Automated classification of textual documents based on a controlled vocabulary in engineering

Koraljka Golub
Postal address: KnowLib Research Group, Lund University, P. O. Box 118, SE-221 00 Lund, Sweden
E-mail: Koraljka.Golub@eit.lth.se

Thierry Hamon
Postal address: Laboratoire d'Informatique de Paris-Nord – UMR CNRS 7030, Institut Galilée, Université Paris-Nord, Avenue J.-B. Clément, 93430 Villetaneuse, France
E-mail: thierry.hamon@lipn.univ-paris13.fr

Anders Ardö
Postal address: KnowLib Research Group, Lund University, P. O. Box 118, SE-221 00 Lund, Sweden
E-mail: Anders.Ardo@eit.lth.se

Abstract. Automated subject classification has been a challenging research issue for many years now, receiving particular attention in the past decade due to rapid increase of digital documents. The most frequent approach to automated classification is machine learning. It, however, requires training documents and performs well on new documents only if these are similar enough to the former. We explore a string-matching algorithm based on a controlled vocabulary, which does not require training documents – instead it reuses the intellectual work put into creating the controlled vocabulary. Terms from the Engineering Information thesaurus and classification scheme were matched against title and abstract of engineering papers from the Compendex database. Simple string-matching was enhanced by several methods such as term weighting schemes and cut-offs, exclusion of certain terms, and enrichment of the controlled vocabulary with automatically extracted terms. The best results are 76% recall when the controlled vocabulary is enriched with new terms, and 79% precision when certain terms are excluded. Precision of individual classes is up to 98%. These results are comparable to state-of-the-art machine-learning algorithms.

1 Introduction

Subject classification is organization of objects into topically related groups and establishing relationships between them. In automated subject classification (in further text: automated classification) human intellectual processes are replaced by, for example, statistical and computational linguistics techniques. Automated classification of textual documents has been a challenging research issue for several decades. Its relevance is rapidly growing with the advancement of the World Wide Web. Due to high costs of human-based subject classification and the ever-increasing number of documents, there is a danger that recognized objectives of bibliographic systems (Svenonius 2000, 20-21) would be left behind; automated means could provide a solution to preserve them (ibid., 30).

Automated classification of text has many different applications (cf. Sebastiani 2002 and Jain et al. 1999); in this paper, the application context is that of information retrieval. In information retrieval systems, e.g. library catalogues or indexing and abstracting services, improved precision and recall are achieved by controlled vocabularies, such as classification schemes and thesauri. The specific aim of the classification algorithm is to provide a hierarchical browsing interface to the document collection, through a classification scheme.

In our opinion, one can distinguish between three major approaches to automated classification: text categorization, document clustering, and document classification (cf. Golub 2006a).

In document clustering, both subject clusters or classes into which documents are classified and, to a limited degree, relationships between them are automatically produced. Labeling the clusters is a major research problem, with relationships between them, such as those of equivalence, related-term and hierarchical relationships, being even more difficult to automatically derive (Svenonius 2000, 168). In addition, "[a]utomatically-derived structures often result in heterogeneous criteria for category membership and can be difficult to understand" (Chen & Dumais 2000, 146). Also, clusters' labels and relationships between them change as new documents are added to the collection; unstable class names and relationships are in information retrieval systems user-unfriendly, especially when used for subject browsing.

Text categorization (machine learning) is the most widespread approach to automated classification of text. Here characteristics of subject classes, into which documents are to be classified, are learnt from documents with human-assigned classes. However, human-classified documents are often unavailable in many subject areas, for different document types or for different user groups. If one would judge by the standard Reuters Corpus Volume 1 collection (RCV1) (Lewis et al. 2004), some 8000 training and testing documents would be needed per class. A related problem is that the algorithm performs well on new documents only if they are similar enough to the training documents. The issue of data collections was also pointed out by Yang (1999) who showed how certain versions of one and the same data collection had a strong impact on performance.

In document classification, matching is conducted between a controlled vocabulary and text of documents to be classified. A major advantage of this approach is that it does not require training documents. If using a well-developed classification scheme, it will also be suitable for subject browsing in information retrieval systems. This would be less the case with automatically-developed classes and structures of document clustering or home-grown directories not created in compliance with professional principles and standards . Apart from improved information retrieval, another motivation to apply controlled vocabularies in automated classification is to re-use the intellectual effort that has gone into creating such a controlled vocabulary (cf. Svenonius 1997).

The importance of controlled vocabularies such as thesauri in automated classification has been recognized in recent research. Bang et al. (2006) used a thesaurus to improve performance of a k-NN classifier and managed to improve precision by 14%, without degrading recall. Medelyan & Witten (2006) showed how information from a subject-specific thesaurus improved performance of keyphrase extraction by more than 1,5 times in F1, precision, and recall.

The overall purpose of this experiment is to gain insights into what degree a good controlled vocabulary such as "Engineering Information thesaurus and classification scheme" (Ei thesaurus 1995) (in further text: Ei controlled vocabulary) could be used in automated classification of text, using string-matching. Vocabulary control in thesauri is achieved in several ways (Aitchinson et al. 2000). We believe that the following could be beneficial in the process of automated classification:

- Terms in thesauri are usually noun phrases, which are content words;
- Three main types of relationships are displayed in a thesaurus:
    1) equivalence (e.g. synonyms, lexical variants);
    2) hierarchical (e.g. generic, whole-part, instance relationships);
    3) associative (terms that are closely related conceptually but not hierarchically and are not members of an equivalence set).

  In automated classification, equivalence terms could allow for discovering concepts and not just terms expressing the concepts. Hierarchies could provide additional context for determining the correct meaning of a term; and so could associative relationships;
- When a term has more than one meaning in the thesaurus, each meaning is indicated by the addition of scope notes and definitions, providing additional context for automated classification.

In a previous paper (Golub 2006b) it was explored to what degree different types of Ei thesaurus terms and Ei classification captions influence performance of automated classification. In short, the algorithm searched for terms from the Ei controlled vocabulary in engineering documents to be classified (cf. 2.1). The majority of classes were found when using all the types of terms: preferred terms, their synonyms, related, broader, narrower terms and captions, in combination with a stemmer: recall was 73%. The remaining 27% of classes were not found because the words in the term list designating the classes did not exist in the text of the documents to be classified. No weighting or cut-offs were applied in the experiment. Apart from showing that all those types of terms should be used for a term list in order to achieve best recall, it was also indicated that higher weights could be given to preferred terms (from the thesaurus), captions (from the classification scheme) and synonyms (from the thesaurus), as those three types of terms yielded highest precision.

The aim of this experiment is to improve the classification algorithm based on string-matching between the Ei controlled vocabulary and engineering documents to be classified. We especially wanted to do the following:

1) Achieve precision levels similar to recall achieved in a previous experiment (Golub 2006b,964) by applying different weights and cut-offs.
2) Increase levels of recall to more than those achieved in a previous experiment (ibid.) by adding new terms extracted using natural language processing methods such as multi-word morpho-syntactic analysis and synonym extraction.

