Data Fusion of Machine-Learning Methods for the TREC5 Routing Task (and other work)

Kwong Bor Ng* David Loewenstern† Chumki Basu‡
Haym Hirsh§ Paul B. Kantor¶

Rutgers University
New Brunswick, NJ 08903

Abstract

The goal of the document routing task is to extrapolate from documents judged relevant or irrelevant for each of a set of topics accurate procedures for assessing the relevance of future documents for each topic. Rather than viewing different approaches to this problem as "winner-takes-all" competitors, we view them as potentially complementary methods, each exploiting different sources of information. This paper describes two quite different machine-learning approaches to the document routing task, and two approaches to combining their results to perform relevance assessments on new documents. We also describe an approach to the the confusion task based on n-grams that allow approximate matches.

1 Introduction

In the TREC routing task there is a large collection of documents, some of which have been assessed for relevance to each of a number of topics. This makes it possible to approach the routing task as a machine-learning problem — to extrapolate from the topics and evaluated documents some method for accurately assessing the relevance of future documents to each topic. However, applying a machine-learning method to this problem begins with the need to define the attributes that will be used to characterize the documents, over which the machine-learning methods will operate.

Historically, since the work of H.P. Luhn [Luhn, 1958] word tokens (or their morphological stems) have been the preferred attributes for characterizing documents. Although commercial retrieval systems also make use of word position (as does the more sophisticated Inquery system [Turtle and Croft, 1991]), many others use a simple vector representation in which each document is

*APLab, SCILS, kbng@kantor.rutgers.edu.

†Computer Science Department, and Bell Laboratories, Room 15F315, 67 Whippany Road, Whippany, NJ 07981, loewenst@paul.rutgers.edu.

‡Computer Science Department and Bellcore, 445 South Street, Morristown, NJ 07962, cbasu@paul.rutgers.edu.

§Computer Science Department, hirsh@cs.rutgers.edu.

¶APLab, SCILS, kantor@scils.rutgers.edu.

represented by the frequencies with which terms (or stems) appear in it. More recently, though, some success has been claimed for another approach, using overlapping sliding windows of length 5 [Damashek, 1995] or 4 [Grossman *et al.*, 1996].

In what follows we distinguish between the representation itself (which is a mapping of documents into some space of attributes whose structure may be quite complex), and a "scheme", which is a method for assessing the relevance of documents for a topic, usually based upon a particular document representation. Although one scheme may be uniformly more effective than another (with respect to both corpora and information requests), sometimes there are systematic dependencies of the effectiveness of the representation on the corpus, the information need, or both, and in some situations the two may vary in effectiveness in ways that are not systematically determined by any known variables.

A key idea in this work is that if one scheme is sometimes superior to and sometimes inferior to a second, then it may be possible to combine the two to achieve better performance than is possible with either in isolation. We have explored two specific methods for such "data fusion" in our TREC5 work. The first is to find, for each topic in isolation, which scheme works best, and use it to perform the routing task for that specific topic. We call this approach "scheme-per-topic", abbreviated *SPT*, since each topic is assigned its own single scheme. Our second approach is to assess the relevance of a document as some combination of the relevance assessments of the different schemes. We call this second approach the "global" method, abbreviated *GLOB*, since our specific combination method uses a single rule of combination across all topics.[1]

One would expect data fusion to be more successful as the difference between the schemes increases. In TREC4 we found that term-based and 5-gram-based representations were not sufficiently different to support interesting fusion experiments [Ng and Kantor, 1996]. We are now exploring data fusion on two schemes that appear to differ more significantly from one another. Although our preliminary work (decribed in the Workshop Paper) examined four different schemes, the fusion experiments reported here focus on only the two we were able to complete in time for the TREC5 conference.

The first of these schemes (called *KB*) is term-based, implemented using mgquery [Witten *et al.*, 1994], with a small number of query terms selected according to their individual ability to discriminate the union of all relevant documents from the union of all not-relevant documents. The second (called *DL*) estimates the compressibility of each document in terms of the dictionary built from compressing the topic statement plus the corpus of judged-relevant documents.

