UNIVERSITÄT MANNHEIM

FAKULTÄT FÜR MATHEMATIK UND INFORMATIK

Thesaurus Analysis and Visualization in Semantic Search Applications

Diploma thesis

by

Kai Eckert

Knowledge Representation and Knowledge Management
Research Group

Prof. Dr. Heiner Stuckenschmidt

May 2007

Supervisor: Prof. Dr. Heiner Stuckenschmidt
Co-Referee: Dr. York Sure, University of Karlsruhe

Acknowledgments

This thesis was written at the Knowledge Representation and Knowledge Management Research Group of the University of Mannheim. I want to thank all co-workers for their support and the pleasant atmosphere.

I want to thank Prof. Dr. Heiner Stuckenschmidt for his excellent supervision of my work and for the confidence shown to me. I also thank Magnus Pfeffer of the Library of the University Mannheim who pursued my work with interest and provided me with his professional competence.

Furthermore, I want to thank SyynX GmbH for their support. They invited me to their in-house conference to get an insight on the state of the technology, to meet other users and to learn more about the different applications that are realized or planned with the Collexis Engine. Special thanks go to Mario Diversy and Georg Lambert who answered all my questions about the use of the technology and provided me with further ideas and suggestions.

I thank all my friends and fellow students, who accompanied me during my studies, especially my girlfriend Florence Görg for her endless patience during the last weeks and my friends Ralf Hettinger and Stefan Schmidt for proofreading and further suggestions.

Last but not least I want to thank my parents Inge and Uli Eckert, who were always supportive of everything I did.

Contents

List of Figures

1 Introduction and Overview

The use of thesaurus-based indexing is a common approach for increasing the performance of information retrieval. In this thesis, we examine the suitability of a thesaurus for a given set of information and evaluate improvements of existing thesauri to get better search results. On this area, we focus on two aspects:

1. We demonstrate an analysis of the indexing results achieved by an automatic document indexer and the involved thesaurus.
2. We propose a method for thesaurus evaluation which is based on a combination of statistical measures and appropriate visualization techniques that support the detection of potential problems in a thesaurus.

In this chapter, we give an overview of the context of our work. Next, we briefly outline the basics of thesaurus-based information retrieval and describe the Collexis Engine that was used for our experiments. In Chapter 3, we describe two experiments in automatically indexing documents in the areas of medicine and economics with corresponding thesauri and compare the results to available manual annotations. Chapter 4 describes methods for assessing thesauri and visualizing the result in terms of a treemap. We depict examples of interesting observations supported by the method and show that we actually find critical problems. We conclude with a discussion of open questions and future research in Chapter 5.

1.1 Introduction

Advanced methods for retrieving documents based on their contents are becoming more and more important in many application areas. Today, information about virtually any topic is accessible in digital form through digital libraries which are accessible over the web.

Beside full-text search, semantic search applications with automatically or manually created annotations of documents in terms of keywords from controlled vocabularies and thesauri are used. In particular, they solve the problem of using different terms to talk about the same topic. This is done by explicitly representing information about synonymous terms and relating them to a preferred term that is used to describe the content of a document.

The benefits of using a thesaurus for annotating documents comes at the price of the effort needed for annotating large document sets. Traditionally, this annotation is done manually by specialists that read the abstract of a document and decide which of the preferred terms in a thesaurus describe its content best. But with the increasing amount of information manual annotation is not feasible any more. Therefore, techniques for automatically annotating documents with terms from a thesaurus become more and more important.

1.2 Motivation

With the increased use of automatic indexing techniques a fundamental problem of thesauri becomes more pressing: while the limitation of annotations to preferred terms in the thesaurus improves document search, the quality of the annotations significantly depend on the quality of the thesaurus. If the terms of the thesaurus do not adequately cover the topics represented in the documents, the annotation will be far away from optimum. This problem is less pressing when annotation is done manually, because the human annotator can compensate for a suboptimal thesaurus by choosing nontrivial combinations of existing terms or even by modifying the thesaurus to better cover a particular topic.

In the presence of fully automatic annotation, the issue of thesaurus quality becomes an essential one, because existing methods for automatic annotation are not able to compensate for missing or misleading terms. We argue that methods for automatic thesaurus-based document annotation require an additional thesaurus assessment step in which the suitability of the thesaurus as a basis for annotating a certain document set is assessed. The goal of this step is to avoid the problems mentioned above and to unveil hidden problems in the thesaurus that first need to be fixed before automatic annotation can be performed successfully. We further argue that this assessment step has to be an interactive one.

There are approaches that aim at computing the quality of a thesaurus or more generally an ontology either simply on the basis of the thesaurus or based on the correspondence between the thesaurus and the corpus of documents. While such methods are a good starting point for an analysis because they can provide interesting statistics about the thesaurus and the documents at hand, they can only provide hypotheses about potential problems. For instance, a certain thesaurus term is very rarely used to annotate documents from the corpus. This can be due to a problem in the thesaurus (for example caused by missing synonyms) or it can just be a consequence of the fact that some particular topic is not mentioned in the corpus. The decision whether a certain observation actually reveals a problem in the thesaurus requires a human in the loop.

1.3 Research Questions and Contributions

In this thesis, we present an approach for interactively assessing the suitability of a thesaurus for automatically annotating a given document set. Our method combines the application of statistics of concepts in the thesaurus backed by information theory and a visualization of the results of the statistical analysis in a way that helps the user to identify and further investigate potential problems in a thesaurus. We describe this method and evaluate it in two experiments with different thesauri and document sets. In particular, we take the following steps:

- We automatically create annotations and compare them to existing manual annotations.
- We develop an approach to evaluate the quality of the automatic annotations and use it to judge the suitability of the German standard thesaurus for the economic domain (Standard Thesaurus Wirtschaft) for automatic indexing compared to the well established Medical Subject Headings (MeSH) for the medical domain.

- We propose a new analysis approach using a combination of information theoretic measures and advanced visualization and use it to analyze the two above mentioned thesauri and to identify some problems and non-problems.
- We use our evaluation approach to show that fixing the problems identified using our method indeed improves the automatic annotations compared to the manually created ones.

2 Thesaurus-based Information Retrieval

According to [BYRN99], Information retrieval (IR) deals with the representation, storage, organization of and access to information items (mostly documents). The representation and organization of the information items should provide easy access to the information which a user is interested is.

Contrary to data retrieval (e.g. with a relational database), information retrieval works on unstructured data with no well defined semantics. Instead of retrieving all objects which match certain conditions with a regular expression or some relational algebra expression, an information retrieval system is used to find relevant information for a given query. The relevance is a rather diffuse criterion and depends on the correct interpretation of the information items by the IR system. This interpretation of a document involves extracting syntactic and semantic information from the document content.

The primary goal of an information retrieval system is to retrieve all relevant documents for a given user query while retrieving as few non-relevant documents as possible.

In an information retrieval system, documents are usually not accessed directly but via a document representation (record). Traditionally, such a record contains all information needed to identify and locate the corresponding documents and additional information to characterize the content of the document, just like a record card in a library. Like these record cards, a record contains information about the author, the title and the publisher of a document, the year of publication, an identification number, the so called signature and the location of the document within the library. The document is characterized by some manually selected keywords, usually contained in a library specific controlled vocabulary, the thesaurus. The process of assigning the proper keywords is called indexing or annotation.

2.1 Problems of Full Text Search

The advancements of computer technology made it possible to save the whole text of a document and use it for information retrieval. Such a representation is called a full text representation and the search in these representations is called full text search.

Obviously, the advantage of a full text search is that the indexing process is very simple and no controlled vocabulary is needed. It is independent of the subject area and even of the document's language. For these reasons, full text search currently is the standard method used by internet search engines.

The problem of full text search is that the user is required to use the same terms in the query as used in the relevant documents. In case of existing synonyms for a term, all synonyms should

be included in the query, as within one document, usually only one of the synonyms is used. This problem is even impaired by different grammatical flections used in the documents. When using the search engine Google[1], the result set depends strongly on the exact search term, as the following example illustrates:

- Search for “heart attack”: 16 million results
- Search for “heart attacks”: 1.1 million results
- Search for “Myocardial infarction”: 4.7 million results.

On the other hand, if a term in the query has homonyms (i.e words with multiple but different meanings), all documents with the given term are found irrespectively of the meaning.

Figure 2.1 demonstrates this for the Google search engine. Searching for “jaguar” reveals four different senses on the first ten result hits: The car, the cat, the nick name as part of the title “Aimée & Jaguar” and a company named “Jaguar Models”.

2.2 Thesaurus-Based Search

In this section, we introduce some of the ideas used for thesaurus-based search with automatically created annotations. In this work we use the following definitions:

Thesaurus A hierarchy of *concepts*. The hierarchy reflects a relation between the concepts depending on the purpose of the thesaurus, like “is a”, “part of” or just “related to”. In literature, sometimes the more general notion *ontology* is used.

Concept A concept consists of several *terms* and represents the meaning of all these terms. Another notion, especially in the environment of the MeSH thesaurus, is *descriptor*.

Keyword Keyword is the traditional notion for a *concept* used to annotate a document. As such, it is the same as a concept. We use the notion *keyword* for a manually selected concept, in contrast to the automatically assigned concepts, which we refer to as *concepts.*

Term A term is a single word, usually in it’s basic form. It is also referred to as *synonym*, *entry term* or just *word*, but especially in the latter case, one often means the actual word in a text in its inflected form.