The paper is structured as follows: the next section, (2 Methodology) describes the applied string-matching classification algorithm, data collection and the evaluation methodology. The third section (3 Improving the string-matching algorithm) describes methods for enhancement of the string-matching algorithm, including the enrichment with automatically extracted terms. In fourth section, (4 Results) analyzes and discusses the results. Major conclusions and implications for further research are presented in the fifth section (5 Conclusion).

2 Methodology

2.1 String-matching algorithm

This section describes the classification algorithm used in the experiment. It is based on searching for terms from the Ei controlled vocabulary, in the field of engineering, in text of documents to be classified (also in the field of engineering). The Ei controlled vocabulary consists of two parts: a thesaurus of engineering terms, and a hierarchical classification scheme of engineering topics. These two controlled vocabulary types have each traditionally had distinct functions: the thesaurus has been used to describe a document with as many controlled terms as possible, while the classification scheme has been used to group similar documents together to the purpose of shelving them and allowing systematic browsing. The aim of the algorithm was to classify documents into classes of the Ei classification scheme in order to provide a browsing interface to the document collection. A major advantage of Ei is that thesaurus descriptors are mapped to classes of the classification scheme. These mappings have been human-derived and are an integral part of the thesaurus. Compared with captions[1] alone, mapped thesaurus terms provide a rich additional vocabulary for every class: instead of having only one term per class (there is only one caption per class), in our experiment there were on average 88 terms per class.

Pre-processing steps of Ei included normalizing upper- and lower-case words. Upper-case words were left in upper case in the term list, assuming that they were acronyms; all other words containing at least one lower-case letter were converted into lower case. The first major step in designing the algorithm was to extract terms from Ei into what we call a term list. It contained class captions, thesaurus preferred terms, their synonyms (Term), classes to which the terms and captions map or denote (Class), and weight indicating how appropriate the term is for the class to which it maps or which it designates (Weight). Geographical names, all mapping to class 95, were excluded on the grounds that they are not engineering-specific. The term list was formed as an array of triplets:

Weight: Term (single word, Boolean term or phrase) = Class

[1] A caption is a class number expressed in words, e.g. in Ei classification scheme “Electric and Electronic Instruments” is the caption for class “942.1”.

Single-word terms were terms consisting of one word. Boolean terms were terms consisting of two or more words that must all be present but in any order or in any distance from each other. Boolean terms in this form were not explicitly part of Ei, but were created to our purpose. They were considered to be those terms which in Ei contained the following strings: "and" (word "and"), "vs." (short for "versus"), "," (comma), ";" (semi-colon, separating different concepts in class names), "(", ")" (parentheses, indicating the context of a homonym), ":" (colon, indicating a more specific description of the previous term in a class name), and "--" (double dash, indicating heading--subheading relationship). These strings we replaced with "@and" which indicated the Boolean relation in the term. All other terms consisting of two or more words were treated as phrases, i.e. strings that need to be present in the document in the exact same order as in the term. Ei comprises a large portion of composite terms (3474 in the total of 4411 distinct terms in our experiment); as such, Ei provides a rich and as such provides a rich and precise vocabulary with the potential to reduce the risks of false hits.

The following are two excerpts from the Ei classification scheme and thesaurus, based on which the excerpt from the term list (further below) is created:

From the classification scheme:
931.2 Physical Properties of Gases, Liquids and Solids
…
942.1 Electric and Electronic Instruments
…
943.2 Mechanical Variables Measurements

From the thesaurus:
TM Amperometric sensors
UF Sensors--Amperometric measurements
MC 942.1
…
TM Angle measurement
UF Angular measurement
UF Mechanical variables measurement--Angles
BT Spatial variables measurement
RT Micrometers
MC 943.2
…
TM Anisotropy
NT Magnetic anisotropy
MC 931.2

All the different thesaurus terms as well as captions were added to the term list. Despite the fact that choosing all types of thesaurus terms might lead to precision losses, we decided to do just that in order to achieve maximum recall, as shown in a previous paper (Golub 2006b). In the thesaurus, TM stands for the preferred term, UF ("Used For") for an equivalent term, BT for broader term, RT for related term, NT for narrower term; MC represents the main class; sometimes there is also OC, which stands for optional class, valid only in certain cases. Main and optional classes are classes from the Ei classification scheme that have been human-derived and are an integral part of the thesaurus. Based on the above excerpts, the following term list would be created:

```
1: physical properties of gases @and liquids @and solids = 931.2,
1: electric @and electronic instruments = 942.1,
1: mechanical variables measurements = 943.2,
1: amperometric sensors = 942.1,
1: sensors @and amperometric measurements = 942.1,
1: angle measurement = 943.2,
1: angular measurement = 943.2,
1: mechanical variables measurement @and angles = 943.2,
1: spatial variables measurement = 943.2,
1: micrometers = 943.2,
1: anisotropy = 931.2,
1: magnetic anisotropy = 931.2,
```

The number at the beginning of each triplet is weight estimating the probability that the term of the triplet designates the class; in this example it is set to 1 as a baseline, and experiments with different weights are discussed later on.

The algorithm searches for strings from a given term list in the document to be classified and if the string (e.g. "magnetic anisotropy" from the above list) is found, the class(es) assigned to that string in the term list ("931.2" in our example) are assigned to the document. One class can be designated by many terms, and each time a term is found, the corresponding weight ("1" in our example) is added to a score for the class. The scores for each class are summed up and classes with scores above a certain cut-off (heuristically defined, discussed later on) are selected as the final ones for the document being classified.

The Ei classification scheme is hierarchical and consists of six main classes divided into 38 finer classes which are further subdivided into 182 classes. These are subdivided even further, resulting in some 800 individual classes in a five-level hierarchy. For this experiment one of the six main classes was selected, together with all its subclasses: class 9, “Engineering, General”. The reason for choosing this class was that it covers both natural sciences such as physics and mathematics, and social sciences fields such as engineering profession and management. The literature of the latter tends to contain more polysemic words than the former, and as such presents a more complex challenge for automated classification. Within the 9 class, there are 99 subclasses. However, for seven of them the number of documents in a database based on which the data collection was created (see 2.2 Data Collection) were few, less than 100. Thus those seven classes were excluded from the experiment altogether. These were: 9 (Engineering, General), 902 (Engineering Graphics; Engineering Standards; Patents), 91 (Engineering Management), 914 (Safety Engineering), 92 (Engineering Mathematics), 93 (Engineering Physics), and 94 (Instruments and Measurement). Of the remaining 92 classes, the distribution at the five different hierarchical levels is as follows: at the fifth hierarchical level 11 classes, at the fourth 67, at the third 14, and at the second hierarchical level 5.

2.2 Data collection

The data collection comprised 35166 bibliographic records[2] from the Compendex database (2006). The records were selected by simply retrieving the top 100 or more of them upon entering the class number. A minimum of 100 records per class were downloaded at several different points in time during the years of 2005 and 2006.

For each record there was at least one of the 92 selected classes that were human-assigned (cf. 2.1). A subset of this collection was created to include only those records where main class[3] was class 9; this subset contained 19237 documents.

