

Anti-Serendipity: Finding Useless Documents and Similar Documents

James W. Cooper and John M. Prager

IBM Thomas J. Watson Research Center

jwcnmr@watson.ibm.com, prager@watson.ibm.com

Abstract

The problem of finding your way through a relatively unknown collection of digital documents can be daunting. Such collections sometimes have few categories and little hierarchy, or they have so much hierarchy that valuable relations between documents can easily become obscured.

We describe here how our work in the area of term-recognition and sentence-based summarization can be used to filter the document lists that we return from searches. We can thus remove or downgrade the ranking of some documents that have limited utility even though they may match many of the search terms fairly accurately.

We also describe how we can use this same system to find documents that are closely related to a document of interest, thus continuing our work to provide tools for query-free searching.

I. Introduction

The problem of finding important and relevant documents in an online document collection becomes increasingly difficult as documents proliferate. Our group has previously described the technique of Prompted Query Refinement (Cooper & Byrd, 1997, 1998) to assist users in focusing or directing their queries more effectively. However, even after a query has been refined, the problem of having to read too many documents still remains.

We have also previously reported the details of the "Avocado" summarization system we developed for producing rapid displays of the most salient sentences in a document. (Neff & Cooper, 1999a, 1999b).

Users would prefer to read or browse through only those documents returned by a search engine which are important to the area they are investigating. In this paper we propose that document retrieval systems can utilize a set of relatively easily derivable numerical parameters to predict which documents will be of most interest to the user.

In this paper, we describe the most important algorithms used in our group's Talent toolkit and outline how the most recent version of our summarizer

algorithm works. We describe the kind of document collection we were studying and why removing useless documents became important.

We then outline the major quantitative parameters we derived for each document and develop an algorithm for identifying useless documents. Finally, we describe our work in document similarity using these same derived parameters and correlate the findings with those from the problem of useless document identification.

II. Background

Finding documents in a collection is a well-known problem and has been addressed by any number of commercial search engine products, including Verity, IBM Intelligent Miner for Text and Alta-Vista. While each system provides some additional features for refining queries, we are not aware of any work in which documents in the hit list are downgraded in ranking because of content once they have been returned by the engine.

There have been a number of approaches to solving document retrieval problems in recent years. For example, Fowler [Fowler, Wilson and Fowler, 1992] has described a multi-window document interface where you can drag terms into search windows and see relationships between terms in a graphical environment. Local feedback was utilized by Buckley [Buckley, et al., 1996] and Xu and Croft, [Xu & Croft, 1996] who also utilized local context analysis using the most frequent 50 terms and 10 two-word phrases from the top ranked documents to perform query expansion. [Schatz, et al. 1996] describe a multi-window interface that offers users access to a variety of published thesauri and computed term co-occurrence data.

The Talent Toolkit

In approaching these document retrieval problems, we have applied a number of technologies developed in our group. In particular, we utilized the suite of text analysis tools collectively known as Talent (Text Analysis and Language Engineering Tools] for analyzing all the documents in the collection.

Texttract

The primary tool for analyzing this collection is **Texttract**, itself a chain of tools for recognizing multiword terms and proper names. Texttract reduces related forms of a term to a single *canonical form* that it can then use in computing term occurrence statistics more accurately. In addition, it recognizes abbreviations and finds the canonical forms of the words they stand for and aggregates these terms into a vocabulary for the entire collection, and for each document, keeping both document and collection-level statistics on these terms. Each term is given a collection-level importance ranking called the IQ or Information Quotient [Cooper & Byrd, 1998; Prager, 1999]. IQ is effectively a measure of the document selectivity of a particular term: a term that appears in only a few documents is highly selective and has a high IQ. On the other hand, a term that appears in many documents is far less selective and has a low IQ. IQ is measured on a scale of 0 to 100, where a value of X means that X% of the vocabulary items in the collection have a lower IQ. Two of the major outputs of Texttract are the IQ and collection statistics for each of these canonical terms, and tables of the terms found in each document. In addition, we computed and stored the tf^*idf value for the major terms found in each document by the summarizer process.

In this project, we entered all of the documents and terms and statistics into a relational database so that the most important terms per document or the documents containing such terms could easily be retrieved. Such terms could be selected either by IQ or by tf^*idf .

Context Thesaurus

We have previously described the context thesaurus [Cooper and Byrd, 1997, 1998] It is computed from a concordance of all the sentences and occurrences of major terms in those sentences. It is an information retrieval (IR) index of the full text of the sentences surrounding these terms and thus provides a convenient way for a free text query to return terms that commonly co-occur with the query phrase. It is similar to and was inspired by the Phrase-finder [Jing and Croft, 1994].

