Automatic Keyphrase Indexing with a Domain-Specific Thesaurus

Automatische Schlagwort-Indexierung
mit einem domänen-spezifischen Thesaurus

Magisterarbeit
zur
Erlangung der Würde des Magister Artium
der Philologischen, Philosophischen und Wirtschafts- und
Verhaltenswissenschaftlichen Fakultät der
Albert-Ludwigs-Universität
Freiburg i. Br.

vorgelegt von

Olena Medelyan
aus Kertsch, Ukraine

WS 2004/2005

Sprachwissenschaft des Deutschen

Contents

Zusammenfassung

Schlagwortindexierung ist die gängige Form der inhaltlichen Beschreibung von Dokumenten in physikalischen und digitalen Bibliotheken. Schlagwörter und -phrasen dienen zur Organisierung der Dokumente entsprechend ihrer Themengebiete, ermöglichen inhalt-basierte Suche in einem Katalog und sind nützlich zur Darstellung und Navigation in den Suchergebnissen. Zur manuellen Verschlagwortung müssen professionelle Indexierer die Dokumente lesen und verstehen, um danach anhand von Katalogregeln passende Schlagwörter aussuchen. Dies ist eine zeit- und kostspielige Aufgabe.

Zwei Ansätze zur Automatisierung dieses Prozesses wurden bislang entwickelt. Bei der Schlagwort*extraktion* werden die intrinsische Eigenschaften von Wortketten in einem Dokument, wie z.B. deren Häufigkeit und Länge, analysiert, um relevante Schlagwörter zu identifizieren. Bei der Schagwort*zuweisung* werden die Phrasen aus einem kontrollierten Vokabular als Kategorien gesehen, in die Dokumente aufgrund ihres Inhalts automatisch klassifiziert werden. Ein entscheidender Nachteil im ersten Verfahren ist die schlechte Qualität der extrahierten Phrasen: sie sind oft grammatikalisch falsch oder unpassend. Die zweite Methode umgeht dieses Problem, benötigt jedoch ein großes Korpus mit manuell klassifizierten Dokumenten zum Trainieren entsprechender Lernalgorithmen.

In Rahmen dieser Magisterarbeit entwickelte ich ein neues Verfahren, dass die Extraktion und die Zuweisung von Schlagwörtern verbindet. Es profitiert von den Vorteilen der beiden Methoden, vermeidet jedoch deren Schwachpunkte. Ein Thesaurus wird als kontrolliertes Vokabular verwendet. Die semantischen Beziehungen zwischen den Thesauruseinträgen, die in einem Dokument wörtlich vorkommen, sowie die üblichen Eigenschaften der Phrasen werden analysiert, um die endgültige Schlagwortmenge zu bestimmen. Auf diese Weise wird aus einem relativ kleinen Korpus manuell indexierter Dokumenten ein Klassifizierungsmodell mithilfe maschineller Lernmethoden erstellt. Bei einem neuen, bislang nicht verschlagwortetem Dokument werden alle Terme, die sowohl im Dokument als auch im Thesaurus

vorkommen, anhand dieses Modells analysiert, um relevante Terme festzustellen. Die resultierenden Schlagwörter sind wohlgeformt und haben eine direkte Beziehung zum Inhalt des Dokuments.

Das entwickelte System wurde anhand einer Dokumentenkollektion aus der Agrikultur-Domäne getestet. Für die Implementierung wurde der domänen-spezifische Thesaurus Agrovoc verwendet. Die Evaluierung ergab, dass über 50% der automatisch identifizierten Schlagwörter exakt oder konzeptuell denen entsprechen, die von professionellen Indexierern manuell zugewiesen wurden. Der Vergleich zur Konsistenz zwischen den Indexierern zeigte, dass das System im Durchschnitt nur 10-15% weniger konsistent mit den Indexierern ist, als diese es untereinander sind. Die manuelle Analyse der Phrasen zeigt auf, wo die Probleme bei der automatischen Indexierung dieser Art liegen und wie die entwickelten Methoden verfeinert werden können.

Acknowledgements

During the course of my studies though especially in the main phase, this past year, I received immense support from many people, and I am grateful to have the opportunity to thank all of them in this part of my thesis.

First of all I would like to thank Prof. Ian H. Witten, my supervisor at the University of Waikato. His invitation to write my master thesis at his research group in New Zealand was the beginning of the very valuable and fruitful time. From the day one, Ian made me feel comfortable and welcome in a foreign country and gave me tremendous support throughout writing this thesis. He has been an encouraging and motivating supervisor, who has taught me the essentials of doing research and academic writing. He gave me directions on how to organize and plan a project and how to communicate with other people so that everyone can reap the most benefits from it. Thank you, Ian, for the priceless experience of working with you and the enjoyable after hours with you and your family.

This thesis wouldn't have been written without the support of Prof. Dr. Günter Kochendörfer. His willingness to supervise me while I was on the other hemisphere and to allow me to write on an interdisciplinary topic deserves my deep respect and appreciation. Thank you very much for your flexibility.

Prof. Dr. Udo Hahn, who supported me in my studies from the very beginning, deserves a special mentioning. The majority of the theoretical knowledge I applied in this thesis I learned in his inspiring courses. Prof. Hahn also supported me financially by offering student assistance work at his research group at the University of Freiburg. I am very grateful for your help in facilitating my studies in Germany and all the valuable knowledge in Computational Linguistics I received from you.

I would also like to thank everyone at the exorbyte Ltd in Konstanz, Germany, where I gained useful practical skills and started investigating the subject of this thesis. I also want to show appreciation to Prof. Dr. Günthner from the University of Munich for valuable discus-

sions and the prompt support in applying for this exchange program.

Thanks to the Department for Computer Science at the University of Waikato for offering me a study award and conference funding. I learned to love the excellent research environment and the friendliness of my colleagues here.

An important part of my studies in Freiburg and Hamilton was the student life beside the university. I was lucky to meet the most exciting people in both towns and to share unforgettable experiences with all of them. Thanks for the great time, your friendship and backing in all the difficult moments.

A special thank you deserves Michael Poprat, who accompanied my studies as a great friend and partner. He was always there for me, when I needed help, and made me feel safe and happy. His support through the last years is beyond all words. I am glad that I met you and very thankful for everything you have done for me.

Finally, I would like to thank my parents, who instilled in me a thirst for knowledge and foreign languages from the young age, which all turned out to be of a great benefit for my studies. Thank you for the permanent encouragement to do great things, teaching me to be confident and stay optimistic and for making my education in Germany possible.

I am fully aware that without all this support I would not have been able to reach this point, though only few months are left until I receive my degree from the University of Freiburg, in a fascinating interdisciplinary field that combines Linguistics and Computer Science in an exciting and challenging way.

Chapter 1

Introduction

Searching for information has become a common activity for everyone who uses computers in everyday life. While the Internet offers an unfathomable source for satisfying most inquiries, finding the documents that contain necessary information is still a challenging task, often compared figuratively to the problem of finding a needle in a haystack. Formulating correct and accurate search terms is the key problem in an electronic search. In online library catalogues users are supported by controlled vocabularies that set forth potential search phrases. The aim of these vocabularies is to organize documents into groups labeled by keywords or keyphrases that correspond to their salient concepts. These keywords are selected manually by professional human indexers and provide quick content-based access to library holdings.

Because of the importance of keyphrase indexing and the impossibility of manually indexing the constantly growing panoply of electronic documents, methods for assigning keyphrases automatically are in great demand. Existing approaches for automatic keyphrase extraction and assignment do not achieve sufficiently high indexing performance to be of practical use. This is obviously caused by their restriction to syntactic and statistical properties of extracted phrases. I propose a new method that enhances keyphrase extraction by semantic information on terms and phrases, contained in a domain-specific thesaurus. Comparing the results with manually-assigned keyphrase sets reveals the utility of the approach.

1.1 Keywords, keyphrases and keyphrase sets

In library and information science terminology, a *keyword* is defined as a "word which succinctly and accurately describes the subject or an aspect of the subject discussed in a document" (Feather and Sturges 1996, 240). For a similar purpose, librarians also use *subject headings*:

terms or phrases that describe the content of a document and, additionally, group and organize library holdings. Documents that deal with the same subject are grouped together and can be accessed quickly by selecting a particular subject heading. Like subject headings keywords and keyphrases can be selected from a standardized set of descriptors called a *controlled vocabulary*. Another way of assigning keywords is to extract them from the document's body or, as frequently done by librarians, solely from its title.[1] Compared to full-text indexing, where all terms that appear in the document are stored in a database, keyword or keyphrase indexing require notably less storage and provide a concise overview of thematic areas covered by the document collection as a whole and each document in particular.

A *keyphrase* implies a multi-word lexeme (e.g. *computer science*), whereas a keyword is a single word term (e.g. *computer*). Humans seem to prefer keyphrases to keywords. In the training collection used in this project about 68% of humanly assigned keyphrases contain more than one word (see Section 3.1 for further details). Hulth (2004) reports that over 86% of the keyphrases in her collection are multi-word lexemes. The number varies depending on the purpose for which keyphrases are intended: Shorter phrases can organize small collections of documents into larger thematic groups, whereas longer ones describe the document's content more precisely. Phrases longer than one word are potentially more useful for keyphrase search. Search engine query logs show that people tend to use longer phrases in order to describe the subject they are looking for (e.g. Silverstein et al. (1998) estimate the average query length in an Altavista query log as 2.4 terms). Therefore this thesis focuses on determining keyphrases rather than keywords, but refers to both terms by the word *keyphrase*.

In this work the main emphasis is to identify keyphrases as descriptors of the document's content. The purpose of the thesis is to detect keyphrases that are terms or phrases from a controlled vocabulary that may or may not appear in the document verbatim. The intention is to determine a *keyphrase set* consisting of several keyphrases. Each keyphrase should represent one of the main topics in a given document, and the keyphrase set should cover the thematic scope of the entire document. In other words, the keyphrase set should summarize what the document is about.

[1] Building keyphrases by reformulating document's title corresponds to three basic keyphrase indexing strategies KWIC, KWOC and KWAC (key word in, out and alongside of context). E.g. a title "Biology of Animals and Plants" would be represented with KWAC technique as three index entries: "Biology - of Animals and Plants", "Animals - and Plants. Biology", "Plants. - Biology of Animals and". Such indexing is a simple and fast way to create library catalogs, but it presumes that titles consist of content words (James 1996).

1.2 Motivation of the research

Keyphrases are useful metadata in physical and digital libraries and help to organize the holdings based on their content. Their most important advantage over other metadata such as title, author name, or date of publication, is that they support thematic access to the documents.[2] As search engines became everyday tools for all who work with the Internet (or corporate intranets), people learn to express their information needs in terms of search queries. Consequently, they prefer to use this kind of search in conventional catalogs as well. User surveys on online catalog systems (e.g. Engl et al. (1997)) show that about 30% of search queries are keyphrases, and users consider keyphrase search as important as title and author search.

Another advantage of keyphrases is their multi-functionality. A series of recent publications explored how keyphrases and keyphrase sets can be used for various natural language processing applications such as text clustering and classification (Jones and Mahoui 2000), content-based retrieval and topic search (Arampatzis et al. 1998), automatic text summarization (Barker and Cornacchia 2000), thesaurus construction (Paynter et al. 2000), representing search results (Hulth 2004), and navigation (Gutwin et al. 1998). The effectiveness and utility of these approaches depend on the quality of the keyphrases that have been assigned. Therefore, it is important to have accurately selected descriptors, ideally provided by human experts.

Depending on the situation, keyphrases are sometimes assigned by the document's authors, e.g. for articles in conference proceedings. In cataloguing, documents are usually indexed by professional indexers, partly because most documents lack author-defined keyphrases, and most importantly because this is the only way to achieve indexing consistency over the entire library holdings. Assigning high-quality keyphrases by humans is an expensive and time-consuming task. Professional human indexers must scan each document and select appropriate keyphrases from the library's own vocabulary according to defined cataloguing rules. Nowadays libraries spend more money on employees than for new books or journal subscriptions (Hilberer 2003). Given the constant growth of the Internet and the popularity of digital libraries, manual keyphrase indexing is completely impractical. Therefore accurate and effective methods for automatic keyphrase indexing are in great demand.

Existing approaches to keyphrase extraction and assignment are based on simple statistical algorithms that analyze the frequencies of character strings extracted from a document collection. Some authors report that they could improve performance by enhancing their system with

[2] Titles, for example, often do not refer the actual topic of the book: "Because he could", by Dick Morris and Eileen McGann (2004) – A bestseller about the life of Bill Clinton.

linguistic knowledge (Hulth (2004); Paice and Black (2003)). At the same time, automatic indexing methods rarely make use of any semantic information. The meaning of the terms in the documents, and the relations between them, are generally ignored. In this thesis, an automatic keyphrase indexing method is designed that analyzes both the statistical behavior of phrases and their semantic characteristics encoded in a thesaurus.

1.3 Thesis statement

Existing approaches to automatic keyphrase indexing are realized either as an extraction process or as an assignment process. In keyphrase *extraction* terms and phrases that appear in the document verbatim are analyzed to select the most representative ones. In *assignment* documents are instead classified into a pre-defined number of categories corresponding to keyphrases (cf. Chapter 2). In this thesis I extend the state-of-the-art keyphrase extraction algorithm KEA[3] into a new version KEA++ that combines these two approaches into a single process, where terms and phrases are extracted from documents but need to be present in a controlled vocabulary in the first place. The indexing process is extended by analyzing semantic information about the document's terms, i.e. their relations among each other and to other terms in the thesaurus.

Another important issue studied in this thesis is measuring the performance of keyphrase indexing. Previous studies in this area evaluated results based on exact matching of phrases assigned from two sources, one human and the other automatic. Semantic similarity of keyphrases was basically ignored. This thesis explores semantic relations between keyphrases as essential constituents in measuring indexing quality between human assessors and the algorithm.

The striking deficiencies in the current manual and automatic keyphrase indexing, as well as the possibilities for improvement mentioned above, suggest the following hypotheses:

1. Keyphrase extraction can be improved by using controlled vocabulary.
2. Using semantic relations between terms enhances indexing performance.
3. Automatic keyphrase indexing can achieve the same degree of indexing consistency as professional indexers.

Although it is apparent that computer systems will never be able to compete with human performance in tasks that require understanding of the meaning of words, the possibility of

[3]The Keyphrase Extraction Algorithm KEA (http://www.nzdl.org/kea/) is developed at the University of Waikato. See Section 2.1.1 on details on this algorithm.

enhancing automatic keyphrase indexing by semantic information suggests that it may finally become utilizable in real-world scenarios. The goal of this thesis is to prove this empirically. Through the implementation and evaluation of KEA++, we determine whether or not the above three hypotheses are valid. The following subsections discuss them in detail.

1.3.1 Controlled keyphrase indexing

Keyphrase extraction implies that all keyphrases appear in the document verbatim. Of course, human authors or indexers are not subject to this constraint. The use of free text rather than a pre-defined set of index terms has the potential disadvantage that keyphrases are neither normalized nor restricted to a particular vocabulary. This reduces consistency compared to keyphrase assignment, even among professional indexers (Leininger 2000). The probability of achieving higher levels of consistency grows as the size of the controlled vocabulary used for indexing decreases. Therefore, automatic controlled keyphrase indexing has a greater chance of being solved in a way that competes with human performance.

Beside the restrictive function of controlled vocabularies they also usually provide a hierarchical structure that can be explored for keyphrase indexing in different ways. The domain specific thesaurus Agrovoc,[4] utilized for indexing of the document repository of the Food and Agriculture Organization of the United Nation (FAO), is probably the best example for this. KEA++ is not restricted to assigning only extracted phrases, but they must appear in Agrovoc. Additionally, KEA++ explores the semantic relations integrated in Agrovoc's structure, which is the basis for the next hypothesis.

1.3.2 Semantic relations

When humans assign keyphrases, they need in the first place to understand what the document is about. Understanding the document's content requires understanding the main concepts in it, and how they relate to each other. Both the text and the knowledge available to the reader can be expressed as a semantic network, in which terms are connected to each other by semantic relations based on the similarity of their meaning and usage.

Agrovoc has such a network similar structure, where terms and phrases are connected by semantic relations such as hierarchical *IS-A* links (between a *broader* and a *narrower* term) and by associative links, if terms are related in any other way to each other. KEA++ extends the number of concepts that are associated with a document due to their verbatim appearance

[4]The Agrovoc thesaurus (http://www.fao.org/agrovoc/) contains over 16,000 descriptors for indexing of agricultural documents. The thesaurus is described in detail in Section 3.2

by terms that are connected to them by these semantic relations. This increases the statistical probability of finding correct keyphrases for indexing. Another enhancement for indexing is to consider the characteristics of semantic relations between terms, i.e. the number of other terms in the document that are connected by different relations to a candidate keyphrase.

1.3.3 Towards improved indexing consistency

Inter-indexer consistency is computed as the average overlap between keyphrase sets assigned by two or more professional indexers to a set of documents, and reflects the potential quality of index terms in a library catalog. Experiments on indexing consistency reveal that humans seldom assign exactly the same keyphrase sets, because their understanding of the documents and their background knowledge differ. Therefore, evaluating the performance of automatic keyphrase indexing algorithms against keyphrases assigned by a single person may be misleading. Moreover, it is important to know how the consistency of automatic indexing compares to that of human indexing. In this project, a document set was indexed by six professional indexers and by KEA++. A comparison of the average consistency between humans and between KEA++ and all indexers will show whether the algorithm can achieve an applicable level of indexing quality. Missing phrases, or phrases that do not match exactly, are unavoidable even in keyphrase sets assigned by professional indexers. Thus, the similarity of keyphrases, estimated through semantic links in the Agrovoc thesaurus, is an important issue investigated in this thesis and will be included in measurements of consistency.

1.4 Basic concepts

This thesis addresses an interdisciplinary topic and requires understanding of basic concepts in both linguistics and computer science. This section presents briefly the main terms that will be mentioned in following chapters.

Term Conflation One of the main problems in automatic text processing is term variation. While humans easily identify morphologically related words as the same concept, an algorithm needs a pre-processing operation that conflates these words to the same surface form. Common variations are inflectional affixes, alternative spellings, spelling errors, and abbreviation forms. While the two later variations are not easy to identify, because they require equivalence dictionaries, simple conflation approaches such as case folding and stemming are widely used.

In case folding, all letters are transformed into either low or upper case. Lovins (1968, 22) defines stemming as “a procedure to reduce all words with the same stem to a common form,

usually by stripping each word of its derivational and inflectional suffixes". Algorithms that iterate from left to right and remove affixes till the word root can be separated use lists with possible inflectional endings. More sophisticated approaches are required for languages with agglutinative features, where words are formed by joining morphemes together (e.g. decompounding algorithms for Germanic languages).

Two different words can be also conflated due to their semantic similarity, for example if they are synonyms. Lexical databases and thesauri are usually required for establishing semantic relations between terms.

Vector Space Modeling To represent and process documents in a vector space three stages are required. In the first stage single terms in a document are identified, e.g. by dividing it into tokens separated by space symbols. It is common to ignore all tokens that are stopwords – very frequent and not content bearing words, e.g. articles, propositions, adverbs, etc. Each document in the collection can be represented then as a vector, where each element indicates presence or absence of a word in a document. This is the so called "bag of words" representation.

The second stage is the weighting of the terms according to their appearance frequency in the entire collection and in the given document. Salton and McGill (1983) showed different weighting schemes, and in particular the popular $TF \times IDF$ technique. *TF* stands for *term frequency* and is computed relatively to the total number of words in the document. *IDF* is the *inverse document frequency*, computed as the number of documents that contain a given term divided by the total number of documents in the collection. It is common to take log_2 of the document frequency to normalize the overall term weight.

The last stage includes computing similarity between vectors in the resulting space to build clusters of similar documents or to rank documents according to their similarity to the user's query, also represented as vector. The most common similarity measure is the cosine coefficient, which measures the angle between two vectors D_a and D_b according to the following formula:

$$Sim(D_a, D_b) = \frac{\sum_{i=1}^{n} w_{ai} * w_{bi}}{\sqrt{\sum_{i=1}^{T} (w_{ai})^2 * \sum_{i=1}^{T} (w_{bi})^2}},$$

where w_{ai} and w_{bi} are the weights of the term t_i in the documents D_a and D_b respectively. This measure returns a value between 0 (no similarity) and 1 (identical documents).

Learning Schemes When computer scientists claim that their system learns, it means that it changes its structure or behavior in such manner that it improves its future performance (Witten and Frank 1999). These changes are based on the regularities in the input data and a learning

scheme used to interpret them. In the first place, the characteristics of the input data are analyzed statistically. Then, this information is used to create a learning model that describes the solution of the problem according to a pre-defined function called learning scheme. Depending on the problem type different learning schemes can be applied. For the task described in this thesis the *Naïve Bayes* scheme is suitable. It is based on conditional probability rule defined as following: Given a hypothesis H and an event E, the probability that H holds given this event is:

$$Pr[H|E] = \frac{(Pr[E|H]Pr[H])}{Pr[E]},$$

where $Pr[A]$ is the probability of event A and $Pr[A|B]$ is the probability of A given another event B. This is a simple method, called naïve because all events are assumed to be independent of one another. However, in practice it was proven to be very effective (Witten and Frank 1999).

Evaluation of training performance To evaluate the performance of the learning scheme the experimental data (e.g. a document collection) needs to be split into two sets, where the first one serves for training and the second for testing purposes. The training set is analyzed to adjust the computational model to the characteristics of the data. The test set remains untouched and serves solely for testing purposes. Splitting is usually made randomly, but especially when the collection is small, the results depend on what document turn to appear in each set. To keep the evaluation unbiased towards data's properties, an extension of this strategy called *10-fold cross-validation* is useful (Witten and Frank 1999). Here, the document collection is divided randomly into ten sets. Ten separate evaluational runs are performed with training on nine sets and testing on the remaining one sets, so that every set is used nine times for training and once for testing. The results are taken as an average over all runs. Further improvement is possible, where the 10-fold cross-validation is repeated ten times on ten random split-ups, with 100 experimental runs in total. Due to practical reasons, only a single 10-fold cross-validation is applied for the automatic evaluation in this thesis.

1.5 Outline

This introductory part of the thesis presented the research problem and its motivation. The remainder of this work will investigate the task of keyphrase indexing in detail and determine whether the hypotheses formulated in the previous section can be substantiated. Chapter 2 presents previous approaches in the field of automatic keyphrase extraction and assignment. After a comprehensive overview of the advantages and limitations of different techniques, this

chapter discusses the indexing task as performed by humans. Because understanding the document's content is the main phase in human indexing, we also introduce some approaches to the study of human text comprehension.