Document A document is the actual manifestation that can be searched for with a search engine. It can be virtually anything, but in the context of this work, documents are electronic representations like abstracts and fulltexts of papers and books, PDFs[2], websites, etc.

Record A record is the representation of a *document* in the database of the search engine. Like a traditional library filecard, it usually consists of bibliographic data like title, author, publisher and a set of manually selected *keywords* or automatically assigned *concepts*.

[1] http://www.google.com
[2] Adobe Portable Document Format

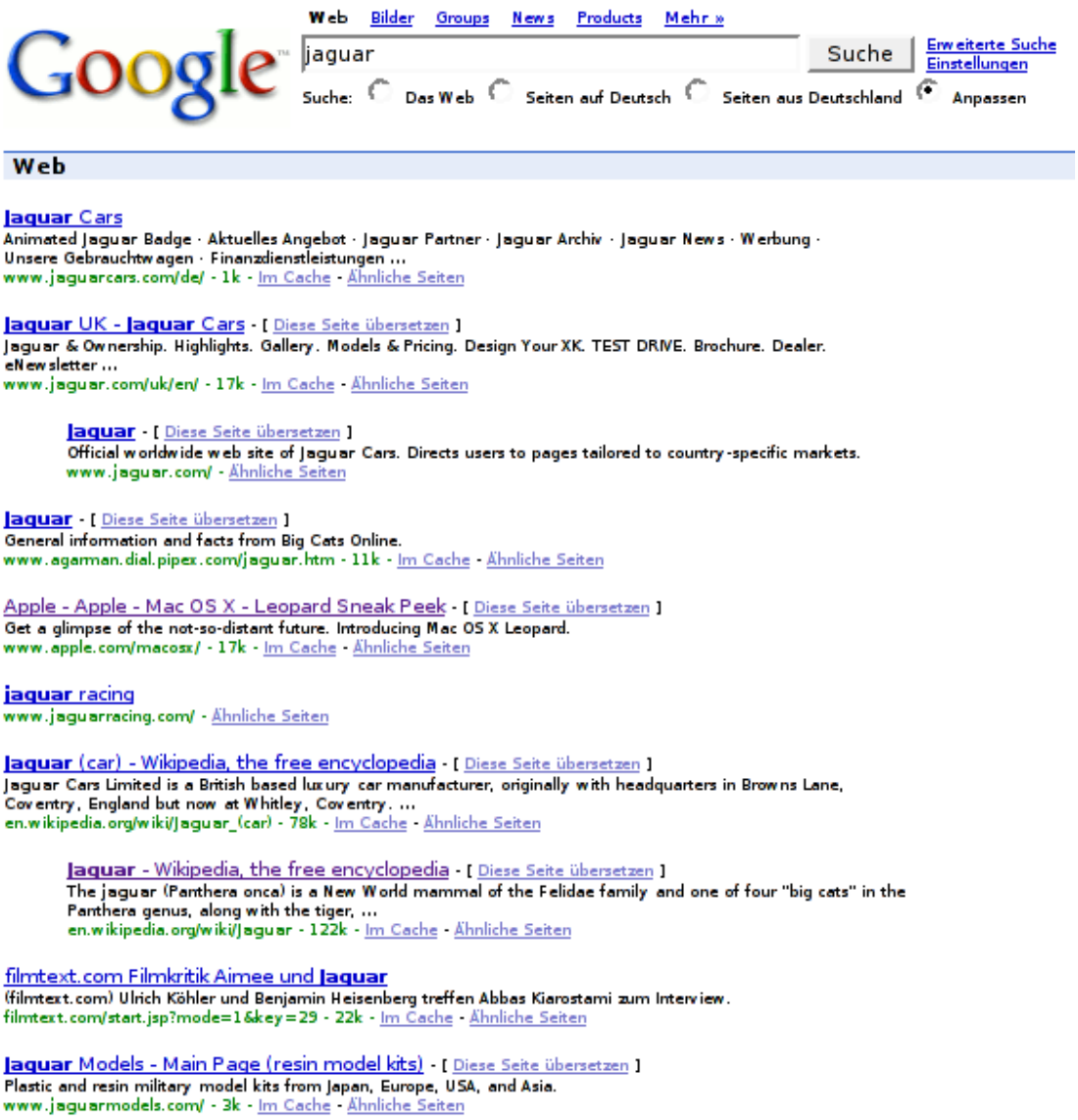


Figure 2.1: Google Results for “jaguar”

A brief example should illustrate the definitions. We have a document base of some scientific papers in PDF format. These are our *documents*. Now we use our IR system to save these documents as *records*. For every PDF, a *record* is created with the title, the author and a reference to the *document*. As we want to search for these *records* via some *keywords*, we select some appropriate *concepts* from a *thesaurus* and assign them as *keywords* to the *record*. Additionally, the IR system provides an automatic indexer, which uses the same *thesaurus* to annotate or index the *document*. For example, the indexer detects the *term* "heart attack" in the document. In the *thesaurus*, we have the following *concept*: "Myocardial infarction" with the synonym *term* "heart attack". As this matches the term found, the indexer assigns the concept "Myocardial infarction" to the *record*. Later, we use our IR system to search for relevant *records* which lead us via the reference to the PDF *document*.

Traditional library search engines generally use a simple full-text search over the elements of the records, usually the title, the author and the keywords. This is very simple and can be done without high computational costs. The quality of the search result depends strongly on the quality of the manually selected keywords. As a human expert will only assign keywords with a high relevance, the documents returned will probably be of high relevance for the user. But the user is required to understand the used thesaurus and to find the appropriate keywords for the query that match the assigned keywords.

A digital IR system that uses automatically annotated records as database needs sophisticated techniques to ensure that the results are relevant for the given search query, as an automatic annotation process usually assigns all concepts found in a document to the record. Moreover, the search engine should also select records that do not contain the concept that is searched for, but other highly related ones. These two aspects of determining the relevance of a concept and searching for related relevant records are described in the next two sections.

2.2.1 The Vector Space Model

With the vector space model (Figure 2.2), the query and each record is represented by a vector in a space spanned by the words in the document base [SWY75].

The vector represents the words in a document in case of a full text search. In this case, the vector space is very high dimensional. The model is especially suitable for thesaurus-based search, where only relevant words are considered. Moreover, as words with the same meaning are represented by a single concept, the number of dimensions is reduced even more.

The vector space model is based on the assumption that the meaning of a document is conveyed by the words used. Near vectors should indicate that the represented documents are related. As the query is represented as a vector as well, relevant documents are determined by searching for the nearest document vectors to the query vector.

Different distance measures can be used to determine the nearest and thus most relevant record vectors to the query vector, for example the scalar product, cosinus measure, Dice coefficient and Jaccard coefficient [Fer03]. A short review of the cosine measure, the Dice- and the Jaccard coefficient can be found at [GF04].

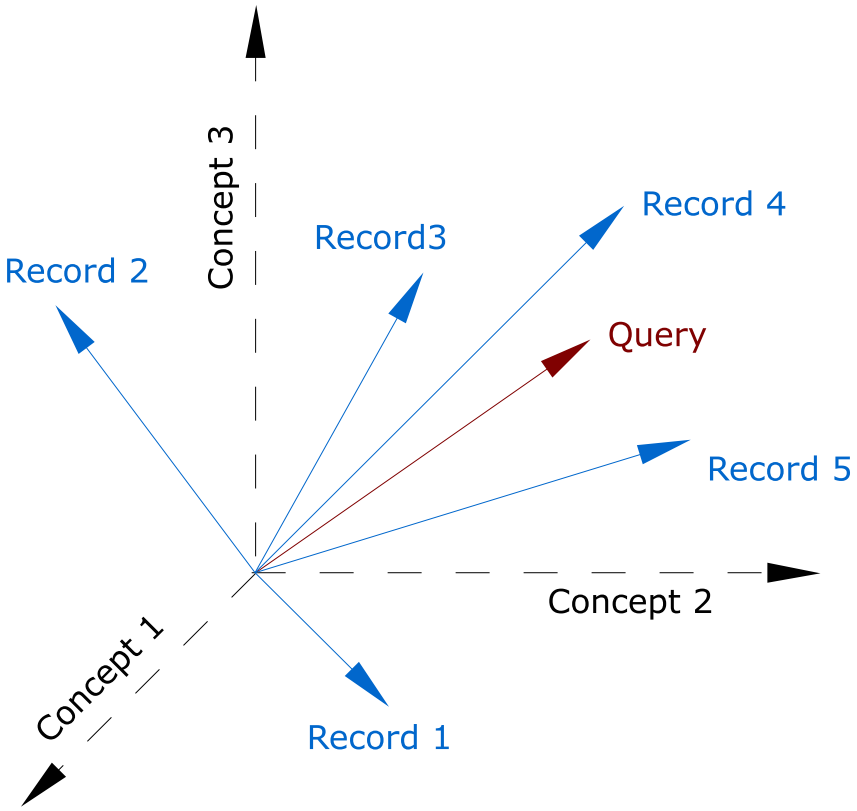


Figure 2.2: Vector space model

The components of the vectors represent the weight of each concept. In the simplest case, this could be just the frequency of the concept in the given document. Consider the following example:

With a thesaurus containing the concepts $(sun, moon, star)$, the following "documents" result in these vectors:

- "The sun is shining."$\rightarrow \overrightarrow{d_1} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$
- "The moon is a moon, whereas the sun is a star."$\rightarrow \overrightarrow{d_2} = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}$

2.2.2 Calculating concept relevance (TF-IDF)

The frequency of a concept in a document alone is not the optimal measure for the relevance of a concept for this document. If the same concept appears in all documents with this frequency, it's relevance for each document would be rather low. So the frequency is usually weighted by the *Inverse Document Frequency*. As the vector space model and this approach of calculating relevance are not restricted to thesaurus-based search, we cite the following generally accepted definitions [GF04], despite the fact that in this work the notion "term" depicts the synonyms of a concept in a thesaurus:

tf_{ij} *Term frequency*, the number of occurrences of a term j in a document i. In our case, it would be more consequent to call it *concept frequency*.

df_j *Document frequency*, the number of documents containing a term j.