Named Relations

Named relations [Byrd and Ravin, 1999] are derived by a shallow parsing of the sentences in each document, recognizing over 20 common English patterns which show a named relation between two terms. Two of the most common of these patterns are appositives and parenthetical phrases such as

Lou Gerstner, CEO of IBM, said today...

The named relation finder recognizes “Lou Gerstner” as a proper name and “IBM” as a proper name, and assigns the named relation “CEO” to these two terms. Note that this subsystem looks for patterns rather than specific English phrases and is quite general. We typically find several hundred different kinds of names for relations in a collection of several thousand documents, and several thousand actual relations. This system also assigns a direction between these relations, so that a relationship like “makes” or “is located in” points from a company to a product or city name. These names and the terms they relate to are entered as rows in relations tables in the relational database mentioned above.

Unnamed Relations

Unnamed relations are strong bi-directional relations between terms which not only co-occur but occur together frequently in the collection. These terms are recognized from the document and term statistics gathered by Texttract and by the relative locations of the terms in the document. The method of weighting the terms is related closely to tf^*idf and amounts to an application of mutual information. [Kazi]. These unnamed relations are also entered in the relational database.

The Summarization Data System

Our summarization system, called **Summarizer** [Neff & Cooper, 1999a, b], is based on a representation of the text produced by the Texttract information extractors. A document structure builder produces a structural representation of the document, identifying sections, headings, paragraphs, tables, etc. Currently, it is rudimentary, preferring text with structural tags to text with white space cues; however, there are plans to make it more robust. Texttract locates, counts, and extracts items of interest, such as names, multiword terms, and abbreviations, allowing all variant items to be counted together. Summarizer compares the frequency of the vocabulary items found in the text (including also single words but ignoring stop words) to the frequency of the same vocabulary in the collection vocabulary, using a tf^*idf measure (proposed by Brandow, et al. (1995), adapted from Salton and McGill (1993)).

Simply described, this version of tf^*idf (term frequency times inverted document frequency) measures how much more frequent, relatively, a term is in the document than it is in the collection. Items whose tf^*idf exceeds an experimental threshold are identified as signature terms. Further, items occurring in the title and in headings are added to the list of signature terms, regardless of their tf^*idf . The salience score for a

sentence (simplified here) is a function of the sum of the $tf*idf$'s of the signature words in it, how near the beginning of the paragraph the sentence is, and how near the beginning of the document its paragraph is. Sentences with no signature words get no "location" score; however, low-scoring or non-scoring sentences that immediately precede higher-scoring ones in a paragraph are promoted under certain conditions. Sentences are disqualified if they are too short (five words or less) or contain direct quotes (more than a minimum number of words enclosed in quotes). Documents with multiple sections are a special case. For example, a longer one with several headings or a news digest containing multiple stories must be treated specially. To ensure that each section is represented in the summary, its highest scoring sentences are included, or, if there are none, the first sentence(s) in the section.

Although earlier researchers (e.g. Brandow, et al.) have asserted that morphological processing and identification of multi-words would introduce complication for no measurable benefit, we believe that going beyond the single word alleviates some of the problems noted in earlier research. For example, it has been pointed out [Paice, 1990] that failure to perform some type of discourse processing has a negative impact on the quality of a generated abstract. Some discourse knowledge can be acquired inexpensively using shallow methodology. Morphological processing allows linking of multiple variants of the same term.

In actual use, we run the summarizer program on all of the documents in the collection, saving a table of the most salient terms (measured by $tf*idf > 2.2$) in each document (we arbitrarily select 10 as the cutoff) and a table of sentence number of the most salient sentences (which contain these terms) and the sum of the $tf*idf$ salience measures of the terms in each selected sentence. We also save the offsets of all of the occurrences of each salient term in the document to facilitate term highlighting. All of these data are then added to the relational database.

III. The Document Collection

The document collection we have been working with consists of about 7500 consultant reports on customer engagements by members of the IBM consulting group. There, reports had been divided into 50 different categories and each category had its own editing and submission requirements. They were originally stored in Lotus Notes databases and were extracted to HTML using a Domino server.

Many of these documents had attachments in a number of formats, including Word, WordPro, AmiPro, Freelance, Powerpoint, PDF and zip. All but the PDF and zip files were converted to HTML, and eliminated if

they were not in English as reported by the Linguini [Prager, 1999] language recognition tool. If they were English, the text was extracted and inserted into the documents.

A number of documents had attachments that were not usable, either because they were not in English, or because they were in a format for which no convenient HTML converter was available. For the purposes of this study, we treated such documents as short, English documents; for the most part they were found to be useless.