Chapter 3 gives an overview of experimental data used in this project and describes the evaluation strategy. A deep analysis of the Agrovoc thesaurus (Section 3.2) and keyphrase sets assigned by professional indexers (Section 3.5) provides valuable insights into semantic relations between terms that are integrated in the experiments described here. The problem of estimating semantic similarity between keyphrase sets, based on the individual relatedness of phrases in one set to phrases in another, takes up an important part of the chapter.

The detailed explanation of the keyphrase extraction algorithm KEA++ in Chapter 4 starts with the description of term extraction and conflation. It demonstrates the main idea behind merging keyphrase extraction and assignment into a single keyphrase indexing procedure, by mapping extracted terms to entries in a controlled vocabulary. Sections 4.2 and 4.3 describe further extensions of the state-of-the-art approach such as candidate selection with respect to semantic relations in the thesaurus and new features implemented in KEA++. The chapter also explains the learning scheme used to build the model from the training collection and its application to new documents, where keyphrases need to be extracted automatically.

Chapter 5 presents the evaluation of the automatic keyphrase indexing method according to the metrics elaborated in Chapter 3. The candidate extraction technique and filtering method are evaluated automatically on the main document collection. These results demonstrate the improvements achieved by controlled indexing strategy and the new developed features. To compute the inter-indexer consistency KEA++ is also tested against data provided by six professional indexers. These results are considered as the main indicator of KEA++'s performance. Manual analysis of the results reveals sources for errors produced by the algorithm, some of which can be avoided with small modifications. The last section repeats the evaluation after these final changes and demonstrates some typical keyphrases extracted with KEA++.

The final Chapter 6 concludes the investigation of the thesis. Based on achieved evaluational results Section 6.1 discusses, whether the hypotheses stated in the first chapter could be proven or not. Section 6.2 provides information on how KEA++ can extract keyphrases from documents in other languages and domains and describes further areas for its application. The thesis ends with the discussion of limitations in current approaches for automatic keyphrase indexing.

Chapter 2

Related work

Due to the increasing popularity of electronically available documents (company databases, digital libraries, and webpages) efficient access methods became necessary. To provide content-based search text processing tools for automatic indexing, terminology extraction, document classification, keyphrase assignment and extraction are required. There is a large variety of methods developed for these tasks. While some of these problems are well solved (e.g. Jacquemin and Tzoukermann (1999) report that their system archives recall and precision near 100% for indexing on terms and their variants within a restricted domain), keyphrase indexing is still a challenging problem: Maximal reported F-measure value for automatic evaluation against author's phrases[1] is 45% (Hulth 2004). This can be explained by the difficulty of the task and the problematic evaluation of indexing algorithms, since even professional human indexers often disagree on what phrases are significant for a given document (See Sections 3.3.2 and 3.5 for further discussion).

This chapter begins with an overview of current achievements in keyphrase extraction, assignment and related areas. The comparison of these two strategies for keyphrase indexing reveals the reasons for sparse efforts in both approaches. The chapter ends with background on human indexing and related work in the area of cognitive linguistics, which sheds some light on the indexing specific cognitive processes and motivates the keyphrase indexing method proposed in this work.

2.1 Keyphrase extraction

Keyphrase extraction is usually realized in two stages:

[1] See Chapter 3 for details on evaluation strategies.

1. **Selecting candidates.** All content words (non-stopwords) and phrases (concatenation of content words or combined with stopwords) are extracted from the document. Alternatively, a part of speech (PoS) tagger and a shallow parser are used to identify noun phrases (NPs).

2. **Filtering.** Analysis of selected candidates and filtering heuristics are applied to determine keyphrases among them.

Filtering methods can be classified into machine learning and symbolic approaches according to the techniques they use. While machine learning methods build a statistical model from the training data (cf. Section 1.4), in symbolic methods researcher deduce best possible filtering heuristics based on manual analysis of documents and their keyphrases.

2.1.1 Machine learning methods

Keyphrase extraction can be seen as supervised learning from examples. The main problem is to define informative *features* that can be determined for each phrase and used to distinguish keyphrases and non-keyphrases among the candidate terms. A learning algorithm needs two sets of documents with manually assigned keyphrase sets, where the first is used for training, i.e. to create *classification model*, and the second one for evaluation of this model. All phrases in the documents of the training set can be seen as positive and negative examples. The training scheme learns the model by analyzing feature values for each example. The performance of this model is then evaluated on unseen documents from the test set.

KEA The research group for machine learning at the University of Waikato developed the Keyphrase Extraction Algorithm KEA,[2] based on robust and simple methods (Frank et al. (1999), Witten et al. (1999)). In the first stage of keyphrase extraction KEA determines textual sequences defined by orthographical boundaries such as punctuation marks, numbers, new lines and splits these sequences into tokens. All n-grams, i.e. single words or concatenations of two and more tokens, that do not begin or end with a stopword are candidate keyphrases. KEA stems each candidate with the iterated Lovins (1968) stemmer, but keeps the most frequent full version of a phrase for the output. In the filtering stage KEA computes two features for each candidate: the $TF \times IDF$ measure (a phrase's frequency in a document compared to its frequency in the document collection, (Salton and McGill 1983)) and the distance of the phrase's first occurrence in the document from its beginning. A *Naïve Bayes* (Domingos and

[2] Further information about KEA and the source code is available under http://www.nzdl.org/kea/.

Pazzani 1997) learning scheme creates training data consisting of two sets of weights: for author's keyphrases and all other phrases appearing in the document. In the filtering stage the overall probability for each candidate being a keyphrase is calculated based on this data. The candidate phrases are ranked according to the computed probability, and the r top ranked phrases are included into the resulting keyphrase set, where r is the number of phrases the user requested.

GenEx P. Turney from the National Research Council Canada developed a hybrid genetic algorithm for keyphrase extraction GenEx, consisting of two components: Genitor and Extractor (Turney 1999). Extractor combines a set of symbolic heuristics to create a ranked list of keyphrases. Turney's candidates are phrases consisting of up to three content words. The candidates are stemmed by truncation: cutting off all terms to the length of 5.[3] To filter out keyphrases among the candidates, each of them is scored as its frequency multiplied by its position in text. The scores of the candidates with more than one word are additionally boosted, because they have usually lower scores than single word terms. After removing duplicates and selecting the most frequent full form for each stemmed phrase, Extractor presents top-ranked phrases as output. Extractor has 12 numeric parameters and flags (e.g. boosting factor for phrases with two stems or parameter for number of resulting keyphrases). The steady-state genetic algorithm Genitor[4] is applied to determine the best parameter settings from the training data.

A. Hulth presented in her PhD thesis (Hulth 2004) another machine learning technique with additional use of NLP tools. She compared different methods to extract candidate words and phrases: NP chunking, PoS pattern matching and, finally, the trivial n-gram extraction. Matching of candidates against the manually assigned keyphrases proved that both linguistic oriented approaches produce more correct phrases than n-grams. NP-approach outperformed other candidate extraction methods additionally by having a smaller amount of incorrect terms. For filtering Hulth uses four features: term frequency, inverse document frequency (unlike KEA, not combined as $TF \times IDF$ measure), position of the first occurrence and the PoS-tag (Hulth's analysis of her experimental data has shown that certain PoS-patterns are more likely to denote

[3] Turney (1999) argues that stemming by truncation is much faster than Porter or Lovins stemmer. But such an aggressive stemming has lots of disadvantages: words with different meaning are stemmed to the same string (e.g. *contact* and *container*), allomorphs are disregarded (*moderate* and *modest* are truncated to different stems) and all words shorter than 5 characters are not stemmed at all (e.g. *terms* and *term*).

[4] In genetic algorithms learning is considered as a competition among a population of possible problem solutions. For each candidate in the population the algorithm computes a "fitness" function representing its contribution for the next generation of solutions. An evolutionary process of selection leads finally to the most functional solution.

keyphrases). A combination of several prediction models, applied on NP candidates (after removing the determiners at the initial position) yielded best results.

KEA is the simplest keyphrase extraction approach among these systems. GenEx is based on more complex filtering heuristics, but it does not outperform KEA (Frank et al. 1999). Although no comparison of all three systems on same document collection exists, Hulth's evaluation results are significantly higher than those reported for KEA and GenEx. She demonstrates that the performance of her algorithm was improved after using linguistic based techniques for candidate selection and classification. Hulth's observations are a good motivation to explore further NLP techniques for keyphrase extraction and assignment.

2.1.2 Symbolic methods

Barker and Cornacchia (2000) developed a tool they call *B&C* that does not incorporate any machine learning techniques. To extract candidates they use a simple dictionary lookup to assign basic PoS-tags and a shallow parser to identify all nouns with zero or more premodifying adjectives and nouns. In the filtering step, Barker and Cornacchia compute the frequency of the head noun for each candidate phrase and keep all phrases with N top ranked heads. For each phrase they compute its frequency times its length. The top K highest scoring phrases are keyphrases for the given document. N and K are thresholds that are set by the user. Evaluation experiments involving human judges have shown that this simple approach performs as well as Turney's Extractor (cf. Subsection 2.1.1).

Paice and Black (2003) extract terms from documents that are specific for a particular domain and can be seen as keyphrases. To archive a higher conflation rate of candidate phrases they transform each extracted n-gram to a *pseudo-phrase* in three basic steps: removing all stopwords from the n-gram, stemming the content terms and sorting them into alphabetical order. This matches similar phrases such as *algorithm efficiency*, *efficiency of algorithms*, *the algorithm's efficiency*, *an efficient algorithm* and even *the algorithm is very efficient* to the same *pseudo-phrase algorithm effici*. Original forms of each pseudo-phrase are stored to include into the final set the most frequent one. This is a more sophisticated conflation method than simple stemming and boosts the overall score of a phrase group, based on the morphological similarity of its members. Paice and Black (2003) score each pseudo-phrase by using formula $score = W * (F - 1) * N^2$, where W is the sum of the weights of all words in the pseudo-phrase, F is the frequency of the phrase in the document, and N is the length of the phrase in words (with maximum value 4). All candidate phrases are sorted by their scores.

In a final step, they apply a pattern-based technique to establish semantic roles and relations among remaining phrases. They concentrate their attention on three main roles (INFLUENCE, OBJECT, and PROPERTY) covered by domain-independent patterns such as effect of INFLUENCE on PROPERTY of OBJECT. Phrases which are not covered by any of the patterns are removed. The remaining phrases are presented with information on the roles that have been detected between them. This approach is an interesting symbiosis between keyphrase and information extraction. The authors do not provide any evaluation of their methods, but they present some striking examples.

Keyphrase extraction methods and especially the last presented one are highly related to the area of automatic terminology extraction. Such approaches are advantageous if a database with all terms in a specific field is required. This data is important for various tasks (e.g. content-based indexing, text classification) and is normally collected manually by human experts in a given field. The main difference to keyphrase extraction is that terms are not extracted from individual documents, but from the entire collection.

2.1.3 Terminology extraction

Terminology extraction and keyphrase extraction share the initial problem of term normalization, which means conflation of morphologically, syntactically or even semantically related terms into the same group (cf. Section 1.4.

FASTR is one of the most prominent examples in this field (Bourigault and Jacquemin (1999), Jacquemin and Tzoukermann (1999), Jacquemin and Bourigault (2003)). FASTR uses a generator of morphologically related terms and a shallow transformational parser with a number of sophisticated conflation rules for terminological indexing of French and English texts. Term extraction starts with statistical PoS tagging. After that FASTR applies derivational analysis on each term in the document collection. This analysis is based on inflectional morphological rules and generates as many lexical forms for a given term as possible. For example, for the French stem *mortel* the following related terms are derived:

[mortel] $\rightarrow$ [im-[mortel]$_A$]$_A$	(immortal)
$\rightarrow$ [[im-[mortel]$_A$]$_A$ -iser]$_V$	(immortalize)
$\rightarrow$ dés- [[im-[mortel]$_A$]$_A$ -iser]$_V$]$_V$	(unimmortalize)
$\rightarrow \ldots$	

Overgenerated forms that either do not exist or are unusual (e.g. *unimmortalize*) are removed afterward by matching them against the dictionary and all terms that appear in the cor-

pus. The remaining forms, grouped into derivational classes, support the morpho-syntactical conflation of terms and phrases. While syntactical variants are identified by coordination or substitution rules (e.g. *high tension* is a variant of *high or medium tension*), morpho-syntactic variants imply additional derivational changes of their components. The derivational equivalence is estimated by using prior determined classes. The following meta-rule[5] identifies variants of the same concept N_1 and N_5:

$$Metarule\ AtoN(N_1 \rightarrow N_2\ A_3) \equiv N_5 \rightarrow N_2\ (A^?\ P\ D^?\ A^?)\ N_4$$
$$< N_4\ deriv\ reference > = < A_3\ reference >$$

For example, it conflates phrases such as *production industrielle* (industrial production) and *production actuelle de l'industrie* (current production of the industry), where the first phrase corresponds to the pattern $N_2\ A_3$ and the second to $N_2\ A\ P\ D\ N_4$. The logical constraint in the second line of the rule above says that it can be applied only if A_3 and N_4 are in the same derivational class, which is the case for *industrielle* and *industrie*.

Conflating semantic variants is an additional constraint for every type of phrase in the document collection. Two phrases can be conflated if their components are located in the same synset of the lexical database WordNet.[6] E.g., the following rule matches phrases such as *rigid lens* and *hard lens* as semantic variants:

$$Metarule\ SemArg(N_1 \rightarrow A_2\ N_3) \equiv N_1 \rightarrow A_4\ N_3$$
$$< A_2 syn > = < A_4 syn >$$

FASTR involves a variety of NLP-tools from a finite-state morphological processor over a parser for extraction of terminological transformations to a lexical database. The high amount of manual work makes this approach quite complex and also expensive. But the results show that the efforts are worthwhile. The authors report that their indexing method has a very high precision (97.7%) and increases the recall rate from 72.4% (for simple indexing) to 93,4%, after term conflation (Jacquemin and Tzoukermann 1999).

MOGA is a simple machine learning based system that was proposed by Wu and Agogino (2004) for the same task. They focused attention not on term conflation but on distinguishing terms specific for a given field from generic ones. For this purpose they use a measure called *dispersion* computed for each candidate noun phrase as $M = \frac{N}{E}$, where N the exact number of units that contain the phrase and E is their expected number. E is computed with a formula that

[5]The word class abbreviations are the following: A for adjective, N for noun, P for preposition and D for article. The sign ? in the meta-rule denotes an optional element.

[6]See Section 3.4

takes into account the total number of all textual units and the total occurrence of the phrase in the entire document collection. The lower is the dispersion M, the higher is the probability that this term is content-bearing.

The authors use a generic algorithm to determine the best cutoff value of dispersion. The resulting terminology set was evaluated by six human experts in the given specific field (engineering design), who considered over 90% of extracted phrases as related to the domain in question. They also cover 80% of the author's phrases assigned to this document collection manually. Although the purpose of this approach differs from the present project, it would be interesting to consider the dispersion as a feature. Unfortunately the definition of E is unclear and the authors also do not clarify their definition of textual units: these could be either paragraphs or sentences in the documents.

2.2 Keyphrase assignment

Keyphrase assignment utilizes keyphrase indexing in an entirely different way. In contrast to keyphrase extraction, it uses a predefined set of keyphrases, a so-called controlled vocabulary. The characteristics of the documents themselves, rather than the individual phrases in them, are analyzed to find appropriate keyphrases from this vocabulary. A keyphrase may or may not appear in the document verbatim. The task of keyphrase assignment is better known as text classification or text categorization, defined as "labelling natural language texts with thematic categories from a predefined set" (Sebastiani 2002, 1). Keyphrase indexing with a controlled vocabulary is referred to as *keyphrase assignment* or *text classification*, depending on the context.

Scientists have been developing methods for automatic text classification for around fifty years. Until the late '80s knowledge-engineering experts manually created logical classification rules that were then automatically applied to electronic texts. Since the early '90s, this task has been explored mainly in the context of machine learning. Different inductive learning schemes have been used to analyze the characteristics of manually classified documents and build the classifier automatically.

A classifier is a function that maps a *document vector* to a confidence value that a document belongs to a given class. This value is a number between 0 and 1. A threshold is used to decide whether a particular document can be assigned to a given category or not. A classifier for the class *interest* could be thus

if (interest AND rate) OR (quarterly), then confidence(interest) = 0.9

The process of building classifiers starts with representation of the document in a vector space (cf. Section 1.4). Because of the high number of index terms in a document collection, the resulting vector space is high dimensional and requires a lot of computational resources. The number of dimensions can be reduced by keeping only those terms that are especially useful for the classification process. The usefulness value can be determined computationally with a specified formula (see Sebastiani (2002) for details).

Dumais et al. (1998) compared the effectiveness of classification when using more sophisticated techniques such as considering multi-word phrases as single index terms (e.g. *new york*) or using shallow or deep parsing. None of these linguistically driven methods could improve the accuracy of classifying functions. Lewis (1992), who had similar experience, explained this by low statistical quality of the linguistic indexing methods.

Different learning methods have been applied for text classification. Dumais et al. (1998) compared all main types of classifiers 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 highest average precision of 0.87 for classifying around 3,300 documents into 90 categories. This technique has the advantage that it can handle more dimensions than other methods, and no parameters need to be set. Another technique is to use the so-called *classifier committees*, where the majority vote among different classifier makes the decision. According to Sebastiani (2002), who summarized the performance of different techniques, SVM and classifier committees outperform all other methods.

These experiments were made on a widely-used collection of Reuter's news articles classified under categories related to economics. The number classification labels is 135. This scenario is far removed from a real-world keyphrase assignment problem, where vocabularies comprise thousands of descriptors. Some of the authors use for their experiments the OHSUMED collection, which is a part of a MEDLINE document repository (Joachims (1998), Ruiz and Srinivasan (1999), Markó et al. (2003)). Every document in this collection has one or more manually assigned descriptors from the MeSH (Medical Subject Headings) vocabulary, that has in total over 20,500 main headings (descriptors) and over 120,000 synonymic non-descriptors. Joachims (1998) and Ruiz and Srinivasan (1999) use only a small fraction of this vocabulary of around 23 and 120 descriptors respectively.

Markó et al. (2003) considers the entire vocabulary for MeSH indexing. Their system combines a sophisticated approach to orthographical, morphological and semantic term normalization with a hybrid weighting technique. First, they map every word in the document collection

and in the indexing vocabulary onto one or several *subwords* encoded in the lexicon constituing the most important part of the MorphoSaurus system.[7] The mapping includes an orthographical normalization of equivalent spellings,[8] morphological segmentation (*leukemia* becomes {*leuk, em, ia*}) and semantic mapping (equivalent forms are grouped in MorphoSaurus under the same ID). Second, they retrieve normalized MeSH headings that appear in every normalized document and score them according to predefined rules. These scores are enhanced by probabilities acquired statistically from a training corpus. The probabilities are computed for every unordered trigram sequence of subwords that co-occur with MeSH headings of the particular document. MeSH descriptors assigned to a given document are ranked according to their total scores. A threshold value defines the resulting set of top ranked keyphrases. The evaluation on 309 abstracts yielded precision of 55% and recall of 26% for the top 5 assigned keyphrases.

The particular advantage of this approach is its language independence. MeSH headings can be assigned to any language that is present in the MorphoSaurus lexicon. Unfortunately, creating such lexical knowledge base requires diligent efforts by people who have to be experts in linguistics and the particular domain (e.g. medicine).

2.3 Extraction vs. assignment

The previous section demonstrates that text classification works well as long as the documents belong to relatively few classes. For example, the text classification software TextCat[9] has been successfully used in applications ranging from language identification (trained for 70 classes) to spam filtering (only 2 classes necessary). Keyphrase assignment is a more problematic version of text classification and cannot be solved with the same strategies, because the amount of training data required increases rapidly with the number of categories grows. Controlled vocabularies, even if used in restricted domains, usually consist of several thousands of keyphrases. For example, the United States National Library of Medicine uses more than 22,500 MeSH descriptors for indexing medical articles.

[7] MorphoSaurus (http://www.morphosaurus.de/) is developed by the Department of Medical Informatics at Freiburg University, Germany, in cooperation with the Language and Information Engineering Lab at Jena University, Germany. It is maintained manually and comprises over 20,000 equivalent classes for English, German, Spanish and Portuguese subwords.

[8] For example, changing of “c” before {a, o, u} to “k” and before {e,i} to “z” helps to conflate different German versions of “cancer”: *Karzinom, Carzinom, Karcinom* etc. There are a lot of similar examples in medical language, since Latin terms are here very frequently.

[9] http://odur.let.rug.nl/ vannoord/TextCat/

Keyphrase extraction is closely related to keyphrase assignment, but it is resolved with a completely different strategy. It is based on analysis of the properties of keyphrases that can be computed relatively simply and quickly. It is notable that both sophisticated machine learning approaches and simple scoring of candidate phrases or their head nouns yield similar levels of accuracy (Barker and Cornacchia 2000). Their performance is still insufficient to replace the expensive manual work of human indexers. Although a rather large percentage of keyphrases assigned by authors have been reportedly extracted automatically (e.g. Hulth (2004)), these numbers should be taken with a pinch of salt, because they do not contain any information about non-matching keyphrases. When two professional indexers assign keyphrase sets to the same document, these sets usually also match only partially, but all keyphrases are highly related to the document and are definitely grammatical phrases. There is no guarantee for this in automatic keyphrase extraction. In fact, extracted keyphrases are often too general or malformed. Even approaches enhanced by some rudimentary linguistic techniques such as PoS patterns matching or NP chunking tend to extract useless or non-grammatical phrases, because they can not avoid errors generated by the linguistic tools.[10]

While these technical errors could be improved by using more accurate NLP-tools, keyphrase extraction still has several disadvantages. For example, there is no homogeneity among selected keyphrases, because the extraction process is restricted to the vocabulary of the document's author. Documents that describe the same subject in different but synonymous words (e.g. *seaweed culture* and *sea weed farming*) receive different keyphrases and cannot be grouped together according to their content. Keyphrase assignment avoids this shortcoming by having a controlled vocabulary of indexing terms. Another problem in automatic keyphrase extraction is that it is restricted to syntactic properties of phrases without considering their meaning. These algorithms also ignore the content of the document as a whole and therefore fail to cover all its topics in the keyphrase set. Keyphrase assignment is more advanced in this sense, because it analyzes the content of the document by taking into account the co-occurrence statistics between terms.