Unfortunately, our implementation of the *DL* and *SPT* methods had bugs not discovered until after our official submission was made. The results reported here represented an exploration of the *GLOB* method, albeit when given inferior data.

We begin this paper with a description of these two machine-learning schemes for the routing task. We then present the two fusion methods we used in this work. Results comparing our two fully implemented (and in one case buggy) routing schemes as well as the *GLOB* fusion method are then presented. We next present some results on the effectiveness of the *DL* method expressed in terms of a signal detection approach. These results help us to understand why the method was not more effective as implemented, and to lay the groundwork for possible modifications that would make it more effective in the IR routing situation. Finally, we discuss some additional work we performed

[1] This is in contrast to adding an additional layer in which a different rule of combination is formed for each topic [Bartell *et al.*, 1994].

on the TREC5 confusion track. We conclude the paper with some final remarks concerning the philosophy underlying our collection of TREC5 efforts.

2 Term-Based Relevance Assessment: Scheme *KB*

To select the terms used in term-based representations of documents, we considered the union of the texts of all the known relevant documents (P), and the union of the texts of all the known non-relevant documents (N). The appearance of each term (stemmed as in the General Electric-Rutgers collaboration, reported elsewhere) was assumed to follow a Poisson distribution, and the parameter of that distribution was estimated (computed on the union of all the texts of a given type). The ratio of the estimated parameters — π for the positive text P and ν for the negative text N — was used to rank the stems for their ability to discriminate between the two texts. To avoid the occurrence of zeros and infinities we used an approximate confidence interval estimator of the Poisson parameter. In the positive texts, the parameter was estimated to be at the lower edge of a confidence interval. For the negative texts, the parameter was estimated to be at the upper edge of the confidence interval.

More specifically, we chose π (resp. ν) to be that point below (resp. above) the unique value of the Poisson parameter that makes the likelihood of the observed value a maximum (that is, $\lambda = k$), and such that the probability of the observed value, conditioned on the parameter, is $e^{-0.5} = 0.607$. Exact solutions are used for $k \leq 5$, and the approximation $k \pm \sqrt{k}$ was used for larger values of k. Selected terms were given equal weights and used as the representation of the query given to the mg system [Witten *et al.*, 1994]. This then produced a “KB”-labeled ranked list of documents for each topic.

3 Compression-Based Relevance Assessment: Scheme *DL*

Our second approach uses the LZW compression method [Welch, 1984] to calculate the entropies of the test documents conditioned on a topic statement plus the documents judged relevant to the topic. These entropies are used to yield probabilities that form the basis for ranking the relevance of new documents for each topic.

In more detail, documents are first preprocessed to remove non-ASCII characters and to replace all whitespace with the space character. SGML heading information is maintained. For each topic the collection of (preprocessed) documents judged relevant to a topic is used to generate a dictionary of strings of various sizes. The topic statement is also included with this collection of documents relevant to the topic, and for ease of exposition, for the remainder of this discussion we will not point this out again — references to collections of documents relevant to a topic should be interpreted to include the topic statement. The dictionaries are formed by initializing them with an alphabet of the printable ASCII characters, plus the space character. The sequence of relevant documents is then parsed greedily into a dictionary: First, the longest substring matching a dictionary entry and starting from the current position (initially, the first character of the document) is found. Then, the substring consisting of the match plus the next character is added as a new dictionary entry. The current position is then moved just after the character following the match, and the process is repeated until end of the sequence of documents has been reached. (See [Ziv and Lempel, 1978] for more details on this dictionary approach for text compression.)

These dictionaries, one per topic, can then be used to compress the test documents (each yielding different degrees of success in compression). Again, the longest substring from the current position (starting at the beginning of the document) matching a dictionary entry is found, and a pointer to the matching entry is output. The current position is moved to just after the match in the document, and the process is repeated until the document has been completely processed. This is the LZW-parse of the document, and compression is achieved when the cumulative size of such pointers is less than the original document length. The entropy estimate for document d conditioned on topic t, written $E(d|t)$, is equal to $p \times \lceil \log_2 |D_t| \rceil$, where $|D_t|$ is the number of entries in the dictionary generated from the documents relevant to topic t, and p is the number of pointers output in the LZW-parse.