IV. Finding Useless Documents

Most of these reports were consultant reports of the usual nature, but some were simply templates for writing such reports, and a few were simply "managementese." While these latter types have a real social purpose in the collection, we needed to develop ways to recognize them and only return them when specific query types warrant them.

In the context of this work, we considered documents to be useless for our user population if

1. They were very short (such as "this is a test document.")
2. They were outlines or templates of how one should write reports.
3. They contained large numbers of management buzzwords but little technical content.
4. They were bullet chart presentations with little meat.
5. They were not in English.

While we could write filters to recognize some of these document types directly, we felt it would be more useful to consider some general approaches to finding and filtering out documents of low content.

We therefore decided to have some human testers evaluate a number of documents for usefulness, using the above-listed criteria as guidelines. We examined the rated-useful and rated-useless sets to determine if there were any properties of these sets that could be used by a classifier to make this distinction automatically.

Evaluating a Collection of Useful and Useless Documents

In order to study the problem of useless documents, we developed a web site powered by a Java servlet (Hunter, 1998; Cooper, 1999) which allowed a group of invited evaluators to view a group of documents and mark them as useful or useless. Figure 1 shows the selection screen where volunteer evaluators could select one of the 50 categories and then view and evaluate documents in that category.

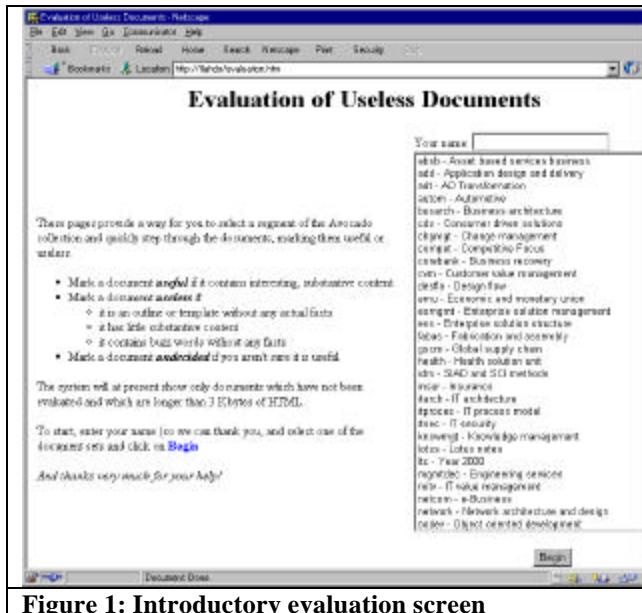


Figure 1: Introductory evaluation screen

Figure 2 illustrates the evaluation interface, where users could quickly rate a document and move on to the next one. In each case a hidden form variable contained the database document ID on the server and passed it back to the Java servlet that recorded the user's rating in the database.

We collected responses from 4 different raters who rated about 300 documents. Of these, about half of them were rated useful and half of them useless. We then extracted data from these documents to develop a statistical model describing the characteristics of useful and useless documents.

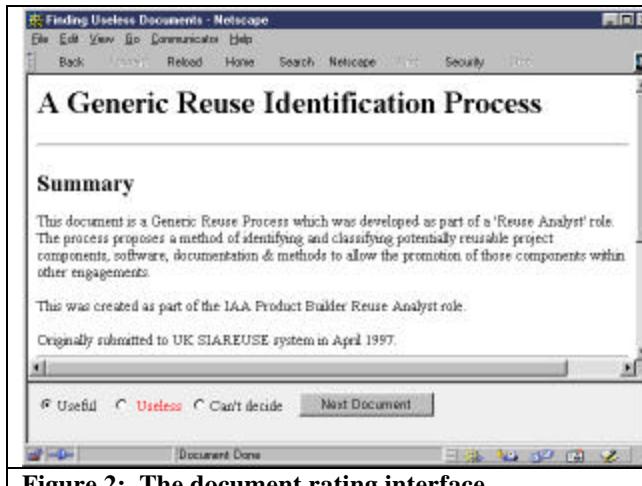


Figure 2: The document rating interface.

Characteristics of Useless Documents

Intuitively, we might expect useless documents to be

short and to contain almost no significant terms. Conversely, we would expect that useful documents would contain a number of significant terms and a number of high scoring sentences. Our goal was to use Textract statistics to develop a description of useful and useless documents.

The following is the entire content of a typical document selected by the evaluators as useless:

Since functional requirements are often specific to an industry, it is useful to capture the "fit" of a product for a given industry or industry segment. These can be narrow analyses based on an industry issue, or more broad based analyses based on actual client experience or detailed review of the software.

The purpose of this category is to hold these analyses, and to make them available for commentary by practitioners who use the information.

Clearly, this is an "orphan document," disconnected from its context and of little value to any reader. It is documents such as these that we would like to eliminate completely from search results.