$idf_j = \log\left(\frac{d}{df_j}\right)$ *Inverted document frequency*, where d is the total number of documents.

The components of the document vectors are then calculated as

$$d_{ij} = \begin{cases} tf_{ij} \cdot idf_j & tf_{ij} > 0 \\ 0 & tf_{ij} = 0 \end{cases}. \tag{2.1}$$

This measure is often referred to as TF-IDF measure. With this, the above example changes to ($df_{sun} = 2, idf_{sun} = 0, df_{moon} = 1, idf_{moon} = 0.3, df_{star} = 1, idf_{star} = 0.3$):

- "The sun is shining."$\rightarrow \overrightarrow{d_1} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$
- "The moon is a moon, whereas the sun is a star."$\rightarrow \overrightarrow{d_2} = \begin{pmatrix} 0 \\ 0.6 \\ 0.3 \end{pmatrix}$

This illustrates two aspects of the TF-IDF measure: First, a value other than zero will only be assigned, if a concept occurs at least once in a document. Then, a concept is considered more relevant for a document, if it has a higher frequency. Second, a value of zero is assigned, if a concept appears in all documents. In this case, it is considered as not relevant at all for the meaning of a single document. Generally, a concept is considered most relevant, if it has a high frequency in a document and a low frequency over all documents.

2.3 Precision and Recall

In this section, we briefly introduce the most common metrics used in information extraction for measuring the quality of search results: *Precision* and *Recall*. Generally, the measures work on the set of documents in the document base and the result set of a query. We will use these measures in a slightly different context, as we do not measure the quality of search results, but the quality of the automatic indexing results. Therefore we describe the measures with abstract object sets.

Figure 2.3 illustrates the object sets defined in an information retrieval application. Generally, there is an *object base* (e.g. a document base for a search application or a thesaurus like in our case). The result of the IR process is a *set of objects found* (a set of documents in case of a search result, a set of concepts in our case). To evaluate the IR process, there has to be a gold standard, i.e. a known *set of relevant objects*, that ideally should match the result. For a search engine, this could be a manually selected set of relevant documents, but in our case the manually selected keywords are used.

The intersection set of objects found and relevant objects contains the *correct* objects. All other objects retrieved (incorrectly) are called *spurious*. On the other side are the *missing* objects, which are relevant, but not found. Equipped with these object sets, we can now define Precision and Recall:

Precision is defined as the ratio of objects found correctly to the total number of objects found (correctly and incorrectly):

$$Precision = \frac{Correct}{Correct + Spurious} \tag{2.2}$$

Recall is defined as the ratio of found correct objects to the total number of correct objects (found and not found):

$$Recall = \frac{Correct}{Correct + Missing} \tag{2.3}$$

For example, we have an thesaurus of 1000 concepts, out of which 10 are used as manually selected keywords by a human expert and thus are relevant for a certain document. Our automatic indexer returns a set of 20 concepts. These concepts contain 8 of the manually selected keywords (*correct*). So we have 12 concepts in the result set considered not relevant (*spurious*).

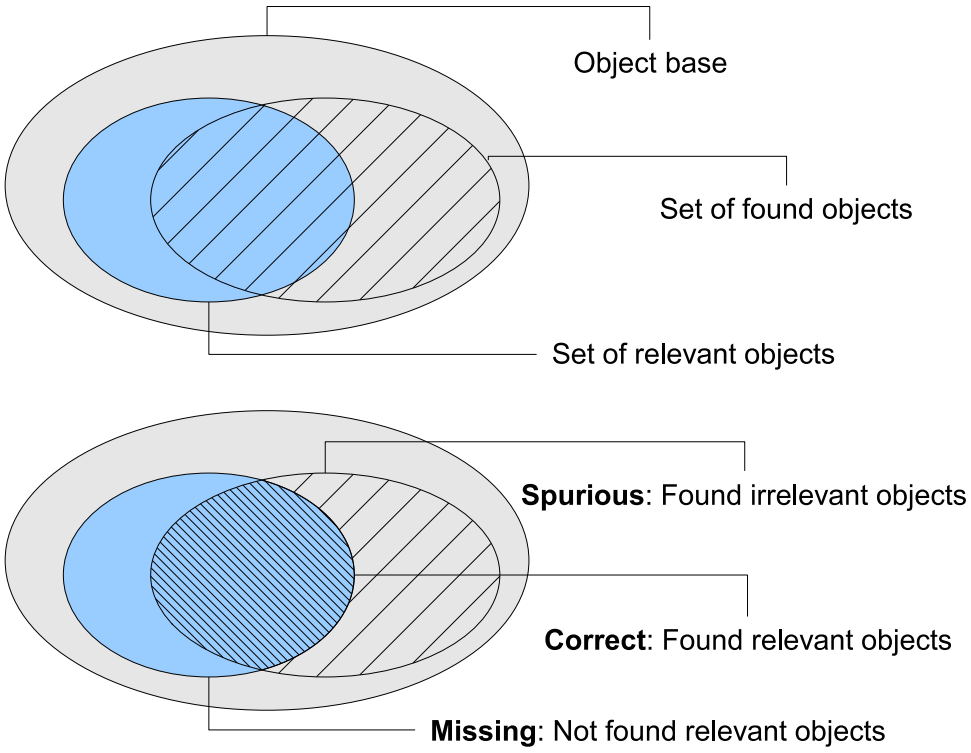


Figure 2.3: Object sets used by precision and recall metrics

2 of the manually selected keywords were not found (*missing*). This leads to the following calculations:

$$Precision = \frac{8}{20} = 0.4$$

$$Recall = \frac{8}{10} = 0.8$$

Precision and Recall are negatively correlated so that a higher Recall generally leads to a lower Precision and vice versa. It depends on the user and the application if a higher Recall or a higher Precision is preferred. If all relevant objects need to be found, we need a maximum Recall. If it is important that only relevant objects are found, the maximum Precision is required. So in practice most often a trade-off between Precision and Recall is used.

2.4 The Collexis Engine and Technology

For our experiments we used a state-of-the-art thesaurus-based search engine, the Collexis Engine by Collexis B.V., Geldermalsen, The Netherlands. Collexis is programmable with APIs for various languages, for example Java, .NET and Python.

2.4.1 Overview

Two companies are involved in the development of the engine:

Collexis B.V. is a global company with worldwide headquarters in the Netherlands and US headquarters in Columbia, South Carolina. Their mission is to develop software that supports knowledge extraction and discovery across multiple industries. The main product of Collexis B.V. is the thesaurus based knowledge extraction and search engine Collexis, referred to as Collexis Engine. Furthermore, special solutions for governmental, medical, pharmaceutical and financial use are based on the Collexis Engine.

The core of the Collexis engine is developed by Syynx Solutions GmbH, settled in Cologne, Germany. Syynx also implements solutions based on Collexis Engine for the public sector, clinical use and life sciences. Syynx provided us with in-depth information about the Collexis engine.

2.4.2 Generating Fingerprints

Collexis uses the vector space model for document retrieval. The vectors representing the records and queries are called fingerprints, the generation of them is called fingerprinting. Such a fingerprint vector contains relevance values for concepts taken from a thesaurus and found in a given text. The Collexis engine is used to create fingerprints of text based information like documents, papers, sheets or web pages.

Figure 2.4 shows the workflow to generate a fingerprint. First of all, the text is broken up into sentences and common words (the stop words) are removed. Stop words are words that don't

