Extracting Semantics of Documents Using Semantic Header Generator

Bipin C. Desai* Sami S. Haddad Tao Wang†

Department of Computer Science
Concordat University
Montreal, QC, Canada H3G 1M8

Abstract

Accurate representation of electronic information on the Internet underlies a solid foundation for precise information retrieval. However, the existing search systems tend to generate misses and false hits due to the fact that they attempt to match the specified search terms without context in the target information resource. It is clear that using traditional keywords-based methods for representing semantics of information items has become a major obstacle to high precision. In this paper, we propose the notion of Semantic Header to replace keyword indexing in extracting the meanings of information resources that marks explicitly the logical structure of a document. The information from the Semantic Header could be used by the search system to help locate appropriate documents with minimum effort. We also introduce an automatic tool, called Automatic Semantic Header Generator (ASHG), used for generating the meta-information for some significant fields of Semantic Header.

1 Introduction

Rapid growth in data volume, user base and data diversity render Internet-accessible information increasingly difficult to use effectively. At this time, a number of information sources, both public and private, are available on the Internet. They include text, computer programs, books, electronic journals, newspapers, organizational, local and national directories of various types, sound and voice recordings, images, video clips, scientific data, and private information services such as price lists and quotations, databases of products and services, and speciality newsletters [12]. There is a need for an automated search system that allows easy search for and access to relevant resources available on the Internet. Proper functioning of this system will require a proper indexing of the available information. Therefore, secondary information must be extracted and used as an index to the available primary resource. Building this index requires information extraction methods tailored to each specific environment. Unfortunately, the currently prevailing keyword-based methods suffer some significant shortcomings in completely representing the semantic information.

*bcdesai@cse.concordia.ca
†wan_tao@cs.concordia.ca

These defects lead most information search systems to have quite poor performance, that is, low precision and low recall.

We propose the notion of Semantic Header to address these problems above. The Semantic Header attempts to capture the semantics of information resource and relationship among different sections in a given information item. The detailed structure of Semantic Header is described in Section 2. We argue that using Semantic Header for supporting information retrieval systems has at least two major advantages over the traditional keyword-based indexing methods: 1) it can represent information resource more completely by including items such as title, author, keywords, subject, date, etc.; 2) it is capable to support flexible retrieval schemes possibly used by search engines according to system interface to users. There is a potential and dramatic increase in system response and retrieval precision.

This paper is organized as follows. Section 2 describes in detail the logical structure of Semantic Header. Section 3 presents our proposed schemes for automatically generating the major information in a Semantic Header. Section 4 tests the implemented system and analyzes the corresponding experimental results. We draw our conclusions in Section 5.

2 Structure of Semantic Header

There is an urgent need for the development of a system which allows easy search for and access to resources available on the Internet. Solving the problem of fast, efficient and easy access to the documents can begin by designing a standard index structure and building a bibliographic system using standardized control definitions and terms. Such definitions could be built into the knowledge-base of an expert system based index entry and search interface. The purpose of indices and bibliographies secondary information is to catalogue the primary information and allow easy access to it.

Preparing the primary source's meta or secondary information requires finding the primary source, identifying it as to its subject, title, author, keywords, abstract, etc. Since it is to be used by many users, it has to be accurate, easy to use and properly classified. Attempts to provide easy search of relevant documents has lead to a number of systems including WAIS, and more recently a number of Spiders, Worms and other creepy crawlers [9, 20, 28, 39, 65, 60, 68, 69, 70]. However, the problem with many of these tools is that their selectivity of documents is often poor [12]. The chances of getting inappropriate documents and missing relevant information because of poor choice of search terms is large. These problems are addressed by Concordia INdexing Discovering System, CINDI for short, which provides a mechanism to register, manage and search the bibliographic information.

For cataloguing and searching, CINDI uses a meta-data description called a Semantic Header to describe an information resource. The Semantic Header includes those elements that are most often used in the search for an information resource. Since the majority of searches begin with a title, name of the authors (70%),

subject and sub-subject (50%) [27], CINDI requires the entry for these elements in the Semantic Header. Similarly, the abstract and annotations are relevant in deciding whether or not a resource is useful, so they are included too.[56, 12]. A brief descripton of the semantic header elements follows:

- **Title, Alt-title:** The title field contains the name of the resource that is given by the creator(s). The alternate title field is used to indicate a secondary title of the resource.
- **Subject:** The subject and sub-subjects of the resource are indicated in the next field which is a repeating group. This field contains a list of possible subject classifications of the resource.
- **Language, Character Set:** The character set and the language are the ones used in resource.
- **Author and other responsible agents:** The role of the person associated with the document, for instance, author, editor, and compiler. This includes fields such as name, postal address, telephone number, fax number, and email address.
- **Keyword:** This field contains a list of keywords mentioned in the resource.
- **Identifier:** The identifiers for the document. Example of identifiers are, ISBN(International Standard Book Number), URL (Universal Resource Locator) of the document. This is a multi-valued slot in case the document is available in many formats or is electronically stored at more than one site.
- **Date:** The date on which the document was created, catalogued, and the date on which the document will expire, if any.
- **Version:** The version number, and the version number being superseded, if any, are given in these elements.
- **Classification:** The legal, security or other type of classification of the document. For each, nature of classification is specified.
- **Coverage:** It indicates the targeted audience of the document or it may indicate cultural and temporal aspect of the document's content.
- **System Requirements:** The document being an electronic one requires certain system requirements for it to be displayed or used. The components are the hardware, the software or the network and for each the minimum needs.
- **Genre:** It is used to describe the physical or electronic format of the resource. It consists of a domain and the corresponding value or size of the resource.

