .01 level. A second criterion variable was created to represent whether the query was in the "failed" category, greater than two standard deviations below the median. A dummy variable was created for each query using zero to represent success and one to represent failure. Correlations of the style variables were also computed with the failure criterion. No correlations were significant at the .01 level. This suggests that query length, complexity, and other stylistic variables are unrelated to retrieval performance.

Query Words.

Table 2: Query style & performance correlations

	Diff.	Failure
Sentences	-.1053	-.1102
Words	-.1139	-.1616
Words/sent.	.0736	-.0004
Long words	-.1574	-.1086
Personal words	.0846	-.0046
Action words	.1192	.2690
Syllables/word	-.1055	-.0763
Reading grade level	-.0256	-.0629

Additional failure analysis was conducted to explore whether there were particular words associated with performance. The frequencies of all words (no stop words) for each query were correlated with both types of performance criteria: 1) continuous difference from the median and 2) failure, indicated by results significantly below median performance. Table 3 presents the correlations that were significant at the .01 alpha level or better across the 98 topics, and which occurred in at least five different topics.

Table 3: Query words & performance correlations

WORD	r	No. of Topics
Difference		
to	-.2743*	15
some	-.2480*	10
who	.3570**	8
more	.2509*	8
type	.3740**	6
following	.3069*	6
been	.2580*	6
two	.3750**	5
Failure		
national	.2479*	11
system	.2479*	9
who	.3828**	8
more	.2426*	8
been	.2545*	6
two	.4100**	5
support	.2479*	5

* p < .01, ** p < .001

The words 'to' and 'some' increased in frequency as performance increased, while frequency of the following words was associated with lower performance: 'who, more, type, following, been, two.' For the failure criterion, 'who, more, been, two' were also significantly associated with lower performance. In addition, 'national, system, support' were also negatively associated with it. This analysis of words from queries associated with performance suggests that the pair matching approach worked best when the documents used a domain-specific vocabulary.

Proper Name Identification.

At the other extreme, topics that used more domain-general words had lower performance. In particular, queries that asked for a category of documents, such as indicated by words such as 'who' and 'type' were more likely in the failure category. Words including: 'system, national, following, been, and two' were also associated with higher failure rates. This suggests that proper noun compounds may require special treatment. The names of organizations, products, locations, etc. cannot apparently be easily identified through direct pair matching when these specific proper nouns are not contained in the query. When such specific results are called for by a query, special procedures are probably desirable for identification of proper nouns in documents that match on other query pairs.

Domain Specificity of Words.

An additional implication is that query expansion may be fruitful when dealing with domain-transcendent words. Through use of thesauri or databases such as WordNet, alternative word meaning senses may be disambiguated. Then synonyms specific to the proper domain could be added to the actual query pairs contained in the original raw query text.

Interestingly, queries that contained the words 'some' resulted in higher performance. This may suggest that the criteria for relevance were less stringent for such queries, in that they asked not for an exhaustive and complete fit of query to documents, but a more partial overlap. The word 'to' in queries was also associated with higher performance. This may be associated with the specificity of this word in discourse, indicating relationships of direction, degree, state, contact, possession, etc.

Natural Language Processing on Queries.

Together, such query-focused results suggest that future work may benefit from performing complex natural language processing such as parsing, sense disambiguation, etc. on the queries themselves to tune them before matching. Sophisticated treatment of queries may improve performance to the point that such treatment of the raw texts themselves, which is expensive, may not add much marginal performance improvement.

Stemming.

Tests were run with the training sets for three queries selected at random: 2, 26, and 49. For query 2 the difference was zero. For query 26, the relevant documents retrieved increased by 43%, while for query 49 there was a 73% improvement. Average improvement for the three queries was 37% using stemming.

All Pairs.

Tests were run for three different queries to examine effects of dropping single occurring word pairs from documents. Queries 51, 71, and 78 were chosen at random. Retrieval of relevant documents increased on average by 75%, with varied results across queries. Query 51 saw relevant documents retrieved increase by 2.25 times, query 71 decreased performance by .93, and query 78 increased by 11 times, for an average of 1.75 times increase in performance.

Indirect Match Tests.

The training set of documents for query 51, about Airbus subsidies, was used to test indirectness effects. One-step indirectness was assessed, meaning that two query pair words were not directly in the document, but were indirectly connected through an intermediary word.