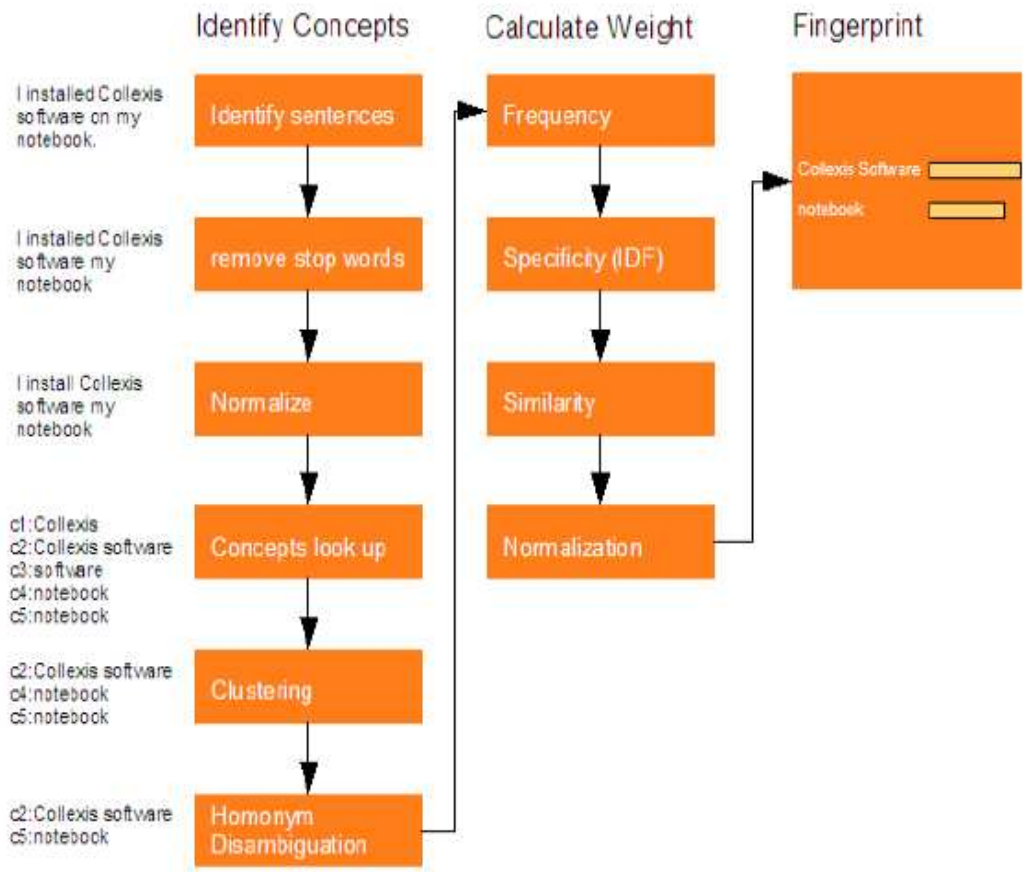


Figure 2.4: Collexis Workflow *(Source: Collexis B.V.)*

contribute to the meaning of the text (like the, on, while...). Stop words are used primarily for performance reasons, as at last only concepts contained in the thesaurus are considered relevant, anyway. So it would be cumbersome and senseless to perform the following steps on these words.

Next, all remaining words are normalized. Different approaches and normalizers can be used. In most cases, every noun is normalized to nominative singular, every verb to its infinitive.

In our case, we used the Lemma normalizer provided by Collexis. Each remaining word is looked up in a database. If it is found the normalized version is returned. If not found then the original word is lowercased and this is looked up in the database. If it is found, the normalized version is returned. If it is not found again, it is searched for a part of the word starting at the end of the lowercase word. If there is a match, the first part and the normalized last part are returned.

After normalization, every word is looked up as a concept in the thesaurus used. Clustering can be used to find combined words, i.e. the most specific concept containing combined words is used instead of single word concepts.

Figure 2.5: Collexis Fingerprint *(Source: Collexis B.V.)*

If more than one concept is appropriate for a word, because its a homonym, a disambiguation step can be performed to determine the right sense in the given context. There are different disambiguation techniques, for example, the semantic distance to other concepts in the environment of the concept in question is used. A comprehensive overview can be found at [IV98]. We used no disambiguation step in our experiments as we wanted most traceable results and in this way we provide a basis for further evaluations of the impact of disambiguation.

These steps have led to a list of concepts found in the text (concept identification). After that, the frequency of each concept is determined and weighted with its specificity. A concept found very often in the text is considered relevant if its not also found very often in the other documents in the document base. Approaches like this are known as TF/IDF (Term Frequency, Inverted Document Frequency), see Section 2.2.2.

Finally, the weights can be smoothed and normalized with different algorithms. This leads to a better usability of the resulting fingerprint for search applications. Figure 2.5 shows an example of a fingerprint generated by the Collexis Engine.

Almost every step above mentioned can be customized and lot of different algorithms can be used. So there are a lot of normalizers for different languages, different approaches for disambiguation, several possibilities to calculate the IDF and the resulting weight and so on. As we do not use the calculated weights in this work, the smoothing technique is not important for our results.

The process of fingerprint creation can involve multiple thesauri and languages. This makes it possible to handle documents from different languages and find relevant documents, even if they are written in a language not used for the query.

2.4.3 Document Retrieval

Once the fingerprints for all documents in the document base are generated and saved in a so called “Collexion”, one can use the Collexis Engine to search for relevant documents with

respect to a given query. In this case, a fingerprint of the query string is generated and a distance measure in vector space is used to find the nearest documents and return them as result.

Documents are stored as records. A record contains only the fingerprint of the document and some additional data, like a unique id, a title and some meta-data. The meta-data can contain arbitrary information, like the publishing date, the author or an organization. A special feature is the use of a direct-link id, with which an URL can be generated to link to the original documents, if the document is available via such an URL.

Beside the vector matching to find relevant documents, the results can be filtered by the meta-data. So it is possible to find documents published before a special date or documents by a specific organization.

2.4.4 Further Possibilities

A very interesting approach of Collexis is the aggregation and clustering of fingerprints to explore hidden knowledge. It can be used for data mining to find new dependencies and for knowledge retrieval (as extension to information retrieval), where not only the information (i.e. the processed document) is found, but also some contained knowledge is used. For example, you could create a map of organization and their specific knowledge areas, if you cluster the documents by similarity and publishing organizations and analyze the overlaps. Of course, these aspects are beyond the scope of this work.

2.4.5 Architecture

The Collexis architecture is developed as a 3-tier environment. Figure 2.6 illustrates this. The core components are the thesaurus component and the matching component. The thesaurus component is used to generate the fingerprints of documents and queries, the matching component executes the document retrieval.

The underlying data (thesauri and collexions with fingerprints) is stored in a proprietary database and can only be accessed via the Collexis Engine.

The clients and tools communicate with the Collexis Engine via the TCP/IP Collexis Gateway. It provides the full API for the Engine. There are different implementations of this API for different languages, like Java, .NET and Python.

2.5 Thesauri

In the experiments, we used two thesauri from different domains and with quite different characteristics in order to be able to generalize our observations. The first thesaurus is MeSH, a well established thesaurus from the medical domain that is extensively used to annotate large collections of medical documents. The second thesaurus is the German standard thesaurus for business and economics that has been created recently to provide the basis for indexing literature in the area of business and economics. In contrast to MeSH the use of this thesaurus is

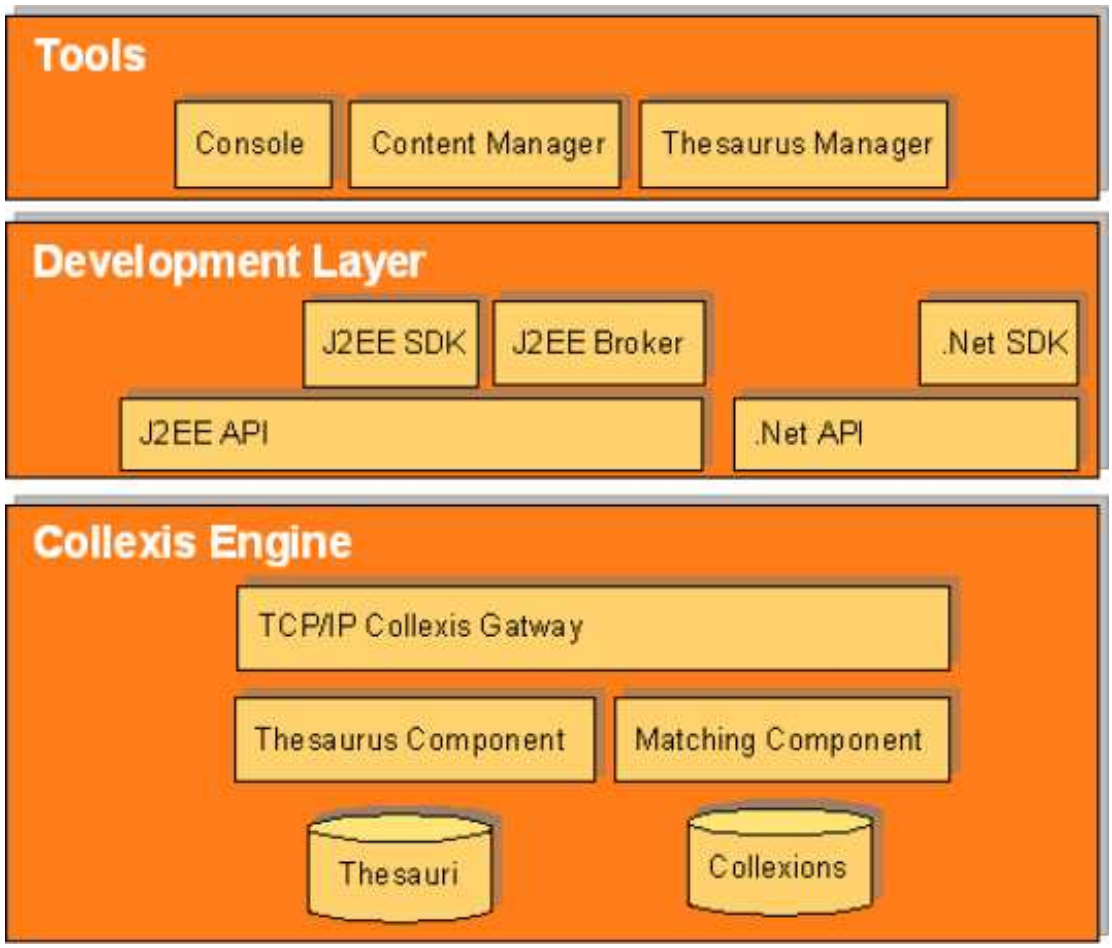


Figure 2.6: Collexis Architecture *(Source: Collexis B.V.)*

limited to a number of specialized libraries and there is no experience with its use for automatic indexing.

