Domain Independent Automatic Keyphrase Indexing with Small Training Sets

Olena Medelyan and Ian H. Witten
Department of Computer Science
The University of Waikato
Private Bag 3105, Hamilton 3240
NEW ZEALAND
Phone: +64 7 838-4246, Fax: 858-5095
{olena, ihw}@cs.waikato.ac.nz

ABSTRACT

Keyphrases are widely used in both physical and digital libraries as a brief but precise summary of documents. They help organize material based on content, provide thematic access, represent search results, and assist with navigation. Manual assignment is expensive, because trained human indexers must reach an understanding of the document and select appropriate descriptors according to defined cataloguing rules. We propose a new method that enhances automatic keyphrase extraction by using semantic information about terms and phrases gleaned from a domain-specific thesaurus. The key advantage of the new approach is that it performs well with very little training data. We evaluate it on a large set of manually indexed documents in the domain of agriculture, compare its consistency with a group of six professional indexers, and explore its performance on smaller collections of documents in other domains and of French and Spanish documents.

1. INTRODUCTION

Seeking information is an important activity for everyone who uses computers in daily life. While the Internet is a bountiful source of all types of knowledge, locating relevant documents is still a great challenge—often compared figuratively to finding a needle in a haystack. The key problem with conventional free text search is formulating correct, accurate, and sufficiently pointed search terms. This

problem is so severe that an alternative approach is beginning to seep onto the Web, one with roots in traditional librarianship: subject indexing or "tagging," e.g. on websites like del.icio.us or technorati.com. Here, each document in a collection is indexed with a set of keyphrases ("tags") that reflect its principal topics. For example, the website of the algorithm described in this paper has been indexed by del.icio.us users with tags like *datamining*, *java*, *keywords*, *nlp*, *software*, *tools*.[1] Collaborative tagging communities are steadily growing in number, and ever more users prefer to explore so-called *tag clouds*, a natural way to discover new material (Begelman et al. 2006), over assiduously formulating free-text search queries.

Keyphrases[2] help organise documents and retrieve them based on content. Additionally, when selected from a controlled vocabulary keyphrases help combat polysemy and synonymy in natural language. But although tags and keyphrases offer significant advantages over free text search, assigning them manually to documents is so time-consuming and expensive that in practice, most electronic documents remain unindexed—and much of the indexing that does take place is done informally, by amateurs, creating so-called "folksonomies." This situation provides strong motivation for techniques that automate keyphrase indexing.

In free keyphrase indexing, or *keyphrase extraction*, phrases that occur in a document are analyzed to identify apparently significant ones on the basis of intrinsic properties such as frequency and length. Controlled indexing is usually performed by *term assignment*, where documents are classified according to their content into classes that correspond to elements of the vocabulary. A disadvantage of keyphrase extraction is that it often produces index terms that are ill-formed or inappropriate. A disadvantage of term assignment is that it requires a vast and accurate corpus of training material, which can only be produced by manually classifying training documents.

We have built a system for *keyphrase indexing*, an intermediate approach between keyphrase extraction and term assignment. It combines the advantages of both, while avoiding their shortcomings. It uses a domain-specific thesaurus as the controlled vocabulary. A machine-learning model is trained based

[1] http://del.icio.us/url/check?url=http://www.nzdl.org/Kea/

on features of related phrases in the thesaurus, along with other characteristics that are used in conventional automatic keyphrase extraction. Vocabulary terms are assigned to a test document by mapping its phrases to those in the thesaurus and using the learned model to decide which ones are significant descriptors of the content. The resulting set contains only well-formed phrases from the thesaurus that are strongly related to the given document.

We report several tests of the new method. First, we evaluate it on a large document collection. Second, we assess its consistency with keyphrase sets assigned independently by six professional indexers using a standard measure of inter-indexer consistency, and find that the new method comes quite close to the consistency level that humans achieve with each other. Third, we experiment with training sets of different sizes and show that good results can be achieved with just a few dozen training documents. Fourth, we investigate how performance changes when the new method is applied to other domains and languages. Although developed for indexing agricultural documents in English, it performs quite well on medical and physics documents, and on agricultural documents in Spanish and French. Finally, sample results are given for three different documents to convey a feeling for the overall quality of the keyphrases.

2. RELATED WORK

The problem of automatically identifying the main topics of documents has been researched for half a century (Luhn, 1959). We summarize recent work under three headings.

2.1 Term Assignment

Term assignment from a controlled vocabulary, or text classification, originally recruited human knowledge-engineering experts to create classification rules manually that could be applied to electronic texts. For example, Fuhr and Knorz (1984) describe a decision-making system that uses approximately 150,000 rules to map physics documents to index terms in a controlled vocabulary, called "descriptors." The rules have the form

IF <property> is identified in the document THEN <descriptor>,

---

[2] In this paper we use *keyphrase* and *index term* (or just *term*) synonymously.

where <property> involves detecting single words or phrases in the text, or recognizing chemical and physical formulas.

During the 1990s the focus of research shifted towards machine learning and other statistical techniques (Sebastiani 2002). Various inductive learning schemes have been applied to analyze the characteristics of manually classified documents and build rules that predict the classification of new documents. The classes correspond to index terms. Dumais *et al*. (1998) compared learning methods such as *Find Similar* (vector similarity with relevance feedback), *Decision Trees*, *Naïve Bayes*, *Bayes Nets* and *Support Vector Machines* (SVMs). The SVM method achieved the best result with an average accuracy of 87% over the 118 categories. Another effective technique is to use *classifier committees* (Sebastiani 2002), where the decision is an ensemble of classifiers that can be combined in different ways.

Plaunt and Norgard (1998) describe a collocation technique that maps lexical cues in the document to a controlled vocabulary using a set of automatically derived association rules. Each rule is a collocation of a document's phrase with a manually assigned descriptor, the strength of the association being computed using the likelihood ratio statistic. The scheme was evaluated by downloading documents from the INSPEC database that contain the words *libraries*, *library*, *information science*, *linguistics*, or *sigir* in their title. From this highly focused collection of 4,100 training documents around 27,500 rules were extracted that covered 1,100 of the 6,500 descriptors in the INSPEC thesaurus. Precision and recall for the top 10 assigned descriptors were 21% and 64% respectively.

Markó *et al*. (2004) assign Medical Subject Headings (MeSH terms) to medical abstracts using a probabilistic technique. Prior to training, they map vocabulary terms to subwords defined in the MorphoSaurus lexicon,[3] which involves orthographic, morphologic and semantic normalization. The probability of a vocabulary term being a keyphrase is the product of the conditional probabilities of its trigrams in documents to which the term was manually assigned, divided by the probability of the

[3] MorphoSaurus is maintained manually and comprises over 20,000 equivalent classes for English, German, Spanish and Portuguese subwords. Subwords are semantic units and morphemes that link to equivalent expressions in each language—for example, *leukemia* becomes *leuk*, *em*, *iag*. See Markó *et al*. (2004) for details.

trigrams over all training documents. They achieve 30% precision and recall for the top 10 terms by combining this method with a weighting scheme that takes into account parameters such as the longest match of a subword sequence to a MeSH term and its appearance in a significant position in a document.