- **Source and Reference:** The Source indicates the documents being referenced or which were required in its preparation. It could also be the main component for which the current document is an addendum or attachment.
- **Cost:** In case of a resource accessible for a fee, the cost of accessing it is given.
- **Abstract:** The abstract of the document is either provided by the author or by ASHG.
- **Annotations:** Annotations put in by readers of the document.
- **User ID, Password:** A Provider ID of at least six characters and a password of four to eight characters. More than one semantic header by the same provider can have the same ID and password.

In the next section, we present a tool for automatically generating semantic information for the Semantic Header.

3 ASHG: Automatic Semantic Header Generator

In this section, we present the Automatic Semantic Header Generator (ASHG) of the CINDI system. This is an important step in building the Semantic Header. The design goal of ASHG is to automatically build a reliable Semantic Header, which includes classifying a document under a list of subject headings. ASHG's scheme measures both the occurrence frequency and positional weight of keywords found in the document. Based on the selected document's keywords, ASHG assigns a list of subject headings by matching those keywords with the controlled terms found in the Controlled Term Subject Association. The major steps applying ASHG are:

1. *Document Type Recognition*: In order to apply the correct ASHG to a document, the type of the document has to be recognised. The system currently understands HTML, LaTeX, RTF, PDF, and plain text documents. The PDF documents are first handled by using a converter to convert the PDF documents into text format and then applying the plain text converter.
2. *ASHG's Extractor Application*: The summariser corresponding to the type of document is applied to the input document.
3. *ASHG's Document Classification*: Each document is assigned a list of subject headings. The procedure of classification consists mainly of:
   (a) *Word stemming*: The system applies the stemming process to map the words found in the extracted fields onto a base root word.
   (b) *A Look up into the Controlled Term Subject Dictionary.*
4. *Semantic Header Validation*: The generated Semantic Header is presented to the user to validate.

3.1 Document Type Recognition

When a document is submitted to the ASHG, the system tries to recognize type of the document through the file-naming conventions. If it fails, the system will then examine this document's contents. The semantics of the ASHG's types are exploited when attempting to recognize the file type. In the case of such content analysis can not reveal the type of document, the author(s) of document recognized is asked to enter mannually the contents of Semantic Header.

3.2 Applying ASHG's Extractors

Based on the document's type uncovered in the document type recognition step, ASHG applies the type specific extractor to documents to extract meta-information for documents, such as title, keywords, dates of creation, author, author's information, abstract and size, etc.. In both *HTML* and *LaTeX* documents, the author might explicitly tag some of the fields to be extracted. In case that these fields were not explicitly tagged, ASHG attempts to extract them using some heuristics. For example, rds, dates of creation, author, author's information, abstract and size extracting the keywords in an *HTML* document, The *HTML_extractor* extracts words that are found in the *meta* tag field, if they were included by the author. However, if the explicit keywords were not found in the document, then words found in the title, abstract and other tagged words would be used to extract an implicit list of keywords.

3.3 Generating an implicit list of keywords and words used in Document classification

ASHG generates an implicit list of keywords in case that explicit keywords were not found in the document. It also derives a list of most significant terms, which is used in the document classification scheme. If keywords were not found in the document, the system derives a list of words from those found in the title, abstract, and other tagged fields. This list of derived words will also be used in classifying the document. However, if the keywords were explicitly stated in the document, then ASHG will include these as well the title, abstract, keywords and other tagged fields. This list is used in generating a list of subject headings for the document.

Generating both lists of words relies on the stemming process that will map the words into their root words, the stemmed word frequency of occurrence and the word location in the document. It uses the following algorithm in generating the list of implicit keywords, in case the keywords were not found in the document, and the words used in the classification scheme:

1. Extract the title, abstract and other tagged fields. If the document wasn't tagged such as in a plain text document, words found in the first two and last paragraphs and in the first sentence of each paragraph are selected. If given, the explicitly provided keywords are also appended.

2. English Noise words constitute usually around 30 to 50 per-cent of a document. The Information Retrieval community calls them the *Stop List*. These words are dropped from the extracted words.

3. The remaining words are sent to the stemming process. This process will remove the words' suffixes and prefixes. For example, the words: *cycled, cycler, cycling* and *cycles* are stemmed to the root term, cycle. The aim of the stemming process is to generate base word class, which include all the forms that could be generated from it.

4. Because the terms are not equally useful for content representation, it is important to introduce a term weighting system that assigns high weights for important terms and low weight for the less important terms [55]. Therefore, the weights constitute the importance of a word. The system assigns weights to both lists of root words. The weight assignment uses the following scheme:

   (a) If a word appears in the explicitly stated keywords, it is assigned a weight of five. Since authors explicitly state the keywords to convey some important terms, which their document covers, it is assigned the highest weight.

   (b) Usually, words found in the abstract are the second most important words, because this is where the author tries to convey his/her idea. Therefore, words found in the abstract are the second most significant and they convey the idea of the article more than any words found in other locations [51]. If the word appears in the abstract, it is assigned a weight of four.

   (c) If the word appears in the title, it is assigned a weight of three. For example, if the word *compute* is found in the title, it is assigned a weight of three.

