Text Summarisation for Knowledge Filtering Agents in Distributed Heterogeneous Environments

H. Leong, S. Kapur and O. de Vel
Department of Computer Science
James Cook University of North Queensland
Townsville 4811 Australia.
{*helen|kapur|olivier*}*@cs.jcu.edu.au*

Abstract

The rapidly growing volume of electronic information available in distributed heterogeneous environments, such as the World Wide Web, has made it increasingly difficult and time-consuming to search for and locate relevant documents (in textual, visual or audio format). To relieve users of the burden of this task, intelligent tools that automate the search and retrieval tasks by generating profiles of user interests with minimal user interaction are required. In this paper, an intelligent knowledge filtering system (SAMURAI) and, in particular, the text summarisation and clustering modules of the system are described. Modules for extracting salient concepts in documents were built and evaluated on a variety of documents from different knowledge domains. A natural language processing approach based on part-of-speech tagging was used and compared with an alternative approach based on the well-known (and commonly-used) TFIDF information retrieval algorithm. Results show that, in the tagger-based approach, more informative keywords and phrases from each document set are extracted than in the TFIDF approach. Furthermore, the tagger-based system has a significantly reduced computation time compared with TFIDF, making it scalable to large document sets.

Introduction

The World Wide Web (henceforth, the Web) is a large number of different files containing graphics, animation or sound stored on a distributed heterogeneous network based on a client/server model. Within the Web, information is organised into a system of distributed hypertext consisting of documents, links and indexes. A great variety of information is typically retrieved and viewed simply by specifying its uniform resource locator (URL), a unique identification of the information. However, the continuous growth in popularity of the Web has caused an explosive increase in the quantity and diversity of Internet resources, and in the number of users who search through it. The large amount of available information has made it increasingly difficult and time-consuming to locate relevant documents on the Web.

To relieve users of the burden of this task, the building of intelligent tools that automate the search and retrieval tasks by generating profiles of user interests with minimal user interaction is required. Knowledge filtering undertakes the task of matching a set of documents retrieved from the distributed environment against a profile of the user's interests that has either been statically defined or in a dynamic model learned by successive queries generated by the user during previous interactions. The matching process is heavily dependent on the quality of the salient concepts extracted from the document set. Building intelligent tools that automate the search and retrieval tasks by focusing on the major issues of extraction of salient concepts from the document sets and of learning of the user's profile is the main goal of the work described in this paper.

Web Search Systems

There are already many systems for locating information available on the Web. A typical Web search system can be decomposed into three main components: a human or automatic component (referred to as *robots* or *spiders*) which gathers resources from the Web and builds an index, a database where this index is stored, and a search engine which allows the user to interrogate the database for information of interest (Internet Search Engines 1996).

The search engine component typically employs a keyword searching strategy to retrieve information from the database. Keywords are specified by the user via a simple interface and passed on to the search engine which searches its index for best matches. Each word in the index has an associated weight and it is this weighting information that determines which of the matches are best. Upon finding these best matches, the search engine usually returns a title, a 'summary' and the URL of the source documents. Many search systems have been implemented, including robot-based

search engines (WebCrawler (Pinkerton 1994), Lycos (Mauldin & Leavitt 1994), AltaVista (AltaVista 1996) etc...), list-based systems (ALIWEB (ALIWEB 1996), CUI W3 (CUIW3 1996)) and proxy search systems that forward a user query to multiple search engines (MetaCrawler (MetaCrawler 1996; Selberg & Etzioni 1995), SavvySearch (SavvySearch 1996)) concurrently.

Users retrieving information with Web search systems must contend with four major issues. Firstly, the user must make a conscious effort to find the information. Secondly, there is the constraint on the user to pass the most appropriate keywords to the system. Another issue is the need to filter out the many irrelevant links from the results obtained, despite the careful selection of keywords. Finally, there is no guarantee to the user that the information returned is the best that can be obtained. Better information could be missed by the system simply due to the fact that the best choice of keywords was not made.

Mainly due to the issues outlined above, it is difficult for a user to locate relevant information quickly using the systems currently available on the Web. The difficulty in producing more sophisticated systems is partly due to two major facets: the problem of indexing and the complexity of linguistic phenomena associated with natural language. Web search systems are inherently limited in their ability to distinguish relevant from irrelevant texts. The reason for this is that any natural language has many complex idiosyncrasies which keyword-based retrieval systems can not easily take into consideration. Mauldin (Mauldin 1991) describes two linguistic phenomena that, in particular, limit the ability of such systems to retrieve relevant information:

- Synonymy: various terms can be used to describe the same object.
- Polysemy: a single word can have multiple meanings.