Such conditional entropies may be used to calculate conditional probabilities. Let us imagine a machine that, when given a dictionary D_t, repeatedly and randomly selects entries from dictionary D_t and outputs the entries concatenated together until a document of length L is created. Such a machine would generate a particular parse π of document d of length $L = length(d)$ with probability $P(d, \pi|t) = |D_t|^{-p_\pi}$, where p_π is the number of dictionary entries selected in parse π. The probability of generating any particular document d from D_t is $P(d|t) = \sum_\pi P(d, \pi|t)$.

If we assume that the LZW-parse described is much better than any other parse (*i.e.*, that the inequallity $p_{LZW} << p_\pi$ holds for $\pi \neq LZW$), then $P(d|t) = kP(d, LZW|t)$, where k is a constant slightly larger than 1. This assumption is known to hold true for natural languages including English [Bell *et al.*, 1990]. Since $P(d, LZW|t) = 2^{-E(d|t)}$, we used a monotonically decreasing function of $E(d|t)$, namely $\rho_{d,t} = length(d)/E(d|t)$ (where $length(d)$ is the length of the document) to rank topic relevance for each new document.

4 Scheme-Per-Topic Data Fusion: Scheme *SPT*

Recall that we use the term "scheme" to denote a method for assessing a document's relevance — computing the similarity (or "proximity") of either a pair of documents or a document and a query/topic. We now also use the term "scheme" to refer to our two fusion methods, since they, too, are methods for assessing document relevance.

The *SPT* approach simply uses as its relevance-assessment scheme for a topic whichever baseline method works best on that topic. Since we apply data fusion to the *KB* and *DL* schemes, to determine which is better we selected a "training corpus" from the full initial corpus of documents, consisting of the union of 90% of the judged relevant documents, 90% of the judged non-relevant documents, and 10% of the non-judged documents, assumed for training purposes to be not-relevant. This latter collection of documents was selected to broaden the distribution of documents on which the methods would be evaluated. Testing of each scheme took place on the disjoint remainder of the judged and not-judged documents, called the "testing corpus", with each scheme using just those documents appearing in the training corpus to assess the relevance of documents in the testing corpus.

The sum, over the top 200 documents, of $(200 - \text{rank})$, evaluated at each rank occupied by a relevant document, was then computed, and whichever yielded a better value for a particular topic was used for our official submission on that topic. In this fusion scheme, yielding the results given in our official submission *rutapspt*, the relevance of each document is determined using the complete corpus — the full collection out of which the scheme-selecting training and testing corpora were selected for our "internal" scheme-selection process.

5 Global Data Fusion: Scheme *GLOB*

For the second fusion scheme we formed a simple global combination rule. Each ranking scheme was used to produce a set of the top 1000 documents. In doing so, each scheme produced a ranking score for each document. (These scores were given by the cosine rule in the *KB* method, and by the compressibility of the document being scored, in the *DL* method.) Scores assigned by each system were normalized, for each topic:

$s_{\max}$ =score of top document
$s_{\min}$ =score of document ranked 1000
$s_{norm} = (s - s_{\min})/(s_{\max} - s_{\min})$.

If a document did not appear in the top 1000 documents retrieved by scheme J it was assigned the normalized score $s^J_{norm} = 0$. The overall score assigned to a document was then computed as $s_{glob} = (s^{DL}_{norm} + s^{KB}_{norm})/2$. Documents in the union of the two lists were ranked in decreasing order of s_{glob} and the top 1000 formed the official submission *rutapglob*.