An example of another typical document rated useless was one called "An Example of an End-User Satisfaction Survey Process." This document was essentially a blank survey form to be used at the end of some consulting engagement. While the questions contain salient terms, the overall impression was of an "empty" document. Accordingly, we investigated the statistical data we had available in our document database and extracted the following indicators that looked like they might be useful in predicting the usefulness of a document.

- The document's length.
- The number of high IQ words in the document as determined by Textract.
- The sum of the scores of sentences selected for summarization.
- The total count of high $tf*idf$ words in the document as determined by the terms selected for highlighting by the summarizer.

We also examined the number of named and unnamed relations that originated in a document, and concluded that this statistic was not at all selective.

Document Length

Document length is a definitive parameter of the extreme cases. All very short documents can be declared *a priori* to be useless, and all very long documents to be

useful without invoking further measures. The size cutoffs that you choose are dependent on the collection and how it was produced. In the case of our collection of consultant, the documents were originally stored in Lotus Notes, and exported to HTML for display and analysis; this export process generates quite a few invisible markup tags. Consequently, the threshold we used for short documents was higher than one might expect for general collections. We set our size window to label all documents less than 2000 bytes as useless and all documents greater than 40,000 bytes as useful. Otherwise, document length is not the best predictor of usefulness

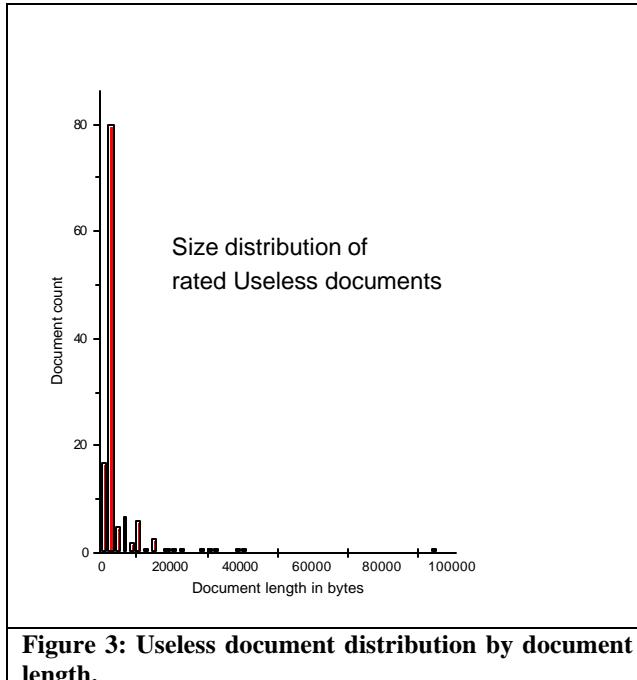


Figure 3: Useless document distribution by document length.

Figure 3 illustrates the document length distribution of the documents rated useless in our test set. While the preponderance of these documents are under 50,000 bytes, this is also true of the length distribution of the useful documents, as shown in Figure 4.

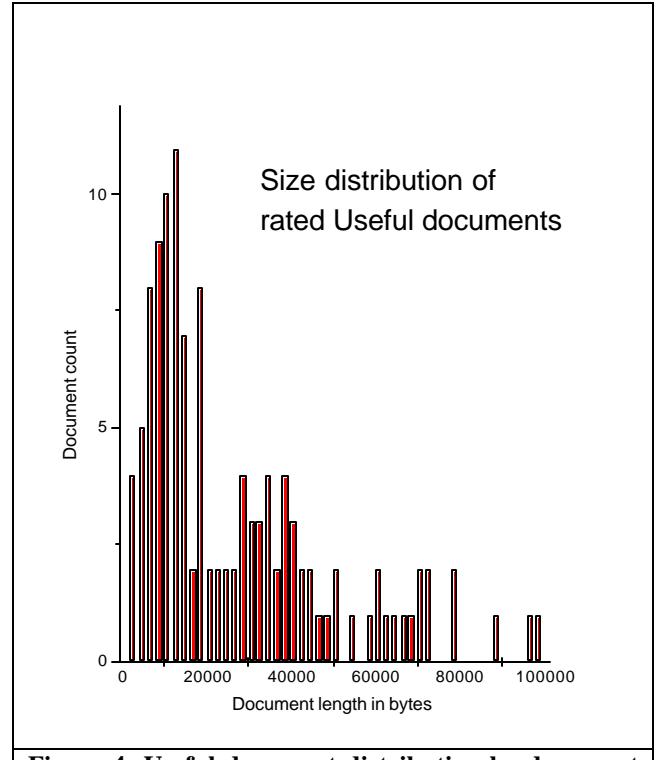


Figure 4: Useful document distribution by document length.