Since the meaning of words is rarely analysed in these systems, due to problems such as polysemy and synonymy, a large amount of information is missed, or a large number of documents returned are irrelevant. A partial solution to the problem of polysemy is to index texts by word senses instead of words themselves (Mauldin 1991). However, it has been observed that the quantity and distributedness of information on the Web prevents the construction of such indices. The problem of synonymy can be partially solved by using a thesaurus to generate synonyms of keywords in a user query (Mauldin 1991). This would prevent relevant information from being missed but is likely to generate many irrelevant links.

Intelligent Knowledge Filtering

To partially overcome some problems associated with purely keyword-based retrieval, many intelligent search systems have been developed recently. These include BORGES (Smeaton 1996), WebWatcher (WebWatcher 1996; Joachims *et al.* 1995), LAW (Bayer 1995), WEBSOM (WEBSOM 1996; Lagus *et al.* 1996), and Syskill & Webert (SYSKILL 1996).

The BORGES system requires a user to specify a set of words or phrases describing their information needs. The system then highlights polysemous words in the user profile, displays alternative meanings for each of these, and asks the user to choose between the meanings. Once the relevant meaning of each polysemous word has been acquired from the user, BORGES expands the user profile to include terms related to the disambiguated meanings. Related terms are obtained using WordNet (WordNet 1996), an online lexical database.

Like the BORGES system, WebWatcher requires the user to specify keywords. To identify pages related to a given page, WebWatcher employs an algorithm which works under the assumption that two pages are related if some third page points to them both. The algorithm uses *mutual information* (Quinlan 1990) as a similarity measure for comparing hyperlinks. The LAW system provides help to a user searching for information on the Web by interactively suggesting links to the user as they traverse through the Web and also by employing a Web robot that autonomously searches for pages that might be relevant. Unlike WebWatcher, the LAW system does not identify relevant information through the structure of the hypertext. Instead the actions of a user (e.g. bookmarking or printing a page) are observed and features are extracted from those documents which the actions suggest are interesting.

Most of these systems have focussed on the learning of user profiles based on the traditional vector space retrieval model (Salton & McGill 1983) using measures of word frequencies. However, not much consideration to natural language processing techniques on a wide variety of documents (as opposed to narrow domains of applications) has been given. Consequently, it is virtually impossible for these systems to isolate relevant information since they have not dealt with the problems arising out of linguistic phenomena.

In this paper, we present an NLP approach to text summarisation in order to extract salient concepts from documents in the Web and use this to filter knowledge.

Extraction of Salient Concepts in Documents

Textual summarisation of documents[1] can be partly achieved by using word frequency heuristics. One common approach is the TFIDF (term frequency–inverse document frequency) word weighting heuristic (Salton & McGill 1983). This heuristic assigns weights to words in a document based on two measures:

1. the *term frequency* of a word w, $TF(w)$, which is the number of times w occurs in the document, and
2. the *document frequency* of a word w, $DF(w)$, which is the number of documents in which the word occurs.

There are many variations of the TFIDF formula. One commonly used version is

$$TFIDF(w) = TF(w) * log(\frac{|D|}{DF(w)}) \qquad (1)$$

where $|D|$ is the total number of documents in the document set (de Kroon, Mitchell, & Kerckhoffs 1996). Terms that appear frequently in one document (as measured by $TF(w)$), but rarely on the outside (as measured by $IDF(w) = log(\frac{|D|}{DF(w)})$, the inverse-document-frequency), are more likely to be relevant to the topic of the document. The words with highest valued weights are selected to form a vector that represents the document.

A rather different approach used for text summarisation is the part-of-speech tagger based HT-summarisation paradigm (Hooper & Theofanes 1995). In this paradigm, the speech tagger BPOST (Lum Wan *et al.* 1994) assigns each word in a document a part of speech in accordance with the Penn Treebank tagging system(Santorini 1991).[2] Four binary search trees are constructed; a tree each for the unigrams, the bigrams, the trigrams and the quadgrams in the document. Each node of a tree is given a score based on the frequency the node-words appear in the document and the number of nouns that feature in the node-words. The words/phrases which score the highest are extracted from the document.