The disadvantage of all these text classification techniques is that they require training data for every potential class—that is, for *every* term in the controlled vocabulary. Several documents are needed per term to form an adequate model. For example, Pouliquen *et al*. (2003) used a training set with 60,000 manually indexed documents for a vocabulary of around 6,000 terms; most domain-specific thesauri are far larger than this. Markó et al. (2004)'s technique appears to be more parsimonious: they used 35,000 abstracts to create a training model for a subset of the 20,000-item MeSH vocabulary (we do not know how many terms were actually assigned to these abstracts). However, their method requires a manually created lexicon for morphological decomposition.

2.2 Keyphrase Extraction

The idea of keyphrase extraction is to analyze phrases that occur verbatim in the document to identify apparently significant ones on the basis of intrinsic properties such as frequency and length. Thus such systems are not restricted to assigning index terms from a pre-defined vocabulary. Extraction takes place in two stages: selecting candidate phrases (n-grams or noun phrases, depending on the approach) in the document, and filtering out the most significant ones to serve as keyphrases.

Methods for candidate selection vary from n-gram extraction (Turney, 1999; Witten *et al.* 1999; Barker and Cornacchia, 2000) to shallow parsing (Hulth, 2004). N-grams often yield ungrammatical phrases; parsing is potentially more accurate because it utilizes part-of-speech information and grammatical rules for identifying valid noun phrases. Filtering uses either simple statistics, where a fixed weighting scheme is applied to rank phrases according to their score (Barker and Cornacchia, 2000; Paice and Black, 2003), or machine learning, where the ranking function is defined by a statistical model derived from a training set with manually-assigned keyphrases (Turney, 1999; Witten *et al.*, 1999; Hulth, 2004). The latter approach has the potential to take account of a particular document set's characteristics.

The disadvantage of these techniques is that the resulting phrases may be ill-formed or inappropriate.[4] Furthermore, whereas controlled vocabularies enforce consistency in the selection of synonyms for the same concept and eliminate polysemy through broader and narrower links between terms, free indexing does not—even when the extracted keyphrases are correct.

2.3 Keyphrase Indexing

Combining keyphrase extraction with ideas from text classification promises a practical solution that does not depend on massive training data. The phrases in a document are first mapped to terms in a controlled vocabulary; then their properties are analyzed to identify the most significant terms.

Tiun *et al.* (2001) describe a mapping from phrases in web pages to categories in the Yahoo directory, via synonyms obtained from WordNet. They achieve a precision of 30% when mapping 200 documents into 100 Yahoo categories (recall was not reported). Golub (2006) indexed documents with a vocabulary consisting of 800 descriptors and 22,000 non-descriptors (not-allowed terms that are synonyms of descriptors). After mapping document phrases to descriptors she applied a weighting scheme that takes into account term frequency and position in the document, yielding an F-measure (defined in Section 4) of 26%.

Aronson *et al.* (2000) describe a sophisticated system for indexing medical texts with MeSH terms. It first identifies potential index terms by mapping the phrases and letter trigrams in the document to concepts in the medical UMLS thesaurus. This list of candidates is then enriched with MeSH terms that have been previously assigned to the 100 closest documents in the PubMed collection. This introduces an element of text classification, because the test document is compared to all manually indexed documents. In a final clustering stage, each term is weighted according to where and how it has been detected, and the weights of semantically similar terms representing a cluster are combined into a final score. In an experiment with 500 full text articles this system achieved 60% recall and 31% precision for the top 25

[4] Even if partial parsing is used, the extracted noun phrases might be ill-formed, because current technology cannot produce completely accurate results (e.g. “first problem concerns”); or inappropriate, because parsing does not necessarily extract

extracted terms (Gay *et al.*, 2005). However, it seems to involve the entire PubMed collection, which includes many million manually pre-indexed documents, and it is not clear whether the results would hold if less training material is used.

Both Tiun *et al.*'s (2001) and Golub's (2006) work involve rather small vocabularies (100 and 800 terms respectively). Aronson *et al.*'s (2000) does not, but it is closely tailored to the medical domain and requires a huge database of manually-indexed documents for training. The scheme proposed in the present paper is suitable for very large vocabularies, requires only a small amount of training data, and can easily be applied to other domains.

3. KEYPHRASE INDEXING ALGORITHM

The new keyphrase indexing algorithm, called KEA++,[5] works in two stages: *candidate identification* and *keyphrase selection.* The first stage identifies candidate terms that relate to the document's content, including ones that appear verbatim as phrases in the document. The second uses a model to identify the most significant terms based on certain properties or "features." This involves first learning a model based on training data, and then applying it to test documents. Candidate identification is described in the next section; following that we define the features that are used to characterize the phrases. Then we explain how the model is learned, and finally show how it is applied to previously unseen test documents.

The candidate identification process, and one of the features, both depend upon a thesaurus. KEA++ can utilize as the thesaurus any knowledge structure expressed in SKOS (Simple Knowledge Organization System) format.[6] This is a simple RDF format for defining individual terms and semantic relations between them. SKOS supports multilingual thesauri, and KEA++ can be used for indexing in different languages (see Section 5).

---

meaningful phrases (e.g. "presented research"). Both examples are keyphrases extracted from Hulth (2004) to demonstrate her approach.

[5] Throughout this paper KEA++ refers to KEA-5.0, which is freely available from *http://www.nzdl.org/Kea* under the GNU General Public License. It is a development of KEA-4.0, a domain-restricted system for controlled keyphrase indexing described by Medelyan and Witten (2006). Earlier versions of KEA perform keyphrase extraction rather than indexing, and are described by Witten *et al.* (1999).

[6] *http://www.w3.org/2004/02/skos/*

3.1 Candidate Identification

Each document is first segmented into tokens on the basis of white space and punctuation. Clues to syntactic boundaries, such as punctuation marks, digits and paragraph separators, are retained. Next, all word n-grams of up to the length of the longest term in the controlled vocabulary[7] that do not cross these boundaries are extracted, and the number of occurrences of each n-gram is counted.

Most extracted n-grams are ungrammatical or meaningless in isolation. Unlike other keyphrase extraction algorithms, which select candidates using either rough heuristics based on punctuation and stopwords or syntactic processing and noun-phrase identification, KEA++ selects them with reference to a controlled vocabulary. To achieve accurate matching with a high degree of conflation, each n-gram is transformed into a *pseudo-phrase* in three steps:

- remove all stopwords from the n-gram;
- stem the remaining terms to their grammatical root (Porter, 1980);
- sort them into alphabetical order (Paice and Black, 2003).