These observations show that both keyphrase assignment and extraction have their advantages, but also their limitations. Combining the approaches into a single keyphrase indexing system would allow one to benefit from both of them and at the same time avoid their shortcomings. The following section starts with a short insight into indexing strategy performed by

[10] Keyphrases extracted automatically from the abstract of the PhD thesis of Hulth (2004) with her tool are a good example for both these limitations: *presented research* is too vague and *first problem concerns* was selected due to the incorrect tagging of the word *concerns*.

librarian experts. On the one hand, they need to understand the content of the document first, on the other hand, their choice of keyphrases are restricted by cataloguing rules. The processes of understanding the content of the indexed document has not been explored in the context of automatic indexing yet. The indexing theory and the text comprehension issues in cognitive linguistics might provide some information for improving current automatic keyphrase indexing strategies.

2.4 Indexing theory and cognitive processes

The goal of keyphrase indexing in library science is to organize library holdings based on their content and to provide fast and unified access to them. Given a document, the task of the indexer is to assign a keyphrase set that reflects its main topics in an exact and precise way and to keep the indexing consistent among all of the documents in the collection.

David et al. (1995) define following key stages in performing this task:

1. Scanning of document: In this stage the indexer recognizes the document's nature (e.g. research report) and its main parts by using his professional knowledge.
2. Content analysis: In this stage the indexer uses his domain knowledge to determine what the document is about.
3. Concept selection: Among all concepts appearing in the document the indexer chooses those that are representative for the given document.
4. Translation into descriptors: This stage is only important for controlled indexing where central concepts need to be translated into descriptors.
5. Revision: The indexer adjusts his selection to his work environment by comparing his index terms to similar documents in the database.

The first and the last two stages are specific for the task of professional indexing. The main phase in the indexing task consists of content analysis and concept selection that both correspond to the general process of text comprehension. The way in which humans understand the document's content is an enormously complex process. Theories in cognitive linguistics describe this process as an interaction between perception organs, text memory and long term memory consisting of general knowledge, episodic memory, frames etc. (van Dijk and Kintsch 1983, p. 347ff). According to van Dijk and Kintsch (1983), large amounts of knowledge that are not expressed in the text itself need to be accessed and retrieved in order to construct a mental representation of the text's content in one's memory and to understand it. While perception

"The man went to the train station. After he bought a one-way ticket, he went to the platform and got into the train."

P1 [GO, MAN, TRAIN-STATION]
P2 [AFTER, P3, P5]
P3 [BUY, MAN, TICKET]
P4 [KIND, TICKET, ONE-WAY]
P5 [AND, P6, P7]
P6 [GO, MAN, PLATFORM]
P7 [GET-INTO, MAN, TRAIN]

Figure 2.1: Example for representation of a text in form of propositions

and text memory are involved in a modified way into current solutions for automatic keyphrase extraction and assignment, the long-term memory is not presented in any of known approaches. To support systems involving automatic text understanding with this kind of external knowledge, an automatically accessible knowledge base, represented in a flexible and usable form, is required. A popular type of such knowledge bases is *semantic networks*, where concepts are connected to each other according to semantic relations between them. Beside the relatively simple and intuitive representation of conceptional knowledge, semantic networks also support another activity involved into text understanding – *spreading activation* (Collins and Loftus 1975). When a particular concept (node) in the network is activated (e.g. the indexer reads a particular term in the text), other nodes that are related to it are activated as well. Activating nodes allows the avoidance of ambiguities and provides a framework, which represents the scope of the text's content and its relatedness to the entire knowledge base.

This process is crucial for determining concepts in the document. Concepts, in turn, build up *propositions* that denote the meaning of sentences and represent the text base (e.g. P1 to P7 in Figure 2.1). According to Kintsch's and van Djik's theory of text understanding (Kintsch and van Dijk 1978), propositions are then transformed mentally into a *macro-structure* representing the gist of the text. This transformation includes a recursive applying of the so-called *macro-rules*, such as *deletion* (elimination of irrelevant propositions), *generalization* (converting a series of specific proposition to a more general one) and *construction* (constructing a new proposition from a series of proposition given in text by activating world knowledge). The macro-rules can be represented as edges that connect the propositional nodes into a network representing the macro-structure (cf. Figure 2.2).

By applying macro-rules, new *macro-propositions* (e.g. M1 and M1 in Figure 2.2) can be constructed. Each macro-proposition summarizes the meaning of proposition from which

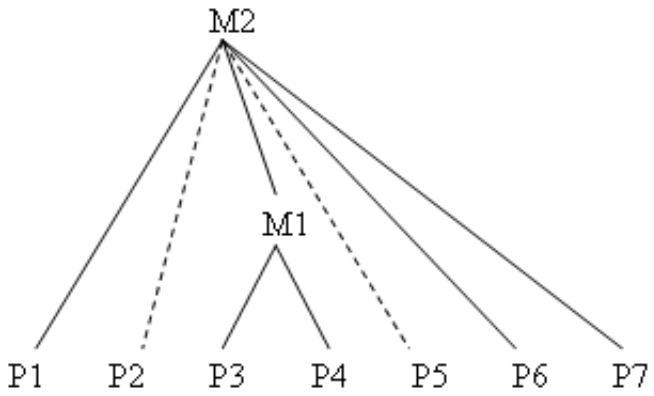

Figure 2.2: Example macro-structure for the text in Figure 2.2

it is constructed. Figure 2.2 shows a macro-structure that could be created from the text in Figure 2.1. The form of the macro-structure depends on the particular content of the text and the long-term knowledge of a given person. For example, Figure 2.2 represents the construction rule that transforms propositions P1, P6 and P7 and the macro-proposition M1 into a new macro-proposition M2. This macro-proposition is the theme of the given text snippet and could be expressed by a phrase *taking the train*. Although this information is not explicitly available in text, the reader is able to deduce it from his world knowledge. David et al. (1995) claim that these operations take place when professional indexers perform the second indexing stage, the content analysis.

The third point, selecting the main concepts in a document, can be also observed from the perspective of text comprehension process. Given a longer text, with more than one global macro-proposition, the question is, which are the most important. (Kintsch and van Dijk 1978, p. 78) proposed different strategies of how humans identify the key subjects in texts. For example, those propositions that are more frequent are obviously more important. Frequency is also a common feature in keyphrase extraction algorithms (cf. Section 2.1.1). Another interesting notice made by Kintsch and van Djik is that macro-propositions with the higher amount of ingoing edges are more significant. When one considers the semantic content of the text not as a macro-structure but as semantic network, this observation would mean that nodes with the highest amount of relational links to other nodes in the document are most relevant for its content.

The main goal in developing text-processing approaches, such as keyphrase extraction, is to supersede expensive and difficult intellectual human work. The natural way of designing such methods would be to simulate cognitive processes used for the same tasks by humans.

The backgrounds of indexing theory and cognitive linguistics presented briefly in this section provide new ideas of how keyphrase indexing task can be solved automatically. The ideal keyphrase indexing system needs, first of all, to detect the main concepts within a text. Second of all, semantic relations between the document's concepts and other information (e.g. general world knowledge or domain specific statistics) available to the system should be determined. Thirdly, this information on concepts and relations between them need to be analyzed to create the final set of most significant terms in a particular document.

The most difficult task in designing such a system is to extend it with a large but flexible database containing background knowledge. In computer science, thesauri and ontologies are typically used to encode world-knowledge into computer. A thesaurus is a database with entries representing concepts connected by semantic relations such as synonymy, antonymy, hyper-/hyponymy, mero-/holonymy and others. One of the most popular thesauri, *WordNet*,[11] developed in the Cognitive Science Laboratory at Princeton University, contains more than 150,000 words that are organized into over 110,000 groups representing concepts. WordNet became primarily a practical tool for knowledge-based natural language application and was explored successfully for different tasks, such as information retrieval, word sense disambiguation and knowledge engineering (Fellbaum 1998). Using domain-restricted thesauri allows one to limit the number of meanings per term and to reduce the size of the required knowledge database. In this project Agrovoc, a domain-restricted thesaurus for the field of agriculture (Section 3.2), was explored as semantic knowledge base. Chapter 4 describes in detail, how theoretical backdrops of this section were included in the design of the keyphrase indexing system KEA++.

[11] http://wordnet.princeton.edu/

Chapter 3

Experimental data and evaluation strategy

All textual resources employed in this project were obtained from the Food and Agriculture Organization (FAO) of the United Nations.[1] The main document collection consists of 200 fulltexts that were downloaded randomly from the document repository of the FAO.[2] Both training and automatic evaluation of KEA++ are accomplished on these documents. The actual indexing is based on another electronic resource from the FAO, the domain-specific thesaurus Agrovoc, which is incorporated in two ways in KEA++: as a controlled vocabulary and as a knowledge base. Agrovoc's semantic links between terms are not only explored in the context of the indexing process, but also extend the evaluation strategy in this thesis. A detailed analysis of Agrovoc's structure is necessary to see how it can be applied for these tasks in the best way. The description of both resources is presented in the first part of this chapter.

The remainder of the chapter investigates the problem of measuring the indexing performance. Matching the automatically extracted index phrases against manually assigned ones is a common way of evaluating indexing algorithms. Given a test set that is large enough, it is a fast way to see if experimental modifications are effective. However, the final testing of the indexing algorithm requires a more accurate and objective evaluation based on judgments of several assessors. In this thesis a small evaluation set with keyphrase sets assigned by several professional indexers is analyzed to find the best way of measuring the indexing quality of KEA++. A special issue of semantic relatedness between keyphrases (vs. exact matching) is studied in a separate section. The resulting evaluation strategy is applied to evaluate KEA++'s

[1] http://www.fao.org/

[2] http://www.fao.org/documents

indexing performance in Chapter 5.

3.1 Document collection

The mandate of the FAO is to increase agricultural productivity and to improve the conditions of rural population all over the world. One of its important activities is to collect, analyze and disseminate data that aid this development. FAO maintains a large and well used (1 Million hits per month) online document repository with technical documents, reports, books in the domain of agriculture, forestry and fishery. To access the documents users can either fill out the search form or select one or more descriptors from the FAO's thesaurus Agrovoc.

In this thesis a part of the document repository, consisting of 200 fulltext documents and their abstracts, serves as a document collection in the experiments. The collection was downloaded randomly with two restrictions. First of all, each document should contain two or more keyphrases from the Agrovoc thesaurus. This restriction is of a practical nature, since many documents in the repository are not indexed. Furthermore, each document should be available in two forms: as fulltext (PDF file with several pages) and as abstract. This restriction was made for experimental reasons. Using both abstracts and fulltexts in experiments allows us to determine how the indexing quality depends on the length of documents in both training and testing stages. This aspect was not investigated in previous experiments despite its importance. Most electronic document collections provide only abstracts. If the indexing quality does not depend on the document's length, the automatic indexing would be even more useful and applicable in real-world cataloguing systems.

Some statistical data about the document collection is presented in the Table 3.1. Lines 1 and 2 correspond to number of words per document and per abstract, whereas line 3 describes how many keyphrases were assigned to each document. The length of fulltexts varies from 95 to about 145,150 words per document and sums up to the total size of the fulltext collection of approximately 3.45 million words. Although abstracts are on average significantly shorter than fulltexts (the total size of the abstract collection is circa 25,650 words), some of them are longer than particular fulltexts, since the maximum size of an abstract is higher than the minimum size of a document. Each document was indexed with 5.4 phrases on average, with a median value of 4 keyphrases. The total number of assigned keyphrases is 1080. However, only 495 keyphrases in this set are unique. The median and mean[3] values demonstrate the

[3]The *mean* was computed as $\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$. The *median* is the $\frac{n+1}{2}$ largest observation. If this formula does not produce a whole number, the median is the average of the two observations on either side (Wild and Seber 1995).

	min	max	median	mean	total
(1) Documents	95	145,150	12,220	17,250	3,450,500
(2) Abstracts	20	501	116	128.3	25,650
(3) Keyphrase Sets	2	15	4	5.4	1080

Table 3.1: Size of text files in words (1,2) and keyphrases (3).

average size of each file in words or keyphrases. Differences in these values (particularly in lines 1 and 3) are caused by a small number of extremely long documents and few documents with high number of keyphrases respectively.

3.2 The Agrovoc thesaurus

The Agrovoc thesaurus has been developed by the FAO and the AGRIS/CARIS centers[4] since the 1980s and is used for both indexing and retrieval. Agrovoc is a multilingual controlled vocabulary designed to cover the terminology of all subject fields of agriculture, forestry, fisheries, food and related domains (e.g. environment). Using Agrovoc as a controlled vocabulary allows the maintainer of the FAO's document repository to organize the collection by grouping documents that deal with similar subject under the same index keyphrase. The most important advantage of having univocal descriptors is that they can reduce searching time. Various ways to express the same topic (e.g. *convenience foods*, *ready meals*, *ready to cook foods* etc.) are covered by a single keyphrase (*prepared foods*). Instead of trying all possible fulltext searches, with the help of Agrovoc users can retrieve all related documents in one instance. Another way to utilize Agrovoc for accessing documents in the online catalog of the FAO is to browse them by topic. Each descriptor in Agrovoc has a number of related phrases, so that users can refine their search by clicking through linked concepts (e.g. selecting *instant foods* from the page on *prepared foods*).

Agrovoc is available in print (AGROVOC 1995) and in a digital form. The current digital version (IMARK 2005) comprises nearly 35,000 entries in English, Spanish, French, Arabic and Chinese languages. Most entries are available in at least English, Spanish and French (E.g. a typical entry from AGROVOC (1995) in Figure 3.1). FAO supports the completion of the thesaurus and the extension of Agrovoc to other languages (e.g. currently Czech and Portuguese) are planned. Each entry consists either of a descriptor and terms that are related to it, or of

[4] AGRIS (The international Information System for the Agricultural Sciences and Technologies) and CARIS (Current Agricultural Research Information System) support the Library and Documentation System Division of the FAO.

En. Descriptor:	**Epidermis**
Scope Note:	*Of plants; for the epidermis of animals use SKIN*
En. BT:	**BT1** Plant tissues
	BT2 Plant anatomy
En. NT:	NT1 Plant cuticle
	NT1 Plant hairs
	NT2 Root hairs
	NT1 Stomata
En. RT:	RT Peel
Fr. Descriptor	Épiderme
Sp. Descriptor	Epidermis

Figure 3.1: Example entry in Agrovoc (1998).

a non-descriptor and the pointer to the corresponding descriptor. Only descriptors are used by human indexers as keyphrases for documents in the repository. The USE relation between non-descriptors and descriptors prevents having different entries of the same semantic concept and supports subject based indexing. Ambiguities are handled by scope notes, if required.

Beside the USE (or the symmetric USED-FOR) relation, which in most cases denotes the synonymy (e.g. *overweight* is the descriptor for *obesity*[5]), three other semantic relations are present in Agrovoc. Hierarchical relations between terms in Agrovoc are expressed by *broader term* (BT) and the symmetric *narrower term* (NT) relations (e.g. *sense organs* is a BT for *eyes*). Usually there is only one broader term for one or more narrower terms (in circa 96% of cases). Rarely, some terms have two and three generic terms: e.g. *Brazil* is NT for both *South America* and for *Latin America*. The last relation RT, *related term*, is vaguer and expresses any kind of relatedness (e.g. *vision disorders* is related to *vision*, or *marine fisheries* is related to *coastal fisheries*).

In this thesis only the English entries in the Agrovoc were considered: in total over 27,500 terms. This number adds up from approximately 16,800 descriptors and 10,600 non-descriptors. Table 3.2 shows how many descriptors are linked to other terms by each relation. It is remarkable that all non-descriptors are mapped to less than 6,000 descriptors. This means that for at least every third descriptor in Agrovoc there are two alternative expressions. This example shows that using a controlled vocabulary for content-based organization of documents is essential. Other relations in Table 3.2 represent links between descriptors. Approximately one half of controlled terms in Agrovoc have RTs, building over 13,700 semantically related

[5]In some cases a descriptor is used for several specific non-descriptors (e.g. *eyes* should be used instead of *pupils*, *eyelids* etc.) to prevent over-specificity in indexing.

Relation	terms with given relation	pairs connected by given relation
BT	15,000	15,700
NT	4,200	
RT	8,800	13,700
USED-FOR	5,900	6,300

Table 3.2: Number of terms connected by each relation in Agrovoc.

Relation	max	avg	stdev
BT	3	1.0	0.2
NT	242	3.7	7.6
RT	152	3.1	5.0

Table 3.3: Number of terms related to a descriptor by each relation in Agrovoc.

pairs. NT and BT relations connect most of the terms (95%) into a hierarchical structure and build around 15,700 symmetric pairs of generic and specific terms.

Each descriptor in Agrovoc is connected by different relations to other terms in the thesaurus. Table 3.3 presents details on the number of terms that are connected by each relation. While a term usually has only one broader term, NT and RT connect between up to 242 or 152 terms respectively. The number of NTs per descriptor varies a lot due to the specificity of the domain. Keyphrases from biological areas in particular have many classificational branches (e.g. the descriptor *saltwater fishes* has 82 narrow terms, which link to further fish species in their turn).

The analysis of the vertical distribution of terms reveals that Agrovoc does not have a tree-like structure like other thesauri (WordNet or Roget's thesaurus). Its relational organization is rather flat: A vertical path from a generic concept on the top to a specific one has an average length of 2.3, and maximum 7 links (c.f. to WordNet, where terms are organized on up to 16 levels).

Another observation is that more than 11,600 descriptors (70% of the total) have no further BTs and build the top level of the Agrovoc tree. Compared to other thesauri, where concepts are organized up to several basic concepts (e.g. 9 unique beginner in WordNet such as *entity*, *abstraction*, *act* etc.), Agrovoc is obviously missing several top levels that would represent relations between generic concepts.

Descriptors in Agrovoc are phrases consisting of one or more words. Row 1 of Table 3.4 shows the percentage of possible phrase lengths in Agrovoc. One and two word phrases form the majority of almost 93% of all descriptors. Comparing these numbers to lengths of keyphrases assigned in the main document collection (Row 2 of Table 3.4) reveals that there

words per keyphrase	1	2	3	4 to 7
% of all Agrovoc terms	45.0	47.7	5.90	1.4
% of assigned terms	34.1	62.4	3.2	0.3

Table 3.4: Distribution of term length in Agrovoc.

is a difference between both distributions. It seems that when indexers select keyphrases from the thesaurus, they prefer to use two-word phrases.

Developed by the same organization, Agrovoc is a very suitable thesaurus for the given document collection. Its structure differs from other popular general thesauri, but it is simple and consists of only four basic relations. This allows us to keep the methodology general enough to be used with any other controlled vocabularies or thesauri. Thesauri, in general, can be incorporated into systems in miscellaneous ways. After introducing several standard measures for indexing performance, this chapter describes how Agrovoc can be used to enhance these metrics.

3.3 Measuring indexing performance

Measuring the quality of assigned keyphrases is an important issue in both automatic and manual indexing. While in automatic keyphrase extraction and assignment evaluating strategies were mainly adopted from the field of Information Retrieval (relations between the algorithm's answer set and the "correct" set specified by humans), in library science performance measures were developed independently, with the main goal of improving indexing in the entire catalog. The following subsections give an overview on developed evaluation methodologies and discuss their limitations.

3.3.1 Indexing performance of algorithms

To evaluate the performance of KEA++ the 10-fold cross-validation, described in Section 1.4, is used. To do this the main document collection (Section 3.1) is divided into ten parts consisting each of 20 documents, and use 180 documents for training and 20 for testing in each evaluational run.

The test collection contains keyphrase sets assigned to every document by a human indexer. The simplest evaluation method is to match the stemmed versions of extracted keyphrases to stems of manual assigned ones. The number of matching ("correct") keyphrases is then computed relatively to the number of all extracted phrases (*Precision,* Equation 3.1) and to the

number of manual assigned phrases (*Recall*, Equation 3.2) for each document separately. The overall precision and recall values are taken as average over the entire test set.

$$Precision = \frac{\#\ correct\ extracted\ keyphrases}{\#\ all\ extracted\ keyphrases} \tag{3.1}$$

$$Recall = \frac{\#\ correct\ extracted\ keyphrases}{\#manually\ assigned\ keyphrases} \tag{3.2}$$

Both formulas formulas return a value between 0 and 1, where 0 stands for complete failure and 1 denotes the ideal case. It is easy to get the highest possible values for both measures separately. For example, using the keyphrase extraction strategy presented in Section 2.1, recall can be maximized by including as many candidate phrases into the keyphrase set as possible; however, this would decrease precision to a minimum. The opposite effect can be achieved by keeping the resulting set as small as possible, which increases precision and decreases recall. Therefore, it is important to consider precision and recall in parallel, or use *F-measure* (Equation 3.3, (van Rijsbergen 1979)) that combines these metrics in a single formula.

$$Fmeasure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{3.3}$$

All three measures were applied in most studies on keyphrase extraction (Turney (1999), Witten et al. (1999), Frank et al. (1999), Hulth (2004)), which allows us to compare results produced by KEA++ with other algorithms.[6]

3.3.2 Inter-indexer consistency

Keyphrase indexing is originally the task of professional human indexers, so it is worthwhile to see, how their performance is measured in library and information science. The indexing quality reflects the degree to which chosen keyphrases accurately and consistently represent the content of every document in a given document collection. Evaluating these characteristics is a subjective task and difficult to quantify. In contrast to automatic evaluation, it is not easy to define which keyphrases are "correct", so that matching an assigned keyphrase set against a "perfect" one is impossible. In its place, measuring the degree of consistency between pairs of human indexers working on the same collection has established as the most appropriate way in estimating the potential indexing quality and the effectiveness of keyphrase retrieval.

[6] This comparison has to be handled with care since the performance of an algorithm depends on the particular test and training data. The ideal way to compare algorithms against each other would be to test them on the same previously unseen document collection.