Thus, it is clear that length is not strongly correlated with document usefulness except at the extremes. In fact, at first glance, this same generalization can be made concerning each of the other quantitative measures we described above.

Many useless documents have a very low number of high-IQ words, but not all of them. Since IQ is developed on the basis of collection-level statistics, high-IQ words could occur in documents of no particular content, such as outlines, questionnaires and empty summaries.

Many useless documents have a low value for salient sentence scores, but not all of them. A salient sentence could be just one containing several high $tf*idf$ terms, but such scoring could occur in sentences of no particularly significant meaning.

Many useless documents contain a low number of high $tf*idf$ terms, but not all of them. Outlines, summaries of bullet-chart presentations and questionnaires could contain individual words of some significance without containing any net meaning.

Therefore, to examine how these parameters interact, it is instructive to plot several of these parameters

against one another and study the document space they occupy.

We have found that by plotting three of the parameters we have described in a 3D plot, we can easily discern the space occupied by rated-useful and rated-useless documents. In Figures 5 and 6, the *left-right* axis is the number of different high IQ words found in each document, the *front-back* axis is the sum of sentence salience scores obtained by the summarizer, and the *up-down* axis the total count of high $tf*idf$ words in each document.

Figure 5 shows the document space occupied by rated-useless documents and Figure 6 shows the space occupied by rated-useful documents. While there is some overlap between the space occupied by useful and useless documents, a careful examination of the documents in the region led us to conclude that most of the useful documents in this overlap area might be rated as useless by other readers.

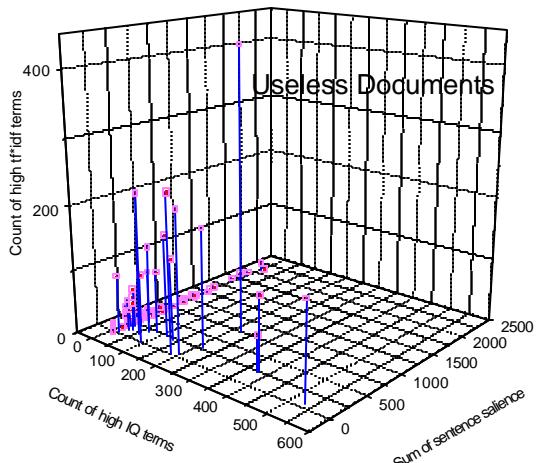


Figure 5: Document space occupied by useless documents

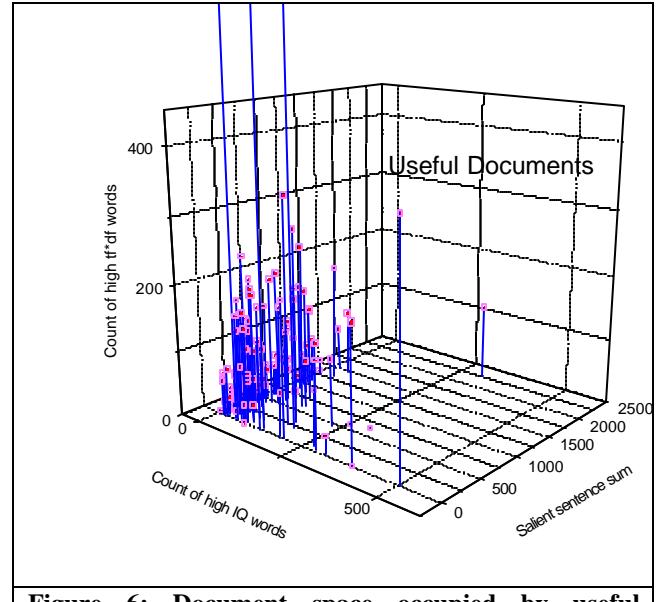


Figure 6: Document space occupied by useful documents.

As a first approximation, then, we can generalize from Figures 5 and 6, that

- All documents having a low count of salient $tf*idf$ terms are useless regardless of the score of sentence salience.
- Documents having a high number of high IQ words, but a low count of high $tf*idf$ words are useless.

At first, there appear to be a dozen or so low salience, low word count documents among the Useful documents which can lead to some ambiguity. However, in actual fact all of the documents in this region are quite long, on the order of 40,000 bytes, and would be selected as useful in any case. These documents are apparently ones written primarily in Italian, but which escaped our language filter for several mechanical reasons.

Experimental Results on the Rated Documents

For each document, we tabulated the number of words it contained having an IQ of 60 or more. Since the IQ measurement is statistically arranged to return values of 50 or less for less selective terms, this eliminated a number of noise words and terms of no consequence,

We also tabulated the document length, the number of occurrences of high $tf*idf$ terms, and the sum of salient sentence values and stored that back into the document table in the database.