This matches phrases such as *algorithm efficiency*, *an efficient algorithm* and even *these algorithms are very efficient* to the same pseudo-phrase *algorithm effici*, where *algorithm* and *effici* are the stemmed versions of the corresponding full forms. In the next step pseudo-phrases in the document are matched against terms in the vocabulary, also represented as pseudo-phrases. Matching n-grams are identified with the corresponding term. Each term receives a count which is the sum of the occurrence counts of its associated n-grams.

Terms are also conflated by replacing non-descriptors, which are synonymous alternative versions, by their equivalent descriptors using the synonymy links in the thesaurus. The occurrence count of the corresponding descriptor is increased accordingly. This operation recognizes terms whose meaning is equivalent, and greatly extends the usual approach of conflation based on word-stem matching. It also allows terms that do not actually appear in the document to be assigned (unlike conventional keyphrase

[7] Five words in the case of the vocabulary we used.

extraction). The result is a set of candidate index terms for a document. These are all grammatical terms that relate to the document's content; each has an occurrence count. The next step is to identify a subset containing the most important of these candidates.

3.2 Feature Definition

A simple and robust machine learning scheme is used to determine the final set of index terms for a document. It uses as input a set of attributes, or features, defined for each candidate term. We have analyzed individual and combined performance of four features: TF×IDF, position of the first occurrence, length and node degree.

The **TF×IDF** score compares the frequency of a phrase's use in a particular document with the frequency of that phrase in general use. General usage is represented by the number of documents containing the phrase in a global corpus of documents. Before starting to calculate any features, KEA++ creates a file that stores each pseudo-phrase and a count of the number of documents in which it appears. With this file in hand, the TF×IDF score for phrase P in document D is:

$$\text{TF}\times\text{IDF} = \frac{\text{freq}(P,D)}{\text{size}(D)} \times -\log_2 \frac{\text{count}(P)}{N},$$

where freq(P,D) is P's occurrence count in D, size(D) is the number of words in D, count(P) is the number of documents containing P in the global corpus, and N is the number of documents in the global corpus. By default the training set serves as the global corpus, but this can be changed if required.

The first component in the expression above is the term frequency TF; the normalized frequency of term P in document D. The second is the logarithm of the inverse document frequency IDF, and is larger for rarer phrases that are more likely to be significant. Note that TF×IDF does not refer to a particular formula: heuristics are called "TF×IDF" whenever they use term frequency in a monotonically increasing way and a term's document frequency in a monotonically decreasing way. We use the logarithm of document frequency because this is common practice in information retrieval (Salton and Buckley, 1988).

The **position of first occurrence** of a term is calculated as its distance in words from the beginning of the document, normalized by the document's word count. The result represents the proportion of the

document that precedes the phrase's first appearance. Candidates that have extreme (high or low) values for this feature are more likely to be valid index terms because they appear either in the opening parts—title, abstract, table of contents, introduction—or the final sections—conclusion and reference list. Professional human indexers commonly focus on these portions in order to assign keyphrases to lengthy documents without having to read them completely (David et al., 1995). A better strategy might be to identify the structural position of a phrase, e.g. in title, abstract, particular sections, figure and tables captions; however, not all documents are available in suitably structured form. We experimented with features that depend upon all occurrences of a phrase, such as the span or standard deviation across occurrence positions, but were unable to obtain any improvement.

The **length** of a candidate phrase in words is used as another feature because statistical analysis of one batch of experimental data revealed that indexers prefer to assign descriptors consisting of two words, whereas most terms in the corresponding thesaurus had one word. Including phrase length as a feature allows the system to take account of such preferences in an empirical way.

Node degree measures how richly the term is connected in the thesaurus graph structure. The degree of a term is the number of semantic links that connect it to other terms—for example, a term with one broader term and four related terms has degree 5. The node degree feature represents the number of links that connect the term to other phrases in the document that have been identified as candidate phrases. A document that describes a particular topic area will cover many thesaurus terms from this domain; therefore candidate phrases with high node degree are more likely to be significant.

KEA++ discretizes these numeric features into a nominal form using the supervised method of Fayyad and Irani (1993). During the training process, a discretization table for each feature is derived from the training data. This table gives a set of numeric ranges for each feature, whose values are replaced by the range into which they fall (cf. Witten et al., 1999, for an example of such a table).

3.3 Training: Building the Model

The model is built from training data, that is, documents to which terms have been manually assigned. For each training document, candidate pseudo-phrases are identified and their feature values calculated as described above. Each phrase is then marked as a positive example if and only if that term has been assigned to the document by a professional indexer. The resulting feature is the *binary class* used by the machine-learning scheme.

For training corpora where documents have been simultaneously indexed by several humans, each positive example is assigned a "degree of correctness." This *numeric class* is determined as the number of human indexers that have chosen the term, divided by the total number of indexers: thus a term chosen by 3 out of 6 indexers receives the class value 0.5. Section 6.7 demonstrates how indexing performance improves when the system has the opportunity to learn from multiple indexers.

The machine learning algorithm generates a model from the training data that predicts the class using the values of the features. We have experimented with several different classifiers, including *Naïve Bayes*, *Support Vector Machines*, *Decision Trees*, *Linear Regression*, and *Bagging* of decision stumps. The best results were achieved with the Naïve Bayes classifier, which assumes that the features are independent of each other, given the class value.

To predict a numeric class we treat the class value as nominal, with possible values equal to the number of indexers plus one. The predictions are the average of the class values, weighted by the predicted probability of each class.

3.4 Extracting Index Terms from New Documents

To select index terms from a new document, its candidate pseudo-phrases and their feature values are determined. In most experiments the document frequency for TF×IDF was calculated from the training set because cross-validation and leave-one-out approaches involve small test sets; though any suitably sized document corpus would suffice instead. Then the Naïve Bayes model built during training is applied to

determine the overall probability that each candidate is an index term. Finally the terms with the greatest individual probabilities are selected as keyphrases.

Suppose (for brevity of exposition) that just two features, TF×IDF and position of first occurrence (*first*), are being used, with a binary class value. When the Naïve Bayes model is used on a candidate pseudo-phrase with these features, two quantities are computed:

$$\mathrm{P}[yes] = \frac{Y+1}{Y+N+2}\ \mathrm{P}_{TF\times IDF}[TF\times IDF \mid yes]\ \mathrm{P}_{first}[first \mid yes]$$

and an analogous expression for P[*no*], where *Y* is the number of positive instances in the training files—that is, manually-assigned terms—and *N* the number of negative instances—that is, candidate phrases that have not been assigned as terms.[8] Here, $\mathrm{P}_{TF\times IDF}[]$ is the probability distribution function computed from the training data for the *TF×IDF* feature, and $\mathrm{P}_{first}[]$ is the analogous function for the *first occurrence* feature. The overall probability that the candidate phrase is a keyphrase is then:

$$p = \mathrm{P}[yes] / (\mathrm{P}[yes]+\mathrm{P}[no])$$

Candidates are ranked according to this value. TF×IDF (in its pre-discretized form) is used as a tiebreaker if two phrases have equal probability.