6 Updated *DL* Results

Nineteen of the routing topics were selected at random. For each topic two dictionaries were formed, one based on the documents that had been judged relevant (called the positive dictionary), and one based on the documents that had been judged not relevant to that topic (call it the negative dictionary). To do this, for each topic half of the relevant, and half of the not relevant documents were selected (once), at random. This represents a single split-half test of the method. The total (9399) of all cases exceed the number of actual documents in the test set (8269) because some documents had been judged not relevant for more than one topic. Each document from the test set for a given topic was compressed with respect to both the positive and negative dictionaries for that topic, and was assigned relevant or not-relevant according to which dictionary gave more compression. The performance is summarized in Table 1, in which the basic data are the numbers of True (False) Positives and Negatives. A True Positive is a relevant document that has been assigned "relevant" by this method.

The results, also presented in Figure 1, are best expressed in terms of the detection rate and the false alarm rate. Together, these define for each topic, a Generalized Retrieval Operating Characteristic (GROC; Kantor and Voorhees, this volume) determined by three points: the indicated operating point, the point $(0, 0)$, and the point $(1, 1)$. For example, the operating point, for this method, for Topic 11 has a false alarm rate of $fa = 52\%$ and a detection rate of $dr = 86\%$. An ideal routing system would have a very low value of fa and a high value of dr.

In the actual TREC situation, the number of non-relevant documents is of the order of 10^6, while the number of relevant documents is of the order of 10^3. Thus performance at this level would produce, in general, 860 relevant documents and 520,000 non-relevant documents, for a precision of 0.16%. Clearly if this method is to become part of a workable system some changes are needed.

Topic	#Pos	#Neg	#	TP	TN	FP	FN	FA	DR	Err
001	161	441	602	129	208	233	32	0.53	0.80	0.440
003	308	323	631	303	35	288	5	0.89	0.98	0.464
004	86	374	460	68	297	77	18	0.21	0.79	0.207
005	184	398	582	182	135	263	2	0.66	0.99	0.455
006	169	326	495	158	106	220	11	0.67	0.93	0.467
011	171	359	530	147	174	185	24	0.52	0.86	0.394
012	219	412	631	195	174	238	24	0.58	0.89	0.415
023	113	481	594	108	162	319	5	0.66	0.96	0.545
024	216	281	497	214	60	221	2	0.79	0.99	0.449
044	293	458	751	283	131	327	10	0.71	0.97	0.449
053	348	126	474	346	6	120	2	0.95	0.99	0.257
054	136	122	258	131	58	64	5	0.52	0.96	0.267
058	111	181	292	108	95	86	3	0.48	0.97	0.305
068	131	222	353	123	105	117	8	0.53	0.94	0.354
077	109	202	311	106	122	80	3	0.40	0.97	0.267
078	112	171	283	108	60	111	4	0.65	0.96	0.406
082	363	50	413	362	19	31	1	0.62	1.00	0.077
094	350	250	600	335	67	183	15	0.73	0.96	0.330
095	325	317	642	40	303	14	285	0.04	0.12	0.466
total	3905	5494	9399	3446	2317	3177	459	0.58	0.88	0.387

Table 1: Updated *DL* method. #Pos=Number of positive test examples. #Neg=Number of negative test examples. #=Number of test examples. TP=True (correct) positive classifications. TN=True negative classifications. FP=False positive classifications. FN=False negative classifications. FA=False alarm rate. DR=Detection Rate. Err=Crude Error Rate.