   (d) If the word appears in the other tagged words, it is assigned a weight of two.

5. Each numeric weight is a class by itself defining the words' location. The system has the following classes:

   (a) A class weight of two defines the OTHER WORDS class. This class contains the terms found in only the Other Words field.

   (b) A class weight of three defines the TITLE class. This class contains all the terms found only in the Title field.

   (c) The class weight of four contains all the terms found only in the Abstract field, which therefore defines the ABSTRACT class.

   (d) A class weight of five includes all the terms found in either the Keywords' field or in the Title and Other Words fields.

   (e) A class weight of six includes all the terms found in both the Abstract field and the Other Words field.

(f) A class weight of seven includes all the terms found in either the Keyword and Other Words fields or the Abstract and Title fields.

(g) A class weight of eight contains all the terms found in Keywords and Title fields. For example, if the word *compute* appears in both the title and explicitly stated keywords, it is assigned a weight of eight. The word *compute* will be an element of the class weight of eight.

(h) A class weight of nine contains all the terms found in either the Abstract, Title and Other Words fields, or Abstract and Keywords fields.

(i) A class weight of ten contains all the terms found in the Other words, Title and Keywords fields.

(j) A class weight of eleven contains all the terms found in the Other Words, Abstract and Keywords fields.

(k) A class weight of twelve contains all the terms found in the Title, Abstract and Keywords fields.

(l) A class weight of fourteen contains all the terms found in the Other Words, Title, Abstract and Keywords fields.

A term appearing in Other Words field is less important than the one appearing in the Abstract field. Furthermore, a term appearing in both Title and Other Words fields is less significant than the one appearing in the Keywords, Abstract and Title field. We would want to extract more terms in higher weighted. We also limit the number of extracted terms using term's frequency of occurrence. Significant terms are those which have the highest frequency of occurrence. On the other hand, as the class weight increases, more of its terms should be regarded as significant ones.

Based on the above assumption, we set the maximum occurrence frequency of all terms found in that class the Maximum Class Frequency. For instance, if, in class four, there are three terms having occurrence freqency two, four and six, respectively, system would select six as Maximum Class Frequency for class four. Each word' frequency is then compared with its associated Maximum Class Frequency for significancy. For low weighted classes such as two and three, significant terms have the Maximum Class Frequencies. Thus, limiting the number of significant terms. However, all terms found in class eight and more are significant regardless of their frequency of occurrence.

6. Two lists of words will be generated. The first one containing only the root words or controlled terms found in CINDI's thesaurus. This list of controlled terms is used in the document's subject classification scheme. The second list contains the most significant root words not found in CINDI's thesaurus.

7. If no keywords were found in the document, ASHG extracts words having a term weight more than four and their corresponding frequencies of occurrence is the same as the ones tabulated. These words are the document's keywords.

Term Weight	Term Frequency
2	Maximum Class 2 Frequency
3	Maximum Class 3 Frequency
4	Greater or equal to Maximum Class 4 Frequency minus 1
5	Greater or equal to Maximum Class 5 Frequency minus 1
6	Greater or equal to Maximum Class 6 Frequency minus 2
7	Greater or equal to Maximum Class 7 Frequency minus 3
8 or more	All

Table 1: Weights and Frequency numbers used in extracting terms

8. In generating a list of controlled terms used to classify the document, terms having weight of two or more are extracted. The extracted words should have the frequencies of occurrence as the ones tabulated.

3.4 ASHG's Stemming Process

Stemming consists of processing a word so that only its stem or root form is left. Plural stemming attempts to identify and index the singular form of a term. Stemming attempts to identify and index the word *stem*. If a word and its stem are different, only the word stem is indexed. The stemming algorithm developed by Porter [44] at Cambridge uses weak stemming to remove common plural endings and other grammatical suffixes like *-ing* and *-ed* and implements strong stemming to remove derivational suffixes like *-ent, -ence*, and *-ision*. Many searchers use right hand truncation to find different variations of a search term that is of interest. The problem with right hand truncation is that it indiscriminately adds words to the query [72]. For example, if a searcher were to search for the truncated form of the word *cover*, the searcher would not only retrieve instances of the terms *covers, covering* and *covered* but also the terms *covert, coverall, coversheet* and *coverage*. QPAT-US [72] helps avoid extraneous right hand truncation terms by automatically performing a process called *stemming*. First, QPAT-US evaluates the terms for common suffixes that indicate plurality, verb tense, etc. If QPAT-US discovers these suffixes, it will strip them to find the root form of the term. For instance, if QPAT-US finds the term *covering* it will strip the suffix to obtain the root word of *cover*. Next, QPAT-US takes the root form of the search terms and, using sophisticated linguistic rules, creates a set of word variants. If the original term is *covering*, QPAT-US will also search for *cover, covers* and *covered*.

ASHG's stemming process implements the removal of both suffixes and prefixes of a given word in order to get the root of the word. For example, applying the stemming process on the words *simulation* and *analogies*, the words *simulate* and *analogy* are generated as their root words respectively. ASHG stores the root forms of the words.