From each bibliographic record (in further text: document) the following elements were extracted: an identification number, title, abstract and human-assigned classes (“Ei classification codes”). Thesaurus descriptors (in Compendex called “Ei controlled terms”) were not extracted since the purpose of this experiment was to compare automatically assigned classes (and not descriptors) against the human-assigned ones. Below is an example of one document:

Identification number: 03337590709
Title: The concept of relevance in IR
Abstract: This article introduces the concept of relevance as viewed and applied in the context of IR evaluation, by presenting an overview of the multidimensional and dynamic nature of the concept. The literature on relevance reveals how the relevance concept, especially in regard to the multidimensionality of relevance, is many faceted, and does not just refer to the various relevance criteria users may apply in the process of judging relevance of retrieved information objects. From our point of view, the multidimensionality of relevance explains why some will argue that no consensus has been reached on the relevance concept. Thus, the objective of this article is to present an overview of the many different views and ways by which the concept of relevance is used - leading to a consistent and compatible understanding of the concept. In addition, special attention is paid to the type of situational relevance. Many researchers perceive situational relevance as the most realistic type of user relevance, and therefore situational relevance is discussed with reference to its potential dynamic nature, and as a requirement for interactive information retrieval (IIR) evaluation.
Ei classification codes: 903.3 Information Retrieval & Use, 723.5 Computer Applications, 921 Applied Mathematics

Automated classification was based on title and abstract, and automatically assigned classes were compared against human-assigned ones (Ei classification codes in the example). On average, 2.2 classes per document were human-assigned, ranging from 10 to 1.

2.3 Evaluation methodology

2.3.1 Evaluation Challenge

According to standard “Documentation – Methods for examining documents, determining their subjects, and selecting index terms” (International Organization for Standardization 1985: 5963-

[2] Compendex being a commercial database, the data collection cannot be made available to others, but the authors are willing to provide documents’ identification numbers on request.

[3] The first one listed in the “Ei classification codes” field of the record.

1985), human-based subject indexing is a process involving three steps: 1) determining subject content of a document, 2) conceptual analysis to decide which aspects of the content should be represented, and 3) translation of those concepts or aspects into a controlled vocabulary. These steps, in particular the second one, are based on a specific library's policy in respect to its document collections and user groups. Thus, when evaluating automatically assigned classes against the human-assigned ones, it is important to know the human-based indexing policies. Unfortunately, we were unable to obtain indexing policies applied in the Compendex database. What we could derive from the data collection was the number of human-assigned classes per document, which were used in evaluation. However, without a thorough qualitative analysis of automatically assigned classes one cannot be sure whether, for example, the classes assigned by the algorithm, but not human-assigned, are actually wrong, or if they were left out by mistake or because of the indexing policy. A further issue is that we did not know whether the articles had been human-classified based on their full-text or/and abstracts; we had, however, only abstracts.

Another problem to consider when evaluating automated classification is the fact that certain subjects are erroneously assigned. When indexing, humans make errors such as those related to exhaustivity policy (too many or too few terms become assigned), specificity of indexing (which usually means that humans do not assign the most specific term), they may omit important terms, or assign an obviously incorrect term (Lancaster 2003, 86-87). In addition, it has been reported that different people, whether users or professional subject indexers, would assign different subject terms or classes to the same document. Studies on inter- and intra-indexer consistency report generally low indexer consistency (Olson & Boll 2001, 99-101). Markey (1984) reviewed 57 indexer consistency studies and reported that consistency levels range from 4% to 84%, with only 18 studies showing over 50% consistency. There are two main factors that seem to affect it:

1) Higher exhaustivity and specificity of subject indexing both lead to lower consistency, i.e. indexers choose the same first term for the major subject of the document, but the consistency decreases as they choose more classes or terms;
2) The bigger the vocabulary, or, the more choices the indexers have, the less likely they will choose the same classes or terms (Olson & Boll 2001, 99-101).

Both of these two factors were present in our experiment:

1) High exhaustivity: on average, 2.2 classes per document had been human-assigned, ranging from 10 to 1.
2) Ei controlled vocabulary is rather big (we chose 92 classes) and deep (five hierarchical levels), allowing many different choices.

Today evaluation in automated classification experiments is mostly conducted under controlled conditions, ignoring the above-discussed issues. As Sebastiani (2002, 32) puts it, "…the evaluation of document classifiers is typically conducted experimentally, rather than analytically. The reason is that… we would need a formal specification of the problem that the system is trying to solve (e.g. with respect to what correctness and completeness are defined), and the central notion… that of membership of a document in a category is, due to its subjective character, inherently nonformalizable." Because of the fact that methodology for such experiments has yet to be developed, as well as limited resources, we followed the common approach to evaluation and started from the assumption that human-assigned classes in the data collection were correct, and compared automatically assigned classes against them.

2.3.2 Evaluation measures

The subset of the Ei controlled vocabulary we used comprised 92 classes that are all topically related to each other. The topical relatedness is expressed in numbers representing the classes: the more initial digits any two classes have in common, the more related they are. For example, 933.1.2 for "Crystal Growth" is closely related to 933.1 for "Crystalline Solids", both of which belong to 933 for "Solid State Physics", and finally to 93 for "Engineering Physics". Each digit represents one hierarchical level: class 933.1.2 is at the fifth hierarchical level, 933.1 at the fourth etc. Thus, comparing two classes at only first few digits (later referred to as partial matching) instead of all the five also makes sense. Still, unless specifically noted, the evaluation in this experiment was conducted based on all the five different levels (later referred to as complete matching), i.e. an automatically assigned class was considered correct only if all its digits were the same as a human-assigned class for the same document.

Evaluation measures used were the standard microaveraged and macroaveraged precision, recall and F1 (Sebastiani 2002, 40-41), for both complete and partial matching:

Precision = correctly automatically assigned classes / all automatically assigned classes

Recall = correctly automatically assigned classes / all human-assigned classes

F1 = 2*Precision*Recall / (Precision + Recall)

In macroaveraging the results are first calculated for each class, and then summed and divided by the number of classes. In microaveraging the results for each part of every equation are summed up first (e.g. all correctly automatically assigned classes are added together, all automatically assigned classes are added together), and then the “aggregated” values are used in one equation. Equations for macroaveraged and microaveraged precision are given below:

$\text{Precision}_{\text{macroaveraged}}$ = sum of precision values for each class / number of all classes

$\text{Precision}_{\text{microaveraged}}$ = sum of correct automated assignments for each class /
sum of all automated assignments for each class

In microaveraging more value is given to classes that have a lot of instances of automatically assigned classes and the majority of them are correct, while in macroaveraging the same weight is given to each class, no matter if there are many or few automatically assigned instances of it. The differences between macroaveraged and microaveraged values can be large, but whether one is better than the other has not been agreed upon (Sebastiani 2002, 41-42). Thus, in this experiment, it is the mean macroaveraged and microaveraged F1 that is mostly used.