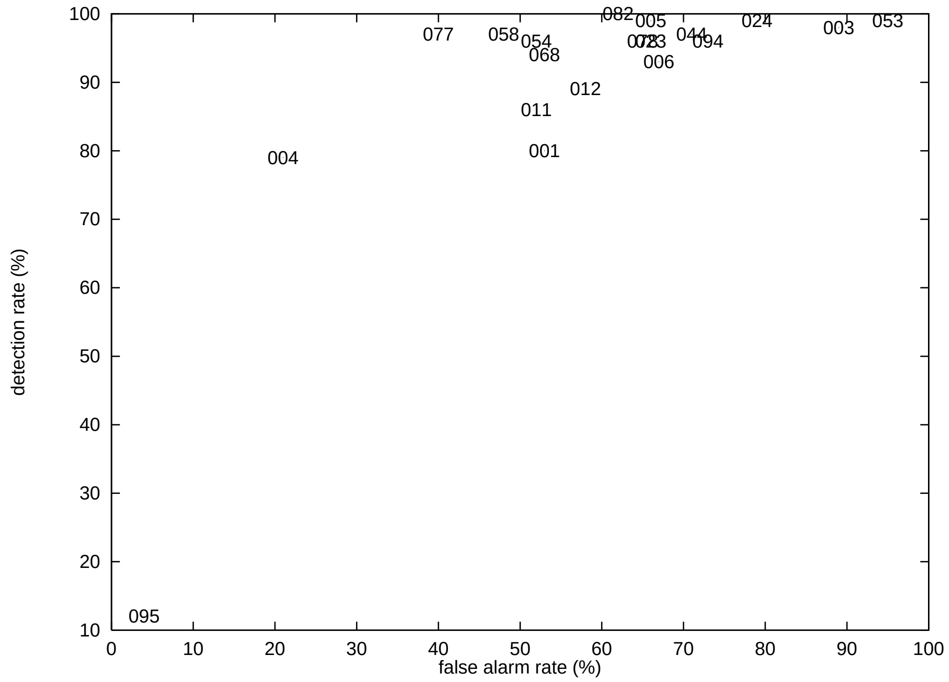

Figure 1: GROC curve for Updated *DL*.

The simplest method to solve this problem would be to convert the compressibilities into conditional probabilities $P(d, LZW|t)$ and calculate *a posteriori* probabilities for the relevant and non-relevant sets:

$$P(t|d, LZW) = \frac{P(d, LZW|t)P(t)}{P(d, LZW)}$$

where $P(d, LZW)$ may be set arbitrarily since it is the same for both the relevant and non-relevant (negative) sets for a given document. Routing would then proceed by classifying as relevant all documents d for which $P(\mathit{relevant}|d, LZW) > P(\mathit{non\text{-}relevant}|d, LZW)$. Another possible source of improvement is to change the assignment rule, by finding the relative compressions which empirically optimize, for example, precision at 1000 documents retrieved. This can be explored by ranking documents in decreasing order of the ratio of compressions achieved by the positive and negative dictionaries, and choosing the cutoff to optimize a measure of precision. Another possibility is to combine this method with some method which is essentially orthogonal in its principles, to see whether fusion of the two methods gives an improvement over either method.

7 Results

Unfortunately, the implementations of our *DL* and *SPT* schemes both had bugs that were not discovered until after the submission of our official runs, and thus the retrieval performance for both schemes *SPT* and *GLOB* were quite poor. Due to the bug in *SPT* we do not report further on it

win/tie/lose	*DL*	*KB*	*GLOB*	median
DL	—	1/2/42	0/2/43	0/0/45
KB	38/6/1	—	34/10/1	11/7/27
GLOB	35/9/1	4/9/32	—	0/4/41
median	42/3/0	33/10/2	39/5/1	—

Table 2: Performance comparisons: win/draw/lose on number of relevant retrieved in top 1000 (above diagonal) and top 100 (below diagonal).

here. In the case of *GLOB*, which combines the ratings of *KB* and (the incorrectly implemented) *DL*, at 1000 documents *GLOB* equaled the median on four topics, and was inferior on the remaining topics. Results were roughly comparable at 100 documents retrieved, with *GLOB* exceeding the median once, tying it five times, and performing worse in the remaining cases. Nonetheless, these results of our official submissions, as well as our subsequent analyses, allow us to reach a number of conclusions.

First, to calibrate how seriously the implementation error affected the *DL* scheme, we determined what its precision rates would have been for each topic at 100 and 1000 documents retrieved. In every case but one *DL* performed far worse than the median of all official submissions, only equaling median performance for 100 retrieved documents when the median precision was 0. We are therefore confident that our implementation error did not in any way reflect a serendipitous invention of a successful new scheme — it is a vastly inferior method.

In contrast, the *KB* scheme at 1000 retrieved exceeded the median in 11 cases, tying it in 7, and was worse in the remaining cases. At 100 retrieved documents it exceeded the median in 2 cases and tied it in 10. Thus, although it did not perform overwhelmingly well either, it did far better than our incorrectly implemented *DL* scheme.