Suppose the word impressionists is in a document for which meta-information is to be extracted. Without stemming, this would match only the keyword impressionists and not the singular form. Now suppose that

the word impressionist was in CINDI's list of controlled terms, then that document will miss that term and will not have it as a keyword. Following stemming, documents having the word impressionistic and impressionism will match the root term that is found in CINDI's list of controlled terms. We have mainly used the *spell* unix command in our system in extracting the root of a word. The *spell* command collects words from an input file and looks them up in a dictionary list. Words that neither occur among nor are derivable (by applying certain inflections, prefixes, and/or suffixes) from words in the spelling list are printed on the standard output. Two options were used along the *spell* unix command in our system: the *-v* option, in which all words not literally in the spelling list are printed, and plausible derivations from the words in the spelling list are indicated, and the *-x* option, in which every plausible stem is displayed, one per line, with = preceding each word. The steps of the ASHG stemming process are:

1. Using the *sort* unix command, sort the input words.
2. Apply the *uniq* unix command to filter out duplicate words.
3. Apply the *spell* command with the *-x* option. Thus, all the plausible stems are stored in an output file.
4. Apply the *spell* command with the *-v* option. Thus, all words not found in the spelling list are stored.
5. Create a file which contains the words found in step 3 but not in step 4.
6. Apply the *spell* command with the *-v* option to each word found in the file that resulted from the previous step. If the resulting output is empty, this means that this root word is found.

3.5 ASHG's Document Subject Headings Classification scheme

An important step in constructing the semantic header is to automatically assign subject headings to the documents. The title, explicitly stated keywords, and abstract are not enough by themselves to convey the ideas or subjects of the document. Since the author tries to convey or to summarise his ideas in the previously mentioned fields, there is a need to use all English none noise words found in those fields. To assign the subject headings, ASHG uses the resulting list of significant words generated from the previous section and CINDI's controlled term subject association. The subject heading classification scheme relies on passing weights from the significant terms to their associated subjects, and selecting the highest weighted subject headings.

3.6 The ASHG Subject Generation

Having the keywords, title words, abstract words and other tagged words, will help us select the most appropriate subjects for a given document. The following algorithm is used:-

1. Three lists of subject headings are to be constructed. The list of Level_0 subject headings, the list of Level_1 subject headings and the list of Level_2 subject headings.

2. For each term found in both CINDI's controlled terms and the generated list of words, the system traces the controlled term's attached list of subjects (list of *level0, level1 and level2*) headings, and adds the subject headings to their corresponding list of possible subject headings.

3. Weights are also assigned to the subject hierarchies. The weight for a subject is given according to where the term matching its controlled term was found. A subject heading having a term or set of terms occuring in both title and abstract, for instance, gets a weight of seven. The matched terms' weights are passed to their subject headings.

4. The system extracts *Level_2, Level_1* and *Level_0* subject headings having the highest weights from the three lists of possible subject headings.

5. After building the three lists for the three level subject headings, the system :

   (a) selects the subjects using the bottom-up scheme.

   (b) Having selected the highest weighted *level_2* subject headings, the system derives their *level_1* parent subject headings.

   (c) An intersection is made between the derived *level_1* subject headings and the list of the highest weighted *level_1* subject headings. The common *level_1* subjects are the document's *level_1* subject headings.

   (d) The system uses the same procedure in selecting *level_0* subject headings.

4 Analysis of ASHG's Results

In this section, we illustrate how the ASHG system extracts the meta-information from the HTML, Latex and text documents, and we demonstrate ASHG's automatic subject headings classification. For each of these document types, we apply ASHG and show the results. We compare the subject classification generated by ASHG with that of INSPEC for the same set of documents. We also compare the results with what the papers' authors would regard as good subject classifications and poor ones.

4.1 Reduction of Controlled Terms

Salton et al [55] introduces the term weighting system that assigns high weights to terms deemed important and lower weights to the less important terms. The term weighting system favours terms with high frequency in particular documents but with a low frequency overall in the collection.

ASHG's controlled terms favours the terms that have low frequency in the ASHG's subject headings over the terms having high frequency. Controlled terms having high frequency are dropped from the ASHG's list of controlled terms. Terms having lower frequency distinguish the subject headings associated for the document. The controlled term *system* occurs two hundred and eleven times in the ASHG's subject headings, which is the highest frequency control term. Therefore, it is dropped from the ASHG's list of controlled terms. Other control terms such as section, two, three, function, and method were dropped due to their ambiguity. The following table shows the words that are dropped and their corresponding frequencies.

Words	Frequency
system	211
power	115
design	106
electric	100
circuit	96
application	93
language	87
device	84
measure	83
general	72
manage	71
information	70
analysis	69
miscellaneous	58
other	47

Table 2: Words Dropped from the list of controlled terms

4.2 Two Experiments

The experiments presented below are designed to test accuracy of contents contained in Semantic Headers generated by ASHG. Of which, the field of Subject Headings is primarily focused on because of its difficulty to extract. Two relevant experiments on two different collections of documents are conducted: one collection consisting of papers submitted by professor at Concordia University, another choosing from autholoty of ACM. The corresponding experimental results are evaluated and tabulated.

4.2.1 Experiment on HTML, LaTeXand Text Documents

This experiment was conducted on thirty three documents, which had been submitted in HTML, LaTeX, or text format, by professors at Concordia University. Due to less number of documents submitted, we convert them into three different format sets: HTML, LaTeX, and text, and then apply the corresponding extractor to them. ASHG can extract all the explicitly stated fields such as title, abstract, keywords, and author's

information with a hundred percent accuracy. If the abstract was not explicitly stated, ASHG was able to automatically generate an abstract that would describe the paper. However, ASHG's implicit keyword extraction generated a list of words which included some words that are insignificant. These insignificant words in turn lead to the diversion in subject classification.