Although there is no standard calculation method for measuring the *inter-indexer consistency* (IIC), there are two widely-accepted formulas. Hooper's measure (Hooper (1965), Equation 3.4) computes the ratio between the overlap of two given keyphrase sets and their difference. The second, Rolling's measure (Rolling (1981), Equation 3.5), is the ratio between double overlap and the total number of keyphrases in both sets. Further analysis of the inter-relation between these formulas (cf. Equation 3.6) reveals that Hooper's measure always gives lower inter-indexer consistency than Rolling's, since it is related by $\frac{R}{(2-R)}$ and R operates on a range [0,1].

$$Hooper's IIC \;\; H = \frac{C}{A+B-C} \tag{3.4}$$

$$Rolling's IIC \;\; R = \frac{2C}{A+B} \tag{3.5}$$

$$H = \frac{1}{\frac{(A+B)}{C} - 1} = \frac{1}{\frac{2}{R} - 1} = \frac{R}{2-R} \tag{3.6}$$

where

- C – number of keyphrases in agreement
- A – number of keyphrases assigned by Indexer 1
- B – number of keyphrases assigned by Indexer 2

Another observation is that Rolling's measure produces the same results as F-measure. If A is the number of keyphrases assigned by the algorithm and B the number of manual keyphrases, precision could be defined as $\frac{C}{A}$ and recall as $\frac{C}{B}$. The Equation 3.7 demonstrates how the reformulated F-measure can be equalized with the Rolling's IIC formula.

$$\frac{2\frac{C}{A}\frac{C}{B}}{\frac{C}{A}\frac{C}{B}} = \frac{\frac{2C^2}{AB}}{\frac{AC+BC}{AB}} = \frac{2C^2}{AC+BC} = \frac{2C}{A+B} \tag{3.7}$$

3.3.3 Limitation of evaluation methods

The advantage of the formulas presented above is that they can be computed quickly and simply. But at the same time, this kind of evaluation has several limitations. First of all, it lacks on objectiveness. When humans select keyphrases, their results are affected by several factors, such as their expertise and experience in a given field, depth of indexing and its purpose, quality of the used controlled vocabulary, availability of additional aids etc. Thus, values produced by Rolling's or Hooper's measures are usually quite low. Leininger (2000) summarizes the achieved results in inter-indexer consistency with following substantial ranges: Consistency of human indexers ranges between 4% and 67% (with an average of 27%) for free indexing and

between 13% and 77% (with an average of 44,3%) for indexing with controlled vocabulary.[7] From this point of view, using only automatic evaluation is not very meaningfull. Instead, one should compare the performance of the developed system to as many other indexers as possible by using the same documents. Ideally, the automatic indexing should have on average as high consistency with human indexers as they among each other.

Another reason why the common computation of these formulas is insufficient, is that terms in agreement, or the "correct" extracted terms, are identified mainly by using exact matching. In automatic evaluation studies stemmed version of keyphrases are usually used, and librarians in some studies consider partial matching between subject headings (exact matching of at least one word), but these alterations are still restricted to word surfaces, which do not provide any information on their meaning. Keyphrases that do not match exactly can be still similar in their meaning, which is relevant for computing the overlap between two keyphrase sets. Semantic relatedness between keyphrases is usually ignored not only in all algorithm evaluation studies, but also in measuring inter-indexer consistency between humans.

Some studies on evaluation of algorithms' indexing performance consider the meaning of keyphrases in human-based experiments. Human assessors read example documents and rate keyphrases assigned to them manually and automatically. For example, they rate the degree, to which the entire keyphrase sets (Jones and Paynter 2003) or each individual keyphrase (Jones and Paynter 2002) reflect the content of the document. The assigned scores, taken as an average over all documents and all subjects involved into the evaluation study, reflect objectively the indexing ability of the algorithm compared to the author's or indexer's performance. Human evaluation of the keyphrase extraction algorithms KEA, B&C and Extractor (cf. Section 2.1) conducted by Jones and Paynter (2003) and Barker and Cornacchia (2000) confirmed that human indexers have consistently higher performance than algorithms. While the individual extracted keyphrases were rated positively by human assessors, algorithms in many cases failed to produce keyphrase sets that cover all important topics of the document.

Experiments involving human judgments are very extensive and costly. It is necessary to find a large group of subjects, who have similar education level and direction, corresponding to a sufficient amount of the experimental data (for example, Jones and Paynter (2003) had 20 students from a Human Computer Interaction course who read and rated 6 documents from a conference on human factors). Another problem with experiments based on human judgments is that the degree of agreement between them is low. Barker and Cornacchia (2000) compared

[7] Of course, the size of the controlled vocabulary should be considered, when comparing these numbers. The larger is the vocabulary, the lower is the probability that indexer will assign same keyphrases.

the agreement between 12 subjects in their study by using *Kappa Statistics*, a measure that takes into account the likelihood of choosing particular answer by chance.[8] The agreement on giving preference to a particular keyphrase set (Extractor or B&C, cf. Section 2.1.1) was 0.06 and on rating of individual keyphrases 0.27. Both values are surprisingly low, since they are closer to chance agreement ($\kappa = 0$) than to perfect agreement ($\kappa = 1$). Finally, these studies do not include a comparison of judgments on keyphrase sets assigned by different humans. Similar to automatic evaluation, they solely compare the indexing quality of an algorithm with the abilities of one particular human indexer.

Many studies on humans' indexing performance have noted the distinction between consistency at the concept and term levels (e.g. Markey (1984); David et al. (1995); Saarti (2002)). For example, Markey (1984) and Iivonen (1995) estimated conceptual consistency by including terms that are synonyms or semantically related, as well as exact matches, when determining the number of terms in common, and showed that consistency is higher at the concept level than at the terminological level. However, they did not quantify the effect in terms of the relative importance of terminological and conceptual matches. Also, judgments of synonymy and semantic relatedness were made subjectively by human evaluators.

To ensure a fair comparison with other systems, the automatic evaluation of KEA++ in Chapter 5 includes precision, recall, and F-measure. At the same time, an important issue in this thesis is to overcome the limitation of existing evaluation metrics by comparing the algorithm to more than one human indexer and taking into account the semantic relatedness of assigned keyphrases.

3.4 Semantic based similarity measures

Computing the semantic similarity between words and phrases is a central problem in computational linguistics. Different metrics have been developed by using additional information sources such as corpus statistics or semantic networks. Although both corpus-based and thesaurus-based approaches yield strong correlation to human assessments (McDonald (1997); McHale (1998)), no generally accepted solution for estimating similarity exists. The problem with corpus-based methods is that they require large amounts of data. If the occurrence frequency of two words in a given corpus is too low then similarity measures are not reliable, as

[8] Kappa is defined as $\kappa = \frac{P(A)-P(E)}{1-P(E)}$, where $P(A)$ is the number of times where two judges agree relative to the total number of judgments; $P(E)$ is the proportion of times judges would agree by chance, which is computed as the number of possible agreements divided by the number of all possible answer combination. $\kappa = 0$ stands for chance agreement.

words may co-occur by chance. Terra and Clarke (2003) showed that acceptable results can be achieved with around 400 GBytes of textual data. This amount of data could not be procured within this project, therefore, similarity is estimated here based on semantic relations encoded in the thesaurus.

3.4.1 Level-based similarity

The majority of similarity metrics developed for other thesauri are based on the hierarchical relations between keyphrases (Resnik (1995); Leacock and Chodorow (1998)). Moreover, it is presumed that the hierarchy in the thesaurus has a complete tree structure, so that there is a path connecting any pair of keyphrases. Due to the restriction on a specific domain, Agrovoc has only a flat hierarchical structure. Also, the associative relation RT in Agrovoc is almost as important as the hierarchical BT and NT, since the number of words connected by RT is the second largest in this thesaurus (cf. Section 3.2). Therefore, all three Agrovoc relations should be considered in the computation of semantic similarity between keyphrases. Rada et al. (1989) applied the same distance measure on a network augmented with associative relations and evaluated the re-calculated distances against human judgments. This experiment showed that distance metrics designed for hierarchical relations cannot be applied on other semantic links due to the differences in their significance. The natural consequence is to weight relation according to their strength. However, the strength of relations is rather of intuitive nature and difficult to quantify automatically.

The simplest way to compute relatedness according to Agrovoc relations is to define similarity levels, as proposed by Markey (1984) and Iivonen (1995) in order to estimate the number of matching keyphrases and concepts between keyphrase sets assigned by two humans. Analogously the similarity levels can be integrated into evaluation measures described above (Section 3.3.3), by computing the number of matching keyphrases according to relations between them as encoded in Agrovoc. The similarity on level one is the traditional way of counting matching stemmed keyphrases in two sets and corresponds to terminological “correctness”. At level two, an automatically extracted keyphrase is considered “correct” if it is related to any manually assigned keyphrase by any one-path relation in the thesaurus (BT, NT or RT in Agrovoc). At the next, a keyphrase is considered “correct” if it is related to a manually assigned keyphrase by a two-path connection that involves the same relation (BT and BT, or NT and NT, or RT and RT). In other words, the three levels of “correctness” are:

- **Level I** keyphrases have equal pseudo-phrases, e.g. *epidermis* and *epidermal*.
- **Level II** keyphrases have equal pseudo-phrases or are one-path related, e.g. *epidermis*

and *peel*, or *plant hairs* and *root hairs*.

- **Level III** keyphrases have equal pseudo-phrases or are one- or two-path related, e.g. *plant cuticles* and *root hairs*.

This is the easiest way to estimate semantic similarity between keyphrases. The standard evaluation metrics that do not consider similarity at all are easily extended from pure terminological to conceptual matching and are still computed automatically. Although the results are not exact, they give an idea about the nature of extracted keyphrases. The limitations of level-based metrics were already mentioned in the previous subsection. They do not consider the strength of relatedness, and the comparison is made only to one keyphrase set, defined as perfect.

3.4.2 Vector based similarity

When multiple indexers have assigned keyphrases to the same document set, their keyphrase sets normally differ and none of the sets can be considered as "perfect". A keyphrase extraction algorithm should ideally assign keyphrases that match exactly or are related to those phrases that were selected by the most indexers, since they are likely to be more important (Zunde and Dexter 1969). An elegant way to grip this observation formally is to represent keyphrase sets in a multi-indexer setting as vectors and compute the similarity between them with the cosine correlation measure (cf. Section 1.4). The firm theoretical foundation of this technique gives it some advantages over ad hoc measures like Rolling's. Couched in the same terms, it is:

$$Cosine(Indexer_1, Indexer_2) = \frac{C}{\sqrt{AB}} \tag{3.8}$$

where C is the number of keyphrases the indexers have in common and A and B the size of their individual keyphrase sets respectively. This equation ranges from 0 when there are no keyphrases in common to 1 when the sets are the same. In effect this uses the geometric mean of A and B in place of Rolling's arithmetic mean. The two measures are the same when the sets are the same size and not much different unless they have radically different sizes. (Even when the set sizes differ by a factor of three, the measures differ by less than 15%.)

For a more general formulation, represent set A by the vector $\underline{A} = [A_1, A_2, \ldots, A_n]$, where n is the vocabulary of keyphrases and the element A_i is 1 or 0 depending on whether keyphrase i is in the set or not. Then

$$Cosine(Indexer_1, Indexer_2) = \frac{\sum_i A_i B_i}{\sqrt{\sum_i A_i^2}\sqrt{\sum_i B_i^2}} = \frac{\underline{A} \cdot \underline{B}}{|\underline{A}||\underline{B}|} \tag{3.9}$$

where $\underline{A} \cdot \underline{B}$ is the *dot product* of vectors. If the elements of the vectors are 0 or 1, their dot product is the number of elements they have in common. Given keyphrase sets from several different indexers, the single vector that represents their average can be used in the cosine measure to determine the similarity of an individual indexer to the group.

The next step is to include the relations between keyphrases into the formula. The idea in this experiment is to use numeric weights $\gamma, \alpha, \beta \in [0, 1]$ that reflect the "importance" of an exact match and the relatedness expressed by the RT and the inverse BT/NT links respectively. To ensure that the importance is relative, the weights are considered to sum to 1, and write $\gamma = 1 - \alpha - \beta$. It makes sense to demand that an increase in one weight necessarily involves decreasing the others; otherwise the measure of similarity could be raised artificially by simply increasing all weight values.

Of course, taking thesaurus relations into account in an inter-indexer consistency measure by including them as linear keyphrases in a weighted sum can only give a gross numeric approximation to the true semantic consistency. In any actual example of indexing, the importance of related keyphrases will depend greatly on the other keyphrases that the indexers have included and the exact nature of the relationship between the keyphrases. However, using a linear sum seems better than the alternative of not considering related keyphrases in the consistency measure at all.

To take account of thesaurus relations, the similarity between two indexers A and B is estimated by computing the cosine measure between A's vector of keyphrases $\underline{A}$ and a version $\underline{B}'$ of B's keyphrase vector that has been adjusted to reflect keyphrases that are related to B's choices. First express the relations RT and BT/NT by $n \times n$ matrices $\underline{\underline{R}}$ and $\underline{\underline{N}}$ whose element at position i, j is 1 if keyphrase i is related to keyphrase j and 0 otherwise. These two matrices are symmetric, the former because RT is a symmetric relation and the latter because it subsumes both the NT and BT relation, which are inverse. Then, using weights γ for identity and α and β for RT and BT/NT respectively, the adjusted version of B's keyphrase vector is $\underline{B}' = (\gamma + \alpha\underline{\underline{R}} + \beta\underline{\underline{N}}) \cdot \underline{B}$. This makes the overall measure

$$\frac{\underline{A} \cdot (\gamma + \alpha\underline{\underline{R}} + \beta\underline{\underline{B}}) \cdot \underline{B}}{|\underline{A}||\gamma + \alpha\underline{\underline{R}} + \beta\underline{\underline{N}}||\underline{B}|}$$

The formula is symmetric: it is the same as the cosine measure between $\underline{B}$ and $\underline{A}' = (\gamma + \alpha\underline{\underline{R}} + \beta\underline{\underline{N}}) \cdot \underline{A}$ because the relationship matrices are symmetric.

To determine suitable values for the coefficients α and β, I choose them to maximize the overall consistency of professional human indexers. The work of human indexers is taken to

be the gold standard, and take thesaurus relations into account in a way that optimizes their performance. The next section gives an overview of the performance achieved by indexers in our experiment and describes how the coefficients for this formula are determined.

3.5 Human experts' keyphrases

The main goal of developing algorithms for automatic keyphrase indexing is to replace expensive and time-consuming work of humans (cf. Section 1.2). Ideally, the quality of keyphrases assigned automatically should be as high as the quality of humans' keyphrases. As noticed in Section 3.3 of this chapter, the indexing quality of professional indexers is commonly estimated with an inter-indexer consistency measure. Given an experimental data of several documents and keyphrase sets assigned by more than one human indexer, the average consistency between keyphrases assigned automatically and manually by each individual indexer should be in best case as high as the average consistency between humans. To determine the IIC between humans, which will provide the baseline for the KEA++'s performance, the second document collection is used. It consists of ten documents indexed by six professional indexers with keyphrases from the Agrovoc thesaurus.

Although ten documents are probably not enough to obtain reliable results, this experiment is important, since none of the known studies on automatic keyphrase indexing provide evaluation of this kind. I believe that this amount of data is sufficient at least to have a gross idea of the true performance achieved by KEA++. The indexed documents were provided by the FAO and cover topics in the agricultural domain; they were selected randomly from two sources: the FAO's document repository, and websites whose content relates to the FAO's interests. This section describes the properties of this data and information that can be additionally deduced from it.

3.5.1 Statistics on indexers' keyphrase sets

The indexers assigned between 5 and 16 keyphrases to each document. Table 3.5 gives statistics on the number of keyphrases per document, broken down by indexer. For example, Indexer1 assigned 4.35 more keyphrases to each document than the average indexer, whereas Indexer3 assigned 3.25 fewer keyphrases than the average. The standard deviation of the Indexer4 is much higher than the others, who have similar values. Table 3.6 concerns the length of keyphrases. All keyphrases were selected from Agrovoc, which is specific to this domain. As in the main document collection, all indexers preferred descriptors consisting of two and

	max	min	mean	stdev
Indexer1	16	11	13.5	1.51
Indexer2	12	8	9.2	1.40
Indexer3	8	5	5.9	0.99
Indexer4	14	5	9.3	2.97
Indexer5	13	8	9.8	1.55
Indexer6	10	5	7.2	1.32
average	12.17	7.00	9.2	1.62

Table 3.5: Size of keyphrase set in keyphrases

	1 word	2 words	3 words	4 words
Indexer1	30	62	7	0
Indexer2	27	66	7	0
Indexer3	34	61	5	0
Indexer4	36	59	4	1
Indexer5	25	67	7	1
Indexer6	30	64	7	0
average	30	63	6	0

Table 3.6: Keyphrase length in words (%)

more words (69% of all keyphrases), although only about 45% of descriptors in Agrovoc exceed one word (cf. Table 3.4 in Section 3.2).

Table 3.7 shows the consistency of each indexer's selections with those of the other indexers, over all test documents, according to Rolling's measure; it also shows the cosine measure between each indexer and the group comprising the remaining indexers. In both cases Indexer1 has the highest average consistency, while Indexer6 has the lowest, although the differences are marginal. The same picture is obtained using both Hooper's measure and the cosine measure between pairs of individuals (not shown), although as noted above all consistency figures are slightly smaller for Hooper's and slightly larger (1%-12%) for the cosine measure.

Next, the overlap between the keyphrases that different indexers assigned to documents is examined. There were 280 different keyphrases in the total of 550 keyphrase assignments made by all indexer. It is remarkable that the majority of these keyphrases (150, or 55%) were assigned to documents by a solitary indexer, and only 10 keyphrases (3.6%) were agreed by all indexers.

To study this striking phenomenon further each indexer's predilection for assigning keyphrases that no one else assigns to that document are analyzed. These are keyphrases that are idiosyncratic to this particular indexer. This figure varied between indexers from an average of one idiosyncratic keyphrase per document to more than four, and averaging over all

Rolling	Indexer							Cosine
	1	2	3	4	5	6	avg	
1		45	37	44	41	43	41.8	54.6
2	45		48	33	36	35	39.4	46.8
3	37	48		36	30	32	36.8	46.8
4	44	33	36		40	31	36.9	52.3
5	41	36	30	40		38	36.9	51.1
6	43	35	32	31	38		35.9	45.5
						total	38.0	49.5

Table 3.7: Inter-indexer consistency according to Rolling's measure, and overall consistency with all other indexers according to the cosine measure (%).

indexers, nearly a third (28%) of each indexer's keyphrase choices were idiosyncratic. (This is the same as the number of idiosyncratic keyphrases, 150, expressed as a percentage of the total number of tem assignments, 550.)

Does the high percentage of idiosyncratic keyphrase choices reflect a major difference in the perception of the document by each indexer? I assume not, and believe instead that it reflects a flaw in the measures used. Rolling's measure (along with Hooper's, and precision and recall) ignores the semantics of the keyphrases because exact phrase matching (after case-folding) is used to quantify the overlap between keyphrase sets. For example, the keyphrases *seaweed culture* and *seaweed products* assigned to a document entitled *Seaweed Farming: An Alternative Livelihood for Small-Scale Fishers?* are considered as non-matching, although they are semantically related.

When comparing keyphrase sets assigned by different indexers to the same document, it is proposed here to take account of whether a non-matching keyphrase in one set is related by any of the three thesaurus links (RT, BT, NT) to any keyphrase in the other sets. Furthermore, certain compositions of these relationships are also investigated: keyphrases related to a related keyphrase (RRT), broader than a broader keyphrase (BBT), narrower than a narrower keyphrase (NNT), and sibling keyphrases (BNT). These particular compositions are chosen because they might occur reasonably often between keyphrase sets assigned by the different indexers.

Table 3.8 shows the distribution of relations among non-matching keyphrases, computed separately for each indexer. Because indexers often select multiple keyphrases for a single topic discussed in a document, intersections between relations often occur. For example, given keyphrase sets $A = a_1 \dots a_n$ and $B = b1 \dots b_m$, keyphrase a_1 may be RT for b_1 and NT for b_2. The number of related keyphrases covered by every relation was computed independently, and some keyphrases may contribute to more than one row-for example, most RRT keyphrases

	RT	NT	BT	RRT	NNT	BBT	BNT	No relation
Indexer1	7	4	2	15	2	1	1	20
Indexer2	6	2	2	14	1	0	1	6
Indexer3	4	0	0	5	0	0	1	4
Indexer4	2	2	2	13	0	0	1	11
Indexer5	8	2	1	15	1	1	2	12
Indexer6	9	2	0	11	0	0	2	3
average	6.0	2.0	1.2	12.2	0.7	0.3	1.3	9.1

Table 3.8: Number of non-matching keyphrases chosen by one indexer that relate to some keyphrase chosen by another.

are also RT. By far the most common relations are RT and RRT – an average of nearly 13 non-matching keyphrases from each indexer are either RT- or RRT-related to keyphrases assigned by other indexers – whereas the grandchild/grandparent relations NNT/BBT are rare overall.

Table 3.8 reveals interesting indexer-specific patterns. For example, Indexers 3 and 6 almost never choose a broader keyphrase (BT and BBT) than another indexer. Indexers 1, 4 and 5 seem to exhibit some inconsistency in specificity level, choosing both broader and narrower keyphrases than other indexers reasonably often.

3.5.2 Relations among assigned keyphrases

To see which keyphrases are more important than others and how they are related among each other according the thesaurus it is instructive to analyze particular documents in detail. Figure 3.2 demonstrates this analysis visually for the document with the title "*The Growing Global Obesity Problem: Some Policy Options to Address It*". It is notable, that some phrases selected by only one indexer (e.g. *disease control*) are less related than those selected by many indexers (e.g. *overweight* or *nutrition policies*). But at the same time, this is not the only one evidence of significance. Keyphrases, that were selected by only one indexer, but are related to other, frequently selected keyphrases, (e.g. *overeating*, *nutrition status*) are obviously more important than single selected but unrelated ones. The fact that particular keyphrases were preferred to others can be explained by difference in the idiolects of indexers. For example, the descriptor *overeating* is in general not as commonly used as *overweight*[9]. Thus, the overall significance of each keyphrase should not only consider the degree of its popularity but also its relatedness to other keyphrases describing the document.

By using Figure 3.2 one can also observe that when indexers select keyphrases from the

[9]While this assumption is made here rather intuitively, one could use a corpus to compare the occurrence frequencies of both terms. A quick check at http://www.google.com approved that approximately 2.7 millions pages are about *overweight*, while only 0.4 million pages contain the word *overeating*.