The user can determine how many keyphrases are included into the final set. This can be done by specifying

- a numeric probability threshold
- or the number of terms required per document.

Of course, in practice human indexers assign different numbers of terms to each document, and the first option permits the machine to do this too. However, we have not yet succeeded in exploiting this flexibility to produce measurably superior keyphrase sets. Our data demonstrates that while the number of terms assigned by human indexers does differ from document to document, there is little variation when these values are averaged over all indexers.

[8] The additional 1 in the numerator and 2 in the denominator appear because the Laplace estimator is used to avoid zero probabilities.

KEA++ can be used for semi-automatic indexing by setting the threshold to generate a long list of potential keyphrases, from which a human indexer selects the most appropriate.

4. MEASURING INDEXING CONSISTENCY

Several formulae have been proposed for measuring indexing quality. *Inter-indexer consistency* and *agreement* are commonly used in the context of human indexing (Zunde and Dexter, 1969), while *precision* and *recall* are used to evaluate automatic systems (van Rjsbergen, 1979). These measures all involve comparing the number of matching ("correct") terms with the sizes of the two term sets being compared. As we shall see, they are closely related.

Inter-indexer consistency is defined as "the degree of agreement in the representation of the (essential) information content of a document by certain sets of indexing terms selected individually and independently by each of the indexers" (Zunde and Dexter, 1969). Hooper (1965) quantified it as

$$Hooper = \frac{C}{C + M + N},$$

where C is the number of terms two indexers have in common, and M and N respectively are the number of idiosyncratic terms that they assign. (Hooper multiplied this figure by 100 to express it as a percentage.) Another measure was proposed by Rolling (1981):

$$Rolling = \frac{2C}{A + B},$$

where again C is the number of terms the indexers have in common and A and B are the total number of terms they assign.

Both measures range from 0 when the two indexers assign disjoint sets to 1 when they assign identical sets. Expressing them in the same terms reveals that Hooper's is always smaller than Rolling's, since

$$Hooper = \frac{C}{A + B - C} = \frac{Rolling}{2 - Rolling}.$$

Furthermore, the two measures are the same as the Jaccard and Dice coefficients used to measure statistical similarity between sets A and B:

$$Jaccard = \frac{A \cap B}{A \cup B} = Hooper \qquad Dice = \frac{2A \cap B}{|A| + |B|} = Rolling$$

In studies of automatic keyphrase indexing, manually defined keyphrases are generally considered as the gold standard against which the automatically generated ones are compared. Two measures are used together: *precision* (*P*) expresses the number of matching ("correct") keyphrases as a proportion of all extracted phrases and *recall* (*R*) is the proportion of manually assigned phrases that are covered:

$$P = \frac{\#\ correct\ extracted\ keyphrases}{\#\ all\ extracted\ keyphrases} \qquad R = \frac{\#\ correct\ extracted\ keyphrases}{\#\ manually\ assigned\ keyphrases}$$

The *F-measure* combines the two, and in its full generality involves a parameter β which allows the evaluator to give more weight to either of them (van Rijsbergen, 1979):

$$F_\beta = \frac{(1+\beta^2)PR}{\beta^2 P + R}$$

In this paper we take β = 1 throughout. Expressing precision and recall in the same terms as the inter-indexer consistency measures, the F-measure coincides with Rolling's measure:

$$F_1 = \frac{2PR}{P+R} = \frac{2C}{A+B} = Rolling$$

The Kappa statistic, which takes into account the proportion of times the indexers would agree by chance, is another way of computing indexing consistency. It involves the negative counts—that is, the number of possible keyphrase choices that the indexers did *not* make. For free indexing, and for indexing with large controlled vocabularies, Kappa is the same as the F-Measure (Hripcsak and Rothschild, 2005).

5. EXPERIMENTAL DATA

While developing the algorithm we evaluated its performance on agricultural documents obtained from the UN Food and Agriculture Organization (FAO). The FAO has a mandate to increase agricultural productivity and improve the conditions of rural populations worldwide. It collects, analyses, and disseminates information, and maintains an online document repository that is large and well used (1M hits/month).[9]

[9] *http://www.fao.org/documents/*

We have obtained several sets of documents from the FAO: 780 English, 60 French and 47 Spanish documents, each indexed by one indexer, and a further 30 English documents indexed by 6 professional indexers working independently. Although the non-English document sets are small, they allow us to assess KEA++'s language independence. Likewise, the multiple-indexer data allows the system's consistency with humans to be compared to their consistency with each other. This dataset has been specifically created for our experiments. To investigate the method's domain-independence, we also use medical and physics document sets. The following sections describe each corpus and the controlled vocabularies in detail. We also give an overview of the evaluation techniques used.

5.1 Agricultural Corpora and the Agrovoc Thesaurus

The documents from the FAO are manually indexed by professional FAO staff with terms from the Agrovoc thesaurus[10], which has been translated into five languages. Agrovoc is specific to the domain of agriculture and defines over 28,000 concepts. All have one preferred term ("descriptor"), and many have several alternative versions ("non-descriptors"), resulting in a total size of around 40,000 terms. The concepts are interconnected by 83,000 semantic relations of three types: bi-directional links between related terms (RT) and inverse links between broader terms (BT) and narrower ones (NT). The BT and NT links build a hierarchical structure of seven specificity levels. Figure 1 shows a typical Agrovoc entry.

The main training and evaluation corpus comprises 780 full-text documents selected randomly from the FAO's repository. The documents average 30,800 words (a total of 24 million), ranging from 1200 to 257,000 words. The FAO indexers have assigned an average of 8 Agrovoc descriptors to each document, ranging from 2 to 23. This total of 6225 term assignments includes 2187 different terms.

A different corpus with 30 new agricultural documents had keyphrase sets independently assigned by 6 professional indexers at FAO, with an average of 9.4 Agrovoc terms to each document, ranging from 4 to 21 terms. We used this collection to compare the consistency of KEA++ with human indexers to their consistency amongst each other.

[10] *http://www.fao.org/agrovoc*

To investigate whether KEA++ is language independent, we used all the Spanish and French documents included in the FAO document repository (January – September 2003) that had at least 3 manually assigned keyphrases. The Spanish collection contained 47 documents, averaging 42,500 words. The French collection contained 60 documents, averaging 22,400 words. These documents had been indexed with English keyphrases, which we mapped to the equivalent Spanish and French terms using Agrovoc, resulting in an average of 10.2 and 11.4 terms for the Spanish and French documents respectively. The number of keyphrases per document in both collections varied from 2 to 35 terms per document, with a standard deviation of 6.5 (cmp. to English standard deviation of 3.5).