In the following two sections, we describe the two thesauri in more detail to provide the necessary background information. Finally, we provide an excursus on polyhierarchic thesauri in the last section, as both of them are polyhierarchic.

2.5.1 MeSH

History. The Medical Subject Headings (MeSH) thesaurus is produced by the National Library of Medicine (NLM) and continuously updated since 1960. It is used for cataloging documents and related media and as an index to search these documents in a database and is part of the metathesaurus of the Unified Medical Language System (UMLS).

This thesaurus originates from keyword lists of the Index Medicus, a comprehensive directory of medical documents, nowadays known as Medline. Medline still uses the MeSH headings as descriptors for the documents. The thesaurus, as well as the Medline database are available online [NLM07a, NLM07b].

Structure. The MeSH thesaurus is a polyhierarchic thesaurus, i.e. every concept can occur multiple times (see Section 2.5.3). It consists of the three parts

1. MeSH Tree Structures,
2. MeSH Annotated Alphabetic List and
3. Permuted MeSH.

The Tree Structures contains various subtrees of descriptors. On top, there are 16 categories:

1. Anatomy [A]
2. Organisms [B]
3. Diseases [C]
4. Chemicals and Drugs [D]
5. Analytical, Diagnostic and Therapeutic Techniques and Equipment [E]
6. Psychiatry and Psychology [F]
7. Biological Sciences [G]
8. Natural Sciences [H]
9. Anthropology, Education, Sociology and Social Phenomena [I]

10. Technology, Industry, Agriculture [J]
11. Humanities [K]
12. Information Science [L]
13. Named Groups [M]
14. Health Care [N]
15. Publication Characteristics [V]
16. Geographicals [Z]

Each category is further divided into subcategories. Each subcategory contains a hierarchic subtree from most general to most specific descriptors in up to eleven hierarchical levels. These subtrees are not an exhaustive classification of the subject matter, but contain only those terms that have been selected for inclusion, representing a compromise among the needs of various disciplines and users.

Each MeSH descriptor appears in at least one place in the trees, and may appear in as many additional places as may be appropriate.

For each appearance of a descriptor, a number is assigned, like

- Abnormalities C16.131
  - Abnormalities, Drug Induced C16.131.42
  - Abnormalities, Multiple C16.131.77
    - Alagille Syndrome C16.131.77.65
    - Angelman Syndrome C16.131.77.95

These numbers are used to locate the descriptors in each tree and to alphabetize those at a given tree level and have no intrinsic significance; e.g., the fact that D12.776.641 and D12.644.641 both have the three digit group 641 does not imply any common characteristic. The numbers are subject to change when new descriptors are added or the hierarchical arrangement is revised to reflect vocabulary changes.

Table 2.1 shows an example of a MeSH Descriptor. The MeSH Heading is followed by several tree numbers denoting the multiple positions in the different subtrees of the MeSH thesaurus. A free scope note is used to describe the heading to the user. The different synonyms for the heading are described by the entry terms. One can use qualifiers to narrow the heading in a search application. And at last there is a unique ID for each heading.

In our experiments, we used the MeSH 2006 thesaurus with 31956 concepts and about 170.000 terms. The language of the MeSH thesaurus is English.

MeSH Heading	Ethics
Tree Numbers	F01.829.500.519 K01.316 K01.752.256 N05.350
Scope Note	The philosophy or code pertaining to what is ideal in human character and conduct. Also, the field of study dealing with the principles of morality.
Entry Terms	Egoism Ethical Issues Metaethics Moral Policy Natural Law Situational Ethics
Allowable Qualifiers	CL HI
Unique ID	D004989

Table 2.1: Example of a MeSH Descriptor

2.5.2 STW

The Standard Thesaurus Wirtschaft (STW) is produced by GBI-Genios Deutsche Wirtschaftsdatenbank GmbH and the Deutsche Zentralbibliothek für Wirtschaftswissenschaften (ZBW). The thesaurus is focused on economic sciences.

Structure. The STW consists of the following main parts:

1. A = Allgemeinwörter (*Common Terms*)
2. B = Betriebswirtschaft (*Economic Science*)
3. G = Geographische Begriffe *(Geographics)*
4. N = Nachbarwissenschaften *(Related Disciplines)*
5. P = Produkte *(Products)*
6. V = Volkswirtschaft *(Political Economics)*
7. W = Wirtschaftszweige und spezielle Wirtschaftslehren *(Industries)*

There is another part, primarily intended as additional access and for cross reference: the NACE concordance (Nomenclature général des activités économiques dans les Communautés Européens).

The parts W and P use the systematics of the German Federal Statistical Office. The geographic part G uses continental grouping starting with Europe. The common terms in part A are unrelated terms with low information content.

Each main part is further divided into hierarchic subparts as in the following example:

- B.01 Unternehmensführung und Organisation
  - B.01.01 Unternehmensführung und Unternehmensplanung
  - B.01.02 Organisation
  - B.01.03 Betriebliche Information und Kommunikation
  - B.01.04 Rechtsformen
  - B.01.05 Unternehmensentwicklung, Betriebsgröße und Standortwahl
  - B.01.06 Umweltmanagement
- B.02 Investition und Finanzierung
  - B.02.01 Kapitalbeschaffung
    - B.02.01.01 Eigenkapitalbeschaffung
    - B.02.01.02 Fremdkapitalbeschaffung
  - B.02.02 Kapitalverwendung
    - B.02.02.01 Investitionsplanung und -rechnung

The concepts (Table 2.2 shows an example concept) are polyhierarchically ordered and can be assigned to several of the subparts. So every subpart can be seen as subthesaurus. To reflect the different subthesauri, we built a single thesaurus and used the hierarchy of subthesauri as additional concepts on top of the STW concepts. The parents of a concept node are the union of broader terms and allocations.

The resulting thesaurus consists of 6294 concepts and 27204 terms. The language of the STW thesaurus is German. Every concept contains only one English term as synonym. The hierarchy of the subparts is only available in German.

Broken Implementation. In our first experiments, we used an implementation of the STW thesaurus that was not built by ourselves. This implementation is broken, as it lacks the complete hierarchy on top of the different subthesauri. Instead, the subthesauri simply are put directly under an artificial root concept. Only the broader terms within one subthesaurus are used to build the hierarchy. The result is a very flat hierarchy with a huge number of single concepts directly located under the root node.

We came across the broken implementation when we got astounding good results in our evaluation with different similarity measures described in 3.2. So we used this implementation to demonstrate some weaknesses and pitfalls of these measures. We mention the broken implementation explicitly, if we refer to it. In all other cases, of course, we always refer to the correct implementation.

Concept	Unternehmensversicherung
Description	Versicherung betrieblicher Risiken
Synonyms	Insurance against business risks
	Betriebsrisikoversicherung
	Betriebsunterbrechungsversicherung
	Gewerbliche Versicherung
Broader Terms	Versicherung
Narrower Terms	Exportkreditversicherung
	Industrieversicherung
	Technische Versicherung
Related Terms	Betriebliche Störung
	Landwirtschaftliche Versicherung
	Risikomanagement
Allocation	B.01.01
	W.16.01
	66.03 [a]

[a] NACE Concordance, not used

Table 2.2: Example of an STW concept

2.5.3 Polyhierarchic Thesauri

In a natural language there are a lot of terms, which cannot placed easily in a monohierarchic tree structure. Especially combined terms like “stomach cancer”, which can be seen as a specialized subconcept of cancer or a hyponym related to stomach diseases. To reflect this, the MeSH as well as the STW thesaurus use a polyhierarchic structure. Figure 2.7 illustrates this for Hordeolum, a bacterial eyelid infection. In the MeSH thesaurus, this concept can be reached via several paths.

In a polyhierarchy, a concept can have more than one parent (or broader term). This leads to some difficulties we have to deal with as we face the following two requirements in this work:

1. We use the distance of concepts in the thesaurus hierarchy, usually defined by the number of nodes or edges between two concepts.

2. We have to visualize and browse a thesaurus as whole.

A polyhierarchic thesaurus can have more than one root concept, i.e. concepts, that have no parents or broader terms. In this case, there are concepts in the thesaurus that have no common subsumer and thus no direct connection. Moreover, there are completely disconnected hierarchies in such thesauri.