The Algorithm

We found that we could predict 96 of the 149 useless

documents by selecting all documents with all of the following characteristics:

- Fewer than 5 terms with IQ greater than 60 (“high-IQ”).
- Fewer than 6 appearances of terms with a $tf*idf > 2.2$.
- size of less than 40,000 bytes

We assumed that all documents of less than 2000 bytes were useless and did not consider them in this study.

All of the documents judged useless by the raters that this algorithm did not select had high-IQ term counts greater than 5, and upon re-examination, we felt that their designation as useless was debatable. This algorithm also selected as useless 8 of the 137 documents that had been rated as useful.

Experiments on the Entire Collection

We then carried out the same experiments on the entire collection of 7557 documents. Using these parameters, the algorithm selected 797 as useless. We then reviewed each of these documents and found that only 20 of them could be considered useful.

This algorithm also helped us to discover nearly 40 documents that were quite long in byte count but appeared to be very short because the translation and combining programs had misplaced some HTML tag boundaries. The distribution of document size of the computed useless documents in the entire collection is shown in Figure 7.

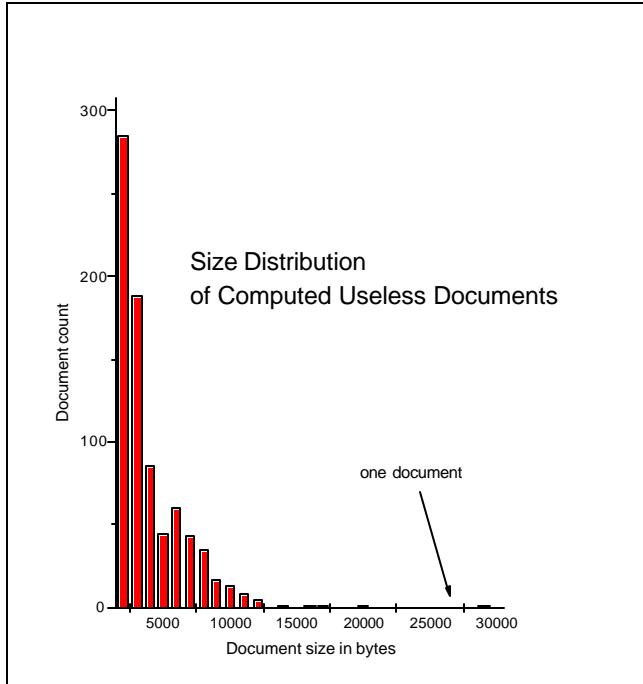


Figure 7: Size distribution of computed useless documents in the entire collection.

Computed-useless documents of substantial length probably deserve special consideration. These may be ones with misplaced tags. In general, when filtering returns of data from search engines, it is probably reasonable to remove useless documents entirely or relegate them to the bottom of the return list. However, documents of substantial length (such as more than 10,000 bytes) that are computed to be useless probably should be demoted, but not as much as shorter useless documents. We also examined the characteristics of the 20 documents we rated as useful, but which the algorithm had selected as useless. Nine of them had salience sum of more than 400. The distribution of salience sums for the computed-useless documents is shown in Figure 8.

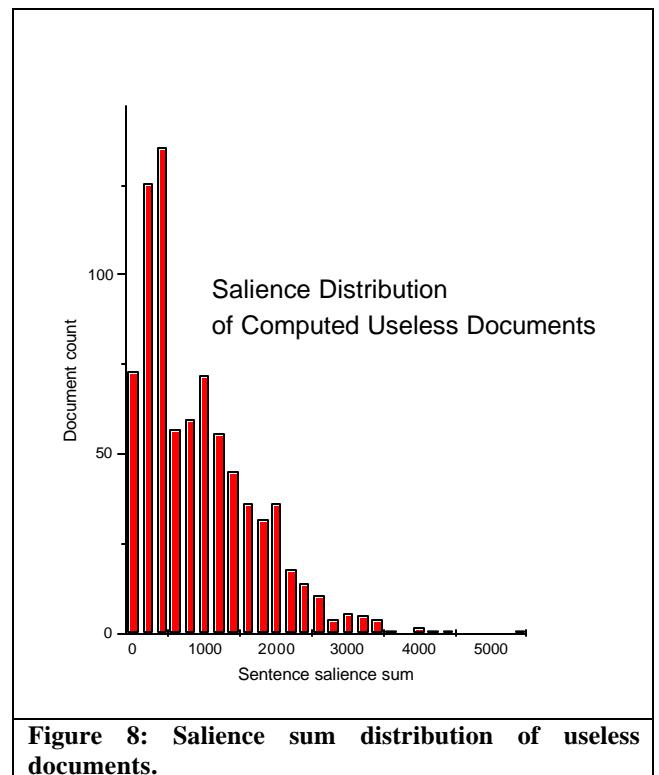


Figure 8: Salience sum distribution of useless documents.