[1] As will become clear later, our usage of the term "text summarisation" is somewhat unconventional. Traditionally the term is used for generation of a summary of a document for human consumption. Since the summaries we need in our system are used exclusively for further processing by the system itself, even though they are in a less natural format, they serve their purpose equally effectively and can be classified as such.

[2] BPOST is a Brill-style Part of Speech tagger (Brill 1993) written in the C programming language.

In this paper, we describe our implementations of both the TFIDF and the HT paradigms for text summarisation and compare their effectiveness.

An Intelligent Knowledge Filtering System

The two text summarisation paradigms were compared in the context of an intelligent information filtering system called SAMURAI (Smart Automatic Modeling of the User for the Retrieval of Appropriate Information) (Leong 1996; Leong, Kapur, & de Vel 1997). SAMURAI is a significant part of a comprehensive project aimed at transforming the Web into a far more effective Virtual Web (VWeb) described fully elsewhere (Kapur & de Vel 1996).

SAMURAI is made up of five major modules: *Text Summarisation*, *Monitoring and User Profiling*, *Search Engine*, *Filter Irrelevant Links*, and *Compile Results*. Figure 1 depicts the functional relationships among these modules.

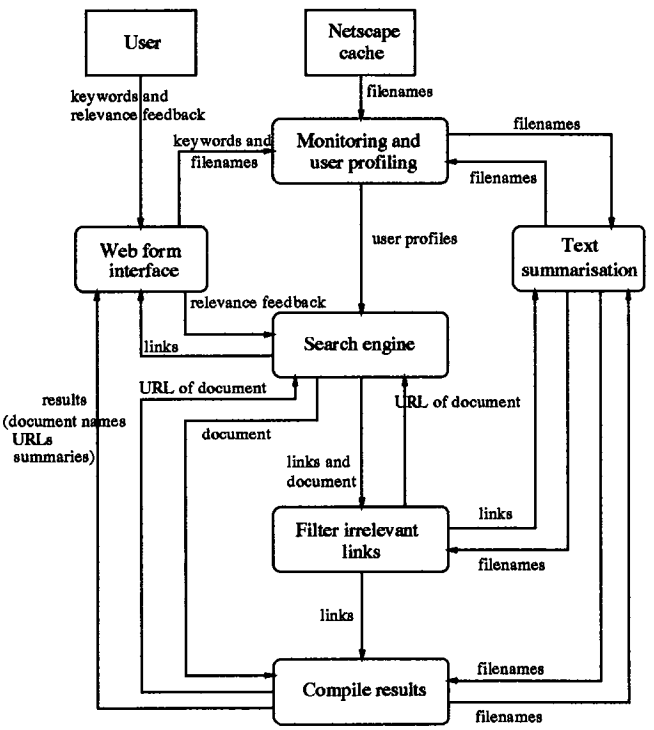

Figure 1: *Functional modules in SAMURAI*

The role of the Text Summarisation module is to extract salient words and phrases from documents that are passed to it. This module interfaces with the Monitoring and User Profiling, Filter Irrelevant

Links, and Results modules. The Monitoring and User Profiling module first determines which documents browsed by the user are relevant. Using these documents as training data, it constructs a *user profile*. The Monitoring and User Profiling module consists of a clustering sub-module based on the AutoClass classifier (Cheeseman 1990), and a generalisation sub-module for query refinement to enable identification of words that subsume the keywords of a query. The generalisation sub-module uses a lexical reference system, WordNet (WordNet 1996) (Miller *et al.* 1993), whose database incorporates word and phrase definitions in addition to semantic links between word definitions. Once the user profile has been generated, it is passed to the Search Engine module which can connect to any appropriate Web search system so that search can be conducted based on the user profile.

The results of the search are sent to the Filter Irrelevant Links module. This module filters out the irrelevant links from the search results. Links to pertinent documents are forwarded to the Results module which ranks and produces 'summaries' of the documents. Further details about rest of the modules and about SAMURAI itself are provided elsewhere (Leong 1996; Leong, Kapur, & de Vel 1997). Since the main focus of this paper is on the use of text summarisation for information filtering, we describe the Text Summarisation module in more detail next.