To circumvent these problems we always use a single artificial root concept, if no such concept exists. We define that every concept without a parent or broader term gets the root concept as the only associated parent node. Note, that we can still decide two concepts as not connected,

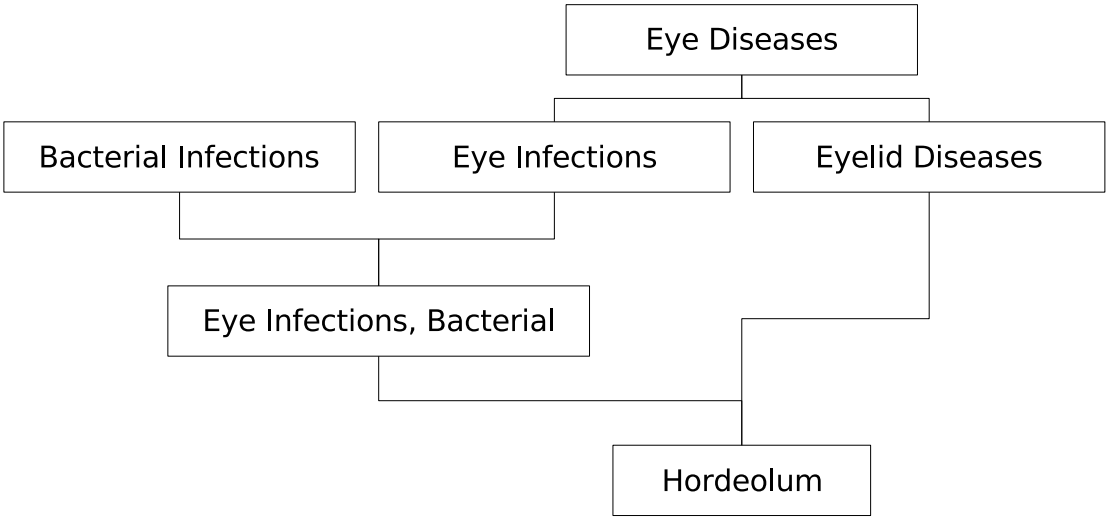


Figure 2.7: Polyhierarchic structure in a thesaurus

if this artificial root node is the least common subsumer (LCS), i.e. the first common term reached when moving from both concepts upwards in the hierarchy. This will be important for the evaluation of the indexing results in Section 3.1.3 and the various measures of semantic similarity in Section 3.2.

Whereas a monohierarchic thesaurus can be displayed as a tree, a polyhierarchic thesaurus is a more general acyclic directed graph. If we use a tree structure to represent the thesaurus, we have to duplicate the concepts with more than one parent node and have to let them appear at multiple locations in the tree. In this work, we use a lot of techniques that are based on a monohierarchic tree structure. For example, we use tree based distance measures which involve the depth of a node in the tree. As the depth depends on the path length from the root node to the concept in question, we have no unique depth in case of multiple paths.

A similar problem is to determine the least common subsumer, as there can be multiple independent subsumers on the different paths. Giving the benefit of the doubt, we decided to use always the closest LCS candidate with respect to the distance and to define the depth of a node in a polyhierarchic structure as the shortest possible distance to the root node. This can be easily achieved by traversing the graph with breadth-first-search [CLR90, 470 ff.].

3 Evaluation of the indexing process

The problem that initially led to this thesis was the planned use of the Collexis Engine in a search application for economic literature. As there are promising results with the MeSH Thesaurus and documents from the medical area, we focused on a comparison with these results.

We assumed that the results would not be as good as in the medical area for the following reasons:

1. The used STW Thesaurus is mainly a German thesaurus. The concepts consist of many German synonyms, but only one corresponding English term. As a lot of the literature is written in English, the thesaurus should to miss a lot of important English synonyms used in the documents.

2. Compared to the medical area, the economic area is more diversified. There are a lot of documents that cover additional areas like economic geography or economic psychology. So the use of additional more specific thesauri could be adequate.

3. In medical documents there is a widely used and well known terminology. Especially the different disease patterns and affected organs are generally identified with one out of few non-ambiguous terms. In medicine it is very important to find all relevant literature to a special case and to compare the results of different therapies all over the world. Classifications like the International Classification of Diseases (ICD) were introduced for that reason. Though there is a well known and defined terminology in the economical area as well, it is supposed to be not used that strictly since the use of the terminology and the comparability of results is not that vitally important.

In order to validate our assumptions and to develop techniques to evaluate a given thesaurus for a document base, at first we focused on the evaluation of the indexing process by comparing the extracted concepts with manually selected keywords by human experts.

The following randomly selected article is used as an example:

Concept
Price
Cigarette
Panel
Regulation
State tax
Tax increase
Retail price
State
Oligopoly

Table 3.1: Concepts found by the indexer

Keyword
Pricing behaviour of firms
Oligopoly
Effects of taxation
Tobacco tax
Cigarette industry

Table 3.2: Manually selected keywords

Title Do cigarette producers price-discriminate by state? An empirical analysis of local cigarette pricing and taxation.

Authors Theodore E. Keeler, Teh-wei Hu, Paul G. Barnett, Willard G. Manning, Hai-Yen Sung

Abstract This study analyzes the interactive effects of oligopoly pricing, state taxation, and anti-smoking regulations on retail cigarette prices by state, using panel data for the 50 US states between 1960 and 1990. The results indicate that cigarette producers do price-discriminate by state, though the effect is not large relative to the final retail price. There are two further results: (1) state taxes are more than passed on - a 1-cent state tax increase results in a price increase of 1.11 cents, and (2) sellers offset state and local anti-smoking laws with lower prices, thereby blunting effects of the regulations.

Journal Journal of Health Economics

Table 3.1 shows the concepts found by the indexer, the manually selected keywords are shown by Table 3.2.

3.1 Evaluation Approach

Performance evaluations are an essential part of developing semantic search applications. Beside measuring the overall quality of the search results, it is important to find metrics to evaluate the single tasks like preprocessing (stemming, normalization steps), indexing and searching.

The general idea for the evaluation of the indexing process is to calculate Precision and Recall (as described in Section 2.3) of the automatically found concepts compared to a gold standard of manually selected keywords by human experts.

We expect that the indexing process finds at least the same keywords as selected by humans if the keyword appears in the abstract. Additional concepts should be found due to the fact that every concept appearing in the abstract is found, regardless of relevance.

The questions to be answered with this evaluation are:

- Is the quality of the automatic indexing process with the STW thesaurus comparable with the quality of the process with the MeSH thesaurus?
- Can we improve the quality if we improve the thesaurus and/or the indexing process according to the results of a detailed analysis of the found problems?

3.1.1 Experimental Setup

For this evaluation, we use two document corpora. On the one hand, there is a randomly selected collection of 706 Medline abstracts [NLM07b]. These articles were annotated with the MeSH thesaurus [NLM07a]. SyynX Solutions provides an own search interface for the Medline database, the SyynX Mediator [Sol07]. Practical experiences with the Mediator have shown that the search engine provides excellent results with this document base. The Medline database contains keywords from the MeSH thesaurus selected by human experts.

On the other hand, we have a document base of 391 economic abstracts, provided by Elsevier B.V. [Els07], indexed with the German-English Standard Thesaurus Wirtschaft (STW) [STW07]. We fetched manually selected STW keywords from the Econis Database of the German central economic library (Deutsche Zentralbibliothek für Wirtschaftswissenschaften) [ZBW07]. The abstracts are from three different journals:

- Journal of Health Economics (JHE)
- Journal of Accounting and Economics (JAE)
- Journal of Financial Economic (FINEC)

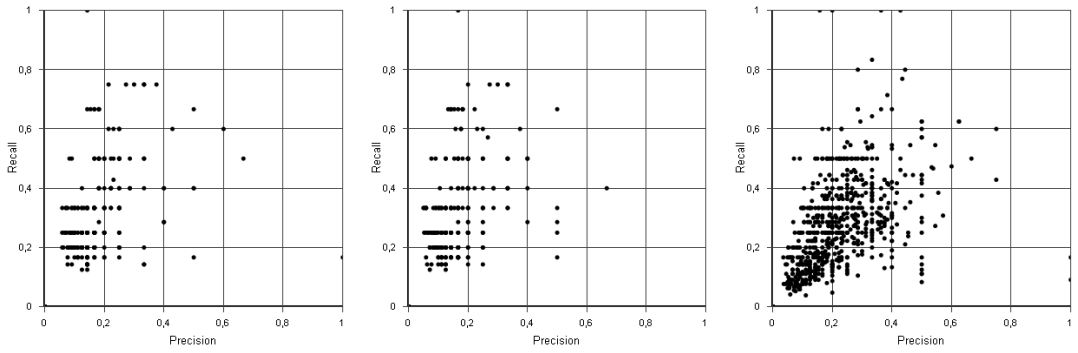
With this experiment, we focus on two aspects. First of all, we want to compare the indexing results with manually selected keywords regarded as a gold standard. Second, we compare these results between the Medline/MeSH articles and the Elsevier/STW articles. As the Medline/MeSH document base is used in real-world solutions, we assume, that this comparison gives us an indication of the practicability of the STW thesaurus for document retrieval in the Elsevier document base.

3.1.2 Evaluation Procedure

Table 3.3 shows the average Precision and Recall for the two document bases used. STW (broken) refers to the broken implementation, as described in Section 2.5.2. With our Medline reference set, only 25% of the human selected keywords are found by the indexer. The result of the Elsevier base is with 18% even worse.

Document Base	# Documents	# Keywords	# Correct	Recall
STW (broken)/Elsevier	391	1658	293	0.18
STW/Elsevier	391	1646	309	0.19
MeSH/Medline	706	8143	2030	0.25
		# Concepts	# Correct	Precision
STW (broken)/Elsevier	391	2980	293	0.1
STW/Elsevier	391	3377	309	0.09
MeSH/Medline	706	10041	2030	0.20

Table 3.3: Binary Precision and Recall results



(a) STW/Elsevier Binary (b) STW2/Elsevier Binary (c) MeSH/Medline Binary

Figure 3.1: Binary Precision and Recall results

Looking at the graph that shows the Precision and Recall of all documents (Figure 3.1), we see, that the results are mostly located in the bottom left quarter, thus indicating a bad result. Only few samples have either a good Recall or a good Precision, no samples show good results for both of them. The Precision is generally lower than the Recall, as there are more found concepts than human selected keywords.