5.2 Medical Documents and the MeSH Thesaurus

The Medical Subject Headings (MeSH) were developed by the U.S. National Library of Medicine (NLM) specifically for indexing medical articles. We use the SKOS version provided by van Assem *et al.* (2006).[11] It contains 24,000 concepts, each defined by one preferred term and (usually) several alternatives, including spelling and formatting variants. The total number of terms (including alternatives) is 140,000, many times more than Agrovoc. The terms are organized into a hierarchy via 32,000 BT/NT links.

We used a collection of 500 medical full texts provided by the NLM Indexing Initiative. The source and content of this corpus is described by Gay *et al*. (2006), who used it to evaluate their indexing system (cf. Section 2.3). The documents average 4500 words and are indexed by 15 MeSH terms, but are quite heterogeneous, with lengths varying from 440 to 24,500 words and the number of assigned terms varying from 2 to 30.

5.3 Physics Corpus and the HEP Vocabulary

The High Energy Physics (HEP) thesaurus was developed by the Deutsches Elektronen-Synchrotron. It was recently converted to SKOS format[12] in a joint effort with the European Organization for Nuclear

[11] *http://thesauri.cs.vu.nl/*

[12] *http://cdsware.cern.ch/tmp/bibclassify/hep.html*

Research, who use it to index the contents of the CERN Document Server. This is the smallest of the three thesauri used in our study. It defines 16,000 concepts with one preferred term each, and includes a further 700 alternative terms for some concepts. Surprisingly, there are only 500 broader, narrower and related links connecting the concepts, while the most common semantic relation is Composite/CompositeOf, of which there are 15,300. The thesaurus is a very specific one—for example, the concept *Einstein equation: solution* has two CompositeOf relations: *Einstein equation* and *Solution*.

The experimental corpus comprises 290 random documents from the CERN Document Server with at least four manually defined HEP terms each. The documents have an average length of 6,300 words and an average of 7 assigned terms.

5.4 Evaluation Techniques

Three standard techniques used in machine learning were used to estimate KEA++'s performance: 10-fold cross-validation, leave-one-out and random sampling (Witten and Frank, 1999).

For *10-fold cross-validation*, the document set is partitioned randomly into 10 sets. Testing is performed on one set and the rest are used for training; the procedure is repeated 10 times so that every set, and every document, is used exactly once for testing. For example, with 780 documents the system is trained on 702 documents and testing on the remaining 78, for each of 10 runs. The results are averaged over all documents and runs. Cross-validation helps mitigate any bias produced by random splits.

Leave-one-out evaluation is *n*-fold cross-validation where *n* is the number of documents in the corpus. Each document in turn is held back and the remainder are used to create a training model which is tested on the final document. This method squeezes the maximum information from small datasets, obtaining the most accurate possible estimate.

Training data—i.e., manually indexed documents—is usually difficult to obtain, and it is interesting to know how much is necessary for good results. Witten *et al*. (1999) showed that for keyphrase extraction without a controlled vocabulary, performance does not improve once the training set reaches 50 documents. We repeat the experiment for controlled keyphrase indexing on our data, by using *random*

samples of different sizes from the training set. For each size, 10 samples are taken and the results are averaged over all runs.

6. EVALUATION RESULTS

In order to provide a baseline for comparison, we used a simplified indexing approach: first map all the phrases in each document to vocabulary terms (using the same strategy as in KEA++), and then select the phrases with the greatest TF×IDF values as index terms. Because it incorporates pseudo-phrase matching and non-descriptor matching, this turns out to be a very powerful baseline technique which it is difficult to improve upon. Any difference in performance reflects the additional features and the advantage of using machine learning techniques to incorporate this new information.

Our evaluation addresses the following questions:

- Does controlled keyphrase indexing outperform free indexing?
- How much does each feature improve indexing performance over the baseline?
- What advantage is gained over the baseline by using all features together?
- How does KEA++'s performance compare to human performance?
- Is KEA++ language independent?
- Is KEA++ domain independent?
- How much training data is required to achieve good results?
- Does simultaneous learning from several humans outperform learning from a single indexer?

The following sections describe experiments with KEA++ that answer these questions.

6.1 Controlled versus Free Indexing

KEA++ differs from other keyphrase extraction systems described in Section 2.2 in its use of a controlled vocabulary for candidate identification. Whereas in free indexing keyphrases are restricted to terms that appear in the document, in controlled indexing only terms listed in the vocabulary are eligible keyphrases. Links between descriptors and non-descriptors allow terms that do not appear in the document verbatim to

be chosen as candidate phrases. In order to determine the advantage of controlled indexing, we compared the candidate identification technique in KEA++ to *n-gram extraction*, which is commonly used in keyphrase extraction. As noted earlier, identification of n-grams is a pre-processing step both in KEA++ and the baseline approach, before document phrases are mapped to vocabulary terms.

Candidates extracted from each of the 780 documents using (a) n-grams ($n \leq 5$) and (b) KEA++ candidate identification (Section 3.1) are compared with manually assigned index terms. On average, both techniques identify around 6 out of the 8 manually assigned keyphrases for each document (75%). However, the n-gram extraction produces 15–20 times as many candidate phrases as KEA++, depending on the stemming method.

Table 1 summarizes precision and recall for the two approaches; the best values are shown in bold. KEA++ outperforms the n-gram method by a factor of 10–13 in precision, depending on the stemming method; its recall is marginally greater too. One reason for the improvement in recall is that non-descriptors appearing in the document are matched to corresponding descriptors. For example, in a document on *smallholders* n-gram extraction failed to identify the candidate term *small farms*, which KEA++ found using a synonymy link. Stemming[13] improves recall by 4.6 further percentage points. However, stemming extracts more candidate phrases, giving a potential source of error in the following stages that offsets the marginal improvement in recall. The recall values give a baseline for the best possible results that could possibly be achieved by the candidate selection stage.

6.2 Evaluation of the Filtering Features

Given the set of candidates, the next step is to identify keyphrases among them. An obvious solution is to use our baseline method: choose the ones with the highest TF×IDF values. Our data set also includes the features defined in Section 3.2. The machine learning scheme explained in Section 3.4 determines a good

[13] We have experimented with different stemming strategies, including removing just the plural endings and the (Iterated) Lovins stemmer. Best results were achieved with the Porter stemmer (Porter, 1980).

way of combining these feature values. Here we assess each feature individually by adding them to the baseline system one by one.