The Text Summarisation Module

The Text Summarisation module accepts a single HTML file as input. This file is converted to text using an HTML to ASCII converter (`lynx`). The module then employs either of the two approaches described above to extract keywords from a document: the TFIDF algorithm or the HT-based tool.

The first step under the TFIDF approach involves filtering out the common words from the text of the document. Weights are then assigned to each of the remaining words using the TFIDF algorithm. The keywords with greatest weights are returned by the module. If the option to personally specify keywords has been selected by the user, then the user-specified keywords will be assigned greater weight during the word-weighting process.

Under the alternative approach, the HT-based tool is used to obtain a set of salient words/phrases from the document. If the set contains any words/phrases that have been specified by the user, then these words are given greater weight. The results returned by the original HT-based tool did not take into account common words or duplicate words/phrases. The Text Summarisation module filters out common words from the results and removes any duplicates that may exist.

The design of the Text Summarisation module not only allows the keywords to be found in a single document, but also enables keywords to be extracted from a set of documents, as well as chapters (major headings), sections (minor headings), paragraphs or even sentences of a document. In other words, it is possible to construct a *summarisation hierarchy*. The top level of this hierarchy would feature summarised sets of documents. As deeper levels are traversed, the summarisation is carried out on smaller and smaller components of a document (e.g., chapter, section, paragraph) until, finally, sentence summaries are reached at the lowest level. Summarisation thus is context sensitive on the basis of this hierarchy. For example, if one of the keywords extracted from a document is 'bank', the context in which the word is used can be determined by traversing deeper into the lower levels of the hierarchy. The section level might feature words such as 'transactions' and 'money' which clarifies the intended sense of 'bank'. In this way, a summarisation hierarchy is used in disambiguating polysemous words.

Experimental Methodology

To evaluate the performance of the Text Summarisation module, the SAMURAI system was tested on a variety of information domains using five human subjects. The human subjects provided feedback on the relevance of the documents returned by SAMURAI.

The information domains consisted of two domain types: *narrow* domains and *wide* domains. The former domain–type covered a narrow information domain whereas the latter was much more general. The narrow domain–type also provided the user with two types of queries: *specific* and *general*. The specific query–type focussed the search for specific information. Examples include "Information about a young female pilot who died in a plane crash earlier this year in the US" or "What four–day Christian conference is held in England at at the Quanta Centre?". The general query–type searched for information that is more general in content. Example general queries covered a variety of topics including "marine life", "mafia", "natural disasters", "outer space", "sport" etc...

The query process consisted of four phases. In the first phase, users sent queries (either narrow or wide domain type) to the search engine. No constraints were imposed on the selection of keywords used by each user other than restriction to the information domain–type selected. The search engine used in all experiments was MetaCrawler (MetaCrawler 1996). In the second phase, the HTML documents returned by the search engine were processed by the Text Summarisation and

the Monitoring and User Profiling modules. In the third phase, the newly extracted keywords were submitted to the search engine for retrieval of a further, more refined, set of documents. Finally, the returned set of documents were processed by SAMURAI and the results are presented to the user for relevance feedback. The first paragraph of each document and the hyperlinks used were also returned. The user was asked to comment on the quality of the links and the quality of the document returned.

Results and Discussion

We present results from three narrow domain–types using a general query–type search: "mafia" (East and West), "natural disasters" (volcanos, tornadoes etc.) and "marine life" (dolphins, turtles etc.). The number of documents returned by each query was 13, 13 and 11, respectively. The total number of words in these document sets was around 20,000. Experiments were run with both the HT-tool paradigm and the TFIDF algorithm used in the Text Summarisation module.

For each document returned by the search engine, the HT-tool in the Text Summarisation (TS) module, TS-HT, produced a set of words with a numeric value indicating a measure of unigram, bigram, trigram or quadgram weight. For example, the bigram weight (BW) was given by

$$\begin{aligned} BW \;=\; & [\textit{freq. of bigram} \\ & + \textit{freq.(firstword)} \times \textit{weight(firstword)} \\ & + \textit{freq.(secondword)} \times \textit{weight(secondword)}] \\ & \times \textit{number of nouns}/2 \end{aligned}$$

and then normalised (on a scale of 0 to 10) by the number of words in the document. The heuristic expressions for the weights of the n-grams and the words were determined based both on intuition and on experimentation.