As we can see in Table 3.4, there are a lot more manually selected keywords per document in the Medline document base than in the Elsevier one (11.5 vs 4.2). Whereas there are more found concepts, too (14.22 vs 7.6), the ratio of keywords to concepts clearly shows, that the ratio is better for the Medline document base. Even with an average Recall of 1, the highest reachable Precision would be 0.55 for Elsevier vs 0.81 for Medline.

An examination of the indexing results shows that the binary approach judging a found concept as correct or wrong with respect to the given keywords is not appropriate: a human can use more abstract keywords not used in the text. For example, in some contexts the keyword *Asia* might get assigned by a human, whereas *China* and *Japan* are used in the text. Table 3.1 and Table 3.2 are a good example for this: In the abstract, the common concepts *Price* and *Retail price* are found, whereas the human selected keywords contain the more specific *Pricing behaviour of firms.*

Document Base	# Documents	# Keywords	∅ Keywords
STW (broken)/Elsevier	391	1658	4.2
MeSH/Medline	706	8143	11.5
		# Concepts	∅ Concepts
STW (broken)/Elsevier	391	2980	7.6
MeSH/Medline	706	10041	14.22
	Ratio of keywords to concepts		
STW (broken)/Elsevier		0.55	
MeSH/Medline		0.81	

Table 3.4: Concept and keyword statistics

To reflect these circumstances, we need a metric to decide, how wrong a found concept is with respect to the given keywords.

3.1.3 Generalized Precision and Recall

If we use a metric to weight the correctness of a found concept, we need a new definition of Precision and Recall. This new definition is no longer set-based, but uses the weight function to quantify the correctness of a set of concepts.

Algorithm 1 outlines the alignment of each keyword with a concept and vice versa.

The Recall is calculated as

$$Recall = \frac{\sum_{k\ in\ keywords}^{n} M_K[k]_w}{|keywords|} \tag{3.1}$$

whereas the Precision reads as

$$Precision = \frac{\sum_{c\ in\ concepts}^{n} M_C[c]_w}{|concepts|}. \tag{3.2}$$

If the weight function $weight(c, k)$ returns 1 in case of equality, respectively 0 in case of inequality, we get the common Precision and Recall.

Hahn and Schnattinger proposed a similar measure to determine the correctness of the position of newly added concepts to a thesaurus hierarchy, called Learning Accuracy, in 1998 [HS98]. Other approaches which focus on the comparability with the traditional Precision and Recall use terms like Generalized Precision and Recall, Augmented Precision and Recall or Semantic Precision and Recall.

Our generalization builds upon the work of Kekäläinen and Järvelin [KJ02] ($r(d_i)$corresponds to our $weight(c, k)$ function):

> The generalized, non-binary recall and precision are defined as follows. Let R be the set n documents retrieved from a database $D = d_1, d_2, ..., d_N$ in response

Algorithm 1 Aligning concepts and keywords

Given:
- A set of found concepts: C
- A set of concepts representing the keywords: K

Define:
- Candidates $Cand$ as objects
  containing a keyword $Cand_k$, a concept $Cand_c$
  and a weight $Cand_w$
- A map M_K that assigns a
  candidate concept to every keyword
- A map M_C that assigns a
  candidate keyword to every concept
- A weight function $weight(c, k)$ that calculates a weight
  for a given concept and a keyword

---

foreach (Keyword k in K) {
  foreach (Concept c in C) {
    new $Cand$ $cand$
    $cand_c = c$
    $cand_k = k$

    $cand_w = weight(w, k)$

    if ($M_K[k]_w < cand_w$): $M_K[k] = cand$
    if ($M_C[c]_w < cand_w$): $M_C[c] = cand$
  }
}

to a query on some topic, $R \subseteq D$. Let the documents d_i in the database have relevance scores $r(d_i)$, being real numbers ranging from 0.0 to 1.0 with as many intermediate points as used in the study, with respect to the request behind the query. Generalized recall gR and generalized precision gP may now be computed by:

$$gP = \frac{\sum_{d \in R} r(d)}{n} \tag{3.3}$$

$$gR = \frac{\sum_{d \in R} r(d)}{\sum_{d \in D} r(d)} \tag{3.4}$$

A generalization by Euzenat to evaluate ontology alignments can be found at [Euz07] (here, $\omega(A, B)$ corresponds to our $weight(c, k)$ function, but is expressed on the whole set):

For a reference alignment B and an overlap function ω between alignments, the precision of an alignment A is given by

$$P_\omega(A, B) = \frac{\omega(A, B)}{|A|} \tag{3.5}$$

and recall is given by

$$R_\omega(A, B) = \frac{\omega(A, B)}{|B|}. \tag{3.6}$$

For a true generalization, ω has to share some properties with $|R \cap A|$:

The measure has to be positive:

$$\omega(A, B) \geq 0 \quad \forall A, B \qquad (positiveness)$$

and has not to exceed the minimal size of both sets:

$$\omega(A, B) \leq min(|A|, |B|) \quad \forall A, B \qquad (maximality)$$

With these constraints the given values are within the unit interval $[0\ 1]$. Further, this measure should only add more flexibility to the usual precision and recall so their values cannot be worse than the initial evaluation:

$$\omega(A, B) \geq |A \cap B| \quad \forall A, B \qquad (boundedness)$$

Hence, the main constraint faced by the proximity is:

$$|A \cap B| \leq \omega(A, B) \leq min(|A|, |R|)$$

This is indeed a true generalization because $\omega(A, R) = |A \cap R|$ satisfies all these properties.

As third approach, we have a closer look at the Augmented Precision and Recall presented by Maynard et al. [May05] [MPL06]: They use a balanced distance metric BDM_i, which corresponds to the relevance score $r(d)$ of Kekäläinen and Järvelin and our $weight(c, k)$ function. The sum of all distances in a given set $BDM = \sum_{i=1..n} BDM_i$ corresponds to the overlap function $\omega(A, R)$ of Euzenat and is used in our implementation as well.

With this distance summation, Maynard et al. calculate the Augmented Precision and Recall as

$$AP = \frac{BDM}{BDM + Spurious} \tag{3.7}$$

and

$$AR = \frac{BDM}{BDM + Missing}. \tag{3.8}$$

These definitions correspond well to the traditional definitions of Precision and Recall (Equation 2.2 and Equation 2.3), but require to identify spurious and missing concepts. In this case, the authors used a BDM_i value of 0 as criteria for a missing respectively spurious concept. But as we will see, there are similarity measures, that never assign a value of 0 to two given concepts, as long as they both exist in the thesaurus.

Considering the subsets $correct$ and $missing$, respectively $correct$ and $spurious$ as the sets of $relevant\,objects$ and $found\,objects$ (Figure 2.3), the generalization of Euzenat corresponds as well.

After this excursus about the theoretical background of the generalization used, we now turn back to our definition and implementation. In the next section, we show the results for different weight functions.

3.2 Results

The results of this evaluation depend strongly on the weight function used. We tested several widely known and well examined similarity measures to determine the degree of correctness for a given concept and its nearest matching counterpart in the reference set. The next section describes the various measures.

Measuring the semantic similarity between words or documents is an important task for information retrieval and natural language processing. The idea behind semantic similarity is to define a metric, that says how similar two words or documents are.

Similarity depends strongly on human perception and intuition, like plausibility or typicality. It is hard to quantify similarity, even for human beings. For example, in some sense the words “car” and “gasoline” are considered similar (one is related to the other), in another sense “car” and “wheel” are considered similar (one is part of the other) and in another sense “car” and “bicycle” are considered similar (both are vehicles). It is not possible to find the best definition of similarity, for each similarity you can think about an application that would prefer this one over the other.

The above mentioned examples lead to the three relations commonly used in thesauri:

1. related-to relationship
2. part-of relationship
3. is-a relationship

Keyword	Matched Concept	Similarity
Pricing behaviour of firms	Retail price	0.86
Oligopoly	Oligopoly	1
Effects of taxation	Oligopoly	0.46
Tobacco tax	State tax	0.44
Cigarette industry	Price	0.23

Table 3.5: Keywords and matched concepts with highest similarity (Intrinsic Jiang Conrath)

So, one general approach for calculating semantic similarity is using a thesaurus and finding some distance measure in this thesaurus. The structure of the thesaurus and the relationship used to build the thesaurus provide the focus of the resulting similarity. Similarity measures using this approach are referred to as network-based, thesaurus-based or ontology-based semantic similarity measures.

To illustrate the values found by such measures, Table 3.5 shows some results from our experiments.

3.2.1 Thesaurus Based Measures

Thesaurus-based measures generally use a distance measure within the thesaurus tree to determine the degree of similarity. The simplest approach would be using the distance between two nodes. The distance is defined by the number of nodes or edges on the shortest path between two nodes. This is called node-counting or edge-counting.

Leacock and Chodorow. Leacock and Chodorow use node-counting for their similarity measure presented in [LC98]:

$$Similarity_{LC}(c_1, c_2) = -\log \frac{distance}{2 \cdot maxdepth} \tag{3.9}$$

where $distance$ is the node-counting distance between the two concepts and $maxdepth$ is the maximum depth of the thesaurus. As $2 \cdot maxdepth$ is the longest possible distance, the values of this measure range from 0 to $\log(2 \cdot maxdepth)$.