Since salience sum does not appear to be a very powerful indicator of document usefulness, we treat it as a secondary, confirming parameter. However, in some cases, if a relatively short document has a high salience sum, it probably should not be eliminated completely. We suggest using it only as a discriminator in these few cases.

V. Document Similarity

There have been a number of approaches to the feature of “more documents like this one.” See, for example, papers on relevance feedback, Rocchio[1971], Harman[1992] and references cited therein. Recently, Allan[1998] *et. al.* have also investigated this problem.

The most common method for finding similar documents is simply to submit the entire document or a processed version of that document back to the search engine as a query. Depending on the search engine employed, this method can have serious performance problems when the original document is long. In the Avocado project, we submitted the top 10 terms from a document as ranked by IQ as a new query, instead of the entire document; however, we did not evaluate this procedure analytically.

As part of this project, we generated a query for each document in the collection based on the top 10 terms found in that document by Textract and submitted it to the Intelligent Miner for Text (IM4T) search engine. The purpose of the experiment was to determine whether these high IQ terms constituted an ideal query for returning similar documents.

One possible metric for arriving at the success of this technique is whether the target document is returned by the search engine near the top of the list of documents. If it is, then a query consisting of the top terms is considered successful. Figure 9 illustrates the success of this technique for 5959 of the documents in the collection.

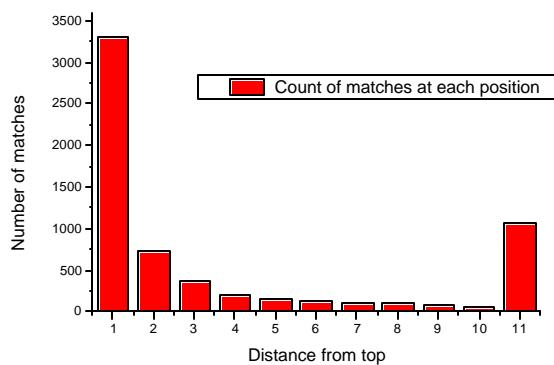


Figure 9: Distribution of documents returned in positions 1-10 after a document similarity query. Position 11 shows those that did not return any matches.

Clearly, for these approximately 6000 documents, the technique is very successful. Of the remaining 1578 documents in the collection, 1092 did not return in the top 10, and for 486, the beta version of the search engine did not complete the query.

The interesting question, however, is why these 1092 documents were not among the top 10 for a query. The most persuasive reason would be that they were useless documents, and in fact, correlating these missing documents with the documents returned by our useless algorithm provided the significant insight that nearly 800 of those were useless. While one could argue that this provides one more way of recognizing useless documents, it is fairly computationally intensive, and is probably better suited to indicate documents where the similarity technique will be less likely to work. In other words, since useless documents do not return themselves in such a process, it is very unlikely that they would return more useful documents from a query, but rather either random documents or none at all.

VI. Conclusions and Further Work

We have examined five predictors of document usefulness: document length, number of high IQ terms found, sum of salience of identified summary sentences, count of high $tf*idf$ terms, and number of terms participating in named or unnamed relations. Very short documents are invariably useless, and we found no correlation between number of terms participating in relations and usefulness. The strongest predictors of usefulness are the IQ word count and the total count of high $tf*idf$ terms.

We then examined using the top IQ words as query terms for document similarity. This was very successful for the preponderance of the documents in the collection, and for those where the query document was not returned in the top 10, 73% of those documents had been found to be useless.

While the cutoff parameters we used in these experiments were determined empirically and would not directly apply to other collections, we believe that these parameters are easily determined for a collection and that filters such as these can definitely improve the quality of documents returned from searches.

Acknowledgments

We'd like to thank Mary Neff, who developed both the summarizer system and the underlying term recognition technology, Yael Ravin, who developed the name and named relations recognizers, Zunaid Kazi, who developed the unnamed relations algorithms, Roy Byrd who developed the architecture for Textract, and Herb Chong and Eric Brown for their helpful discussions on finding similar documents. We are also particularly grateful to Len Taing, who helped devise Java programs to automate the search and ranking process. Finally, we'd especially like to thank Alan Marwick for his continuing support and encouragement of our efforts and

for helpful discussions on representing and analyzing these data.