Table 2 presents the results obtained on the main (780-document) collection, using 10-fold cross-validation. Precision and recall were computed for each document individually and then averaged over the test collection; the F-Measure combines the two into a single figure. For each document the top 8 keyphrases were extracted, 8 being the average number of terms assigned by human indexers. The most powerful feature (besides TF×IDF) is *first occurrence*, which adds 5 percentage points to the overall F-Measure. *Node degree* contributes 1.5 percentage points. *Length* does not provide any significant improvement by itself, but when added as a fourth feature to TF×IDF, *first occurrence* and *node degree*, it improves result by 1.5 percentage points (cf. the difference between the last two rows of Table 2).

Table 2's last row shows KEA++'s performance on the 780-document set when all 4 features are used. Combining the features results in a total gain of 7.4 percentage points in F-measure over the baseline performance. According to a one-tailed t-test, KEA++ outperforms the baseline method at a significance level substantially better than 0.01%.

6.3 KEA++ vs. Professional Human Indexers

The results in Section 6.2, precision of 27.0% and recall of 33.4%, are disappointingly far from the 100% in each that would reflect a perfect score. However, indexing is an inherently subjective task. The evaluation method used so far assumes that the terms assigned manually by one indexer are "correct," and that ideally automatically extracted keyphrases would match this indexer exactly. However, analyzing terms assigned to documents by human indexers reveals that their selections seldom agree, and the definition of the "correctness" of a keyphrase is subjective. In this section we regard the level of inter-indexer consistency reached by several professional indexers as a gold standard. The ultimate goal is to develop an automatic indexing method that is as consistent with a group of indexers as they are amongst each other.

We used a second document collection, consisting of 30 English agriculture documents indexed by six professionals, to compute inter-indexing consistency using Rolling's measure, and applied the same measure to keyphrases assigned by KEA++ after training it on 200 randomly selected documents from the main collection. The optimal settings determined in the previous section were used for this experiment: pseudo-phrase conflation, the Porter stemmer and all four features. Ten keyphrases were included in the final set because this was the average number of manually assigned keyphrases in this set.

Table 3 shows the results. The *vs. Indexers* column summarizes the average percentage consistency of each human indexer individually, compared with the other five. The other two columns give consistency values obtained when keyphrases were extracted automatically using the baseline approach and KEA++ and compared to all 6 human indexers. While it did not achieve the same consistency level as humans, KEA++ does not fall far behind—it is on average 30% consistent with humans, whereas they are 39% consistent amongst each other. The consistency of the baseline is distinctly lower, at 25.5%.

6.4 Experiments with Other Languages

Table 4 summarizes the results obtained when assigning terms to Spanish and French documents, after training on 200 randomly selected English documents. We trained on English because the Spanish and French sets were too small to support both training and testing, and because we believe that the properties used by KEA++ to identify terms are not language specific, at least to a first approximation.[14] The only language specific element, TF×IDF, was based on the test collection in each language, rather than using the values provided in the English model. For each document the top 10 Spanish terms and the top 11 French terms (which were the average number of manually-assigned terms in each collection) were extracted.

Table 4 compares KEA++'s performance on these sets with the baseline. The results are significantly worse than those achieved on English documents, and the difference between the baseline and KEA++ is less than 2 percentage points on both languages (and is not statistically significant at the

5% level). Analysis of the manual keyphrases shows that on average only 66% of Spanish and 74% of French keyphrases actually appear in the document: this is the maximum recall that our system can achieve. This partially explains why recall is significantly lower than on English documents, where 81% of keyphrases appear in text.

As mentioned in Section 5.1, the original keyphrase sets for Spanish and French documents contained English keyphrases. We surmise that they have been assigned by English indexers to the English versions of these documents, not by native speakers of Spanish and French. Also, the number of keyphrases per document varies significantly more than for the English collection. These observations indicate that these two non-English sets are of poorer quality than the English one. The positive findings are that while on average two out of 10 automatically assigned keyphrases match exactly, two further keyphrases are semantically related to at least one of the manual keyphrases.

In general, our findings are similar to those of Markó *et al.* (2004), who tested a multi-language indexing system on German and Portuguese medical articles and achieved significantly lower results than on English documents.

6.5 Experiments with Other Domains

As noted earlier, KEA++ can be used with any controlled vocabulary in SKOS format. Many existing vocabularies—including WordNet—are now freely available in this format.[15] We tested KEA++'s performance on medical documents indexed with MeSH terms and physics documents indexed with HEP terms (cf. Sections 5.2 and 5.3 respectively). In both cases 10-fold-cross validation was applied, and the precision, recall and F-measure values in Table 5 were averaged over all documents and test runs. For the evaluation we used the top 13 MeSH terms and the top 8 HEP terms, the average size of the term sets in the two test collections. We compare results with the baseline approach, and with the performance of systems developed specifically for indexing documents in these two domains.

---

[14] The phrase length feature is probably less useful for languages like German where compounds are written in one word, but this could be rectified by decompounding words first.

[15] http://esw.w3c.org/topic/SkosDev/DataZone

In experiments with medical documents and the MeSH thesaurus, KEA++'s F-measure (reported in Table 5) is only 3.2 percentage points below what was achieved for agricultural documents (Table 2). However, there is no improvement over the baseline, indicating that the additional features developed for agricultural data do not help much when filtering candidate phrases in MeSH. Additional investigation of the characteristics of manually assigned MeSH terms is required to improve the performance.

It is interesting to compare KEA++'s performance (26.0% precision and 27.3% recall) with that of other systems that were specifically developed for indexing with medical terms and use far more manually indexed documents for training. Gay *et al*. (2005) use the same 500 documents for testing their Medical Text Indexer and report precision of 31% and a remarkable recall of 60% on the top 25 terms. These results are impressive; however, this system was trained on the entire PubMed collection—millions of manually indexed documents. Markó *et al.* (2004) achieved 30.2% precision and 32.8% recall on the top 10 keyphrases (in a different test set) after training on 35,000 medical abstracts.

The second part of Table 5 compares KEA++'s performance on physics documents with the baseline and with the bibClassify system.[16] BibClassify is a module of CDS Invenio, a digital library system developed at CERN, and was developed specifically for indexing physics documents: it combines term frequency statistics with the compound relation defined in the HEP thesaurus. Surprisingly, its performance (F-measure of 14.7%) is outclassed not only by KEA++ (21.2%) but also by the baseline (17.1%). The difference probably reflects the fact that the baseline uses semantic conflation—it replaces non-descriptors that appear in the document by their preferred labels. It seems that BibClassify does not take advantage of these links.