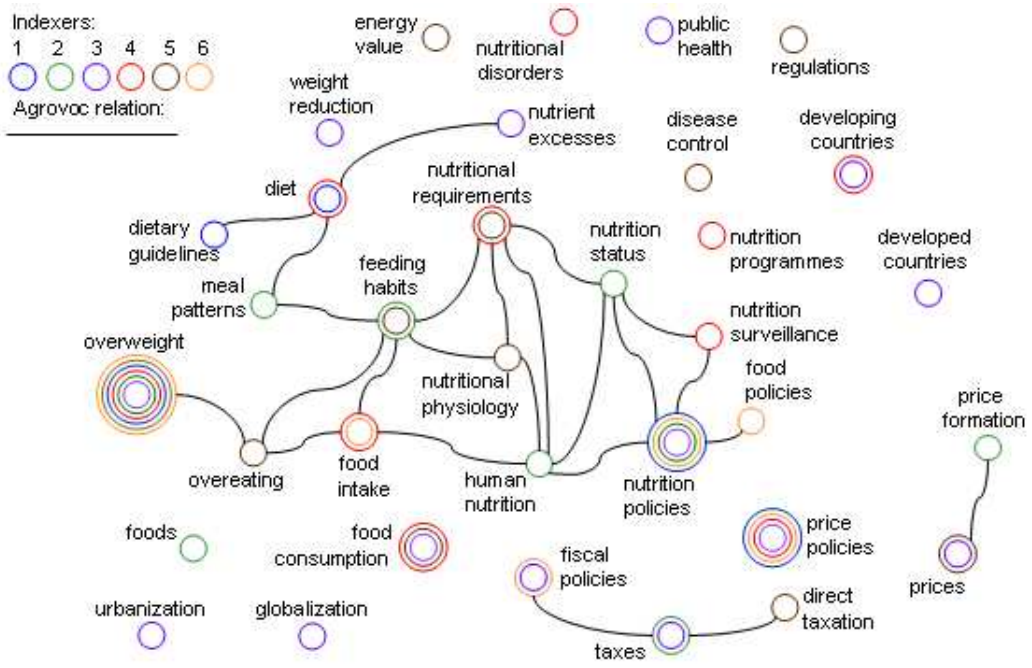

Figure 3.2: Relations between keyphrases assigned to the document on *global obesity problem*

thesaurus, some kind of activating of nodes in the knowledge networks is happening. This phenomenon was described in Section 2.4. The knowledge activation while performing indexing is here supported additionally by the links in the thesaurus. The formation of the almost complete network representing a keyphrase set as in Figure 3.2 is seldom. The analysis of other keyphrase sets revealed that they rather contain several subnets, representing the separate topics in the document. For example, Figure 3.3 shows the excerpt from the keyphrase set for the document on *Overview of Techniques for Reducing Bird Predation at Aquaculture Facilities*. Several subnets such as {*aquaculture, fish culture, ... , fisheries*}, {*bird control, noxious birds, ... , scares*}, {*equipment, fencing, ..., protective structures*} etc. are recognizable in the Figure 3.3. Every indexer seems to have mapped the content of the document to these subnets by selecting one or more descriptors in each of them. Which descriptor in each subnet was selected is not that relevant, as soon as the topic in the document is represented by at least one keyphrase from a given subnet. These observations demonstrate the importance of considering semantic relations between keyphrases.

3.5.3 Estimating similarity coefficients

To take into account semantic relations while computing similarity between two keyphrase sets, the vector model and the cosine measure are explored in this thesis (cf. Section 3.4.2). It

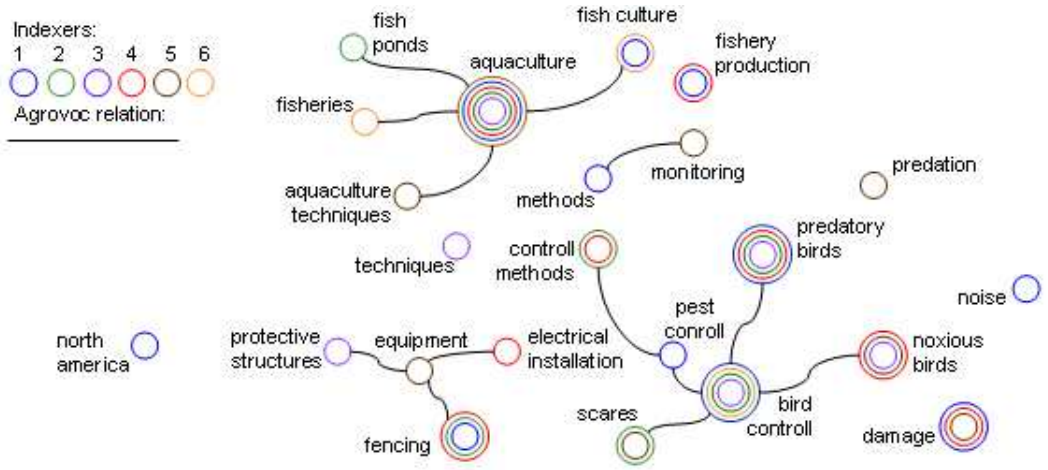

Figure 3.3: Relations between keyphrases assigned to the document on *bird predation*

is assumed that coefficients for different relations included into this measure can be estimated through analysis of manually assigned keyphrases. The idea of maximizing the similarity between indexers' keyphrase sets is described in following.

Given keyphrases assigned by a group of indexers to a group of documents, calculate the similarity between each indexer and all the others taken together, summed over all documents. This measures the degree to which that indexer agrees with the rest. Then choose α and β to maximize the total of these agreement values, in other words, maximize

$$SIM = \sum_{indexers\ i} \sum_{documents\ D} \frac{I_i^D \cdot (\gamma + \alpha\underline{\underline{R}} + \beta\underline{\underline{N}}) \cdot \sum_{indexers\ j} I_j^D}{|I_i^D||\gamma + \alpha\underline{\underline{R}} + \beta\underline{\underline{N}}|| \sum_{indexers\ j} I_j^D|}$$

where I_i^D is the vector of keyphrases that indexer I assigns to document D (and $\gamma+\alpha+\beta = 1$).

Figure 3.4 plots this value, normalized by the number of indexers and documents, against α (when $\beta = 0$) and β (when $\alpha = 0$) respectively. The line whose points are marked is the overall result for all 6 indexers, and to illustrate the robustness of the procedure the other lines show each subset of 5 indexers. The curves for β have a less distinct maximum than those for α because the BT/NT relationship occurred less frequently in the data than RT. The optimal values in the joint distribution for all indexers are $\alpha = 0.20$ and $\beta = 0.15$ respectively. As the shallow peaks in Figure 3.4 indicate, these values are approximate. The optimal values of α and β for the 5-indexer subsets range throughout the intervals [0.2,0.25] and [0.15,0.24] respectively.

When computing the cosine measure between indexers using the best overall values, if a

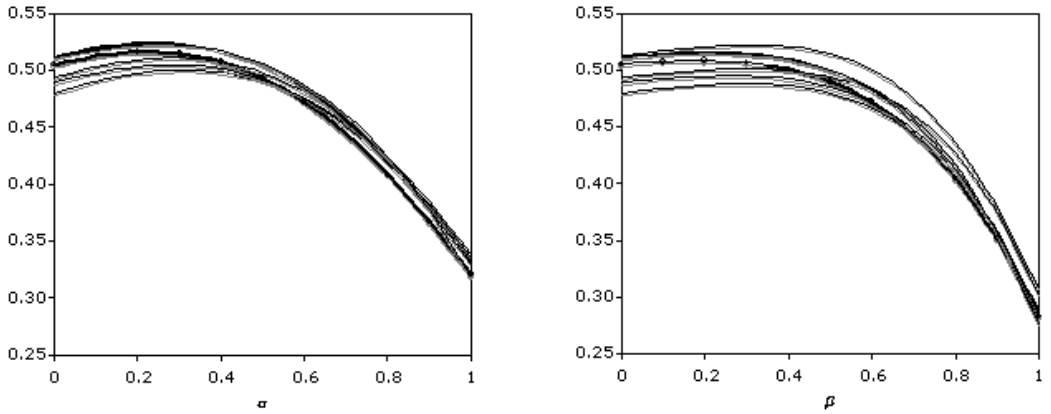

Figure 3.4: Average similarity plotted against α (with $\beta = 0$) and β (with $\alpha = 0$) respectively

Cosine	Indexer						Overall
	1	2	3	4	5	6	similarity
1		49	43	50	45	48	54.6
2	49		53	41	42	41	48.3
3	43	53		43	36	40	47.5
4	50	41	43		45	37	53.1
5	45	42	36	45		47	53.0
6	48	41	40	37	47		49.0

Table 3.9: Inter-indexer consistency and overall consistency with all other indexers according to the cosine measure with adjusted similarity coefficients (%).

keyphrase is the same in both sets it counts with a weight of 65%; if a keyphrase in one set is RT with a keyphrase in the other it counts with a weight of 20%; if it is BT/NT with the other keyphrase it counts with a weight of 15%. If it is not related to any keyphrase in the other set, its weight is 0. This simple and intuitive interpretation of weights demonstrates the advantage of working with the vector space model.

Table 3.9 shows the consistency of each indexer's selections with those of the other indexers, over all test documents, according to the new similarity measure with settings $\alpha = 0.20$ and $\beta = 0.15$. As expected, virtually all figures increase over the original version with $\alpha = \beta = 0$ (the only exception is Indexer 1's overall similarity to all other indexers, which deceases very slightly (by 0.1%). The overall ranking of indexers based on consistency is the same as before. In Table 3.7, Rollin's measure produced the ranking 6, 3, 5, 4, 2, 1 and the overall cosine measure was very similar (6, 3, 2, 5, 4, 1), while Table 3.9 shows exactly the latter ranking.

The coefficients reflect the importance of different thesaurus relations, and once suitable values have been determined from a small set of multiple-indexed documents they can be used

to evaluate the consistency of any pair of keyphrase sets. Beside the standard evaluation techniques, the inter-indexing consistency with Rolling's and cosine measures and the determined similarity coefficients are applied in Chapter 5 to estimate KEA++'s performance.

Chapter 4

KEA++: Keyphrase Extraction Algorithm for controlled indexing

This chapter presents the new algorithm for automatic indexing with keyphrases. It is called KEA++, because it improves the original keyphrase extraction algorithm KEA, described in Section 2.1.1 (Witten et al. 1999). KEA++ combines the positive elements of keyphrase extraction and term assignment into a single scheme. It is based on machine learning and works in two main stages: *candidate identification*, where candidate terms that appear in the document verbatim, or are related to the document's content, are identified, and *filtering*, which uses a learned model to identify the most significant keyphrases based on certain properties or *features*.

The candidate identification process, which is normally reduced to extraction of terms that appear in the document (Section 4.1), can be extended by including related terms from the thesaurus into the candidate list (Section 4.2). This stage includes the crucial conversion from free indexing to indexing with a controlled vocabulary.

After all controlled terms that are related to the document's content are extracted, they need to be filtered to determine the keyphrases among them. The filtering operation involves two phases: learning a suitable model based on training data, and applying it to new test documents. Section 4.3 defines features that are used to characterize the candidate phrases. The next section explains how the model is learned based on the document collection with manually assigned keyphrases. This model is then applied to unseen test documents, where keyphrases are unknown to the system.

4.1 Identifying candidates and term conflation

KEA++ retains mostly the same phrase identification strategy as implemented in the KEA keyphrase extraction system (Frank et al. (1999), Witten et al. (1999), cf. Section 2.1.1). Each document in the collection is segmented into individual tokens on the basis of white space and punctuation. Lexical clues to syntactic boundaries such as punctuation marks, digits and paragraph separators are retained. Words that contain numeral characters are discarded, since none of the descriptors in Agrovoc contain digits.

Next, all concatenations of one, two and three words – that is, word n-grams – that do not cross phrase boundaries are extracted, and the number of occurrences of each n-gram in the document is counted. The n-grams are restricted to seven words because that is the maximum length of index terms in the controlled vocabulary. Figure 4.1 illustrates the proceeding on a typical sentence from one of the documents in the FAO collection.

Although n-grams that end or start with a stopword are ignored beforehand, as in KEA, many of n-gram based phrases are still ungrammatical or meaningless in isolation. Simple keyphrase extraction algorithms like KEA and GenEx, that are based on pure n-gram extraction and rely exclusively on frequency statistics, run the risk of including such sequences into candidate phrase sets, resulting in ill-formed keyphrases. Normalization and selection of identified n-grams is essential to reduce such errors. Hulth (2004) explored two normalization techniques, including syntactic pre-processing of a document's text. After each word in a document is tagged with its grammatical word category (Part-of-Speech tag), the first alternative is to restrict candidates only to those with pre-defined PoS sequences. This technique would, for example, exclude phrases such as *United* or *reviews the recent*, because PoS Patterns *Adjective* or *Verb-Article-Adjective* are invalid, i.e. they do not appear in manually assigned keyphrases. Another way to exclude ungrammatical phrases is to apply a shallow parsing on tagged document, as done in Barker and Cornacchia (2000), Paice and Black (2003), Hulth (2004). An NP-chunker extracts only those candidates that are noun phrases consisting of one noun and zero or more pre-modifying words.

There are several problems with this kind of syntactical pre-processing. First of all, they require additional computational time and need to be applied on every new document in the collection. A keyphrase extraction system also becomes clumsy due to the additionally integrated tagging and parsing modules. Furthermore, PoS tagging and parsing algorithms are error-prone, which influences the indexing quality. But the main problem with syntax based normalization is that it cannot eliminate meaningless or unrelated candidates. For example, a

Sentence:
"As part of a regional review of aquaculture being undertaken by the Inland Water Resources and Aquaculture Service, through the Regional Office for Africa (Accra, Ghana) of the Food and Agriculture Organization of the United Nations (FAO), this document reviews the recent history of aquaculture extension in five representative countries of sub-Saharan Africa."

Sequences:
1. "As part of a regional review of aquaculture being undertaken by the Inland Water Resources and Aquaculture Service"
2. "through the Regional Office for Africa"
3. "Accra"
4. "Ghana"
5. "of the Food and Agriculture Organization of the United Nations"
6. "FAO"
7. "this document reviews the recent history of aquaculture extension in five representative countries of sub-Saharan Africa"

N-grams in Sequence 5: "of the Food and Agriculture Organization of the United Nations"
Food
Agriculture
Food and Agriculture
Organization
Agriculture Organization
Food and Agriculture Organization
United
Organization of the United
Agriculture Organization of the United
Food and Agriculture Organization of the United
Nations
United Nations
Organization of the United Nations
Agriculture Organization of the United Nations
Food and Agriculture Organization of the United Nations

Figure 4.1: Example for n-gram selection strategy in KEA++

grammatically correct phrase *recent history* would pass both tagging and parsing, but is not relevant for the given sentence. KEA++ selects n-grams with reference to a controlled vocabulary, so that most of these problems are avoided.

To achieve the best possible matching and also to attain a high degree of conflation, each n-gram is transformed into a *pseudo-phrase* according to the technique proposed in Paice and Black (2003), cf. Section 2.1.2. In the next step each pseudo-phrase is matched against vocabulary terms, also represented as pseudo-phrases. If they are identical, the n-gram is identified with the corresponding vocabulary term. Each vocabulary term receives an occurrence count which is the sum of the occurrence counts of the n-grams that were mapped to this descriptor. E.g. if the n-gram *development in the agriculture* appears twice in the document and the n-gram *agriculturally developed* three times, the corresponding descriptor *agricultural development* receives the count five.

The following step is semantic conflation that was underlined by many authors as a desired feature for index term extraction (Jacquemin and Tzoukermann (1999), Paice and Black (2003)). Other systems use either manually pre-defined rules (Hlava and Heinebach 1996) or constrains automatically extracted from a thesaurus (Jacquemin and Tzoukermann 1999). I adopt the latter idea and use the encoded semantic similarity relation in the controlled vocabulary for this purpose. The strongest similarity relation in Agrovoc is implemented as the USE link between a descriptor and a non-descriptor to prevent indexing of the same concept by distinct phrases. If a non-descriptor was identified in text, it is replaced it by its equivalent descriptor, e.g. *Food and Agriculture Organization* by *FAO* in the Table 4.1, Agrovoc ID 10960. The occurrence counter of the corresponding descriptor is increased by the sum of the counts of all its associated non-descriptors. This operation recognizes terms whose meaning is equivalent, and greatly extends the usual approach of conflation based on word-stem matching. The result is a set of candidate keyphrases for a document, and their occurrence counts.

Table 4.1 shows candidate keyphrases and their counts that were identified in the example sentence in Figure 4.1. Two terms that do not appear in the document were included into the candidate list: *Irrigation* (descriptor for *Water*) and *UN* (descriptor for *United Nations*). The count of the descriptor *FAO* was increased by the count of the non-descriptor *Food and Agriculture Organization*. Because of a stemming mistake, the false descriptor was identified: *Organic Agriculture*.[1]

[1]Porter Stemmer stems the British spelling *organisation* correctly to *organis*, but it fails to stem the American spelling correctly, while the Lovins stemmer stems both words to the same *organ*.

Agrovoc ID	Agrovoc term	pseudo-phrase	count
165	Africa	africa	2
203	Agriculture	agricultur	1
550	Aquaculture	aquacultur	3
2762	extension	extens	1
2791	FAO	fao	2
3032	Food	food	1
3253	Ghana	ghana	1
3635	History	histori	1
3876	Inland Water	inland water	1
3954	Irrigation	irrig	1
8069	UN	un	1
8332	Water	water	use 3954
8325	Water Resources	resourc water	1
10960	Food and Agriculture Organization	agricultur food organ	use 2791
15078	United Nations	nate unit	use 8069
15911	Organic Agriculture	agricultur organ	1
25804	Service	service	1

Table 4.1: Candidates extracted from example sentence in Figure 4.1.

4.2 Extending candidate selection

As an optional extension, the candidate set is enriched with all terms that are related to the candidate terms, even though they may not correspond to pseudo-phrases that appear in the document. Each candidate's one-path related terms, i.e. hierarchical neighbors (BT and NT in Agrovoc) and associatively related terms (RT), are included. For example, the descriptor *West Africa* is included, since this is NT for the candidate term *Ghana*.

If a term is related to an existing candidate, its occurrence count is increased by that candidate's count. For example, suppose a term appears in the document 10 times and is one-path related to 6 terms that appear once each in the document and to 5 that do not appear at all. Then its final frequency is 16, the frequency of the other terms that occur is 11 (since the relations are bidirectional), and the frequency of each non-occurring term is 10.

This technique helps to cover the entire semantic scope of the document, and boosts the frequency of the original candidate phrases based on their relations to other candidates. Including terms linked to the current one is similar to the spreading activation idea (cf. Section 2.4), where neighbor nodes of a currently processed concept are activated in semantic memory for quicker access. In our case only one-path neighbors are considered. Other terms (e.g. siblings, connected by BT-link to the same parent node or other two-path related terms[2]) are not

[2]Two-path means that terms that are related to related terms by the same or inverse links are included. E.g. if a term A appears in the document, and A is RT of B and C is RT of B, both C and B are included into the candidate list. Inverse relation is used to include siblings: E.g. if A is NT of B and B is BT of C, C is A's sibling (BNT).

included into candidate list due to their low semantic relatedness to a current concept, and also due to the rapidly increasing number of the candidate set and the instantaneous worsening of the overall quality of the resulting candidate set (cf. Section 5.1.1 for the evaluation of the candidate selection technique implemented in KEA++).

In both cases – with and without related terms – the resulting candidate descriptors are all grammatical terms that relate to the document's content, and each has an occurrence count. The next step is to identify a subset containing the most important of these candidates.

4.3 Feature definition

A simple and robust machine-learning scheme is used to determine the final set of keyphrases for a document. It uses as input a set of attributes, or *features*, defined for each candidate term. If the distribution of values for a feature varies significantly for positive and negative example candidates, these features will be probably useful in filtering candidates in unknown documents. In this project the following features were investigated:

The ***TF*** $\times$ ***IDF*** score (cf. Section 1.4) 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 *document frequency* – the number of documents containing the phrase in the document collection. KEA++ creates a document frequency file that stores each candidate keyphrase and a count of the number of documents in which it appears. With this file in hand, the $TF \times IDF$ for phrase P in document D is:

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

where $freq(P, D)$ is the number of times P occurs in D, $size(D)$ is the number of words in D, $df(P)$ is the number of documents containing P in the global corpus, and N is the size of the global corpus.

The second term in the equation is the log of the probability that this phrase appears in any document of the corpus (negated because the probability is less than one). This score is high for rarer phrases that are more likely to be significant.

The **position of the first occurrence** of a term is calculated as the distance of a phrase from the beginning of a document in words, normalized by the total number of words in the document. The result represents the proportion of the document preceding the phrase's first appearance.

Candidates that have very high or very low values for this feature are more likely to be valid keyphrases, because they appear either in the opening document parts such as a title, abstract, table of contents, and introduction, or in its final sections such as conclusion and reference lists. These are usually document parts that are examined by professional human indexers in order to assign keyphrases to long documents without having to read them completely.

The **length** of a candidate phrase in words is another feature in KEA++. The statistical analysis of the experimental data revealed that indexers prefer to assign descriptors consisting of two words, whereas one word terms are the majority in Agrovoc (cf. Table 3.6 in Section 3.5.1). Using phrase length in words as a feature boosts the probability of two-word candidates being keyphrases.

The **node degree** reflects how richly the term is connected in the thesaurus graph structure. The *degree* of a thesaurus 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. This feature can be calculated in three different ways:

- as the number of links that connect the term to other thesaurus terms,
- as the number of links that connect the term to other candidate phrases,
- as the ratio of the two.

Preliminary experiments have shown that the second variant demonstrates better performance than the others. This corresponds to our observation on keyphrases assigned by the six independent indexers in the second document collection (cf. Section 3.5.1). The most keyphrases that were selected by more than one indexer, which is an indicator of their relevancy (Zunde and Dexter 1969), are related to other chosen keyphrases (see Figure 3.2 in Section 3.5.2). Therefore, those candidate nodes in the activated subnet of the Agrovoc thesaurus that are connected to the most others candidate terms are more probably relevant keyphrases.

Appearance is a binary attribute that reflects whether the pseudo-phrase corresponding to a term actually appears in the document. Using the optional extension of candidate terms mentioned above, some candidate terms may not appear in the document. This feature boosts the probability of those terms that appear in the document to be better keyphrases compared to those that are only related to document's content but were not mentioned in its actual text.