The application of the TFIDF algorithm in the Text Summarisation module (TS-TFIDF) on each document set produced a set of words with a numeric value indicating the TFIDF value for each word (see Equation 1). The values obtained by both TS-HT and TS-TFIDF modules have different meanings and are therefore not comparable. However, the words or phrases generated and their relative ordering within each list are significant.

Example TS-HT module outputs taken from the document set in each domain–type are given below.

The New Mafia Order	
Weight	*Keywords Extracted*
4.28	moscow reketiry
4.23	castellammare coast aliya ibvagimovich
4.12	scores
4.12	counterparts crime
4.12	languages police reports
4.04	mafia there's
4.00	sicily turkey russia
4.00	studies bank
3.95	mafia murders
3.90	aliya's

Drought Effects Felt on the Farm	
Weight	*Keywords Extracted*
4.06	berks farmers
4.06	corn prices
3.99	crop insurance
3.91	livestock
3.91	county
3.91	cantaloupes pumpkins
3.74	county's
3.64	yields
3.64	losses
3.54	sweet

Dolphin Abstracts	
Weight	*Keywords Extracted*
6.38	california bottlenose dolphins tursiops
5.98	san diego county
5.96	coastal
5.94	study
5.91	truncatus
5.80	dolphin
5.73	pacific
5.69	southern bight
5.68	fidelity
5.67	orange

Results indicate that the TS-HT module extracts informative keywords or phrases from each document set. This is particularly true of the "marine life" and "mafia" subject areas. In the case of the "disasters" topic, the speech-tagger seems to attribute a large weight to abbreviations even though they are neither nouns nor adjectives. Also, redundancy in some of the keywords was observed as well as the improper inclusion of some common words. The latter probably could be overcome with an improved stop–list.

For comparison, we include a sample output of the same files for the "mafia" and "marine life" domains using the TS-TFIDF module:

The New Mafia Order	
Weight	*Keywords Extracted*
0.09	moscow
0.08	mafia
0.07	sicily
0.07	russia
0.06	sicilian
0.06	aliya
0.05	palermo
0.05	organized
0.05	kosovo
0.05	italian

Dolphin Abstracts	
Weight	*Keywords Extracted*
3.63	dolphins
2.21	san
1.95	california
1.76	bottlenose
1.59	diego
1.24	study
1.06	tursiops
1.06	surveys
0.97	pacific
0.97	individuals

In some cases, it has been observed that the top four keywords extracted by both TS-HT and TS-TFIDF modules are the same. However, the ranking of the words may be different. Furthermore, the TS-HT module also extracts phrases and seems to extract more meaningful concepts than the TS-TFIDF module, primarily due to its ability to extract phrases rather than just single keywords. The TS-TFIDF module also extracts verbs which, in some cases, are not suitable for use as keywords for search engines. The extraction of phrases from documents may be more important for the search engine since providing a phrase generally returns more relevant documents than a single unqualified word. The computation time required by the TS-TFIDF module is significantly larger than that of the TS-HT module. Computation times for the TS-HT module was of the order of one minute maximum (for the document sets used) and up to forty minutes for the TS-TFIDF module. Consequently, the TFIDF algorithm does not scale well for large document sets.

Subsequent to processing by the TS-HT and Monitoring and User Profiling (MUP) modules, the best document clustering results were obtained for the "marine life" topic:

Class 0: *sea turtles, sea grass ecosystems: productivity and physiology, green sea turtles, leatherback sea turtles.*

Class 1: *dolphin abstracts, the wild dolphin project, report from annual dolphin conference, dolphins and man.*

Class 2: *seagrass ecosystems: systematic ecology, dugong, seagrass: the wasting disease.*

This classification was judged to be highly meaningful by all users. The worst clustering results were obtained for the "natural disasters" topic,

Class 0: *storm spotter's guide, volcanex, mount Erebus volcanic observatory, student's tornado experience, asian floods kill 1,100.*

Class 1: *floods kill hundreds in China, earthquake glossary, intro to tornadoes, earthquakes.*

Class 2: *Batur volcano, drought effects felt on the farm, the drought of 1996, Mount Vesuvio.*

There seem to be two classes for volcanoes (Class 0 and Class 2). However, upon examining the contents of the files it was observed that the "volcanex" document referred to a mining company (with no interest in volcanoes) and the "Mt Erebus" document contained many abbreviations which, as mentioned previously, the speech-tagger gave a disproportionate weighting. Removing these discrepancies, Class 0 and Class 1 collapse into a single class. Also, it was observed that improved classification performance was achieved if the AutoClass clustering time was increased (often by just a few seconds).