In order to examine different aspects of the automated classification performance, several other factors were also taken into consideration:

- Whether the (human-assigned) main class is found;
- The number of documents that got automatically assigned at least one class;
- Whether the class with highest score was the same as the human-assigned main class;
- The distribution of automatically versus human-assigned classes; and,
- The average number of classes assigned to each document. There were 2.2 human-assigned classes per document, and our aim was to achieve similar. In the context of hierarchical browsing based on a classification scheme, having too many classes assigned to a document would place one document to too many different places, which would create the opposite effect of the original purpose of a classification scheme, that of grouping similar documents together.

3 Improving the algorithm

The major aim of the experiment was to improve the algorithm that was previously experimented with in Golub 2006b, where highest (microaveraged) recall was 73% when all types of terms were included in the term list. In that experiment neither weights nor cut-offs were experimented with, so all the classes that were found for a document were assigned to it. Here we wanted to achieve as high as possible precision levels by use of term weighting and class cut-offs. In order to also allow for better recall, the basic term list was enriched with new terms extracted from documents in the Compendex database, using multi-word morpho-syntactic analysis and synonym acquisition.

3.1 Term weights

The aim of this part of the experiment was to achieve as high as possible precision levels by use of weighting and cut-offs. As shown in Golub 2006b, all types of terms need to be used in the term list for maximum recall. Thus, all the different types of terms and their mappings to classes were merged into the final term list. This resulted in a number of duplicate cases which were dealt with in the following manner:

- If one term mapping to the same class was a caption, a preferred term, and a synonym at the same time, the highest preference was, based on their performance (see Table 4), given to captions, followed by preferred terms, followed by synonyms, while others were removed from the list;
- If one term mapping to both optional class (OC) and main class (MC) was a caption, a preferred term, and a synonym at the same time, the highest preference was, based on their performance (see Table 2), given to captions, followed by preferred terms, followed by synonyms, while others were removed from the list;
- If one thesaurus term of the same type mapped to both optional class (OC) and main class (MC), the one that mapped to the optional class was removed (based on their performance, see Table 2).

The final term list consisted of 8099 terms, out of which 92 were captions (all mapped to main class (MC)), 668 were broader terms, 729 narrower, 1653 preferred, 3224 related, and 1733 were synonym terms. This big number of terms that have been human-mapped to classes indicates potential usefulness of such a controlled vocabulary in a string-matching algorithm for automated classification.

In order to systematically vary different parameters, the following 14 weighting schemes evolved:

1) **w1**: All terms in the term list were given the same weight, 1. This term list served as a baseline.

2) **w134**: Different term types were given different weights: single-word terms 1, phrases 3, and Boolean terms 4.

These weights were heuristically derived in a separate experiment (Table 1). Three different term lists were created, each containing only single-word terms, phrases or Boolean terms. Weight 1 was assigned to all of them. The documents were classified using these three terms lists and their performance was compared for precision.

Table 1. Single, phrase and Boolean term lists and their performance as a basis for weights.

	Single	Phrase	Boolean
Avg. precision (%)	8	26	33
Derived weight	1	3	4

Avg. precision (%) is mean microaveraged and macroaveraged precision. Derived weights were based on dividing precision values (Avg. precision) by the lowest precision value (in this case 8).

3) **w12**: Terms mapping to a main class (MC) were given weight 2, and those mapping to an optional class (OC) were given weight 1.

These weights were heuristically derived in a separate experiment (Table 2). Two different term lists were created, one containing only those terms that map to a main class, and another one containing only those terms that map to an optional class. Weight 1 was assigned to all of them. The documents were classified using these two terms lists and their performance was compared for precision.

Table 2. Main code and optional code term lists and their performance as a basis for weights.

	MC	OC
Avg. precision (%)	13	6
Derived weight	2	1

Avg. precision (%) is mean microaveraged and macroaveraged precision. Derived weights were based on dividing precision values (Avg. precision) by the lowest precision value (in this case 6).

4) **w134_12**: This list was a combination of the two preceding lists. Weights for term type 1, 3, and 4 for single, phrase or Boolean term were multiplied by the weight for the type of class to which the term mapped – 1 or 2 for optional or main class.

5) **wOrig**: As used in the original term weighting scheme when the string-matching algorithm based on Ei was first applied (Koch & Ardö 2000). These weights were intuitively derived. They combined types of terms depending if it were a single-word term, Boolean or phrase, and whether the assigned class was main (MC) or optional (OC).

Table 3. Weights in the original algorithm.

	Phrase	Boolean	Single
OC	4	2	1
MC	8	3	2

6) **w1234**: With weights for different term relationships and captions as experimented with in Golub 2006b (captions are from the classification scheme, all others are thesaurus terms).

Table 4. Different types of thesaurus terms captions and their performance as a basis for weights.

	Broader	Captions	Narrower	Preferred	Related	Synonyms
Avg. precision (%)	10	43	25	39	10	35
Derived weight	1	4	2	4	1	3

7) **w134_1234**: This list was a combination of two previous lists, w134 and w1234. Weights for term type 1, 3, and 4 for single, phrase or Boolean term were multiplied by the weight for the type of term relationships as given in Table 4.

8) **w134_12_1234**: This list was a combination of two previous lists, w134_12 and w1234. Weights for term type 1, 3, and 4 for single, phrase or Boolean term were multiplied by the weight for the type of class to which the term mapped – 1 or 2 for optional or main class, and by the weight for the type of term relationships as given in Table 4.

9) **wTf10**: In this list weights were based on the number of words the term consisted of, and of the number of times each of its words occurred in other terms (cf. *tf-idf*, term frequency – inverse document frequency, Salton & McGill 1983, e.g. 63,205). If *f* were the frequency with which a word *w* from the term *t* occurred in other terms, term *t* consisting of *n* words, then the weight *weight* of that term was calculated as follows:

$$\text{weight}_t = \log(n) \cdot (1/f_{w1} + 1/f_{w2} + \ldots + 1/f_{wn})$$

Log was applied in order to reduce the impact of parameter *n*, i.e. to avoid getting overly high weights for terms consisting of several sparse words. In order to get integers as weights, the weights were multiplied by 10, rounded and increased by 1 to avoid zeros.

10) **wTf10Boolean**: As in wTf10, with all the phrases modified into Boolean terms. This list was created in order to study the influence of phrases and Boolean terms on precision and recall.

11) **wTf10Phrases**: As in wTf10, with all the Boolean terms modified into phrases. This list was created in order to study the influence of phrases and Boolean terms on precision and recall.

12) **wTf10_12**: As in wTf10, with those weights multiplied by the weight for the type of class to which the term maps – 1 or 2 for optional or main class. The multiplication was done before the rounding.

13) **wTf10_1234**: As in wTf10, with those weights multiplied by the weight for the type of relationship (Table 4). The multiplication was done before the rounding.

14) **wTf10_12_1234**: As in wTf10_12, with those weights multiplied by the weight for the type of relationship (Table 4). The multiplication was done before the rounding.