Of these features, four ($TF \times IDF$ score, position of first occurrence, length in words, and node degree) are numeric, and the KEA++ algorithm converts them into nominal form for use by the machine-learning scheme. This process is called *discretization* and implies automatic detection of numeric ranges for each feature from the training data. A discretization table that is derived for each feature from the training set gives a set of numeric ranges for each feature,

whose values are replaced by the range into which they fall. In the test stage same ranges are used to replace numeric feature values of candidates that need to be filtered. The first part in the Table 4.2 shows the discretization boundaries for each feature in the final version of KEA++. E.g. the discretized values for the *Length* feature fall into two ranges: for keyphrases that consist of one term and for those that are phrases of two or more words. Discretization is accomplished using the supervised method of Fayyad and Irani (1993).

4.4 Building the model

In order to build the model, a training set with documents for which the author's keyphrases are known is required. For each training document, candidate pseudo-phrases are identified and their feature values are calculated as described above. Optionally the size of the training set can be reduced by discarding those pseudo-phrases that occur only once in the document. Each phrase is then marked as an index term or not, using the actual index terms that have been assigned to that document by a professional indexer. This binary feature is the class feature used by the machine-learning scheme.

The prediction model is constructed automatically from these training instances with the data mining software WEKA.[3] WEKA applies a machine learning scheme to learn two sets of numeric weights from the discretized feature values, one set applying to positive ("is an index term") and the other to negative ("not an index term") instances, which is specified in the class feature. KEA++ uses the Naïve Bayes technique (e.g. Domingos and Pazzani (1997)) because it is simple and yields good results.

A sample model, build on the basis of 200 documents of the main collection, is shown in Table 4.2. It consists of the discretization table, mentioned in the previous section. The second component of the model is a set of feature weights learned for positive and negative examples in the training data, shown in the middle part of the Table 4.2. For example, $P_{1st\ occur}[2|yes]$ is the proportion of positive examples that have a discretized *First Occurrence* value of 2: 14% of positive instances have First Occurrence values ranging from 0.0020 to 0.0086.

The final component of the learned model is the number of positive and negative examples in the training data, cf. the bottom of Table 4.2. These are the prior probabilities of a candidate phrase being a keyphrase, in the absence of any other information.

[3]WEKA is an open-source collection of machine learning algorithm for data mining tasks, developed at the Machine Learning Lab of the Waikato University and available from http://www.cs.waikato.ac.nz/ml/weka/.

Discretization table

Feature	Discretization ranges					
	1	2	3	4	5	6
$TF \times IDF$	≤ 0.0001	(0.0001,0.0003]	(0.0003,0.0008]	(0.0008,0.0027]	(0.0027,0.01]	> 0.01
1st occurrence	≤ 0.0020	(0.0020,0.0086]	(0.0086,0.0443]	(0.0443,0.256]	> 0.256	
Node degree	≤ 1	(1,2]	(2,3]	> 3		
Length	≤ 1	> 1				

Class probabilities

Feature	Values	Discretization ranges					
		1	2	3	4	5	6
$TF \times IDF$	$P[TF \times IDF\|yes]$	0.0056	0.1063	0.1944	0.2615	0.2657	0.1664
	$P[TF \times IDF\|no]$	0.1348	0.4105	0.2538	0.1522	0.0426	0.0062
1st occurence	$P[1st\ occur\|yes]$	0.2661	0.1442	0.2675	0.2353	0.0868	
	$P[1st\ occur\|no]$	0.0241	0.0412	0.1513	0.3923	0.3910	
Node degree	$P[node\ degree\|yes]$	0.1542	0.2539	0.2034	0.3885		
	$P[node\ degree\|no]$	0.4261	0.2959	0.1454	0.1325		
Length	$P[length\|yes]$	0.4459	0.5541				
	$P[length\|no]$	0.7175	0.2825				

Prior probabilities

Class	Training instances	Prior probability
yes	710	P(yes) = Y/(Y+N) = 0.01399
no	50009	P(no) = N/(Y+N) = 0.98601

Table 4.2: Output of the classifier in KEA++, generated by WEKA.

4.5 Computing feature values

To select keyphrases from a new document, KEA++ determines candidate pseudo-phrases and their feature values, and then applies the model built during training. The class feature is unfilled. The idea is to determine the overall probability that each candidate is a keyphrase according to the model, and to set the value of the class feature to "positive" for those candidates that obtained the highest probability values.

Suppose just the two features $TF \times IDF$ (t) and position of first occurrence (f)are being applied. For the Naïve Bayes model I first determine feature values t and f, respectively, for each candidate phrase, and then compute two quantities:

$$P[yes] = \frac{Y}{Y+N} P_{TF \times IDF}[t|yes] P_{distance}[f|yes]$$

A similar expression is used for $P[no]$, where Y is the number of positive instances in the training files – that is, author-identified keyphrases – and N is the number of negative instances – that is, candidate phrases that are not keyphrases. The overall probability that the candidate phrase is a keyphrase can then be calculated:

$$p = P[yes]/(P[yes] + P[no])$$

Candidate phrases are ranked according to this value, and two post-process steps are carried out. First, $TF \times IDF$ (in its pre-discretized form) is used as a tiebreaker if two phrases have equal probability (common because of the discretization). Second, any phrase that is a subphrase of a higher-ranking phrase is removed from the list. From the remaining ranked list, the first r phrases are returned, where r is the number of keyphrases requested. KEA++ can be used for both automatic and semi-automatic indexing, with a smaller value in the first case (e.g. r = 5) and a longer list of ranked keyphrases, from which a human indexers selects the most appropriate ones.

Chapter 5

Evaluation and analysis

This chapter presents the results for the candidate identification and filtering techniques implemented in KEA++ according to the evaluation strategy defined in Chapter 3. Two document collections with 200 (first collection) and 10 (second collection) documents, each indexed manually with terms from the Agrovoc thesaurus, serve as the basis for the evaluation.

The first collection, described in Section 3.1, is used for automatic evaluation via 10-fold cross-validation (cf. Section 1.4). This is a fast way to measure the overall performance of different parameter settings and the effectiveness of individual features. The results presented in Section 5.1 conform to standard evaluation and can be compared with those reported for other keyphrase extraction systems. Each document in the second collection has keyphrases assigned independently by six professional indexers, cf. Section 3.5. Section 5.2 evaluates automatically extracted keyphrase sets against those assigned by the indexers to determine the consistency of the algorithm with humans, and to compare it with the consistency of humans among each other. Both sections compare the original keyphrase extraction algorithm KEA with the new version KEA++ that is enriched by the thesaurus-based indexing strategy and additional features described in Section 4.3.

KEA++'s keyphrase sets assigned to the second document collection were also analyzed manually, and the results are presented in Section 5.3. This section summarizes the reasons for deficiencies in KEA++'s keyphrases and proposes further optimizations – some of which are implemented, and evaluated in Section 5.4. Also, it demonstrates several example keyphrase sets assigned to the documents of the second collection.

		# all candidates	# correct candidates	Precision	Recall
fulltexts	KEA	5766.8	3.98	0.14%	76.1%
	KEA++	407.6	3.96	1.34%	75.35%
	Extended KEA++	1765	**4.99**	0.37%	**92.99**%
abstracts	KEA	51.12	0.64	1.34%	12.03%
	KEA++	15.85	1.09	**8.80**%	22.56%
	Extended KEA++	118.62	2.27	2.38%	43.64%

Table 5.1: Performance of candidate identification in KEA and KEA++

5.1 Automatic evaluation

KEA++ works in two main steps: candidate identification and filtering. The filtering process uses the learned model to select keyphrases from the candidates, and cannot compensate for any deficiencies in the first step. This section evaluates the candidate identification step first and then presents the results obtained by the entire system.

Candidates extracted from the collection of 200 documents are evaluated against manually assigned keyphrases using precision and recall (cf. Section 3.3.1), for each document individually, and then averaged over all documents. To evaluate the overall system precision, recall and F-measure are averaged over the 20 documents in each of the 10 test sets used in the cross-validation, and further averaged over the ten runs. Both automatically and manually assigned keyphrases are sorted into pseudo-phrase form, and stemmed with the Porter (1980) stemmer, before being compared with each other.[1]

5.1.1 Evaluation of candidate identification

As explained in Sections 4.1 and 4.2, KEA++ matches n-grams of a predefined length to index terms in the Agrovoc thesaurus, conflating them with the pseudo-phrase method combined with either Porter (1980) or iterated Lovins (1968) stemmers. Tests using the original algorithm KEA and the new KEA++ revealed no significant difference between the two different stemmers, although Lovins performs slightly better for KEA and Porter for KEA++ at the 5% level according to a one-tailed paired t-test ($p > 0.15$). For both systems the pseudo-phrase technique was advantageous.

Table 5.1 compares the candidate sets with (KEA++) and without (KEA) using a controlled vocabulary, and after extending the set with terms that are related to the candidates. Results are given for both fulltexts and abstracts. The first column (all candidates) gives the average

[1]Terms are normalized using Agrovoc, so it is unnecessary to use pseudo-phrases here. I do so to ensure fair comparison with the KEA system, which does not use a controlled vocabulary.

number of phrases extracted and conflated. The second presents the average number of manually assigned keyphrases contained in the set, out of 5.4 possible phrases on average. In each scenario the best stemmer, as described above, is chosen. The numbers in bold give the best results in each column.

The analysis of candidates extracted from fulltexts shows that KEA extracts over 14 times as many candidates as KEA++ (unextended version). However, the number of "correct" terms is almost the same in both cases. Thus KEA++'s precision is almost a tenfold improvement on KEA's, although its recall is slightly lower. The extended version of KEA++, which includes one-path related terms, increased recall dramatically, at the cost of some precision – although the precision of this technique is still over twice as great as KEA's.

The discrepancies between the results obtained from these three systems are even greater when candidate phrases are extracted from the abstracts compared to those from fulltexts. This time recall and precision of both KEA++ versions are considerably higher than KEA's. Indeed (unextended) KEA++'s precision is over 6 times greater than KEA's. Because the average length of abstracts in the FAO collection is small, only a few authors' keyphrases can be identified in the text. Including semantically related terms into the candidate list improves this shortage, so that their number is twice as high than before extending.

Using a controlled vocabulary increases the precision of the candidate identification spectacularly, while the recall does not decrease or even increases, when extracting candidates from abstracts. Furthermore, good index terms do not necessarily appear in the document and using related terms from the thesaurus improves recall and increases the probability of extracting more relevant keyphrases in the filtering step. The extended candidate selection seems to be especially beneficial for keyphrase extraction from abstracts. The recall values give a baseline for the best possible results that could possibly be achieved in the filtering step. In other words, the results presented in this section are the baseline for the following evaluation.

5.1.2 Evaluation of the filtering technique

The performance of the filtering process corresponds to the overall performance of the system, since the final keyphrase set is specified in this step. This section evaluates the filtering techniques implemented in KEA and KEA++, when different features were activated. All results are obtained via 10-fold cross-validation (Section 1.4).

KEA vs. KEA++

In the first scenario, only $TF \times IDF$ and the position of the first occurrence are included as features for building the model and filtering candidates from test documents. These features are

	Level I			Level II			Level III		
	P	R	F	P	R	F	P	R	F
KEA	13.3	12.4	12.0	17.9	16.0	15.4	21.9	20.0	19.5
KEA++	**20.5**	**19.7**	**18.7**	31.0	28.1	27.8	45.6	41.0	40.2
Extended KEA++	13.1	12.1	11.8	**34.3**	**31.4**	**30.6**	**53.0**	**47.9**	**46.9**

Table 5.2: KEA vs. KEA++, $TF \times IDF$ and the first occurrence as features, fulltexts (%)

implemented in the original KEA algorithm. The minimum occurrence frequency was set to 2, and the 5 highest-scoring phrases were selected as the automatic keyphrase sets in both KEA and KEA++, as this is the average size of the manually assigned keyphrase sets. Table 5.2 summarizes the precision (P), recall (R), and F-measure (F) values that were achieved with these settings by KEA (row 1), KEA++ (row 2) and KEA++ with extended candidate selection (row 3). These results demonstrate the advantage of using controlled vocabulary for index term extraction in KEA++, compared to the pure extraction used by the original system.

At Level I, where thesaurus links are not taken into account in the evaluation, there is no advantage in artificially augmenting the candidate set with related terms as the extended version of KEA++ does, and this is confirmed by the evaluation. The main result is that the basic KEA++ roundly outperforms the original KEA, achieving level of recall, precision, and F-measure that are all over 1.5 times as high. Table 5.2 gives the overall recall figures for the candidate identification and filtering processes together: the recall of the filtering technique alone is obtained by expressing the values in the table in terms of the baseline determined in Section 5.1.1. The revised figures are a recall of 17% of the baseline for KEA and 26% for KEA++, while the extended version of KEA++ fares even worse than KEA (13%) in terms of its improvement over baseline recall.

Levels II and III take account of the conceptual similarity between keyphrase sets. Here the improvement of KEA++ over KEA is even more significant, and the extended version of KEA++ yields an even greater improvement. At Level II, where the one-path relations between automatic and manual keyphrases are considered, KEA++'s precision, recall, and F-measure values are over 1.7 as high as those of KEA, and for the extended version the figures are almost double those of KEA. At Level III, which takes into account more distant relationships between terms, the improvement is even greater, and the values achieved by KEA++ on Level III for all three performance indicators are almost 2.5 times those for KEA. In other words, only one out of five phrases extracted by KEA is related to any manually assigned index term by a two-path link, while around half of KEA++'s terms are closely related to ones assigned by professional indexers.

Adding further features

The previous experiments demonstrate the advantage of using a controlled vocabulary for automatic keyphrase extraction. The following presents the evaluation of KEA++ after adding each of the further features defined in Section 4.3 separately and together, which helps to gain further improvements of the extraction technique.

Tables 5.3 and 5.4 describe the results of adding the node degree, the length of keyphrases in words, and the appearance of a term in the document, both as individual features and in combination with each other. The first lines in both tables repeat the results from the Table 5.2 and serve as starting points for the evaluation of the new features. The best values in each column are shown in bold.

Table 5.3 does not contain the appearance feature, because all candidate phrases appear in the document verbatim. For this scenario the node degree feature has higher impact on the quality of the results on each evaluation level, compared to the length feature. In terms of terminological "correctness" on Level I, both features contribute to additional 4 percentage points for the F-measure individually, and improve the performance of the system in combination by over 5 percentage points, resulting in the F-measure of almost 24%. The highest improvement of over 12 percentage points is achieved for precision on Level III. With these new features, KEA++ extracts keyphrase sets with on average three out of five phrases that are the same or semantically similar to manually assigned keyphrases.

Similar improvements are achieved for the extended candidate selection. The appearance feature, combined with the node degree, turned out to be especially useful for this scenario. This setting helped to increase the F-measure for the terminological evaluation on the Level I from 11.8% to 18.9% compared to the original features. The length feature does not improve the performance of the extended KEA++ on Level I, but it helps to extract more semantically similar keyphrases. Combined with the node degree, this feature covered over 61.3% of manually extracted keyphrases or topic areas they are related to (recall at Level III). The standard candidate selection covers only half of manually indexed topics.

The terminological consistency is probably more important than the conceptual consistency, because it is crucial for retrieval effectiveness.[2] Combined with the fact that the improvements on the Levels II and III achieved by the extended KEA++ do not exceed 10 percentage points, I conclude that the standard candidate selection performs best for automatic keyphrase

[2] According to Leonard (1975), there is a direct correlation between the agreement on terms among indexers and the agreement between searchers and indexers. Therefore, the degree of inter-indexer consistency is positively associated with retrieval effectiveness.

	Level I			Level II			Level III		
	P	R	F	P	R	F	P	R	F
Tfidf, 1st occ	20.5	19.7	18.7	31.0	28.9	27.8	45.6	41.0	40.2
+ Node degree	25.3	23.5	22.6	36.3	33.1	32.2	56.5	50.0	49.3
+ Length	25.1	23.2	22.4	34.7	31.2	30.5	51.0	45.0	44.4
+ Node degree and Length	**25.3**	**23.5**	**22.6**	**36.2**	**33.1**	**32.1**	**56.4**	**50.0**	**49.3**

Table 5.3: Evaluation of new features in KEA++, standard candidate selection, fulltexts (%)

	Level I			Level II			Level III		
	P	R	F	P	R	F	P	R	F
Tfidf, 1st occ	13.1	12.1	11.8	34.3	31.4	30.6	53.0	48.0	46.9
+ Node degree	15.3	13.6	13.5	38.2	34.0	33.7	70.4	61.1	61.2
+ Appearance	18.2	18.0	17.0	39.6	35.7	35.0	55.5	49.8	49.0
+ Node degree and Appearance	**21.4**	**19.3**	**19.0**	38.5	33.9	33.7	63.7	55.4	55.4
+ Length	11.0	11.2	10.4	**41.0**	**36.5**	**36.0**	62.0	55.5	54.7
+ Node degree and Length	15.3	13.5	13.4	38.7	34.1	33.9	**71.0**	**61.3**	**61.5**

Table 5.4: Evaluation of new features in KEA++, extended candidate selection, fulltexts (%)

extraction from fulltexts. Compared to keyphrases extracted by KEA, its figures for precision, recall and F-measure on the terminological level are twice as high.

Extracting keyphrases from abstracts

As already mentioned in the previous section, many digital document collections only contain abstracts. In an abstract only a small part of relevant index terms appears verbatim (cf. Table 5.1), which makes the task of automatic indexing especially difficult. Table 5.5 summarizes the results achieved by KEA++ with optimal settings determined above on the same 200 documents of the main collection by using their abstracts. The results were obtained with 10-fold cross-validation, where the system was trained both on abstracts and fulltexts, but only abstracts were used for testing. Only the node degree and the keyphrase length were used as additional features. The minimum occurrence of a term was set to 1.

The first row shows the performance of KEA, when keyphrases are extracted from abstracts without controlled vocabulary, with only $TF \times IDF$ and the position of the first occurrence features. KEA performs poorly and there is no difference, when it is trained on abstracts or fulltexts. The following two rows contain the evaluation of KEA++ trained abstracts, with both methods for candidate selection. The results in rows four and five were obtained after training on fulltext documents with both candidate selection scenarios respectively. The best

		Level I			Level II			Level III		
Trained on	Algorithm	P	R	F	P	R	F	P	R	F
abstr. or fullt.	KEA	7.2	7.2	6.7	8.5	8.4	7.8	11.3	11.0	10.3
abstracts	KEA++	16.6	16.5	15.4	23.6	22.7	21.6	34.4	32.8	31.3
	Extended KEA++	13.1	12.7	12.0	**26.4**	23.9	**23.4**	**45.2**	**40.6**	**39.9**
fulltexts	KEA++	**18.5**	**18.6**	**17.2**	25.3	**24.5**	23.0	36.6	35.2	33.3
	Extended KEA++	13.6	13.5	12.6	26.2	24.7	23.5	41.2	38.3	36.8

Table 5.5: Evaluation of KEA and KEA++, when keyphrases are extracted from abstracts (%)

	1 word	2 words	3 words	3 words and more	Total
Agrovoc	(45%)	(48%)	(6%)	(1%)	
Doc Rep	(34%)	(62%)	(3%)	(0%)	
Indexers (avg)	27 (30%)	59 (64%)	6 (6%)	0 (0%)	92
KEA	34 (38%)	38 (42%)	18 (20%)	0 (0%)	90
KEA++	41 (46%)	47 (52%)	1 (1%)	1 (1%)	90
Extended KEA++	42 (47%)	47 (52%)	1 (1%)	0 (0%)	90

Table 5.6: Length of keyphrases in words, in Agrovoc and assigned to both document collections by indexers and the algorithms

performance in terms of terminological "correctness" is achieved, when KEA++ is trained on fulltexts and only those terms are selected as candidates, which appear in the abstracts. However, more semantically similar keyphrases are extracted, when the system is trained on abstracts and the related terms are included. Compared to the standard selection its precision on the Level III is improved from 34.4% to 45.2%, which means that almost half of automatically assigned terms are exact or related to the "correct" ones selected by the indexers. Compared to the KEA++'s keyphrases extracted from fulltexts, these results are lower. It is notable though that controlled indexing is beneficial and returns results, which are even better than KEA's trained and tested on fulltexts.

5.2 Indexing consistency

The results presented in the previous section should be judged carefully. They are based on the assumption that keyphrases assigned manually by just one indexer are "correct" and the automatically extracted keyphrases, in an ideal case, should match them exactly.[3] However, the analysis of index terms assigned to the same documents by different indexer independently has shown that people differ in their index term selection (cf. Section 3.5), and the definition of the "correctness" of a keyphrase is subjective.

[3]Other reasons for insufficiencies in automatic evaluation are discussed in Section 3.3.3.

	RT	NT	BT	RRT	NNT	BBT	BNT	No relation
Indexers (avg)	8	3	2	11	1	1	1	9
KEA	1	0	0	3	0	0	0	77
KEA++	11	2	5	19	1	0	0	24
Extended KEA++	17	5	3	37	1	2	3	19

Table 5.7: Number of non-matching keyphrases chosen by one indexer that relate to some keyphrase chosen by another

Instead, it is reasonable to define the "gold standard" as the level of inter-indexer consistency that was reached by several professional indexers. The goal is to develop an automatic indexing method which is as consistent with a group of indexers as they are among each other. To access the performance of KEA++ from this perspective, its results are evaluated against the data set analyzed in Section 3.5, where six indexers assigned keyphrases to ten documents. The same evaluation was applied on keyphrases extracted from these documents by KEA++, after it was trained on the 200 documents from the main collection. The optimal settings estimated in the previous section were used in this experiment: pseudo-phrase based conflation with the Porter Stemmer, the minimum occurrence of a keyphrase set to 2, and the features: $TF \times IDF$, position of the first occurrence, the length of keyphrase in words and the node degree. The number of keyphrases in the final set was adjusted to 9, which is the average number of manually assigned keyphrases.

Despite the number of words per keyphrase feature, KEA++ still extracts shorter phrases on average compared to human indexers, as shown in Table 5.6. This is probably caused by the high variation in ways to express longer concepts and by difficulties of the system to identify these variants in text. Therefore, the system selects on average more general index terms, while humans are more precise.

Table 5.7 shows the distribution of relations among non-matching terms and confirms this assumption, since the average number of broader terms in the automatic assigned keyphrase sets is more than double as high as on average in human indexers' sets. This analysis also reveals that keyphrases extracted automatically with both candidate selection techniques have similar distribution of semantically related phrases among non-matching ones as human do, although the values for the extended candidate selections are considerably higher.