However, even though incorrectly implemented, the *DL* scheme does represent a scheme that ranks documents, however poorly, and thus we can still consider how well our the *GLOB* data-fusion scheme reconciles the rankings of the *KB* and (incorrect) *DL* schemes. Our ideal result would be to find that *GLOB* performed better than either individual scheme in isolation, demonstrating that fusion can yield better results than either method in isolation. Unfortunately, this was not the case. Occasionally *GLOB* exceeded *KB*, but for the most part *KB*'s performance substantially exceeded that of either fusion method.

On the other hand, the worst-case scenario would be if the fusion method performed worse than the incorrect *DL* scheme. Our results show that this was not the case: *GLOB* exceeded *DL* on every topic but 2, which were ties, at 1000 documents retrieved, and tied in 9 topics, lost in one, and exceeded *DL* on the rest for 100 documents.

These results are summarized in Table 2. Each entry "$a/b/c$" represents the fact that the scheme labeling that row retrieved more relevant documents than the scheme labeling that column on a topics, tied it on b topics, and retrieved fewer relevant documents on c topics. Entries above the diagonal represent data for the number of relevant documents in the top 1000 for each scheme, and entries below the diagonal represent data for the top 100 documents retrieved by each scheme.

However, these results favor cases where few relevant topics were retrieved by any method. In

win/tie/lose	*DL*	*KB*	*GLOB*	median
DL	—	0/0/31	0/0/29	0/0/33
KB	25/0/0	—	34/0/0	11/2/21
GLOB	15/0/0	3/1/21	—	0/0/32
median	26/0/0	25/2/2	28/0/1	—

Table 3: Performance comparisons for cases with more than 10 relevant documents retrieved: win/draw/lose on number of relevant retrieved in top 1000 (above diagonal) and top 100 (below diagonal).

such situations (for example, when the median number of retrieved documents was 0) it is more likely for even a poor method to match the median. To account for this we report the same data in Table 3, but only for those cases where the better of the two schemes being compared in an entry returned more than 10 relevant documents. These results only strengthen the conclusions reached from inspection of Table 2, in most cases skewing data even farther to favor the better of the two schemes.

8 Approximate Term Matching for Retrieval of Degraded Documents

In addition to the routing task, we developed two approaches to the TREC5 confusion track, both based on the idea of approximate matching of substrings in a document. One first approach was based on the *DL* scheme for routing, loosened by allowing approximate matches on dictionary entries in assessing probabilities. Unfortunately we were unable to complete this effort by the submission deadline.

Our second approach, which led to an official submission, was to form 5-grams and base retrieval on how many matches and approximate matches can be found in a document. In more detail, we define a "double-dot-5-gram" to be a 5-gram — a string of five characters — with one position comprising a "don't care" character that can match ¿from 0 to 2 characters. A double-dot-5-gram matches a string if any substitution of 0, 1, or 2 characters for the don't care causes the result to be a substring of the string.

A set of double-dot-5-grams was generated for each topic statement by first removing stop words (using the same stopword list used by the SMART system [Salton, 1971], available via ftp from Cornell) and then generating every 5-gram that exists in the resulting document. Thus, for example, given a document containing just the word "orange", two 5-grams would be generated, "orang" and "range". Next, each 5-gram is converted into five double-dot-5-grams by one-by-one replacing the character in each position with a don't care character, generating in our example the double-dot-5-grams "*rang", "o*ang", "or*ng", "ora*g", and "oran*" from "orang", if "*" is used to represent a don't care.

Retrieval was accomplished using a simple match-and-count method. The set of double-dot-5-grams generated from each topic was compared to each line of each corrupted document, and the number of matches was determined. The documents with the top 1000 cumulative counts across each topic's double-dot-5-grams were returned as the result of the retrieval and were then ordered in decreasing value of the ratio of the number of hits to the number of lines in the document (hits per line).