3.1.1 Stop-word list and stemming

Although the terms and captions in the Ei controlled vocabulary are usually noun phrases which are good content words, they can also contain words which are frequently used in many contexts and as such are not very indicative of any document's topicality (e.g. the word "general" in the Ei class caption "Engineering, General"). Thus, a stop-word list was used. It contained 429 such words, and was taken from Onix text retrieval toolkit (2006). For stemming, the Porter's algorithm (Porter 1980) was used. The stop-word list was applied to the term lists, and stemming to the term lists as well as documents.

3.2 Cut-offs

In a previous experiment (Golub 2006b) cut-offs were not used – instead, all the classes that were found for a document were assigned to it. In the context of hierarchical browsing based on a classification scheme, having too many classes assigned to a document would place one document to many different places, which would create the opposite effect of the original purpose of a classification scheme (grouping similar documents together). In the data collection, there were 2.2 human-assigned classes per document, and the aim of automated classification was to achieve similar. The effect of several different cut-offs was investigated:

1) All automatically derived classes are assigned as final ones (no cut-off).
2) In order to assign a certain class as final, the score of that class had to have a minimum percentage of the sum of all the classes' scores. Different values for the minimum percentage were tested: 1, 5, 10, 15 and 20, as well as some others (see section 4 Results).
3) The second type of cut-off in combination with the rule that if there were no class with the required score, the one with the highest score would be assigned.
4) In order to follow the subject classification principle of always assigning the most specific class possible, the principle of score propagation was introduced. The principle was implemented so that the scores for classes at deeper hierarchical levels were a sum of their own score together with scores of classes at upper hierarchical levels if such were assigned.

3.3 Enriching the term list with new terms

In the previous experiment (Golub 2006b), highest achieved recall was 73% (microaveraged), when all types of terms were included in the term list. In order to further improve recall, the basic term list was enriched with new terms. These terms were extracted from bibliographic records of the Compendex database, using multi-word morpho-syntactic analysis and synonym acquisition, based on the existing preferred and synonymous terms (as they gave best precision results).

Multi-word morpho-syntactic analysis was conducted using a parser FASTER (Jacquemin 1996) which analyses raw technical texts and, based on built-in meta-rules, detects morpho-syntactic variants. The parser exploits morphological (derivational and inflectional) information as given by the database CELEX (Baayen et al. 1995). Morphological analysis was used to identify derivational variants, such as:

effect of gravity: gravitational effect
architectural design: design of the proposed architecture
supersonic flow: subsonic flow
structural analysis: analysis of the structure

Syntactical analysis was used to:

a) insert word inside a term, such as:

flow measurement: flow discharge measurements
distribution of good: distribution of the finished goods
construction equipment: construction related equipment
intelligent distributed control: intelligent control

b) permute components of a term, such as:

control of the inventory: inventory control
flow control: control of flow
development of a flexible software: software development

c) add a coordinated component to a term, such as:

project schedule and management: project management
control system: control and navigation system

Synonyms were acquired through a rule-based system SynoTerm (Hamon & Nazarenko 2001) which infers synonymy relations between complex terms by employing semantic information extracted from lexical resources. First the documents were preprocessed and tagged with part-of-speech information and lemmatized. Then terms were identified through the YaTeA term extractor

(Aubin & Hamon 2006). The semantic information provided by the database WordNet (Fellbaum 1998; WordNet Search 2007) was used as a bootstrap to acquire synonym terms of the basic terms. The synonymy of the complex candidate terms was assumed to be compositional, i.e. two terms were considered synonymous if their components were identical or synonymous (e.g. building components: construction components, building components: construction elements).

Although verification by a subject expert is desirable for all automatically derived terms, due to limited resources only the extracted synonyms were verified. Checking the synonyms is also most important since computing those leads to a bigger semantic shift than morphological and syntactical operations do. The verification was conducted by a subject expert, a fifth-year student of engineering physics. Suggested synonym terms were displayed in the user interface of SynoTerm. The verification was not strict: derived terms were kept if they were semantically related to the basic term. Thus, hyperonym (generic/specific) or meronym (part/whole) terms were also accepted as synonyms. The expert spent 10 hours validating the derived terms. Of the 292 automatically acquired synonyms, 168 (57,5%) were validated and used in the experiment.

4 Results

4.1 Weights and cut-offs

Based on each of the 14 term lists, the classification algorithm was run on the data collection of 35166 documents (see 2.2). As described earlier (2.3.2), several aspects were evaluated and different evaluation measures were used; thus, for each term list, the following types of results were obtained:

1) **min 1:** if no classes were assigned because their final scores were below the pre-defined cut-off value (described in 3.2), the class with the highest score was assigned;
2) **cut-off:** the applied cut-off value;
3) **min 1 correct:** number of documents that were assigned at least one correct class;
4) **min 1 auto:** number of documents that were assigned at least one class;
5) **avg auto/doc:** average number of classes that were assigned per document, based on documents that were assigned at least one class;
6) **macroa P:** macroaveraged precision;
7) **macroa R:** macroaveraged recall;
8) **macroa F1:** macroaveraged F1;
9) **microa P:** microaveraged precision;
10) **microa R:** microaveraged recall;
11) **microa F1:** microaveraged F1;
12) **mean F1s:** arithmetic mean of macroaveraged and microaveraged F1 values.

Table 5 shows results for list w134_12_1234 which has combined weights for term type (single, phrase or Boolean), type of class, and type of term relationships. In order to provide an example of how results for every other term list were analyzed, we discuss results for this list in detail.

Best recall is achieved when no cut-off is applied, 0.54, but in that case on average 17 classes are assigned per document, which is too many in comparison to 2.2 that are human-assigned. This setting is appropriate in applications such as focused crawling where documents are ranked based on weights. When the most appropriate number of classes for our purpose is assigned (2.63), recall is 0.22. Best precision is gained when cut-off value is highest (20): 0.37 macroaveraged, 0.28 microaveraged. In that setting the average number of classes assigned per document is 1.5. Best mean macroaveraged and microaveraged F1 is 0.22, when cut-offs are 10 or 15.

Best precision results are gained when cut-off is highest, best recall when there is no cut-off. More than twice as many documents are assigned correct classes when no cut-off is used. All these results suggest that weights are not very appropriate. Still, when looking at the F1 values, in comparison to the baseline (first column), an improvement of six percent is achieved when using the w134_12_1234 term list.

Table 5. Results for term list w134_12_1234.

min 1	no						yes				
cut-off	0	1	5	10	15	20	1	5	10	15	20
min 1 correct	**24036**	21403	17403	14339	12320	10278	21403	17403	14425	12774	11606
min 1 auto	34053	34053	34050	33270	30433	26587	34053	34053	34053	34053	34053
avg auto/doc	16.65	9.77	5.02	2.69	1.91	1.47	9.46	4.86	**2.55**	1.65	1.11
macroa P	0.11	0.14	0.18	0.25	0.32	**0.37**	0.14	0.18	0.25	0.31	0.35
macroa R	**0.54**	0.42	0.29	0.21	0.17	0.14	0.42	0.29	0.22	0.18	0.15
macroa F1	0.19	0.21	0.22	**0.23**	0.22	0.20	0.21	0.22	**0.23**	**0.23**	0.21
microa P	0.07	0.10	0.13	0.19	0.24	**0.28**	0.10	0.13	0.19	0.23	0.27
microa R	**0.54**	0.43	0.30	0.22	0.18	0.14	0.43	0.30	0.22	0.18	0.16
microa F1	0.13	0.16	0.19	0.20	0.20	0.19	0.16	0.18	0.20	**0.21**	0.20
mean F1s	0.16	0.19	0.20	**0.22**	0.21	0.19	0.19	0.20	**0.22**	**0.22**	0.21

The same experiment was run on all the other term lists. When looking at mean F1 values, the differences between the term lists are not larger than four percent. Performance of the different lists measured in precision and recall is also similar. Three lists that perform best in terms of mean F1 are w1234, w134_1234 and w134_12_1234 – all of them based on weights for different term relationships. The biggest number of correct classes is found with the wTf10Boolean list in which all phrases were converted into Boolean terms.