We have consulted the authors of papers on the ASHG's subject classification results. Their response was divided into three categories: *good, OK/Not sure* and *poor* subject hierarchy selection. Good subject hierarchy selection implied that the authors would have chosen them as subject hierarchies for the documents. *OK/Not sure* subject hierarchy selection implied that the authors doubt the results and they would not choose them. Finally, the *poor* subject hierarchy selection implied that the selected subject hierarchies described another different subject. We compared the ASHG's subject classification results against the INSPEC's classification done by expert cataloguers and thesaurus. Some of the ASHG's subject classification had different words than INSPEC's even though they described the same subject. That was due to the fact that our computer science subject classification was built from ACM and not from INSPEC. The tabular results of experiment are presented by three tables, each of which corresponds to the specific docuement format.

4.2.2 Experiment on PDF Documents from ACM

It is evident that the PDF format is becoming the dominant one for document publishing since its quality, flexibility, and readiness. Processing documents stored in PDF format is therfore a necessary functionality for an applicable inforamtion system. Given the facts above, we apply a collection of five hundreds of PDF documents from ACM archive [73]. The major procedure of experiment includes:

- Picking a PDF document from ACM authology
- Converting the selected PDF document into a text one using PDF-to-Text converter
- Applying ASHG to the converted document to generate Semantic Header
- Evaluating experimental results with original documents and INSPEC's results

The PDF-to-Text converter adopted in this experiment is Xpdf-2.01, an open source sofeware package. We have also made some modifications to enhance converting quality, which include consistent word spacing, more suitable two-column conversion, and superscript/subscript seperation. These improvements made help ASHG extract more precise contents for each field of Semantic Header.

The experimental results are eventually evaluated using two methods: 1) for fields of Title, Author, Abstract, keywords, we compare manually the generated contents with their couterparts on original documents using distinct notations for performance indication. For example, we put for field of Abstract a notation "G=E" representing case that the generated abstract is exactly the same as one stated on document. A thorough explanation for all notations used in evaluation is given at end of subsection; 2) for field of Subject

Headings, we compare with INSPEC's classification results. These results are given by domain experts and therefore are viewed as being authoratative. However, CINDI's subject hierarchy has a different struture from one used in INSPEC. We judge the coverage of subject headings generated for those given by INSPEC at our discretion.

We excerpt a frame of experimental results for tewenty-two documents for presentation. Evaluation data in the table shows a significant accuracy on fields of Title, Abstract, Author in Semantic Header. Nonetheless, Subject Heading field give a relative low percentage of correctness, which is 38.9. The main reason is that subject hierarchy in ASHG differs from one in INSPCT. It leads to a different collection of control terms to make a classification decision. One possible approach to improving classification accuracy is to adopt the current subject hierarchy in INSPCT.

AS completeness, we give explanation for all notations used in evaluation statistics:

1. **Title**
   "G=E" means that title generated is exactly the same as title stated;

2. **Author**
   "m/n" means that m authors are extracted while there are n authors stated;

3. **Abstract**
   "G=E" means that abstract generated is exactly the same as one stated;
   "G" means that ASHG generates an abstract while no abstract on document;

4. **Keywords**
   "G=E" means that keywords generated is exactly the same as one stated;
   "G" means that ASHG generates a set of keywords while no keywords explicitly stated on document;

5. **Subject Headings**
   "m/n" means that m subjects headings are judged being suitable in comparision with n subject headings given by INSPEC.

5 Conclusions

In this paper, we present a method and its implementation, known as ASGH, for automatically generating Semantic Headers for documents in HTML, LaTeX, Text, and PDF format. We also construct CINDI's three-level subject hierarchy for domains of Computer Science and Electrical Engineering, under which documents are properly classified (categorized) by ASHG. CINDI's computer science subject hierarchy is

Format	Average Number of Subject Headings Generated by ASHG	Average Author's Opinion			Average Accuracy A: Author I: INSPEC	Average OK/Good's Accuracy
		Good	OK/Not Sure	Poor		
HTML	5.38	1.27	2.00	2.11	23.9% (A) 20.7% (I)	66.1%
LaTeX	6.31	0.95	2.00	3.36	15.1% (A) 24.6% (I)	46.8%
Text	8.91	1.29	1.86	5.76	14.5% (A) 18.7% (I)	35.4%

Table 3: Test Results of 32 Documents Submitted by Authors and Converted into Different Formats

Document No.	Title	Author	Abstract	Keywords	Subject Headings
D001	G=E	3/4	G=E	G=E	1/3
D002	G=E	1/2	G=E	G=E	1/4
D003	G=E	1/1	G=E	G=E	2/3
D004	G=E	1/2	G=E	G=E	1/2
D005	G=E	1/3	G	G=E	1/4
D006	G=E	1/2	G=E	G=E	1/3
D007	G=E	1/2	G=E	G=E	1/4
D008	G=E	1/1	G=E	G	1/3
D009	G=E	2/2	G=E	G	1/5
D010	G=E	1/1	G=E	G=E	1/3
D011	G=E	0/3	G=E	G=E	2/4
D012	G=E	3/4	G	G=E	2/5
D013	G=E	3/4	G	G=E	1/4
D014	G=E	3/3	G=E	G=E	1/3
D015	G=E	1/2	G=E	G	1/1
D016	G=E	2/2	G=E	G=E	2/2
D017	G=E	4/4	G=E	G=E	1/3
D018	G=E	3/4	G	G=E	1/3
D019	G=E	2/2	G	G=E	1/4
D020	G=E	0/2	G=E	G=E	2/5
D021	G=E	1/2	G=E	G=E	1/3
D022	G=E	4/4	G=E	G=E	1/2
Average	100%	78%	100%	100%	38.9%