We made no modifications to KEA++ before applying it to these domains. Terms assigned from the MeSH and HEP thesauri are quite complex and therefore difficult to match (e.g. *Delivery of Health Care, Integrated*; *gauge field theory: Yang-Milles*). Sophisticated techniques are needed to match document phrases to these terms, especially to yield recalls as high as the 60% achieved by Gay *et al*. (2005). Also,

[16] http://cdsware.cern.ch/tmp/bibclassify/bibclassify.html

when indexing physics documents human indexers prefer to assign compound terms (65% of index terms are compound terms), and KEA++'s performance would be enhanced by including a feature that utilized this characteristic of the training data.

6.6 How much Training Data is Needed?

Witten *et al.* (1999) investigated how the number of training documents affects the performance of their keyphrase extraction algorithm KEA, the parent of KEA++. They varied the training set from 1 to 130 documents and found that the performance improves steadily up to 20 documents, making smaller gains until 50 documents, and changing little thereafter.

We repeated this experiment by reserving 500 documents from the main corpus exclusively for testing, and taking training sets of different sizes by sampling the remaining 280 documents, repeating 10 times for each training set size. Figure 2 demonstrates, how precision and recall increase with the size of the training set. A steady increase of both precision and recall can be observed up to 50–100 documents, with continued smaller gains thereafter. It seems likely that more training data will provide small further increases.

6.7 Learning the Degree of Correctness

KEA++ implicitly assumes that all manually assigned terms in the training set have the same degree of correctness. However, the extent of agreement amongst human indexers varies from one term to another. Among our 30 multiply-indexed documents, there are only one or two terms per document that all indexers agree on, while most terms are idiosyncratic choices by individual indexers. We hypothesize that KEA++'s performance could be improved if the learning process took the degree of correctness of each manually-assigned term into account.

To test this, we used the 30 document agriculture collection in the following way. For each document we create a joint keyphrase set consisting of all keyphrases assigned by all 6 indexers. The first version of KEA++ uses the original training strategy: if a document phrase appears in the manual keyphrase set, its class value is *true*, otherwise *false*. We call this the "binary class" version. For the

second version we change this strategy: the class value of a keyphrase is the number of human indexers that have chosen it divided by the total number of indexers. We call this the "nominal class" version because although the class is expressed as a real number, it can only take on 7 possible values (one more than the number of indexers). As explained in Section 3.3, we continue to use Naïve Bayes with these class values, but when making a prediction we average the predicted values, weighted by the prediction probability of each one. Due to the small size of our dataset with multiple-indexer judgments we apply the leave-one-out evaluation technique (see Section 5.4).

Table 6 details the professional human indexers' performance, giving for each pair their percentage consistency. The lower part of the table shows the consistency of the automatic indexing systems with each human indexer. The consistency between human indexers varies from 26% to 47%, and averages 39% over the group. The baseline system's overall consistency with human indexers is the same as the lowest consistency achieved by two humans (indexers 3 and 5), and is 13 percentage points lower than the average consistency among humans. KEA++ comes closer to human performance, the nominal-class version falling only 7 percentage points below their overall consistency. The improvement of 2 percentage points obtained by training on nominal classes supports our hypothesis that performance is improved by taking into account the degree of correctness for each keyphrase in the training data. Note that these results are achieved with a very small training set (29 documents): the findings of the previous section indicate that a larger training set of multiply-indexed documents would reap additional improvements.

6.8 Analysis of Non-Matching Terms

The above evaluations only consider whether or not two terms match exactly. Additionally, we analyze how many of the non-matching terms are semantically related by links in the controlled vocabulary. To do so, for each non-matching term we check whether it is broader, narrower or related to any of the keyphrases chosen by other indexers. Table 7 summarizes the number and percentage of matching, non-matching but related, and non-matching unrelated terms, between humans and the KEA++ algorithm

trained on the nominal-class data. The algorithm selects phrases that in 62% of cases exactly match indexers' choices. Almost two thirds of the remainder (25% of the total) are related to at least one of the manually selected phrases by Agrovoc relations, directly or via another term. The remaining 13% of phrases have not been chosen by any of the 6 human indexers and must be considered incorrect. In other words, 78% of phrases are identical or conceptually related to terms selected by professional indexers (50%+28%). These figures are comparable to the results achieved by humans, who agree on 90% of each document's concepts (72%+18%).

Table 8 shows the keyphrases assigned to three sample documents concerning some of today's burning issues. The *Indexer* column shows all terms assigned to each document by at least three indexers, and includes a few other terms as indicated. They are compared to the ten top-ranked phrases selected by KEA++ (trained on the nominal class data): ten is the average number assigned by professional indexers. The terms labeled *Exact* and *Similar* make good sense according to the documents' titles—some of the non-matching ones do too.

Figures 3 and 4 display typical keyphrase sets in a graph form with nodes representing terms and edges their semantic relations. Terms picked by KEA++ are marked by green centers. Circles show the number and identities of indexers who assigned that term. Edges show thesaurus relations: broader/narrower term and related term (the two kinds are not distinguished); they divide the space into disconnected regions. The document of Figure 3 is titled *Home Gardens: Key to Improved Nutritional Well-Being*: all regions covered by keyphrase sets assigned by at least 3 professional indexers are covered by KEA++ keyphrases that are the same or similar to those chosen by the professional indexers. The document of Figure 4 is titled *The Growing Global Obesity Problem*: here, KEA++ failed to cover the topic area of nutritional policies and taxes.

7. SUMMARY

We have presented a new approach to thesaurus-based indexing of documents using machine learning. Its controlled vocabulary is any thesaurus, encoded in a standard way, and the method takes advantage of the

semantic information implicit in the thesaurus. The main advantage over conventional keyphrase extraction is the use of a controlled vocabulary. This helps eliminate the occurrence of meaningless or obviously incorrect phrases, and also yields a dramatic improvement in performance. The main advantage over conventional term assignment, which uses a controlled vocabulary, is that far less training data is needed.

KEA++ is a keyphrase extraction system that embodies the new technique. Comparison with a baseline approach, two other indexing systems, and a group of human indexers, has shown that:

- Automatic controlled-vocabulary indexing outperforms automatic free-text indexing.
- Besides TF×IDF, which is the basis of our baseline model, three additional features corresponding to different aspects of typical keyphrases—namely *importance* (first occurrence), *specificity* (length) and *topic coverage* (node degree)—contributed a noticeable improvement in most of the experiments.
- KEA++'s performance on Spanish and French documents is not comparable to its performance on the English collection, although the extracted keyphrases are usable. We conclude that the results are not indicative due to the anomaly in the data.
- The indexing techniques used by KEA++ are domain independent and the system can be applied to new domains without any modifications at all, so long as their vocabularies are available in the SKOS format.
- A few dozen manually indexed documents are enough to achieve good results. Only a few minutes' training time is needed to learn a model from 50 documents.
- Simultaneous learning from training data provided by several humans for the same documents outperforms learning from a single indexer by smoothing out idiosyncrasies.
- The examples show that most terms identified by KEA++ either match those assigned by human indexers exactly, or are similar to them. However, the system fails to extract some of the terms that

several humans identified as important. The main reason for this failure is that the terms in question never appear in the document, or appear only rarely.