For the "mafia" topic, the classifications results were as follows:

Class 0: *the new mafia order, godfather IV, Alphonse 'Scarface' Capone, Charles 'Lucky' Luciano, Al Capone and friends*

Class 1: *the empire of crime, Yakuza - past and present, Yakuza - the Japanese mafia, Yakuza interview*

Class 2: *Lucky Luciano, mafia history, Al Capone, Zedillo's choice*

For comparison, the "mafia" and "marine life" document sets were also clustered after having been processed by the TS-TFIDF module:

Mafia:

Class 0: *the new mafia order, godfather IV, Charles 'Lucky' Luciano, mafia history, Lucky Luciano*

Class 1: *the empire of crime, Yakuza - past and present, Yakuza - the Japanese mafia, Yakuza interview*

Class 2: *Al Capone, Zedillo's choice, Alphonse 'Scarface' Capone, Al Capone and friends*

Marine Life:

Class 0: *sea turtles, green sea turtles, seagrass: the wasting disease, leatherback sea turtles*

Class 1: *seagrass ecosystems: systematic ecology, dolphin abstracts, the wild dolphin project, report from annual dolphin conference*

Class 2: *seagrass ecosystems: productivity physiology, dugong, dolphins and man*

In the "marine life" domain, the classification produced after the TS-TFIDF module was inferior to that obtained with the TS-HT. Turtle documents seemed to be clustered together, as were most of the dolphin documents. However, the information about the seagrass was spread out into each of the classes. This could be attributed to the superior performance of TS-HT on this set of documents.

In general, however, it was observed that the AutoClass-based MUP module produced better clustering of the output from the TS-TFIDF module rather than the output from the TS-HT module. This is thought to be due to the fact that AutoClass is biased towards TFIDF because it requires the use of input vectors with single keywords as vector components.

Conclusions and Future Work

In this paper we have presented an intelligent knowledge filtering system (SAMURAI) and, in particular, described the text summarisation and clustering modules of SAMURAI. The performance of the text summarisation and clustering modules was evaluated by testing the system on a variety of knowledge domains. Comparative performance evaluation of two alternative approaches to text summarisation was also undertaken. Results indicate that the TS-HT module successfully extracts informative keywords and phrases from each document set. The TS-HT module was able to extract more meaningful words than the TS-TFIDF module, partly due to its ability to extract phrases rather than just single keywords. Furthermore, the TS-HT module has a significantly reduced computation time compared with the TS-TFIDF module. The TFIDF algorithm does not scale well with large document sets.

Clustering with the AutoClass-based MUP module generally obtained better results with output from the TS-TFIDF module than output from the TS-HT module, this being mainly due to the bias of AutoClass towards TFIDF output consisting of vectors with single keywords as vector components. However, the extraction of phrases from documents may be more important for the search engine than single keywords since providing a phrase generally returns a more semantically-relevant document set than a single unqualified word.

More extensive analyses on large document sets with more detailed examination of user relevance feedback need to be undertaken to provide more evidence of the superior performance of the HT paradigm for text summarisation. An improved clustering algorithm that uses both keywords and keyphrases also needs to be developed. Development and evaluation of the generalisation sub-module (which follows on from the MUP module) for query refinement to enable identification of words that subsume the keywords of a query also needs to be carried out. It is hoped the result would be an improvement in the quality of keywords and phrases provided to the search engines.

References

ALIWEB. 1996. Aliweb. [web page]. Available at http://www.nexor.co.uk/public/aliweb/doc-/search.html.

AltaVista. 1996. Altavista. [web page]. Available at http://altavista.digital.com/.

Bayer, D. 1995. A Learning Agent for Resource Discovery on the World Wide Web. Master's thesis, Department of Computer Science, University of Aberdeen, Scotland.

Brill, E. 1993. *A corpus-based approach to language learning.* Ph.D. Dissertation, Department of Computer and Information Science, University of Pennsylvania.

Cheeseman, P. 1990. Bayesian Classification Theory. Technical Report FIA-90-12-7-01, NASA Ames Research Center.