Table 4: Table of Evaluating Accuracy of PDF Documents' Semantic Header Generated by ASHG

Document	Title	Journal	Volume
D001	Broadcast Protocols to Support Efficient Retrieval from Database by Mobile Users	TODS	Vol 24, 1999
D002	Database Design for Imcomplete Relations	TODS	Vol 24, 1999
D003	Temporal FDs on Complex Objects	TODS	Vol 24, 1999
D004	Optimization of Queries with User-Defined Predicts	TODS	Vol 24, 1999
D005	GIOSS: Text-Source Discovery over the Internet	TODS	Vol 24, 1999
D006	Distance Browsing in Spacial Database	TODS	Vol 24, 1999
D007	Supporting Valid-Time Indeterminacy	TODS	Vol 24, 1999
D008	Safe Query Language for Constraint Databases	TODS	Vol 24, 1999
D009	Safe Stratified Datalog with Integer Order Does Not Have Syntax	TODS	Vol 24, 1999
D010	Optimizing Techniques for Queries with Expensive Methods	TODS	Vol 24, 1999
D011	Multi-View Access Protocol for Large-Scale Replication	TODS	Vol 24, 1999
D012	Ensuring Consistency in Multi databases by Preserving Two-Level Serialization	TODS	Vol 24, 1999
D013	An Access Control Model Supporting Periodicity Constraints and Temporal Reasoning	TODS	Vol 24, 1999
D014	Conceptual Schema Analysis: Techniques and Application+N31	TODS	Vol 24, 1999
D015	An Efficient Method for Checking Object-Oriented Databases Schema Correctness	TODS	Vol 24, 1999
D016	Information Gathering in the World Wide Web: The W3QL Query Language and the W3QL System	TODS	Vol 24, 1999
D016	Information Gathering in the World Wide Web: The W3QL Query Language and the W3QL System	TODS	Vol 24, 1999
D017	Towards a Theory of Cost Management for Digital Libraries and Electronic Commere	TODS	Vol 24, 1999
D018	Inverted Files Versus Signature Files for Text Indexing	TODS	Vol 24, 1999
D019	Extended Ephemeral Logging: Log Storage Management for Applications with Long-Lived Transactions	TODS	Vol 24, 1999
D020	Outerjoin Simplification and Recording for Query Optimization	TODS	Vol 24, 1999
D021	An Axiomatic Model of Dynamic Schema Evolution in Objectbase Systems	TODS	Vol 24, 1999
D022	Logical Design for Temporal Databases with Multiple Granularities	TODS	Vol 24, 1999

Table 5: Table of Source of Documents Used in PDF Experiment

based on ACM and CINDI's electrical engineering subject hierarchy is based on INSPEC. LCSH is used to augment both subject hierarchies. We derive control terms from CINDI's subject headings. These control terms were associated with their subjects in CINDI's thesaurus. In addition, we presented a method of generating a Semantic Header, called ASHG. This scheme automatically extracts and generates an index or meta-information.

ASHG exploits the file naming conventions and the data within a document to determine the document's file type. ASHG exploits the semantics of the document's types in extracting the meta-information. It also applies automatic abstracting proposed by Luhn in generating document's abstract. It also assigns weights for terms depending on their location in the document. Both term weight and occurrence frequency were used in assigning terms for a document. These extracted terms were used to classify a document using the association between CINDI's controlled term and their subject headings in the thesaurus.

Finally, we apply ASHG to two collections of documents for evaluation. For the contents of Subject Headings, We compare the results with actual assignments made by INSPEC. We also consulted the papers' authors on ASHG's subject classification results. The results show hundred percent accuracy in extracting the explicitly stated fields such as the title, abstract, author and keywords. They also showed some level of accuracy in generating the abstract.

Because our controlled terms were composed of terms found in CINDI's subject headings, ASHG's results showed a low degree of accuracy in classifying a document. The main reason was that some of the extracted terms were misleading. For example, the term *wire* should not be extracted unless it is followed by another term such as *wire grid*. The classification scheme used by ASHG showed some ineffectiveness, because it was based on term frequency and location information. For example, term-based retrieval cannot handle the following properties:

1. Different words may be used to convey the same meaning.
2. The same words may be used but they can have different meanings.
3. Different people may have different perspectives on the same single concept.
4. The same words may have different meanings in different domains.

Another weakness with ASHG is that it has not considered the issue of synonymity between words or between the subject headings.

In conclusion, we believe that resolving word senses and determining the relationships that those words have to one another will have the greatest impact on refining the ASHG's subject classification scheme. Therefore, the semantic level language processing should be handled by ASHG in the future.

References

[1] Alvarado S., et al, *Argument comprehension and retrieval for editorial text*, Knowledge Based Systems 3 (3), pp. 139-162, 1990.

[2] Andrews K., *The development of a fast conflation algorithm for English*, Dissertation submitted for the Diploma in Computer Science, University of Cambridge, 1971.

[3] *Automatic indexing and classification of the WAIS databases*:
http://www.ub2.lu.se/autoclass.html.