Table 5.8 summarizes the exact number and percentage of matching, non-matching but related and non-matching unrelated keyphrases, assigned by humans and automatically. With the standard candidate selection KEA++ selects phrases that in 47% of cases match exactly the indexers' choices. Exactly half of the non-matching phrases (24 out of 48) are related

	Totaly assigned	Exact match	Related	Unrelated
Indexers (avg)	92	66 (72%)	16 (18%)	9 (10%)
KEA	90	10 (11%)	3 (3%)	77 (86%)
KEA++	90	42 (47%)	24 (27%)	24 (27%)
Extended KEA+	90	25 (28%)	48 (53%)	19 (19%)

Table 5.8: Distribution of exact matches and related and unrelated terms among non-matching keyphrases

	vs. Indexers		vs. KEA		vs. KEA++		vs. Extended KEA++	
Indexers	Rolling	Cosine	Rolling	Cosine	Rolling	Cosine	Rolling	Cosine
1	42	47	7	11	26	28	19	19
2	39	45	8	11	25	28	15	16
3	37	43	9	11	23	27	19	19
4	37	43	6	8	30	30	13	14
5	37	43	6	9	22	24	16	16
6	36	43	4	6	19	19	15	15
avg	38	44	7	9	24	26	16	17
overall	-	51	-	13	-	35	-	25

Table 5.9: Average inter-indexer consistency on the second document collection achieved by professional human indexers, the old version KEA, and the both versions of KEA++ (with and without extended candidate selection). The consistency is measured with the Rolling's and cosine formulas (%)

by one and two path relations in Agrovoc to manually selected phrases, so that only 27% phrases are "incorrect" and 74% are terminologically or conceptually related. The extended candidate selection causes a lower number of exact matches (19 percentage points less than the unextended version), but over 80% of all assigned terms are either the same or related to human indexers' phrases. Both numbers are comparable to the results achieved by human indexers, who agree on 90% of concepts describing document's content.

The terminological consistency is computed with the standard Rolling's and the cosine measure defined in Section 3.3. The results are shown in Table 5.9. The first column "vs. Indexers" summarizes the average consistency of each of the human indexers individually with the other five. Other columns contain consistency values, obtained in the same manner, when keyphrases were extracted automatically by KEA, KEA++ and its extended version. Since the cosine measure can be used to compute both pairwise and overall consistency, where several vectors are compared to a single one in one instance, the last row in Table 5.9 presents these values as well.

The results show that in this evaluational settings KEA performs poorly, KEA++'s val-

ues are significantly lower than humans', and the extended version of KEA++ is somewhere in-between. The results of the latter are less than half as high as those obtained for humans. While the standard version of KEA++ does not achieve the same consistency levels as humans, its average values are still comparable with human performance, since they are only 14 and 18 percentage points lower than Rolling's and cosine consistency of professional indexers respectively. The overall consistency of KEA++ with all indexers is 35%, which is 16 percentage points lower than the overall consistency among humans.

5.3 Manual analysis of results

The evaluation in the previous section shows that KEA++ cannot achieve the same degree of the inter-indexer consistency with human indexers, as they do among each other. To ascertain the reasons for this failure, it is interesting to compare which keyphrases were erroneously extracted automatically by KEA++ and which were missed, despite their selection by more than one indexer. For each negative case the reasons are analyzed, e.g keyphrase's appearance in the document, in the candidate set and its feature values. The results of this manual analysis of keyphrases assigned for one of the ten documents are summarized in Table 5.10. They reveal interesting patterns and can be used to improve the system's performance.

One of the errors was easy to identify and to correct: The algorithm for KEA++ was adapted from the keyphrase extraction system KEA. As noted in Section 4.5, after computing the probabilities for each candidate, KEA removes those terms that are subphrases of higher ranked terms (e.g. the term *methods* in Table 5.10). After disabling this feature the results for 10-cross validation on 200-item collection improve from 22.6% to 23.8%.

While problems of this kind are trivial and can be fixed easily, others are caused by inadequacy in the stemming or by restrictions of the extraction process. The following subsections analyze each error source and discuss how these errors can be avoided.

5.3.1 PDF to text conversion errors

After the documents were downloaded from the FAO document repository as PDF files, they were converted into a textual form. For this, the standard tool available on Linux machines, *pdftotext*, was used. Deficiencies in this preparatory stage caused unclean input, which in turn produced erroneous keyphrases.

KEA++'s vs Indexers' keyphrases				Indexers' vs. KEA++'s keyphrases			
KEA++'s rank	automatically assigned	# of indexers	reason for inclusion	manually assigned	# of indexers	KEA++'s rank	reason for missing
1	birds	0	stemming error	bird control	5	18	
2	predators	1		aquaculture	5	5	
3	wildlife	0	frequent in proper names	predatory birds	4	0	stemming error
4	fishing operations	0	descriptor for *fishing*	noxious birds	3	0	freq 0
5	aquaculture	5		damage	3	9	
6	scares	2		fencing	3	14	
7	horse mackerel	0	descriptor for *jack*, frequent in proper names	fishery production	2	0	freq 1
8	utah	0	proper name	control methods	2	12	
9	damage	3		fish culture	2	0	stemming error
10	noise	1		scares	2	6	
11	fisheries	1		electrical installations	1	0	frequency = 0, cf. KEA++'s rank 16)
12	control methods	2		noise	1	10	
13	technology	0	proper name	north america	1	0	frequency = 1, related to *utah* (rank 8) and *america* (rank 20)
14	fencing	3		protective structures	1	0	frequency = 0
15	ropes	0	high frequency	fish ponds	1	0	frequency = 0
16	electricity	0	high frequency	pest control	1	0	frequency = 0
17	aircraft	0	descriptor for *airplane*	techniques	1	13	UF *technology* cf. rank 13
18	bird control	5		predation	1	2	stemmed to *predators* , cf. rank 2
19	vehicles	0	high frequency	aquaculture techniques	1	0	frequency = 0
20	electronics	0	high frequency	equipment	1	74	
...				monitoring	1	0	frequency = 0
				fisheries	1	11	
				methods	1	0	superphrase for *controlled methods* (rank 12)
154 candidates in total							

Table 5.10: Analysis of keyphrases assigned manually and automatically to the document on *bird predation*. *UF* means that a descriptor is used for this term, *freq* denotes the frequency of this term in the document.

Inclusion of irrelevant textual items in the document. Some structural elements, such as header, page number and page title, are not relevant to the content of the document. For example, *e-JOURNAL OF AGRICULTURAL AND DEVELOPMENT ECONOMICS, Vol. 8(1), March 2003* appears on the top of every page in the one of the PDF files, and therefore appears very frequently in the converted text document. This made KEA++ select the keyphrase *agricultural development*, although it is too general for the document in question.

Interruption of the text flow by figures, footnotes or page breaks. Phrases, and even words, are sometimes broken by what has incorrectly been identified as a paragraph break. This caused incorrect frequency counts for candidate keyphrases.

Erroneous conversion of text style. The program *pdftotext* attempts to approximate the formatting of the original document. For example, some words were converted with inter-letter spaces: *b o l d*. Such words and phrases could not be identified as valid terms by the algorithm, which influenced the candidate selection and frequency counts.

Most of these mistakes could be fixed automatically with a better conversion program that dismisses unnecessary information about text style or graphical layout, but identifies structural elements in the documents and extracts only the content bearing parts of the text such as title, abstract, headings, and sections. Because such tools are not yet available, I hand-edited the 10 documents of the second test set to remove conversion mistakes. This modification improved the inter-indexer consistency considerably: the F-measure increased by almost 3 percentage points.

5.3.2 Conflation mistakes

KEA++ uses the pseudo-phrase technique to conflate words with similar stems but different syntactical roles. This stems individual words in a phrase, removes stopwords and reorders the words alphabetically. Each of these operations is a potential error source, especially when combined with each other.

No or erroneous mapping of stemmed n-grams to Agrovoc terms. Understemming and overstemming are often the reason why n-grams are not mapped to corresponding descriptors in Agrovoc, or why they are mapped to false ones. For example, the descriptor *predatory birds* that was selected by most indexers appears in this form only once in the document. The semantically similar phrase *bird predation* is more frequent in the text, but was stemmed by the Porter stemmer to a different form. As Table 5.11 shows, the iterated Lovins stemmer

Original phrase	Porter	Iterated Lovins	Agrovoc descriptor?
bird predation	bird predat	bird pr	no
predatory birds	predatori bird	pr bird	yes

Table 5.11: Example of understemming by the Porter stemmer, and KEA++'s failure to assign the corresponding descriptor

stems these phrases to the same form. However, re-running keyphrase extraction with this stemmer did not improve results, because it produces incorrect candidates in other cases by overstemming. In fact, overstemming is a common error of both stemmers. For example, the phrases *animal communities* and *communications between animals* were erroneously conflated by both stemmers to the pseudo-phrase *anim commun*.

Incorrect stopword removal. In another document, the frequent term *meat* is mapped by mistake to the descriptor *canned meat*, because the word *canned* was stemmed to *can* and discarded as a stopword. This must be seen as a methodological error, since stopword removal should be done before the tokens are stemmed.

False conflation between non-descriptors and descriptors. In many cases non-descriptors that appear in text are mapped to false descriptors in the Agrovoc thesaurus, which produced false keyphrases. Stemming increases the number of these errors. For example, the descriptor *viverridae* (used for *genets*) was assigned, because the frequent term *genetic* stemmed to the same form as *genets*.

False conflation of Agrovoc terms. In some cases Agrovoc terms were stemmed to the same pseudo phrase, so that they were conflated with each other. This meant only one of the original descriptors was used for the automatic indexing, while the others (normally just one) were lost. For example, the Porter stemmer conflates *forestation* and *forest*, and as a result KEA++ assigned term *forestation* at rank 1, although the term *forest* is more appropriate and was selected by four indexers.

It is difficult to "correct" stemming mistakes, since the Porter stemmer used here is already the most accurate. Fuller and Zobel (1998, 12) report the conflation accuracy of this stemmer of 97% and state that it "leaves little scope for practical improvement". However, the manual analysis of top 10 keyphrases assigned automatically in this experiment revealed that in 20% of cases keyphrases were incorrect due to stemming mistakes, and in many further cases keyphrases selected by the indexers are not identified by KEA++, for the same reason.

To avoid stemming errors, several methods were explored. In many examples, a wrong stemmed non-descriptor was mapped to an unrelated descriptor. Deactivating this mapping operation prevented this error, but turned out to be disadvantageous for the overall performance of the algorithm. In some cases, keeping stopwords in the pseudo-phrase or changing the order of the stemming and stopword-removal operations prevent false mapping to a wrong index term. However, these modifications have no significant influence on the quality of the results.

The manual analysis of stemming mistakes has shown that overstemming is a more frequent source of errors than understemming. Therefore, it is reasonable to reduce stemming to a single *s*-removal operation according to four simple rules[4]:

- if a word ends with *sses*, cut off the ending *es*;
- if a word ends with *ies*, cut off the ending *es*;
- if a word ends with *s*, cut off the ending *s*;
- otherwise, no changes on the word.

In the final trial stemming was turned off completely. Only case folding, word re-ordering and stopword removal were retained as conflation strategy.

These settings were made for experiments with both document collections. Surprisingly, there was no considerable difference between the quality of the results produced with stemming, with simple s-removing and no stemming at all. In fact, keyphrase sets assigned automatically to the 10 documents contained even more related index terms than before deactivating the stemming, and the precision and recall values for the 200 assigned keyphrase sets increased by 1.5 percentage points without stemming as well.

These experiments demonstrate that stemming does not have as high impact on automatic keyphrase extraction with controlled vocabulary, as it did for the free keyphrase indexing. On the one hand, this is a disappointing finding, which proves once again that some of the NLP tasks do not require linguistic pre-processing. On the other hand, the stemmer produced too many errors, so that a more elaborated stemming algorithm could probably improve the results. Developing of better stemming procedures and their evaluation in conjunction with practical tasks is therefore desirable.

5.3.3 Proper nouns

Proper nouns or their parts are often mapped to false or irrelevant descriptors. For example, the descriptor *horse mackerel* was assigned to a document about predatory birds (cf. Table 5.10), because of the frequent appearance of the word *jack*, as part of the phrase *Jack H. Berryman*

[4]These are the rules in the first step implemented in the Porter (1980) algorithm.

Institute, the author organization of the publication. According to the Agrovoc thesaurus, *jack* is a fish species related to *horse mackerel*.

In another example, the term *wildlife* was included as descriptor to the same document, because the research in the area of bird predation was conducted by various institutes or agencies that deal with *wildlife* and have this word in their title. Related but not representative keyphrases, such as *technology* or geographical names in the titles (*Utah, Colorado*) were also assigned due to their appearance in proper names.

The open source NLP tool *OpenNLP*[5] includes a named entity recognition module that was used here to remove all identified proper names except stand-alone geographical names (tagged with $< location >< /location >$ by the tool) and re-run KEA++ on these documents. The results showed that excluding proper names does not improve the algorithm's performance. Firstly, many of the proper names do contain words related to document's content, and secondly, the quality of the named entity recognition tool is not good enough to identify all the proper names and classify them accurately, so that only those without any meaning could be removed, e.g. person's name such as *Jack*.[6]

5.3.4 Partial matching

Some descriptors that were selected by the indexers do not appear in the documents completely. For example, the keyphrase *plant genetic resources* was selected by all six indexers and the keyphrase *socioeconomic development* by four indexers, despite the total absence of these phrases in both documents. The system was unable to identify both keyphrases, because only their variants *plant genetic diversity* and *socioeconomic status* appear in the texts, and these variants are not present in the thesaurus.

To solve this problem, one could consider partial matching between terms that are longer than two or three words, in addition to the pseudo-phrase matching technique. However, because almost half of all terms in the Agrovoc thesaurus consist of two or more words, considering partial matches would decrease the quality of the candidate set. Another solution would be to improve the thesaurus by adding terms that are frequent in the given document collection, since common terms are good candidates for being descriptors, or they should be at least linked to the already used descriptors. A keyphrase extraction system (for example, KEA++'s prototype KEA) could help the developers of Agrovoc in solving this task effectively.

[5] http://opennlp.sourceforge.net/

[6] The OpenNLP Named Entity Recognition tool was trained on the Wall Street Journal corpus, which could be the reason for the poor performance on the documents from the agricultural domain.

5.3.5 Low or zero occurrence frequency

The most frequent reason why KEA++ missed many of the manually assigned descriptors is because professional indexers often select terms that do not or very seldom appear in the document. For example, three indexers agreed on descriptor *noxious birds*, although even the word *noxious* by itself does not appear in the document in any form and the frequent term *birds* is not modified by any adjectives related to it. In such cases it is impossible for the system to establish any connection between these concepts. Two-word index terms are often missed by the system, because only their heads appear frequently, whereas the modifying part is seldom or not mentioned in same context at all. While the human knows that if the term *agreements* appears in the document on *greenhose effect*, the descriptor *international agreements* is suitable, KEA++ is unable to identify this relatedness and fails to cover this topic in the keyphrase set completely, since the term *agreements* is not present in Agrovoc at all.

It is difficult to find a solution for this problem. Such examples demonstrate that automatic keyphrase indexing systems like KEA++ are still restricted to wording in the document and lack of deep understanding of the document's content, which is necessary to deduce correct index terms from the meaning of the document.

5.4 Final results

The manual analysis led to small changes in the final code of KEA++, and contributed to some additional improvements. Table 5.12 demonstrates the results after evaluating these changes on the main document collection. The first row repeats the values obtained with the final settings described in Section 5.1.2. The second row shows KEA++'s results after removing the subphrase feature and the stemming procedure. These modifications turned out to be advantageous, especially for the terminological evaluation, where F-measure on Level I increases from 22.6% to 25.2%. Now, more than a quarter of extracted keyphrases (26.0%) match exactly the choice of professional indexers and around half of them (55.7%) are highly related to manually assigned keyphrases. As indicated in the previous section these results can be further improved with a better conversion program and a more accurate term conflation algorithm, which could not be completed in the scope of this thesis.

In the second document collection the converted files were hand-edited, so that no structural elements like page header influenced the extraction process. Table 5.13 demonstrates that this and other final modifications improved KEA++'s performance considerably, since the number

	Level I			Level II			Level III		
	P	R	F	P	R	F	P	R	F
KEA++ before analysis	25.3	23.5	22.6	36.2	33.1	32.1	**56.4**	50.0	**49.3**
KEA++ after analysis	**28.3**	**26.1**	**25.2**	**39.2**	**35.6**	**34.6**	55.7	**50.1**	49.2

Table 5.12: Evaluation of last modifications on KEA++ for the main document collection (%)

	Totally assigned	Exact match	Related	Unrelated
Indexers (avg)	92	66 (72%)	16 (18%)	9 (10%)
KEA	90	10 (11%)	3 (3%)	77 (86%)
KEA++ before changes	90	42 (47%)	24 (27%)	24 (27%)
KEA+ final	90	45 (50%)	25 (27%)	20 (22%)

Table 5.13: Distribution of exact matches and related and unrelated terms among non-matching keyphrases in the final version of KEA++

of totally unrelated terms decrease from 24 to 20 out of 90. Table 5.14 compares the values for inter-indexer consistency of KEA++ before and after performed changes with each indexer and on average according to both Rolling's and cosine measure. In each case the final version of KEA++ performs slightly to considerably better than before the changes.

These final consistency values should now be compared to the inter-indexer consistency estimated in Section 3.5.2. According to the Rolling's measure the average consistency among these indexers is 38%. KEA++ achieves on average the consistency of 27%, which is only 11 percentage points lower than humans' performance. The cosine measure with adjusted similarity coefficients resulted in an average overall similarity among humans of 51%. KEA++'s correlation with the overall indexers' vector is 37%, which is 14 percentage points lower. To compute the overall cosine value for humans, the term vector of each indexer was compared to the average vector of his five colleagues and selected α and β in a way that maximizes the similarity between them. Therefore, it is perspicuous that the difference between humans' and KEA++'s performance according to the cosines measure are lower than according to the Rolling's consistency. However, values achieved by KEA++ are still remarkable high given the difficulty of the task and the performance of the state-of-the-art system KEA (difference of 31% and 38%, cf. Table 5.9).

Table 5.15 shows keyphrases that were assigned to three selected documents by at least two indexers and those that were selected by KEA++. Only 9 top ranked KEA++'s keyphrases are presented here, since this is the average number of keyphrases that are assigned by professional

		Indexer							
		1	2	3	4	5	6	average	overall
Rolling	KEA++ before changes	26	25	23	30	22	19	24	-
	KEA++ after changes	29	28	26	31	25	20	27	-
Cosine	KEA++ before changes	28	28	27	30	24	19	26	35
	KEA++ after changes	32	31	30	33	27	19	29	37

Table 5.14: Average inter-indexer consistency among the indexers and compared to KEA++ before and after last modifications, according to Rolling's and Cosine measure (%)

indexers. Most phrases (exact and non-matching but similar) make sense according to the documents' titles. A few erroneously extracted keyphrases (e.g. *viverridae* in the third document) are extracted due to the stemming mistakes and should not be seen as algorithm's failures.

Figures 5.1 and 5.2 demonstrate typical keyphrase sets that are now extracted with KEA++ in an extended form of a diagram used in Section 3.4. Terms that were identified by KEA++ (top 9 ranked candidates) are marked by gray filling. The circles around each term show the number of indexers that assigned them to both documents. While in the first document on *global obesity problem* KEA++ failed to cover the topic area of nutritional policies and requirements, in the document on *bird predation* all areas that present in manually assigned keyphrase sets are covered by the same or similar keyphrases as chosen by the professional indexers.

"Overview of Techniques for Reducing Bird Predation at Aquaculture Facilities"

	Indexer	**KEA++**
Exact	aquaculture	aquaculture
	damage	damage
	fencing	fencing
	scares	scares
	noise*	noise*
Similar	bird control	birds
	predatory birds	predators
	fish culture	fishing operations
	fishery production	
No match	noxious birds	
	control methods	
		ropes

*Selected by only one indexer.

"The Growing Global Obesity Problem: Some Policy Options to Address It"

	Indexer	**KEA++**
Exact	overweight	overweight
	food consumption	food consumption
	taxes	taxes
Similar	developed countries*	developing countries
	prices	price fixing
		controlled prices
	price policies	policies
	fiscal policies	
	nutrition policies	
	diets	body weight
No match	feeding habits	
	food intake	
	nutritional requirements	
		saturated fats

*Erroneous stemming of these phrases to the same form, so that only one form only one form could be extracted.

"Conserving Plant Genetic Diversity for Dependent Animal Communities"

	Indexer	**KEA++**
Exact	arthropoda	arthropoda
	species	species
	population genetics	population genetics
	populus*	populus
	hybridization*	hybridization
	hybrids*	hybrids
Similar	plant animal relations	plants
	plant genetic resources	
No match	animal population	
	biodiversity	
	genetic variation	
	resource conservation	
		viverridae**
		communication between animals**

* Selected by only one indexer.
** Erroneous stemming and conflation of the extracted phrases with the Agrovoc terms.