- KEA++'s average indexing consistency with professional indexers is only 7 percentage points lower than their consistency with each other. Since it is known that professional indexers outperform amateurs (Saarti, 2002), it follows that KEA++'s consistency is even closer to that of amateur human indexers.

To further improve the system's performance we will pursue two major lines of research in future. First, we will seek ways of learning the degree of correctness of a keyphrase for domains in which multiple-indexed data which is difficult to obtain, and at the same time investigate domains where such data is freely available (e.g. collaborative-tagging websites). Second, we will work on improving recall, which at present falls significantly below that of text categorization approaches. Rather than using categorization methods, which drastically increase the amount of training data needed, we will focus on further exploration of the controlled vocabulary itself. We believe that this work stands a chance of raising automatic indexing to the same level of performance as human professionals, in terms of its consistency with the judgment of an independent group of indexers.

8. ACKNOWLEDGMENTS

We gratefully acknowledge the UN Food and Agriculture Organization (Gudrun Johannsen, Johannes Keizer and Margarita Sini), the U.S. National Library of Medicine Indexing Initiative (Alan Aronson and Jim Mork) and CDS Invenio Team (Alberto Pepe) for providing us with the experimental data. Finally, we thank the anonymous reviewers for their suggestions for improving this paper. This work is funded by a scholarship from Google.

9. REFERENCES

Aronson, A. R., Bodenreider, O., Chang, H. F. et al. (2000) The NLM indexing initiative. In Proc. of the Annual Fall Symp. of the American Medical Informatics Association, pp. 17–21.

van Assem, M., Malaise, V., Miles, A.J. & Schreiber, G. (2006). A method to convert thesauri to SKOS. In Proc. of the 3rd Annual European Semantic Web Conf., pp. 95–109.

Barker, K., & Cornacchia N. (2000). Using noun phrase heads to extract document keyphrases. In Proc. of the 13th Canadian Conf. on Artificial Intelligence, pp. 40–52.

Begelman, G., Keller, P., & Smadja, F. (2006) Automated tag clustering: Improving search and exploration in the tag space. In Proc. of the Collaborative Web Tagging Workshop, WWW Conf.

David, C., L. Giroux, S. Bertrand-Gastaldy, & D. Lanteigne (1995). Indexing as problem solving: A cognitive approach to consistency. In Forging New Partnerships in Information, Medford, NJ, Information Today. pp. 49–45.

Dumais, S.T., Platt, J., Heckerman, D., & Sahami, M. (1998). Inductive learning algorithms and representations for text categorization. In Proc. of the 7th Int. Conf. on Information and Knowledge Management (CIKM), pp. 148–155.

Fayyad, U.M., & Irani, K.B. (1993). Multi-interval discretization of continuous-valued attributes for classification learning. In Proc. of the 13th Intern. Joint Conf. on Artificial Intelligence, pp. 1022–1027.

Fuhr, N. & Knorz, G. (1984). Retrieval test evaluation of a rule based automatic index (AIR/PHYS). In Proc. of the ACM SIGIR Conf. on research and development in information retrieval, pp. 391–408.

Gay C.W., Kayaalp M. & Aronson A. R. (2005). Semi-automatic indexing of full text biomedical articles. In Proc. Annual Fall Symp. of the American Medical Informatics Association, pp. 271–275.

Golub, K. (2006). Automated subject classification of textual Web pages, based on a controlled vocabulary. New review of hypermedia and multimedia. 12(1), 11–27.

Hooper, R. S. (1965). Indexer consistency tests—Origin, measurements, results and utilization. IBM Washington Systems Center, Bethesda, Maryland. Presented at the 1965 Congress International Federation for Documentation.

Hripcsak G. & Rothschild, A. S. (2005). Agreement, the F-Measure, and reliability in information retrieval. Journal of American Medical Information Association, 12(3), 296–298.

Hulth, A. (2004). Combining machine learning and natural language processing for automatic keyword extraction. Ph. D. thesis, Stockholm University.

Luhn (1959). Keyword in context index for technical literature (KWIC Index), Yorktown Heights, NY: IBM, Report RC 127.

Markó, K., Hahn, U., Schulz, S., Daumke, P., & Nohama, P. (2004). Interlingual indexing across different languages. In Proc. of the Int. Conf. „Recherche d'Information Assistée par Ordinateur“, pp. 100–115.

Medelyan, O. & Witten, I. H. (2006). Thesaurus based automatic keyphrase indexing. In Proc. of the Joint Conference on Digital Libraries, pp. 296–297.

Paice, C., & Black, W. (2003). A three-pronged approach to the extraction of key terms and semantic roles. In Proc. of the Int. Conf. on Recent Advances in NLP (RANLP), pp. 357–363.

Plaunt, C. & Norgard, B. A. (1998). An association based method for automatic indexing with a controlled vocabulary. Journal of the American Society for Information Science, 49 (10), 888–902.

Pouliquen, B., Steinberger, R., & Camelia, I. R. (2003). Automatic annotation of multilingual text collections with a conceptual thesaurus. In Workshop on Ontologies and Information Extraction, at the EUROLAN Conference, pp.19–28.

van Rijsbergen, C.J. (1979). Information Retrieval. Butterworths, London.

Rolling, L. (1981). Indexing consistency, quality and efficiency. Information Processing and Management, 17, 69–76.

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

Saarti, J. (2002). Consistency of subject indexing of novels by public library professionals and patrons. Journal of Documentation, 58, 49–65.

Salton, G. and Buckley, C. (1988) Term-weighting approaches in automatic text retrieval. Information Processing and Management, 24 (5), 513–523.

Sebastiani, F. (2002). Machine learning in automated text categorization. ACM Computing Surveys 34(1), 1–47.

Tiun, S., Abdullah, R. & Kong, T. E. (2001) Automatic topic identification using ontology hierarchy. In Proc. 2nd Int. Conf. on Computational Linguistics and Intelligent Text Processing, pp. 444–453.

Turney, P. (1999). Learning to extract keyphrases from text. Tech. report, Nat. Research Council Canada.

Witten, I.H., & Frank, E. (1999). Data mining: Practical machine learning tools and techniques with Java implementations. Morgan Kaufmann, San Francisco, CA.

Witten, I.H., Paynter, G.W., Frank, E., Gutwin, C., & Nevill-Manning, C.G. (1999) Kea: Practical automatic keyphrase extraction. In Proc. of the ACM Conf. on Digital Libraries, pp. 254–255.

Zunde, P., & Dexter, M.E. (1969). Indexing consistency and quality. American Documentation, 20, 259–267.