When using cut-offs, two sets of experiments were conducted: one with assigning at least the class with highest score, and the other following the threshold calculation only. Because the former results in more documents with assigned correct classes, in further experiments the rule to assign at least the class with highest score is applied.

4.1.1 Stop-words removal and stemming

Next, the influence of stop-words removal and stemming was tested (as described in 3.1.1). For this experiment three lists that performed best in the previous one were chosen: w1234, w134_1234 and w134_12_1234. Every list was run against stop-words removed, stemming, and both the stop-words removed and stemming, each in combination with different cut-off values: 5, 10 and 15.

Improvements when using either stemming or stop-words removal or both are achieved in majority of cases up to two percent. There is also a slight increase in the number of correctly found classes without finding more incorrect classes. The differences between the three term lists measured in mean F1 are minor – one or two percent. The best term list is w134_12_1234 used in combination with stemming and stop-words removal and cut-off 10 – best mean F1 is 0.24. For this list more cut-offs were experimented with for better results; the value of 9 proved to perform best but better only on a third decimal digit than that of 10.

4.1.2 Individual classes, partial matching, distribution of classes

We further wanted to investigate performance at the level of individual classes, partial matching as well as how automatically assigned classes are distributed in comparison to human-assigned ones. We used the best-performing w134_12_1234 term list and setting (applying stemming and stop-words removal, cut-off 9).

It was shown that certain classes perform much better than the average. Performance of different classes varies quite a lot. For example, top three performing classes as measured in precision are different from top three classes for recall or F1:

- Top three in precision:
  - Cellular Manufacturing (913.4.3), precision 0.98;
  - Electronic Structure of Solids (933.3), precision 0.97; and,
  - Information Retrieval and Use (903.3), precision 0.82.
- Top three in recall:
  - Amorphous Solids (933.2), recall 0.61;
  - Crystal Growth (933.1.2), recall 0.52; and,
  - Manufacturing (913.4), recall 0.50.

- Top three in F1:
  - Crystal Growth (933.1.2), F1 0.45;
  - Amorphous Solids (933.2), F1 0.44; and,
  - Optical Variables Measurement (941.1), F1 0.40.

As expected, the algorithm performs better when evaluation is based on partial matching between automatically and human-assigned classes. As seen from Table 6, at the second hierarchical level F1 is up to 0.66 and at third 0.59. At the second hierarchical level the best F1 is achieved by classes "Engineering mathematics" (represented by number 92) and "General engineering" (90), both of which have by far the smallest number of terms designating them (**terms**), while other three classes have many more terms and similar performance measured in mean F1. At the third hierarchical level, the class that performs best of all is 921 "Applied Mathematics", while the worst one is 943 "Mechanical and Miscellaneous Instruments". In conclusion, for the 14 classes at top three hierarchical levels mean F1 is almost twice as good as for the complete matching, which implies that our classification approach would suit better those information systems in which fewer hierarchical levels are needed, like the Intute subject gateway on engineering (Intute: Engineering 2007).

Table 6. Results for partial matching at the second and third hierarchical levels, and number of terms per each class.

	General			*Management*				*Maths*		*Physics*			*Instruments*			
	90			*91*				*92*		*93*			*94*			
F1	0.65			0.5				**0.66**		0.51			0.49			
terms	679			1922				848		2902			1748			
	901	*902*	*903*	*911*	*912*	*913*	*914*	*921*	*922*	*931*	*932*	*933*	*941*	*942*	*943*	*944*
F1	0.35	0.27	**0.53**	0.32	**0.36**	0.26	0.29	**0.59**	0.33	0.44	0.33	**0.48**	0.28	0.36	0.2	**0.44**
terms	275	241	163	237	596	393	696	628	220	1648	801	453	422	373	604	349

The variations in performance between individual classes for both complete and partial matching are quite big, but at this stage it is difficult to say why. Further research is needed to explore what the factors contributing to performance are.

Using the same best setting achieved so far, the algorithm was also evaluated for distribution of automatically assigned classes in comparison to that of the human-assigned ones. The comparison was based on how often two classes get assigned together when using the algorithm in comparison to when they get human-assigned. Figure 1 shows the frequency distribution of assigned class pairs. The x-coordinate presents human-assigned class pairs ordered by descending frequency. One point represents one class pair: e.g. the pair of classes 912.2 and 903 occurs most frequently in human-based classification (48 times, as marked on the y-coordinate) and is represented by point 1 on the x-coordinate; point 500 on the x-coordinate represents the 913.5 and 911 pair that occurs 3 times, as marked on the y-coordinate. Thus, the smoothest line (Human-assigned) represents the human-assigned classes. The minimum of 2538 pairs of classes that both the algorithm and humans have produced are shown.

A correlation of 0.38 exists between the human-assigned classes and automatically assigned classes (Automated). However, for the 100 most frequent pairs, the correlation drops to 0.21. In the top 10 most frequent pairs of classes, there is no overlap at all. In conclusion, the distribution of human-assigned and automatically assigned classes is more correlated when looking at all pairs of classes occurring together, but less so for more frequently occurring pairs.

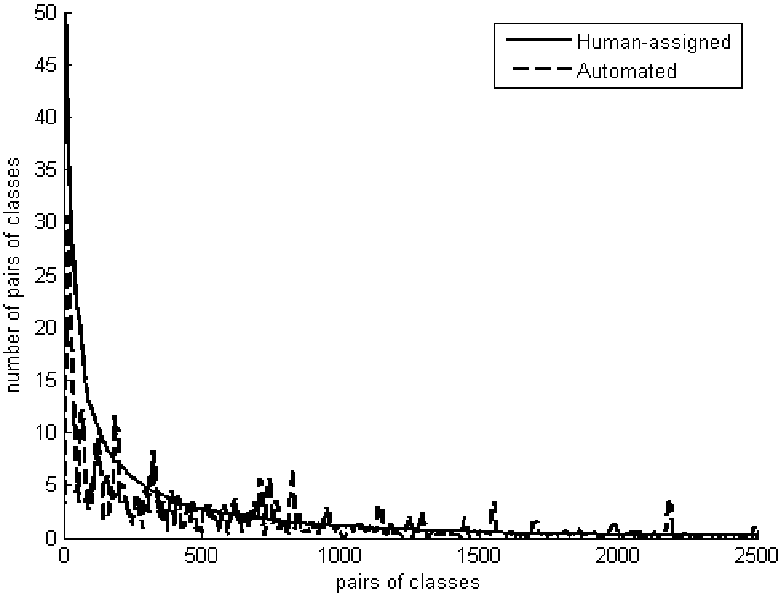

Figure 1. Frequency distribution of assigned pairs of classes (2538 pairs).