Table 5.15: Comparison of keyphrases that were selected by at least two indexers and that were among 9 top ranked KEA++ phrases in example documents

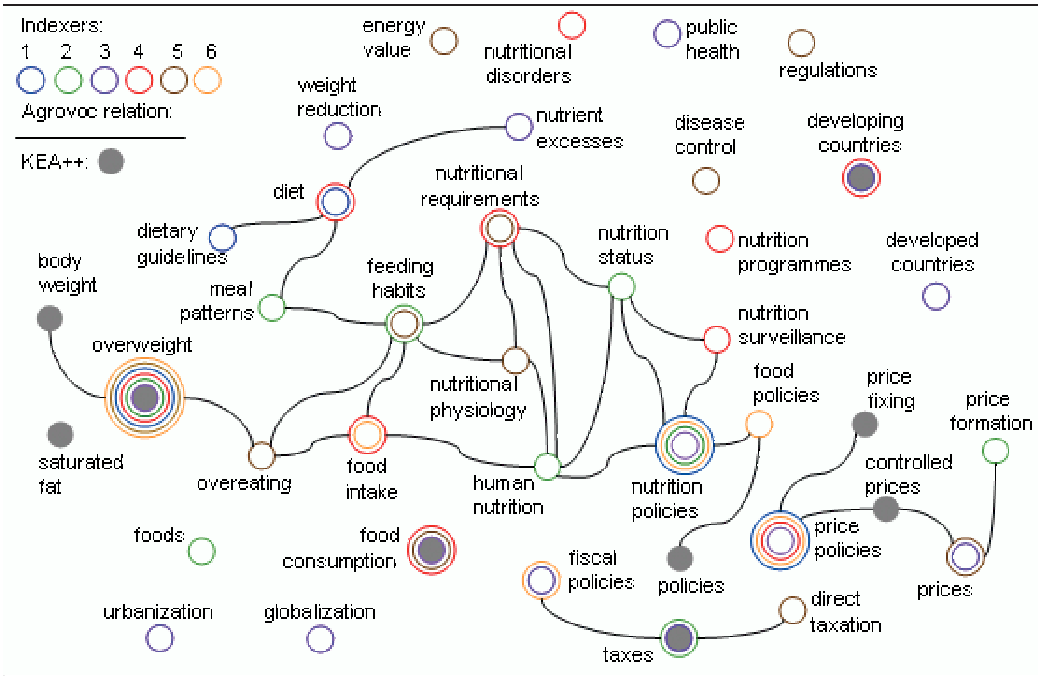

Figure 5.1: Keyphrases assigned to the document on *global obesity problem* by professional indexers and KEA++

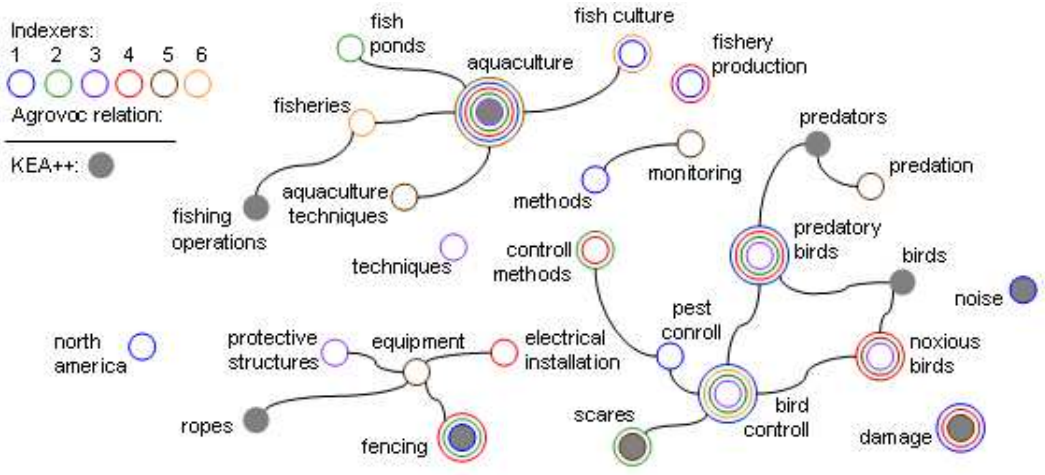

Figure 5.2: Keyphrases assigned to the document on *bird predation* by professional indexers and KEA++

Chapter 6

Conclusions

This thesis investigated the problem of controlled keyphrase extraction and presented a system for solving this task automatically. KEA++ takes into account semantic relations between terms that appear in the document, and was trained and tested on documents from the agricultural domain. The domain-specific thesaurus Agrovoc was used as the controlled vocabulary and semantic database. Evaluating keyphrase sets extracted by KEA++ against humans' keyphrases, and the manual analysis of the algorithm's results, demonstrates that the proposed technique is successful and significantly outperforms KEA, the state-of-the-art algorithm for free indexing. This chapter draws conclusions from the research, and summarizes the findings.

In Section 6.1, I discuss the hypotheses stated at the beginning of the thesis and consider the evaluation of the algorithm as proof for their validity. Section 6.2 contains ideas on how KEA++ can be applied in other domains and languages, and considers other practical utilizations of keyphrase indexing are possible. The final Section 6.3 focuses on limitations of automatic keyphrase indexing that could not be resolved in this project and remain open for further investigation.

6.1 Proof of the hypotheses

Section 1.3 set forth three hypotheses that catch the essence of the research problem of this thesis. They were formulated intuitively, based on findings of previous experiments on keyphrase indexing by humans and machines. The evaluation of KEA++ revealed that each hypothesis can be confirmed, either completely or at least partially. The analysis of the experimental results is used as the basis for this judgment. The outcome is discussed.

6.1.1 Free vs. controlled indexing

The first hypothesis stated that automatic keyphrase extraction with a controlled vocabulary outperforms free indexing. This was an intuitive guess, suggested by experiments on inter-indexer consistency of human indexers. According to Markey's (1984) review of these experiments, professional indexers are more consistent with each other when their choice of index terms is restricted by a controlled vocabulary.

The state-of-the-art system for automatic keyphrase extraction KEA was compared with its extended version KEA++ that matches all candidate keyphrases to the descriptors in the Agrovoc thesaurus. Under the same conditions such as the similar candidate extraction technique, and using the same learning scheme, KEA performs poorly compared to KEA++. The evaluation of both systems on the same document collection demonstrates that the new system achieves precision, recall, and F-measure values that are over 1.5 times greater than those obtained with free indexing. The analysis of concepts covered by keyphrase sets assigned by both systems revealed that KEA++ extracts on average twice as many keyphrases that are equal or semantically similar to those assigned by humans. KEA's relatively poor performance is mainly caused by noise in the keyphrase sets it extracts, since many of its keyphrases are malformed or unrelated phrases, whereas KEA++ only extracts accurate and precise keyphrases from the vocabulary and they all belong to the same domain as the document. Section 5.1.1 demonstrated and discussed these results in detail.

6.1.2 Semantic relations

The next hypothesis is that considering semantic relations between keyphrases improves indexing performance. Before starting experiments with KEA++ and the Agrovoc thesaurus, this statement was supported by the theory of text comprehension, where a human is assumed to understand the document by mapping the network of semantic concepts representing document's content to the global conceptual network of its world knowledge. The assumption in this thesis is that information on relations between concepts in both networks gives valuable insights into which concepts are significant for a particular document.

The spreading activation theory suggests including thesaurus terms that are related to the document's phrases in the set of candidate keyphrases. While this technique reduces the number of terminological matches in all scenarios, compared to the original candidate extraction method, it extracts more keyphrases that are semantically similar to manually assigned ones. These findings are too vague to determine conclusively whether or not this supplementary tech-

nique is advantageous for the system's overall performance.

Another way to prove the hypothesis is to analyze the new features in KEA++ based on semantic relatedness of the document's terms. This feature is defined as the node degree, and represents the number of candidate keyphrases related to the keyphrase in question. The evaluation revealed that node degree contributes to a significant improvement of the keyphrase's quality. While the terminological consistency increased from 18.7% to 22.6%, the number of conceptual matches could be increased from 45.6% to 56.5% (cf. F-measure on Level I and precision on Level III in Table 5.3, Section 5.1.2 respectively). These results confirm that the hypothesis is valid.

6.1.3 Towards improved indexing consistency

The discussion of the evaluation methods for automatic indexing algorithms led to the standard measure in the library science called inter-indexing consistency. A reasonable way to evaluate an algorithm is to compare the consistency between it and the group of professional indexers with the inter-indexer consistency among the indexers. As the later is usually quite low and KEA++ is the first system that includes semantic relations encoded in the thesaurus into automatic indexing process, the arisen hypothesis was that the consistency degrees might be same or similar.

The average consistency between six professional indexers on 10 documents in this project is 38%, when they are compared pairwise with Rolling's measure. Their overall semantically enhanced similarity according to the proposed cosine measure is 51%. The final version of KEA++ achieved 27% and 37% for both measures respectively, which is significantly lower than the consistency between humans (cf. Section 5.4). The conclusion is that the stated hypothesis fails: The algorithm performs worse than professional indexers and cannot be employed as a perfect replacement for human labour.

However, only ten documents might be insufficient for a reliable evaluation. Additionally, I conjecture that the human indexers who took part in this experiment were particularly carefully. They assigned on average 9 keyphrases per document, while the excerpt from the FAO's document repository shows that only 5 keyphrases on average are assigned normally. Unfortunately, it is expensive and difficult to collect a sufficient amount of real-world data for reliable experiments on inter-indexer consistency.

Manual analysis of the results in Section 5.3 revealed that in the majority of cases where KEA++ selected wrong keyphrases, the problem was caused by stemming errors. A stemmer is not an essential part of the system. Using a better stemming algorithm would definitely

improve the performance of KEA++, because these mistakes would be avoided. This might decrease the gap between the degrees of inter-indexer consistency that were achieved.

6.2 Possible extensions

KEA++ was trained and tested on documents in the English language, describing topics related to the agricultural domain. Language and domain independence is not a compulsory requirement for natural language processing systems. However, it is important to know how one can apply the same techniques to other languages and domains.

6.2.1 Other languages

Because the candidate extraction method does not require any NLP tools with exception of stemming (if selected), the approach presented here is to a great extend language independent. The Agrovoc thesaurus is being developed and improved constantly by the FAO. Currently, almost each Agrovoc entry contains alternatives in other languages: Spanish, French, Portuguese, German, Czech, Arabic and Chinese. Given a document in one of these languages, the extracted n-gram can be mapped to a corresponding Agrovoc descriptor in the same manner, as it was done for English. Because most Indo-European languages are inflectionally rich, stemming algorithms are essential for high term conflation. For German, the additional use of a decompounding algorithm would be advantageous, because compounds in this language are often written as one word. For example, *Schlagwortindexierung* and *Indexierung mit Schlagwrtern*, that are both equivalent versions for the English phrase *keyphrase indexing*, cannot be conflated without preliminary decompounding of the first expression.

The advantage of using the Agrovoc thesaurus in KEA++ is that only a small modification is required for cross-language indexing, when texts in English language need to be indexed with keyphrases in other languages or vise versa.

6.2.2 Domain-independence

KEA++ extracts keyphrases only from agricultural documents, because it uses the domain-specific Agrovoc thesaurus as controlled vocabulary. Agrovoc contains index terms from various related fields and has broader nets of geographic names, biological termini, various plant and animal names, but also economical and political concepts. However, when documents from other domains (e.g. computer science) need to be indexed, KEA++ fails to assign appropriate terms.

The structure of Agrovoc is simple (cf. Section 3.2), so that KEA++ can be easily adjusted to any another controlled vocabulary or thesaurus. While preferential and associative relations are commonly used in different kinds of electronic dictionaries, BT and NT relations between terms can be deduced from their hierarchical structure.

The largest electronic dictionary of English language is the WordNet database (Fellbaum 1998). It contains over 152,000 concepts of general and specific nature that are connected by several different relations to each others. Converting this thesaurus for KEA++ would require following steps:

- **Extract all nouns.** Beside nouns WordNet contains verbs, adjectives and adverbs, but only nouns can be considered as useful keyphrases.[1]
- **Extract all hierarchically related term pairs.** In WordNet these terms are called hyper- and hyponyms instead of BT and NT.
- **Conflate semantic relations.** All relations between terms other than hyper- and hyponyms can be mapped to a single associative relation RT.

After these steps all required vocabulary files for KEA++ can be generated. Similar processing is possible for any other electronic vocabulary. Because such modifications are easy to render, the approach can be seen as domain independent.

6.2.3 Other applications

Beside automatic keyphrase extraction, KEA++ can be employed as a system for semi-automatic indexing, where a professional human indexer analyzes top 20 or top 50 from ranked keyphrases identified for a document in question and then selects then the most appropriate ones among them. This saves time for the pre-scanning of documents: an indexer can immediately see, to which sub-area of the domain the documents belongs to and which keyphrases in the thesaurus describe this area. Together with document's title and abstract, an indexer can assign appropriate keyphrases easier and faster. From this perspective, KEA++ can not only help to save time required for manual indexing, but also to increase the inter-indexers consistency. To use KEA++ for semi-automatic indexing only one parameter needs to be adjusted (the number of terms in the final keyphrase set).

For other tasks, e.g. automatic document classification, further modifications are possible. To assign more general terms to documents and to assure a smaller set of index terms associated

[1] While this decision is rather intuitive, Hulth (2004) reports that only few manually assigned keyphrases in her data are not nouns (adjectives and gerund verbs).

with a document collection, KEA++ can be extended with the following feature: Each time, when a candidate term is determined as a valid thesaurus term, it should be mapped to its broader term (if available), similar as the non-descriptors are mapped to their descriptors in the current version of KEA++. This technique would reduce the number of candidate terms to most general terms in the thesaurus. Few significant terms (e.g. three or five keyphrases) in the final keyphrase set can be used as categories for the document classification.

Due to the lack of time available for this project, no empirical results on how effective KEA++ performs for these tasks can be provided. Beside extensions to other languages and vocabularies these are areas for future experiments.

6.3 Limitations of automatic indexing

A novel algorithm for automatic keyphrase extraction was presented and evaluated in this thesis. The results reveal that the performance of the system is acceptable, but it is still lower than that of human indexers. Manual analysis of the automatically extracted keyphrases explains cases where KEA++ failed to assign the same phrases as humans, or where it extracted false terms (Section 5.3).

While some of KEA++'s mistakes are due to reasons that can be overcome (e.g. messy input, erroneous stemming), term appearance in the document, which is a crucial pre-condition for automatic keyphrase extraction, can hardly be influenced by an algorithm. KEA++ will fail to relate correct Agrovoc descriptors to a document if they do not appear there verbatim, and some subjects of the document will not be covered in the assigned keyphrase set.

Covering all the document's topics is one of the most important requirements for keyphrase indexing. While the manual analysis unveils that KEA++ did not extract some of the human indexers' keyphrases or phrases related to them, it is difficult to judge to what extent the algorithm succeeds in this task, compared to human indexers. More elaborate evaluation techniques that make use of the thesaurus tree's structure would be necessary to determine the limitations of automatic indexing in a more tangible way.

The experiments conducted in this project do not consider topic coverage, since each candidate keyphrase is analyzed individually, without analyzing its connection to other phrases in the document. The main conclusion from the manual analysis is that KEA++ is not able to identify dependencies between a document's terms and concepts covered by Agrovoc if there is no exact match between word stems. The ability to understand the real meaning of terms remains solely in the domain of humans, and cannot yet be mimicked by machines.

References

AGROVOC (1995). Multilingual agricultural thesaurus. Food and Agricultural Organization of the United Nations.

Arampatzis, A. T., T. Tsoris, C. H. A. Koster, and T. P. van der Weide (1998). Phrase-based information retrieval. *Information Processing and Management 34*(6), 693–707.

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

Bourigault, D. and C. Jacquemin (1999). Term extraction + term clustering: An integrated platform for computer-aided terminology. *Proc. of the 9th Conference of the European Chapter of the Association for Computational Linguistics (EACL 99)*, 15–22.

Collins, A. and E. F. Loftus (1975). A spreading-activation theory of semantic processing. *Psychological Review 82*(6), 407–428.

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

Domingos, P. and M. Pazzani (1997). On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning 29*(2/3), 103–130.

Dumais, S. T., J. Platt, D. Heckerman, and M. Sahami (1998). Inductive learning algorithms and representations for text categorization. In *Proc. of the 7th International Conference on Information and Knowledge Management (ACM-CIKM 98)*, pp. 148–155.

Engl, D., J. Friedl, J. Labner, M. Sandnerand, W. Schlacher, A. Schmidt, and A. Zartl (1997). Schlagwort “Benutzerforschung”. Beobachtungen bei der sachlichen Suche im OPAC des österreichischen wissenschaftlichen Bibliothekenverbundes. *Mitteilungen der Vereinigung Österreichischer Bibliothekarinnen & Bibliothekare. 50*(3/4), 28–49.

Fayyad, U. and K. Irani (1993). Multi-interval discretization of continuous-valued attributes for classification learning. In *Proc. of the 13th International Joint Conference on Artificial Intelligence (IJCAI 93)*, pp. 1022–1027.

Feather, J. and P. Sturges (1996). *International Encyclopedia of Information and Library Science.* London & New York: Routledge.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press.

Frank, E., G. W. Paynter, I. H. Witten, C. Gutwin, and C. G. Nevill-Manning (1999). Domain-specific keyphrase extraction. In *Proc. of the 16th International Joint Conference on Artificial Intelligence (IJCAI 99)*, pp. 668–673. San Francisco, CA: Morgan Kaufmann.

Fuller, M. and J. Zobel (1998). Conflation-based comparison of stemming algorithms. In *Proc. of the 3rd Australian Document Computing Symposium*, pp. 8–13.

Gutwin, C., G. Paynter, I. Witten, C. Nevill-Manning, and E. Frank (1998). Improving browsing in digital libraries with keyphrase indexes. Technical report, Department of Computer Science, University of Saskatchewan, Canada.

Hilberer, T. (2003). Aufwand vs. Nutzen: Wie sollen deutsche wissenschaftliche Bibliotheken künftig katalogisieren? *Bibliotheksdienst 37. Jg.*, 754–758.

Hlava, M. M. K. and R. Heinebach (1996). Machine aided indexing. European parliament study and results. In *Proc. of the 17th National Online Meeting*, pp. 137–158. Medford, NJ: Information Today.

Hooper, R. S. (1965). Indexer consistency tests–origin, measurements, results and utilization. Technical report, IBM Corp.

Hulth, A. (2004). *Combining Machine Learning and Natural Language Processing for Automatic Keyword Extraction.* Ph. D. thesis, Computer and Systems Sciences, Stockholm University.

Iivonen, M. (1995). Consistency in the selection of search concepts and search terms. *Information Processing and Management 31*(2), 173–190.

IMARK (2005). Module for ”digitalization & digital libraries”. UNESCO, FAO and the National Centre for Science Information (NCSI) at the Indian Institute of Science (IISC). CD-ROM.

Jacquemin, C. and D. Bourigault (2003). Term extraction and automatic indexing. In R. Mitkov (Ed.), *The Oxford Handbook of Computational Linguistics*, pp. 559–616. Oxford: Unviversity Press.

Jacquemin, C. and E. Tzoukermann (1999). NLP for term variant extraction: Synergy between morphology, lexicon and syntax. In T. Strzalkowski (Ed.), *Natural Language Information Retrieval*, pp. 25–74. Dordrecht: Kluwer Academy Publishers.

James, D. (1996). Organization of knowledge. In J. Feather and P. Sturges (Eds.), *International Encyclopedia of Information and Library Science*, pp. 336–353. London & New York: Routledge.

Joachims, T. (1998). Text categorization with support vector machines: learning with many relevant features. In *Proc. of the 10th European Conference on Machine Learning (ECML 98)*, pp. 137–142.

Jones, S. and M. Mahoui (2000). Hierarchical document clustering using automatically extracted keyphrases. In *Proc. of the 3rd International Asian Conference on Digital Libraries*, pp. 113–120.

Jones, S. and G. Paynter (2002). Automatic extraction of document keyprases for use in digital libraries: evaluation and applications. *Journal of the American Society for Information Science and Technology (JASIST) 53*(8), 653–677.

Jones, S. and G. W. Paynter (2003). An evaluation of document keyphrase sets. *Journal of Digital Information. 4*(1).

Kintsch, W. and T. van Dijk (1978). Toward a model of text comprehension and production. *Psychological Review 85*(5), 363–394.

Leacock, C. and M. Chodorow (1998). Combining local context with wordnet similarity for word sense identification. In C. Fellbaum (Ed.), *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press.

Leininger, K. (2000). Interindexer consistency in PsycINFO. *Journal of Librarianship and Information Science 32*(1), 4–8.

Leonard, L. E. (1975). *Inter-indexer consistency and retrieval effectiveness: measurement of relationships*. Ph. D. thesis, Graduate School of Library Science, University of Illinois, Urbana-Champaign, IL.

Lewis, D. D. (1992). An evaluation of phrasal and clustered representations on a text categorization task. In *Proc. of the 15th Annual International Conference on Research and Development in Information Retrieval (ACM SIGIR 92)*, pp. 37–50.

Lovins, J. B. (1968). Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics 11*(1-2), 11–31.

Markey, K. (1984). Inter-indexer consistency test: A literature review and report of a test of consistency in indexing visual materials. *Library and Information Science Research 6*, 157–177.

Markó, K., P. Daumke, S. Schulz, and U. Hahn (2003). Cross-language mesh indexing using morpho-semantic normalization. In *Proc. of the American Medical Informatics Association Symposium (AMIA 2003)*, pp. 425–429.

McDonald, S. (1997). Exploring the validity of corpus-derived measures of semantic similarity. In *Proc. of the 9th Annual CCS/HCRC Postgraduate Conference*.

McHale, M. (1998). A comparison of wordNet and Roget's taxonomy for measuring semantic similarity. In *Proc. of COLING/ACL Workshop on Usage of WordNet in Natural Language Processing Systems*, pp. 115–120.

Paice, C. and W. Black (2003). A three-pronged approach to the extraction of key terms and semantic roles. In *Proc. of the Conference on Recent Advances in Natural Language Processing (RANLP 03)*.

Paynter, G., S. J. Cunningham, and I. H. Witten (2000). Evaluating extracted phrases and extending thesauri. In *Proc. of the 3rd International Conference of Asian Digital Library*.

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

Rada, R., H. Mili, E. Bicknell, and M. Blettner (1989). Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man, and Cybernetics 19*(1), 17–30.

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In *Proc. of the 14th International Joint Conference on Artificial Intelligence (IJCAI 95)*, pp. 448–453.

Rolling, L. (1981). Indexing consistency, quality and efficiency. *Information Processing and Management, 17*(2), 69–76.

Ruiz, M. E. and P. Srinivasan (1999). Combining machine learning and hierarchical indexing structures for text categorization. In *Proc. of the 10th ASIS/SIGCR Workshop on Classification Research*.

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

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

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

Silverstein, C., M. Henzinger, H. Marais, and M. Moricz (1998). Analysis of a very large AltaVista query log. Technical Report 1198-014, Digital SRC.

Terra, E. L. and C. Clarke (2003). Frequency estimates for statistical word similarity measures. In *Proc. of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 03)*.

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

van Dijk, T. A. and W. Kintsch (1983). *Strategies of Discourse Comprehension*. New York: Academic.

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

Wild, C. J. and G. A. F. Seber (1995). *Introduction to Probability and Statistics*. University of Auckland, Dept. of Statistics.

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

Witten, I. H., G. W. Paynter, E. Frank, C. Gutwin, and C. G. Nevill-Manning (1999). Kea: Practical automatic keyphrase extraction. In *Proc. of the 4th ACM Conference on Digital Libraries (DL 99)*, pp. 254–255. Berkeley, CA: ACM Press.

Wu, J. and A. M. Agogino (2004). Automating keyphrase building with multi-objective genetic algorithms. In *Proc. of the Hawaii International Conference on System Science (HICSS 04)*.

Zunde, P. and M. Dexter (1969). Indexing consistency and quality. *American Documentation 20*(3), 259–267.