To illustrate, here are the query pairs including the word, "aid," none of which have any direct matches in the documents:

- AID LOAN
- AID TRADE
- AID FINANCING
- AID SUBSIDIES
- AID ASSISTANCE
- AID GOVERNMENT

Table 4 contains the direct (one-step) and indirect (two-step) links that "aid" had in the documents. The leftmost pairs are direct links, while the rightmost words were directly linked only to the second word of the direct pairs, thus forming a two-step indirect link to the first word in the pair. For example, "aid" is linked to "government" only indirectly through "Airbus." Also, "aid" is linked to "subsidies" only indirectly through "Airbus." These two sets of indirect links, aid-(Airbus)-subsidies and aid-(Airbus)-government are meaningful in terms of the content of the query, which generally concerns government aid and subsidies to Airbus. If we had used only directly matching pairs, we would have missed these two conceptually meaningful sets of links. After identifying all indirect pairs in documents matching query pairs in this way, retrieval of relevant documents was 12% higher.

Shortest Paths.

WORDij does not restrict detection of indirect phrases to these dual bi-gram cases. Rather, indirectness can be of n-step lengths [Danowski and Martin 79, Van Rijsbergen 77]. For example, if there is an intermediate term between two other terms not otherwise linked, then these two other terms have an indirect step linkage of two. If the connection is only through two intermediaries, then the indirect linkage is at step three, and so on. Shortest path algorithms [Gabow & Tarjan 89] find the best set of all direct and indirect links connecting all nodes in a network. Here, this is all words in the query.

We expect that indirectness at the two-step level may contribute most to recall-precision effectiveness. At larger numbers of steps the value of indirect information diminishes. This is because at the extreme lengths, every word is indirectly connected to every other word. This is equivalent to a simple within-document cooccurrence of words, such as in traditional approaches. It renders useless the local cooccurrence constraints. Note also that stop word removal from texts is necessary to represent higher degrees of indirectness. When stop words are present, they increase the connectivity of the word network.

Structural Equivalence and Meaning.

In network analysis, attention to the direct links in a network is called a "cohesion" approach while examining the degree of similarity in two-step links is called "structural equivalence" [Burt 90]. Two nodes are structurally equivalent to the extent that they share the same indirect links, though they may not be directly linked themselves. For example, if word A is linked to words C,D,E and word B is linked to words C,D,E, then although A and B are not directly linked (i.e. show no cohesion), they are structurally equivalent and maximally similar because they share the same links.

Research in mathematical sociology and network analysis has found that structural equivalence is usually equal to or better than cohesion in accounting for system behavior. In text analysis using words as nodes, two words can be considered to share more meaning to the extent they have overlapping two-step links. Therefore, structural equivalence of words is meaning equivalence.

Latent Semantic Indexing and Indirect Pairs.

It is interesting that another approach to indirectness is Latent Semantic Indexing (LSI) [Deerwester et al. 90; Dumais 92]. Instead of than using a network approach, however, it uses an eigenvector model. Eigenvectors represent the combined effects of direct and indirect associations among elements in the matrix. "Latent" refers

Table 4: Direct and indirect links to the word "aid"

			FREQUENCY
aid	airbus		1
		fdp	1
		back	1
		jets	2
		adams	1
		board	1
		crash	1
		group	1
		plans	1
		airbus	1
		boeing	2
		family	1
		german	1
		member	1
		planes	2
		dispute	1
		mandate	1
		nations	2
		partner	2
		percent	1
		program	2
		provide	1
		aircraft	1
		european	1
		products	1
		projects	2
		amendment	1
		executive	1
		industrie	9
		initially	1
		ministers	1
		spokesman	2
		structure	1
	*	subsidies	2
		violating	1
		consortium	5
	*	government	1
		management	1
aid	package		1
aid	guarantee		1
aid	consortium		1
		airbus	1

* These are indirect links that create pairs contained in the query pairs: aid-(Airbus)-subsidies and aid-(Airbus)-government. The other indirect links are not meaningful because they do not relate to the query at the two-step level. Nevertheless, they are listed to show the larger context of identifying meaningful indirectness.

to indirect association patterns below the manifest or direct level. Currently, eigenvector solutions to large matrices are more computationally limited than shortest path network solutions. There has been more development of large scale, parallel algorithms for shortest paths, due to the practical needs to aid routing of information in telecommunications networks. Some work, however, suggests that there is a mathematical equivalence between eigenvector and network approaches to reducing matrices of associations to a simpler underlying structure [Barnett & Richards 91].

Shortest Path Weighting.