4.1.3 Score propagation and main classes

A relevant subject classification principle is to always assign the most specific class available. This principle provided us with a basis for the so-called score propagation, in which scores of classes at narrower (more specific) hierarchical levels were increased by scores assigned to their broader classes (later referred to as "propagated down"). In another run, this was slightly varied, so that the broader classes from which scores were propagated to their narrower classes were removed ("propagated down, broader removed").

These types of score propagation were tested on the best performing term list and setting (w134_12_1234 with stemming and stop-words removal). In complete matching, "propagated down" performs best. However, it is slightly worse than when not using score propagation at all. In partial matching, both "propagated down" and "propagated down, broader removed" perform slightly better than the original on the first two or three hierarchical levels, and slightly worse on the fourth and fifth ones. These not-so-good results with score propagation can be partially explained by the fact that the term list contained both broader and narrower terms, which was done in order to achieve best recall (Golub 2006b).

We further analyzed the degree to which the one most important concept of every document is found by the algorithm. To this purpose, a subset of (19153) documents was used which had the human-assigned main class in class 9 (there is one main class per document). In complete matching 78% of main classes are found when no cut-offs are applied. When cut-offs are applied, 22% of main classes are found. In partial matching, more main classes are found at the second and third hierarchical levels when using both types of score propagation, up to 59% and 38% respectively. Thus, score propagation could be used in services for which fewer hierarchical levels are needed (e.g. Intute 2007).

4.2 Enhancing the term list with new terms

In the previous experiment (Golub 2006b), highest achieved recall was 73% (microaveraged), when all types of terms were included in the term list. In order to further improve recall, the basic term list was enriched with new terms. These terms were extracted from bibliographic records of the Compendex database, using multi-word morpho-syntactic analysis and synonym acquisition, based on the existing preferred and synonymous terms (as they gave best precision results). The number of terms added to the term list was as follows:

1) Based on multi-word morpho-syntactic analysis:
   - derivation: 705, out of which 93 adjective to noun, 78 noun to adjective, and 534 noun to verb;
   - permutation: 1373;
   - coordination: 483;
   - insertion: 742; and
   - preposition change: 69.
2) Based on semantic variation (synonymy): 292 automatically extracted, out of which 168 were verified as correct by the subject expert.

In order to examine the influence of different types of extracted terms, nine different term lists were created and the classification was based on each of them. It was shown that the number of terms is not proportional to performance, e.g. permutation-based extraction comprises 1373 terms, and, when stemming is applied, has performance as measured in mean F1 of 0.02, whereas coordination comprises 403 terms, with performance of 0.07. These two cases can be explained by the fact that permutation also implies variation based on insertion and preposition change (e.g. “engineering for commercial window systems:” “system engineering”) which leads to bigger semantic shift than the identification of term variant based on the coordination. By combining all the extracted terms into one term list, the mean F1 is 0.14 when stemming is applied, and microaveraged recall is 0.11, which would imply that enriching the original Ei-based term list with these newly extracted terms should improve recall. In comparison to results gained in Golub 2006b, where microaveraged recall with stemming is 0.73, here the best recall, also microaveraged and with stemming, is 0.76.

The next step was to assign appropriate weights to the newly extracted terms (Table 7). We used the w134_12_1234 term list, earlier shown to perform best. The result as measured in mean F1 is the same as in the original, 0.24 (cut-off 10, stemming applied but not stop-word removal). The difference is that recall and the number of correctly assigned classes increases by 3%, but precision decreases. Thus, depending on the final application, terms extracted in this way could be added to the term list or not.

Table 7. Performance of the w1 term list enriched with all automatically extracted terms.

	all combined			
stemming	no	yes	no	yes
stop-words out	no	no	yes	yes
min 1 correct	24479	29639	26039	30466
min 1 auto	34086	34966	34425	34987
avg auto/doc	16.79	28.61	18.06	29.68
macroa P	0.11	0.09	0.11	0.09
macroa R	0.54	0.71	0.55	0.72
macroa F1	0.19	0.16	0.18	0.15
microa P	0.07	0.06	0.07	0.06
microa R	0.55	0.73	0.59	**0.76**
macroa F1	0.13	0.11	0.13	0.10
mean F1	0.16	0.13	0.16	0.13

4.3 Terms analysis and shortened term lists

In the original term list there were 4411 distinct terms. In the data collection, 53% of them were found. The average length of the terms found was between one and two words, while the longer ones were less frequently found.

Of the terms found in the collection, based on 16% of them correct classes were always found, while based on 43% of them incorrect classes were always found. For a sample of documents containing terms that were shown to always yield incorrect results, we had a male subject expert confirm whether the documents were in the wrong class according to his opinion. For 10 always-incorrect terms with most frequent occurrences, the subject expert looked at 30 randomly selected abstracts containing those terms. Based on his judgments, it was shown that 24 out of those 30 documents were indeed incorrectly classified, but there were also 6 which he deemed to be correct. This is another indication of how problematic it is to evaluate subject classification in general, and automated subject classification in particular. Perhaps one way would be to have a number of subject experts agree on all the possible subjects and classes for every document in a test collection for automated classification; another way could be to evaluate automated classification in context, by end-users.

Based on the term analysis, three new term lists were extracted from the original one, and tested for performance:

1) Containing only those terms that found classes which were always correct (1308 terms). When cut-off is between 5 and 10, macroaveraged precision reaches 0.89, and microaveraged 0.99, when neither stemming nor stop-words removal are applied. Stemming does not really improve general performance because recall increases only little, by 0.03, while precision decreases by 0.2. However, when using only those 1308 terms, only 5% of documents are classified. The best mean F1, 0.15, is achieved when stemming and the stop-word removal are used.
2) Containing those terms that found classes which were correct in more instances than they were incorrect (1924 terms). This list yields best mean F1, 0.38. This value is achieved when stemming is used but no stop-words are removed. There are 65% of documents that are classified, with the average number of classes 1.7. When stemming is not used, precision levels are 0.75 for microaveraged, and 0.79 for macroaveraged.
3) Containing all terms excluding those that found classes which were always incorrect (4751 terms). The mean F1 is 0.25, when cut-off is 10 and both stop-words removal and stemming are used. The slight improvement in comparison to the original list is due to increase in precision.