Although we were pleased to note that for the 5% corrupted output there are 7 cases in which our scores are equal to the best scores and for the 20% corrupted output there are 3 cases in which our scores are equal to the best scores, these were primarily cases in which the median showed that for most methods the document was found amongst the first few returned results. When we compare our results using median ranks, for the 5% corrupted output there are only 3 cases in which we are better than the median, and for the 20% corrupted output there are only 8 cases where we exceed the median.

9 Final Remarks

This paper has described our efforts on the TREC5 routing and confusion tracks. One theme running through these efforts is a search for alternatives to the term-based schemes that presently dominate the field of information retrieval. Whether it is the *KB* scheme's modification of weighting methods to select terms for retrieval, or the more drastic change of perspective embodied in the *DL* method, our attempt has been to expand the variety of document representations with which these problems are approached. We rest firmly on a "statistical" approach to information retrieval, with no overt pretext of addressing semantic issues. However, we do try to exploit other sources of information that could potentially assist in information-access tasks, such as sequences comprising arbitrarily long sets of words that can be considered by, for example, the *DL* scheme. Whether any such automated schemes can achieve the stunning independence of interpretations by distinct humans, which forms the basis for the positive results reported in our earlier data-fusion work [Saracevic and Kantor, 1988, Belkin *et al.*, 1995] remains to be seen.

Acknowledgments

The LZW code used for our *DL* analysis is a modified version of code written by Michiel Noordewier. We thank William Cohen and Daniel Kudenko for their participation in our TREC5 efforts. The support of the Alexandria Project Laboratory for travel expenses is gratefully acknowledged.

References

[Bartell *et al.*, 1994] B. Bartell, G. Cottrell, and R. Belew. Automatic combination of multiple ranked retrieval systems. *Proceedings of the 17th Annual International AMC SIGIR Conference on Research and Development in Information Retrieval*, 1994.

[Belkin *et al.*, 1995] N. Belkin, P. Kantor, E. Fox, J. Shaw. Combining the evidence of multiple query representations for information retrieval. *Information Processing and Management*, 31(3)431–448, 1995.

[Bell *et al.*, 1990] T. Bell, J. Cleary, and I. Witten. *Text Compression*. Prentice Hall, Englewood Cliffs, NJ, USA, 1990.

[Damashek, 1995] M. Damashek. Gauging similarity via n-grams: Text sorting, categorization and retrieval in any language. *Science* v267 (10 Feb 1995) 843–848.

[Grossman *et al.*, 1996] D, Grossman, O. Frieder, D. Holmes, M. Nguyen, C. Kingsbury. Improving accuracy and run-time performance for TREC-4. In Harman D (Ed.) Proceedings of TREC-4, 1996.

[Li and Vitányi, 1993] M. Li and P. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer-Verlag, New York, 1993.

[Luhn, 1958] H. Luhn. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2):159-165 and 317, April 1958.

[Ng and Kantor, 1996] K.B. Ng and P. Kantor. Two experiments on retrieval with corrupted data and clean queries in the TREC4 adhoc task environment: Data fusion and pattern scanning. In Harman D (Ed.) Proceedings of TREC-4, 1996.

[Salton, 1971] G. Salton (ed.) *The SMART Retrieval System – Experiments in Automatic Document Processing*. Prentice Hall, Englewood Cliffs, NJ, 1971.

[Saracevic and Kantor, 1988] T. Saracevic and P. Kantor. A study of information seeking and retrieving: III Searchers, searches and overlap. *JASIS* 39:178–194, 1988.

[Turtle and Croft, 1991] H. Turtle and W. Croft. Evaluation of an inference network-based retrieval model. *ACM Transactions on Information Systems*, 9(3)187–222, 1991.

[Welch, 1984] T. Welch. A technique for high-performance data compression. *IEEE Computer*, 17(6):8–19, 1984.

[Witten *et al.*, 1994] I. Witten, A. Moffat, and T. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images*. van Nostrand Reinhold, New York, 1994.

[Ziv and Lempel, 1978] J. Ziv and A. Lempel. Compression of individual sequences via variable-rate coding. *IEEE Trans. Information Theory*, 24:530–536, 1978.