Given a set of query word pairs and a list of all documents that contain each word pair--both directly and indirectly-- we can take all pairs of nodes and identify the shortest path linking them in the network. These paths are measured for length according to Euclidean distance in graph terms. Such distance is a direct function of the minimum number of link steps it requires to connect two nodes on their geodesic. Directly linked nodes have a distance of one, nodes linked through one common intermediary node have a distance of two, etc. Documents are counted that were "passed through" or "activated" as each step in the shortest path is traversed. Shortest path algorithms can find these indirect paths with large data sets provided parallel algorithms and hardware are used. We are further developing such experiments.

After IDF weighting, ranking, and selection of the best words, network analysis is conducted on the word pairs they form. The shortest paths linking every word in the set are found, and the word centrality in the network is indexed via the average of the minimum number of steps between that word and all other words in the set.

Then, for each document, it is given a weight that is based on the centrality of the words from the query it contains. The retrieved documents found along the shortest paths between all query pairs are counted and weighted by their constituent word centrality. ank ordered for each query. Documents are then rank-ordered for each query.

CONCLUSION

Results showed that even with unexpected limitations due to the mid-project death of the lead research assistant, Nainesh Khimasia, we succeeded in processing the entire TREC collection and doing direct matching of query word pairs to document word pairs. For 15% of the topics, our results can be considered failures. Failure analysis suggests that improvements in future research may result from:

- query tuning based on natural language processing
- using special procedures for treating proper noun names for organizations, products, locations, etc.
- retaining and using word pairs occurring only once in documents
- stemming the documents and queries
- doing indirect document frequency (IDF) or entropy weighting on words and using these to weight query pairs
- computing additional weights based on shortest paths.

ACKNOWLEDGEMENTS

The author is grateful for the contributions of the following University of Illinois at Chicago faculty, students, and staff to this project: John Andrews, Robert Goldstein, Alan Hinds, Nainesh Khimasia, Jin Hong Meng, Stephen Roy, Gary Singer, Anand Sundaram, George Yanos, and Clement Yu.

REFERENCES

Barnett, G.A. & Richards, W.D. (1991, February). A comparison of NEGOPY'S clique detection algorithm with correspondence analysis. Paper presented to the International Social Networks Conference, Tampa, Florida.

Burt, R.S. (1990). Structure. New York: Center for Social Sciences, Columbia University.

Croft, B., Turtle, H. & Lewis, D. (1991). Proceedings of the SIGIR '91, 32-45.

Danowski, J. (1982). A network-based content analysis methodology for computer-mediated communication: An illustration with a computer bulletin board," Communication Yearbook, 6, 904-925.

Danowski, J. (1988). Organizational infographics and automated auditing: Using computers to unobtrusively gather and analyze communication. In G. Goldhaber and G. Barnett (eds.) Handbook of organizational communication (pp. 335-384). Norwood, NJ: Ablex.

Danowski, J. & Andrews, J. (1985, February). A method for automated network analysis of word cooccurrences. Paper presented to the International Social Networks Conference, San Diego.

Danowski, J. & Martin, T.H. (1979). Evaluating the health of information science: Research community and user contexts. Final report to the Division of Information Science of the National Science Foundation, no. IST78-21130.

Deerwester, S., Dumais, S.T., Landauer, T.K., Furnas, G.W. & Harshman, R.A. (1990). Indexing by latent semantic analysis. Journal of the Society for Information Science, 41:6, 391-407.

Dumais, S.T. (1992). LSI meets TREC: A status report. Paper presented to TREC.

Fagan, J. (1989). The effectiveness of a nonsyntactic approach to automatic phrase indexing for document retrieval. Journal of the American Society for Information Science, 40:2,115-132.

Gabow, H.N. & Tarjan, R.E. (1989). Faster scaling algorithms for network problems. SIAM Journal on Computing. 18(Oct),1013-36.

Salton, G. & McGill, M. (1983). Introduction to modern information retrieval. New York: McGraw-Hill.

Salton, G. & Buckley, C. (1991). Automatic text structuring and retrieval: Experiments in automatic encyclopedia searching. Proceedings of the SIG-IR '91, 21-30.

van Rijsbergen, C. (1977). A theoretical basis for the use of cooccurrence data in information retrieval. Journal of Documentation, 33,106-119.

Yu, C., Buckley, C., Lam, H. & Salton, G. (1983). A generalized term dependence model in information retrieval. Information technology: Research and development, 2,129-154.