5 Conclusion

In comparison to previous results (Golub 2006b) the experiment showed that the string-matching classification algorithm could be enhanced in the following ways:

1) Weights: adding different weights to the term list based on whether a term is single, phrase or Boolean, which type of class it maps to, and type of term relationship, improves precision, mean F1, and relevance order of assigned classes, the latter being important for browsing;
2) Cut-offs: selecting as final classes those above a certain cut-off level improves precision and F1. Assigning at least the class with highest score improves the number of documents that are classified, and the number of documents that are correctly classified;
3) Converting all phrases into Boolean terms increases the number of correct classes;
4) Stemming, stop-words removal or the two in combination improve precision and recall;
5) Score propagation improves finding the main class at the top three and two hierarchical levels;
6) Enhancing the term list with new terms based on morpho-syntactic analysis and synonyms acquisition improves recall;
7) Excluding terms that in most cases gave wrong classes yields best performance in terms of F1, where the improvement is due to higher precision levels; and
8) Best precision levels are achieved when only those terms that always gave correct classes are used.

The best achieved recall is 76%, when the basic term list is enriched with new terms, and precision 79%, when only those terms previously shown to yield correct classes in the majority of documents are used. Performance of individual classes, measured in precision, is up to 98%. At third and second hierarchical levels mean F1 reaches up to 60%.

These results are comparable to machine-learning algorithms (cf. Sebastiani 2002), which are considered to be the best ones but require training documents and are collection-dependent. Another benefit of classifying documents into classes of well-developed classification schemes is that they are suitable for subject browsing, unlike automatically-developed controlled vocabularies or home-grown directories often used in document clustering and text categorization (cf. Golub 2006a).

The experiment has also shown that different versions of the algorithm could be implemented so that it best suits the application of the automatically classified document collection. If the application requires high recall, such as for example in focused crawling, cut-offs would not be used. Or, if one provides directory-style browsing interface to a collection of automatically classified Web pages, Web pages could be ranked by relevance based on weights. In such a directory, one might want to limit the number of Web pages per class, e.g. assign only the class with highest probability that it is correct, as it is done in the "Thunderstone's web site catalog" (About the Thunderstone web site catalog 2007).

Most appropriate weights have still to be discovered. Future research should also involve testing automated classification in the context of an application and by end users, because of the problem of aboutness. The applicability of the string-matching approach mostly depends on the controlled vocabulary itself. While Ei proved to be suitable, which characteristics of controlled vocabularies are beneficial for automated classification needs to be further studied.

Acknowledgement

Many thanks to Traugott Koch for providing us with detailed comments on the manuscript, which helped improve the paper considerably. The authors also wish to thank two subject experts who helped in evaluation. This work was supported by the IST Programme of the European Community under ALVIS (IST-002068-STP).

References

About the Thunderstone web site catalog. 2007. Available: http://search.thunderstone.com/texis/websearch/about.html.

Aitchinson, J., Gilchirst, A., Bawden, D. 2000. *Thesaurus construction and use: A practical manual.* 4th ed. London: Aslib.

Aubin, S., Hamon, T. 2006. Improving term extraction with terminological resources. In Tapio Salakoski, Filip Ginter, Sampo Pyysalo, Tapio Pahikkala. (Eds.). *Advances in Natural Language Processing: Proceedings of the 5th International Conference on NLP, FinTAL*. Springer. 380-387.

Baayen, R. H., Piepenbrock, R., Gulikers, L. 1995. *The CELEX Lexical Database*: Release 2 [CD-ROM]. Philadelphia, PA: Linguistic Data Consortium, University of Pennsylvania.

Bang, S. L., Yang, J. D. Yang, H. J. 2006. Hierarchical document categorization with k-NN and concept-based thesauri. *Information Processing and Management, 42.* 387-406.

Chan, L. M. 1994. *Cataloging and classification: An introduction.* 2nd ed. New York: McGraw-Hill.

Chen, H., Dumais, S. T. 2000. Bringing order to the web: Automatically categorizing search results. In *Proceedings of International Conference on Human Factors in Computing Systems*. 145-52.

Compendex database. 2006. Available: http://www.engineeringvillage2.org/.

Ei thesaurus. 1995. 2nd ed. Edited by Milstead, J. Castle Point on the Hudson Hoboken: Engineering Information.

Fellbaum, C. 1998. *WordNet: An electronic lexical database.* Cambridge, MA: MIT Press.

Golub, K. 2006a. Automated subject classification of textual Web documents. *Journal of Documentation, 62(3)*. 350-371.

Golub, K. 2006b. The role of different thesauri terms in automated subject classification of text. In *Proceedings of the International Conference on Web Intelligence, Hong Kong, 2006*. 961-965.

Hamon, T., Nazarenko, A. 2001. Detection of synonymy links between terms: Experiment and results. *Recent Advances in Computational Terminology*. Amsterdam: John Benjamins. 185-208.

International Organization for Standardization. 1985. ISO 5963-1985: Documentation – Methods for examining documents, determining their subjects, and selecting index terms.

Intute: Engineering. 2007. Available: http://www.intute.ac.uk/sciences/engineering/

Jacquemin, C. 1996. A symbolic and surgical acquisition of terms through variation. In S. Wermter, E. Riloff & G. Scheler (Eds.). *Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing*. 425-438.

Jain, A. K., Murty, M. N., Flynn, P. J. 1999. Data clustering: A review. *ACM Computing Surveys, 3(31)*. 264-323.

Koch, T., Ardö, A. 2000. Automatic classification. *DESIRE II D3.6a, Overview of Results*. Available: http://www.it.lth.se/knowlib/publ/DESIRE36a-WP2.html.

Lancaster, F. W. *Indexing and abstracting in theory and practice*. 2003. 3rd ed. London: Facet.

Lewis, D. D., Yang, Y., Rose, T., Li, F. 2004. RCV1: A new benchmark collection for text categorization research. *The Journal of Machine Learning Research, 5*. 361-397.

Markey, K. 1984. Interindexer consistency tests: A literature review and report of a test of consistency in indexing visual materials. *Library & Information Science Research 6.* 155-77.

Medelyan, O., Witten, I. 2006. Thesaurus based automatic keyphrase indexing. In *Proceedings of the Joint Conference on Digital Libraries, Chapel Hill, NC, 2006.* 296-297.

Olson, H. A., Boll, J. J. 2001. *Subject analysis in online catalogs.* 2nd ed. Englewood, CO: Libraries Unlimited.

Onix text retrieval toolkit: Stop word list 1. 2006. Available: http://www.lextek.com/manuals/onix/stopwords1.html

Porter, M. F. 1980. An algorithm for suffix stripping. *Program, 14(3).* 130-137.

Salton, G., McGill, M. J. 1983. *Introduction to modern information retrieval*. McGraw-Hill, 1983.

Sebastiani, S. 2002. Machine learning in automated text categorization. *ACM Computing Surveys, 34(1)*. 1-47.

Svenonius, E. 1997. Definitional approaches in the design of classification and thesauri and their implications for retrieval and for automatic classification. In *Knowledge Organization for Information Retrieval, Proc. of the Sixth International Study Conference on Classification Research*. 12-16.

Svenonius, E. 2000. *The intellectual foundations of information organization.* Cambridge, MA: MIT Press.

WordNet Search: 3.0. 2007. http://wordnet.princeton.edu/perl/webwn.

Yang, Y. 1999. An evaluation of statistical approaches to text categorization. *Journal of Information Retrieval, 1(1/2)*. 67-88.