CUIW3. 1996. Cui W3 Catalog. [web page]. Available at http://cuiwww.unige.ch/w3catalog.

de Kroon, H. C. M.; Mitchell, T. M.; and Kerckhoffs, E. J. H. 1996. Improving Learning Accuracy in Information Filtering. In *Thirteenth International Conference on Machine Learning Workshop on Machine Learning meets Human Computer Interaction.*

Hooper, S., and Theofanes, D. 1995. Save the World Wide Web. Computer Science Department project document at James Cook University of North Queensland.

Internet Search Engines. 1996. Internet Search Engines. [web page]. Available at http://rs.internic.net:80/scout/toolkit/search.html.

Joachims, T.; Mitchell, T.; Freitag, D.; and Armstrong, R. 1995. WebWatcher: Machine Learning and Hypertext. *Fachgruppentreffen Maschinelles Lernen.*

Kapur, S., and de Vel, O. 1996. Knowledge Structuring and Retrieval in a Virtual World Wide Web. Techical Report, James Cook University of North Queensland.

Lagus, K.; Honkela, T.; Kaski, S.; and Kohonen, T. 1996. Self-Organising Maps of Document Collections: A New Approach to Interactive Exploration. In *Proceedings of Second International Conference on Knowledge Discovery and Data Mining, KDD-96.*

Leong, H.; Kapur, S.; and de Vel, O. 1997. Intelligent Knowledge Filtering in Massively Distributed Heterogeneous Environments. In *Proceedings of 1997 Australasian Natural Language Processing Summer Workshop.*

Leong, H. 1996. SAMURAI - an intelligent system for the discovery of world-wide web resources. BSc (Hons), Computer Science Department, James Cook University, Australia.

Lum Wan, A.; Theofanes, D.; Taylor, D.; and Tracey Patte, T. 1994. Brill-Type Rule-Based Part of Speech Tagger. Computer Science Department project document at James Cook University of North Queensland.

Mauldin, M. L., and Leavitt, J. R. R. 1994. Web-Agent Related Research at the Center for Machine Translation. In *Proceedings of ACM Special Interest Group on Networked Information Discovery and Retrieval, SIGNIDR.*

Mauldin, M. L. 1991. Retrieval Performance in FERRET: A Conceptual Information Retrieval System. In *Proceedings of the 14th International Conference on Research and Develpment in Information Retrieval,* ACM SIGIR.

MetaCrawler. 1996. Metacrawler. [web page]. Available at http://metacrawler.cs.washington.edu:8080/.

Miller, G. A.; Beckwith, R.; Fellbaum, C.; Gross, D.; and Miller, K. 1993. Introduction to WordNet: An On-Line Lexical Database. Available at http://www.cogsci.princeton.edu/w̃n/.

Pinkerton, B. 1994. Finding What People Want: Experiences with the WebCrawler. In *Proceedings of the First International Conference on the World Wide Web.*

Quinlan, J. R. 1990. *C4.5 Programs for Machine Learning.* San Mateo, CA: Morgan Kaufmann.

Salton, G., and McGill, M. 1983. *Introduction to Modern Information Retrieval.* New York: McGraw-Hill.

Santorini, B. 1991. Part of speech tagging guidelines for the Penn Treebank project. Techical Report, Department of Computer and Information Science, University of Pennsylvania, Scotland.

SavvySearch. 1996. Savvysearch. [web page]. Available at http://guaraldi.cs.colostate.edu:2000/.

Selberg, E., and Etzioni, O. 1995. Multi-Service SErach and Comparison Using the MetaCrawler. In *Proceedings of the Fourth International Conference on the World Wide Web.*

Smeaton, A. F. 1996. Using NLP Resources in the BORGES Information Filtering Tool. In *Proceedings of the Third International Conference on Electronic Library and Visual Information Research, ELVIRA-3.*

SYSKILL. 1996. Syskill. [web page]. Available at http://www.ics.uci.edu/pazzani/Syskill.html.

WEBSOM. 1996. Websom. [web page]. Available at http://websom.hut.fi/websom/.

WebWatcher. 1996. Webwatcher. [web page]. Available at http://www.cs.cmu.edu/afs/cs.cmu.edu-/Web/People/webwatcher/.

WordNet. 1996. Wordnet. [web page]. Available at http://www.cogsci.princeton.edu/