[4] Baxendale P. B., *Man made Index for Technical Literature - An Experiment*, IBM Journal of Research and Development, 2:4, pp. 354-361, 1958.

[5] Belkin N., Croft W. B., *Retrieval techniques*, Annual review of information science and technology (ARIST), 22, pp. 109-145, 1987.

[6] Blair D. C. , *Language representation in Information Retrieval*, Elsevier Science publishers, New York, 1990.

[7] Brandow R. , Mitze K., Rau L. F., *Automatic condensation of electronic publications by sentence selection*, Information Processing and management, Vol. 31, No. 5., pp. 675-685, 1995.

[8] Chiaramella Y. et al., *IOTA: a full text information retrieval system*, In proceedings of the ACM conference on research and Development in information retrieval, edited F. Rabitti, Pisa, pp. 207-213, 1987.

[9] De Bra, P., Houben, G-J., & Kornatzky, Y., *Search in the World-Wide Web*,
http://www.win.tue.nl/help/doc/demo.ps

[10] Desai, B. C., *An Introduction to Database Systems*, West, St. Paul, MN 1990.

[11] Desai B. C., *Cover page aka Semantic Header*,
http://www.cs.concordia.ca/~faculty/bcdesai/semantic-header.html, July 1994, revised version, August 1994.

[12] Desai B. C., *The Semantic Header Indexing and Searching on the internet*, Department of Computer Science, Concordia University. Montreal, Canada, February 1995.
http://www.cs.concordia.ca/ faculty/bcdesai/cindi-system-1.1.html

[13] DuRoss Liddy E., *Anaphora in natural language processing and information retrieval*, Information Processing and Management, Vol. 26, No. 1, pp. 39-52, 1990.

[14] Earl L. L., *Experiments in Automatic Extracting and Indexing*, Information Storage and Retrieval, 6:4, pp. 313-334, October 1970.

[15] Edmundson H. P., *Problems in Automatic Abstracting*, Communications of the ACM, 7:4, pp. 259-263, April 1964.

[16] Edmundson H. P. and Wyllys R. E., *Automatic Abstracting and Indexing Survey and Recommendations*, Communications of ACM, 4:5, pp. 226-234, May 1961.

[17] Edmundson H. P., *New methods in Automatic Extracting*, University of Maryland, college park, Maryland, Journal of the Association for computing machinery, Vol. 16, No. 2, pp. 264-285, April 1969.

[18] Evans D. A. , *Concept management in text via natural language processing: the CLARIT approach*, In working notes for the AAAI spring symposium on Text-based intelligent systems. Stanford 1990.

[19] Fung R. and Del Favero B. , *Applying Bayesian Networks to Information Retrieval*, Communications of the ACM, Vol 38, No. 3, pp. 42-57, March 1995.

[20] Fletcher, J. 1993., Jumpstation,
http://www.stir.ac.uk/jsbin/js

[21] Graham I., *Introduction To HTML and URLs*:
http://www.utoronto.ca/webdocs/HTMLdocs/NewHTML/intro.html, Last Update: 24 January 1997.

[22] Hardy D. R., Shwartz M. F., *Customized Information Extraction as a Basis for Resource Discovery*, Department of Computer Science, University of Colorado. March 1994; Revised February 1995.

[23] Jacobs, P.S. and Rau, L.F., *SCISOR: extracting information from online news*, Communications of the ACM, Vol. 33, No. 11, 1990.

[24] Jacqueline W. T. Wong, W. K. Kan, Gilbert Young, *ACTION: Automatic Classification for full-text documents*, ACM Transactions on information systems, pp. 26-41, 1997.

[25] Jing Y. and Croft B. W., *An Association Thesaurus for Information Retrieval*, Department of Computer Science, University of Massachusetts at Amherst, Amherst, MA 01003.

[26] Johnson F. C., Paice C. D., Black W.J, Neal A. P., *The application of linguistic processing to automatic abstract generation*, Journal of Documentation and Text management, 1993.

[27] Katz, W. A., *Introduction to Reference Work*, Vol. 1-2 McGraw-Hill, New York, NY.

[28] Koster, M., *ALIWEB(Archie Like Indexing the WEB)*,
http://web.nexor.co.uk/aliweb/doc/aliweb.html

[29] Krovetz R., Croft W. B.. *Lexical ambiguity and information retrieval*, ACM Transactions on information systems, Vol. 10, No. 2, pp. 115-141, April 1992.

[30] Kupiec, J. , Pederson, J., and Chen F., *A trainable document summarizer*, In preceedings of the 18th ACMSGIR Conference, 1995.

[31] Lamport L., *LATEX: A Document Preparation System*, Addison_Wesley, Reading, Massachusetts, second edition, 1994, ISBN 0-201-52983-1.

[32] Lebowitz M., *The use of memory in text processing*, Communications of the ACM, Vol. 33, No. 8, pp. 30-49, 1990.

[33] Lehnert, W. G. and Sundheim, B. 1991. *A Performance Evaluation of Text Analysis Technologies*, AI Magazine 12(3):81-94.

[34] Lewis D. D. , Jones K. , *Natural Language processing for information Retrieval*, Communications of the ACM, Vol 39, pp. 92-101, January 1996.

[35] Lovins J. B., *Development of a Stemming Algorithm*, Mechanical Translation and Computational Linguistics, Vol 11, January 1968.