References

- Allan, James, Callan, Jamie, Sanderson, Mark, Xu, Jinxi and Wegmann, Steven. *INQUERY and TREC-7*. Technical report from the Center for Intelligent Information Retrieval, University of Massachusetts, Amherst, MA. (1998)
- Brandow, Ron, Karl Mitze, and Lisa Rau, 1995. Automatic condensation of electronic publications by sentence selection. *Information Processing and Management*, 31, No. 5, 675-685.
- Buckley, C., Singhal, A., Mira, M & Salton, G. (1996) "New Retrieval Approaches Using SMART:TREC4. In Harman, D, editor, Proceedings of the TREC 4 Conference, National Institute of Standards and Technology Special Publication.
- Byrd, R.J. and Ravin, Y. Identifying and Extracting Relations in Text. *Proceedings of NLDB 99*, Klagenfurt, Austria.
- Cooper, James W. "How I Learned to Love Servlets," *JavaPro*, August, 1999.
- Cooper, James W. and Byrd, Roy J. "Lexical Navigation: Visually Prompted Query Expansion and Refinement." *Proceedings of DIGLIB97*, Philadelphia, PA, July, 1997.
- Cooper, James W. and Byrd, Roy J., OBIWAN – "A Visual Interface for Prompted Query Refinement," *Proceedings of HICSS-31*, Kona, Hawaii, 1998.
- Edmundson, H.P., 1969. New methods in automatic abstracting. *Journal of the ACM*, 16(2):264-285.
- Harman, Donna, Relevance Feedback and Other Query Modification Techniques, in Frakes and Baeza-Yates, (Ed), *Information Retrieval*, Prentice-Hall, Englewood Cliffs, 1992.
- Hendry, David G. and Harper, David J., "An Architecture for Implementing Extensible Information Seeking Environments," *Proceedings of the 19th Annual ACM-SIGIR Conference*, 1996, pp. 94-100.
- Hunter, Jason *Java Servlet Programming*, O'Reilly, 1998.
- Jing, Y. and W. B. Croft "An association thesaurus for information retrieval", in *Proceedings of RIAO 94*, 1994, pp. 146-160.
- Justeson, J. S. and S. Katz "Technical terminology: some linguistic properties and an algorithm for identification in text." *Natural Language Engineering*, 1, 9-27, 1995.
- Kazi, Zunaid and Byrd, R. J., paper in preparation.
- McKeown, Kathleen and Dragomir Radev, 1995. Generating summaries of multiple news articles. In *Proceedings of the 18th Annual International SIGIR Conference on Research and Development in Information Retrieval*, pages 74-78.
- Morris, A.G., Kasper, G. M., and Adams, D. A. The effects and limitations of automated text condensing on reading comprehension performance. *Information Systems Research*, pages 17-35, March 1992
- Neff, Mary S. and Cooper, James W. 1999a. Document Summarization for Active Markup, in *Proceedings of the 32nd Hawaii International Conference on System Sciences*, Wailea, HI, January, 1999.
- Neff, Mary S. and Cooper, James W. 1999b. A Knowledge Management Prototype, *Proceedings of NLDB99*, Klagenfurt, Austria., 1999.
- Paice, C., 1990. Constructing literature abstracts by computer: Techniques and prospects. *Information Processing and Management*, 26(1): 171-186.
- Paice, C.D. and P.A. Jones, 1993. The Identification of important concepts in highly structured technical papers. In R. Korfhage, E. Rasmussen, and P. Willet, eds, *Proceedings of the Sixteenth Annual International ACM Sigir Conference on Research and Development in Information Retrieval*, pages 69-78. ACM Press.
- Prager, John M., Lingui: Recognition of Language in Digital Documents, in *Proceedings of the 32nd Hawaii International Conference on System Sciences*, Wailea, HI, January, 1999.
- Ravin, Y. and Wacholder, N. 1996, "Extracting Names from Natural-Language Text," IBM Research Report 20338.
- Reimer, U. and U. Hahn, 1988. Text condensation as knowledge base abstraction. In *IEEE Conference on AI Applications*, pages 338-344, 1988.
- Rocchio, J.J., Relevance Feedback in Information Retrieval, in Salton G. (Ed.) *The SMART Retrieval System*, Prentic-Hall, Englewood Cliffs, 1971.
- Salton, Gerald and M. McGill, editors, 1993. *An Introduction to Modern Information Retrieval*. McGraw-Hill.
- Schatz, Bruce R, Johnson, Eric H, Cochrane, Pauline A and Chen, Hsinchun, "Interactive Term Suggestion for Users of Digital Libraries." *ACM Digital Library Conference*, 1996.
- Xu, Jinxi and Croft, W. Bruce. "Query Expansion Using Local and Global Document Analysis," *Proceedings of the 19th Annual ACM-SIGIR Conference*, 1996, pp. 4-11