[36] Luhn, H. P., *The automatic creation of literature abstracts*, IBM Journal of Research and Development, 2, pp. 159-165, 1958.

[37] Maron, M. E. and Kuhns, J. L., *On relevance, probabilistic indexing and information retrieval*, Journal of the ACM, 7, pp. 216-244, 1960.

[38] Mauldin M. 1991. *Retrieval Performance in FERRET: A conceptual Information Retrieval System*, In Proceedings, SGIR 1991. pp. 347-355.

[39] McBryan, Oliver A., *World Wide Web Worm*,
http://www.cs.colorado.edu/home/mcbryan/WWWW.html

[40] O'Brien T., *Oracle ConText, Text looms as the next frontier in Information Management*, prepared by Oracle Corporation, April 1996.

[41] Oracle Corporation, *ConText: Introduction to Oracle ConText*, Oracle Corporation, Sept. 1993.

[42] Paice C. D., *Automatic Generation of Literature Abstracts - An Approach Based on the identification of self indicating phrases, in information retrieval research*, R.N. Oddy, S.E. Robertson, C.J. van Rijsbergen and P.W. Williams, editors, Butterworths, London, pp. 172-191, 1981.

[43] Paice C. D., *Constructing Literature Abstracts by Computer: Techniques and Prospects*, Information Processing and Management, 26:1, pp. 171-186, 1990.

[44] Porter, M. F., *An Algorithm For Suffix Stripping*, Program 14 (3), pp. 130-137, July 1980.

[45] Rau, L. F., Jacobs, P. S. and Zernik, U., *Information extraction and text summarization using lingustic knowledge acquisition*, Information processing and management, Vol. 25, No. 4, pp. 419-428, 1989.

[46] Rijsbergen C. J. van, *Information Retrieval*, second edition, Butterworths, pp. 17-22, 1979.

[47] Riloff E. and Hollar L., *Text Database and Information Retrieval*, ACM computing surveys, Vol. 28, No. 1, pp. 133-135, March 1996.

[48] Riloff E. and Lehnert W., *Information Extraction as a basis for High-Precision Text Classification*, ACM Transactions on Information Systems, Vol. 12, No. 3, pp. 296-333, July 1994.

[49] Rush J. E., Salvador R., and Zamora A., *Automatic Abstracting and Indexing-Production of Indicative Abstracts By Application of Contextual Inference and Syntactic Coherence Criteria*, Journal of the ASIS, 22:4,pp. 260-274, July-August 1964.

[50] Salton G. and Lesk M. E., *Computer Evaluation of Indexing and text processing*, Journal of ACM, Vol 25, No. 1, pp. 8-36, 1968.

[51] Salton G., *The SMART Retrieval System*, Prentice-Hall Inc.,4-6, 1971.

[52] Salton G., McGILL M. J., *Introduction to Modern Information Retrieval*, McGraw-Hill Book Company, pp. 87-89, 1983.

[53] Salton G., *Automatic Text Processing: The Transformation, Analysis, and Retrieval of information by Computer*, Addison-Welsey, Reading, MA., 1989.

[54] Salton G., Allen J. , Buckley O. , *Automatic Structuring and Retrieval of Large Text Files*, Department of Computer Science, Cornell University. 1992.

[55] Salton G., Allan J. , Buckley C., and Singhal A. , *Automatic Analysis, Theme Generation, and Summarization of Machine-Readable Texts*, Science, Vol264, pp. 1421-1426, June 1994.

[56] Shayan N., *CINDI: Concordia INdexing and DIscovery system*, Department of Computer Science, Concordia University, Montreal, Canada, 1997.

[57] Smeaton A. F., *Progress in the Application of Natural Language Processing to Information Retrieval tasks*, The Computer Journal, Vol. 35, No. 3, pp. 268-271, 1992.

[58] Stiles, H. F., *The association factor in information retrieval*, Journal of the ACM, 8, pp. 271-279, 1961.

[59] Teufel S. and Moens M., *Sentence extraction as a classification task*, ACL/EACL'97, Intelligent Scalable Text Summarization, Workshop Program, JULY 11, 1997.

[60] Thau, R., *SiteIndex Transducer*, http://www.ai.mit.edu/tools/site-index.html

[61] Turtle H. R. and Croft, W. B., *Efficient Probabilistic Inference for Text Retrieval*, In Proceedings of RIAO 91. pp. 644-661, 1991.

[62] Computer and Control Abstracts, Produced by INSPEC, No. 10, October 1997.

[63] http://www.oracle.com.sg/products/oracle7/oracle7.3/html/conTxtDS.html.

[64] http://www.oracle.com.sg/products/oracle7/oracle7.3/html/context_seybold.html.

[65] *Experimental Search Engine Meta-Index*, http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/Demo/metaindex.html

[66] Library of Congress Subject Headings, September 1996.

[67] http://www.acm.org/class/1998/ccs98.txt.

[68] Search WWW document full text, http://rbse.jsc.nasa.gov/eichmann/urlsearch.html

[69] WebCrawler, http://www.biotech.washington.edu/WebCrawler/WebQuery.html

[70] World Wide Web Catalog, http://cuiwww.unige.ch/cgi-bin/w3catalog

[71] http://web.soi.city.ac.uk/research/cisr/okapi/stem.html

[72] http://www.qpat.com/info/help/stemhelp.html

[73] http://www.cs.concordia.ca/~bcdesai/publication/ashg-test-document.html