We normalized this measure by dividing by the maximum possible value:

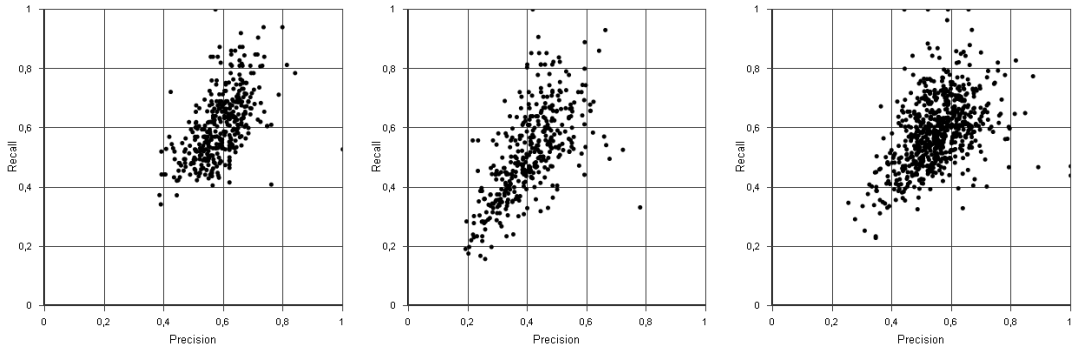
$$Similarity_{LCNORM}(c_1, c_2) = \frac{Similarity_{LC}(c_1, c_2)}{\log(2 \cdot maxdepth)} \tag{3.10}$$

After normalization, this measure has the value 1, if and only if both input concepts are the same. This holds for the following measures as well.

As Table 3.6 and Figure 3.2 indicate, the best results are achieved using the broken STW implementation (see Section 2.5.2). This implementation lacks the additional top-level hierarchy of the single sub-thesauri and is thus a rather flat hierarchy. As the Leacock Chodorow measure does not punish the involvement of the root node as least common subsumer, all the concepts near the root of a subthesaurus are considered very similar.

Document Base	# Documents	# Keywords	∑Similarity	Recall
STW (broken)/Elsevier	391	1658	1005	0.61
STW/Elsevier	391	1646	849	0.52
MeSH/Medline	706	8143	4754	0.58
		# Concepts	∑Similarity	Precision
STW (broken)/Elsevier	391	2980	2980	0.59
STW/Elsevier	391	3377	1371	0.41
MeSH/Medline	706	10041	4578	0.55

Table 3.6: Generalized Precision and Recall (Leacock Chodorow)

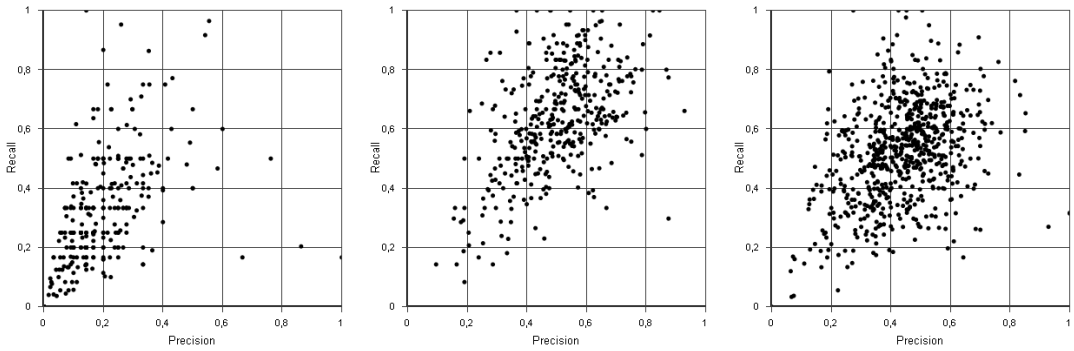


(a) STW Leacock Chodorow (b) STW2 Leacock Chodorow (c) MeSH Leacock Chodorow

Figure 3.2: Generalized Precision and Recall (Leacock Chodorow)

Document Base	# Documents	# Keywords	∑Similarity	Recall
STW (broken)/Elsevier	391	1658	388	0.23
STW/Elsevier	391	1646	1045	0.63
MeSH/Medline	706	8143	4380	0.54
		# Concepts	∑Similarity	Precision
STW (broken)/Elsevier	391	2980	420	0.14
STW/Elsevier	391	3377	1688	0.5
MeSH/Medline	706	10041	4200	0.52

Table 3.7: Generalized Precision and Recall (Wu Palmer)



(a) STW Wu Palmer (b) STW2 Wu Palmer (c) MeSH Wu Palmer

Figure 3.3: Generalized Precision and Recall (Wu Palmer)

Wu and Palmer. A normalized measure with values between 0 and 1 was presented by Wu and Palmer 1994 [WP94]:

$$Similarity_{WP}(c_1, c_2) = \frac{2 * depth(LCS)}{depth(c_1) + depth(c_2)} \tag{3.11}$$

They used the least common subsumer (LCS) of two concepts, which is the most specific concept (with the highest depth) in the tree that has both input concepts as child nodes. As the root node has a depth of 0 in our implementation, this measure always assigns 0 to concepts from different subtrees of the root node. So the results indicate clearly the weakness of the broken STW implementation (Table 3.7 and Figure 3.3).

3.2.2 Information Based Measures

There is a problem with the above described measures. They assume, that the nodes in the thesaurus have the same distance according to similarity. Or in other words, with these measures, two concepts have the same degree of similarity if they have the same distance in the thesaurus.

In real-world thesauri, there is a wide variability in the semantic similarity between adjacent nodes, especially, if the thesaurus is combined from different source thesauri. Generally there

are areas with a high density of differentiating concepts and other areas where only some common concepts exist.

To overcome this problem, Philip Resnik introduced a new way to measure the semantic similarity of words, based on the notion of information content [Res95]:

$$IC(c) = -\log P(c) \tag{3.12}$$

The information content of a given concept is derived from its probability to encounter an instance of this concept or one of its child concepts in a document base. So, the probability is monotonic increasing as one moves up in the thesaurus hierarchy. If c_2 is a child concept of c_1 then $P(c_2) < P(c_1)$. And with a single root thesaurus we have $P(root) = 1$.

In consistence with information theory, a concept has a higher information content, if its probability for being encountered is lower. So the information content of the root concept is 0.

Resnik. The similarity measure of Resnik reads as

$$Similarity_R(c_1, c_2) = \max_{c \in S(c_1, c_2)} (IC(c)) \tag{3.13}$$

with $S(c_1, c_2)$ denoting the set of subsumers from the LCS to the root concept.

As the information content decreases with the level of the concept in the thesaurus hierarchy, we can use the least common subsumer:

$$Similarity_R(c_1, c_2) = IC(LCS) \tag{3.14}$$

Note, that this simplification only holds for monohierarchical thesauri. In a polyhierarchical thesaurus, two concepts can have more than one LCS with different information content. In this case, again the maximum information content has to be chosen, like in Equation 3.13.

To implement Equation 3.12, Resnik used the frequency of a concept in a document base:

$$P(c) = \frac{\sum_{i \in C*} freq(i)}{N} \tag{3.15}$$

with $C*$ as the set of concepts subsumed by the given concept and $freq(concept)$ giving the number of occurrences of the concept. N denotes the count of words in the document base.

For further reference, we denote the sum of frequencies of a concept and its child node as

$$freq*(c) = \sum_{i \in C*} freq(i). \tag{3.16}$$

If the originating document base is not accessible, a reasonable choice for N would be

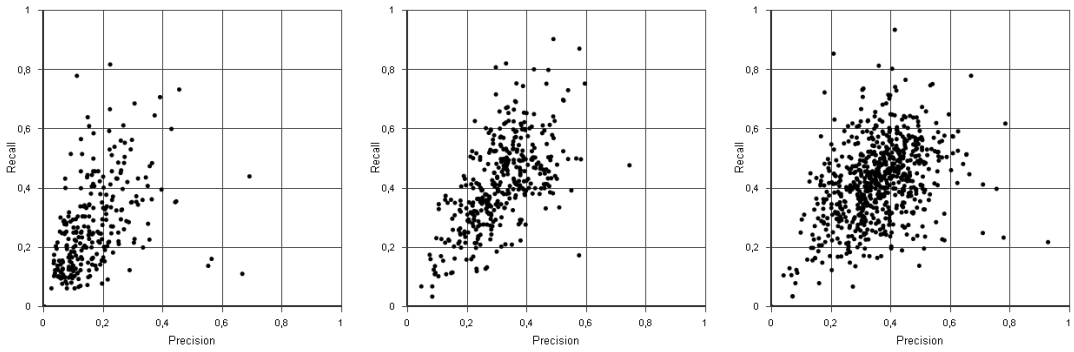
$$N = \sum_{c \in concepts} freq(c) \tag{3.17}$$

with $concepts$ as the set of all concepts in the thesaurus.

The application of this approach to natural language processing and its evaluation can be found at [Res99].

Document Base	# Documents	# Keywords	∑Similarity	Recall
STW (broken)/Elsevier	391	1658	316	0.19
STW/Elsevier	391	1646	686	0.42
MeSH/Medline	706	8143	3406	0.44
		# Concepts	∑Similarity	Precision
STW (broken)/Elsevier	391	2980	343	0.12
STW/Elsevier	391	3377	1049	0.31
MeSH/Medline	706	10041	3562	0.35

Table 3.8: Generalized Precision and Recall (Resnik)



(a) STW Resnik (b) STW2 Resnik (c) MeSH Resnik

Figure 3.4: Generalized Precision and Recall (Resnik)

Normalized Information Content. A drawback of Resnik's measure is, that the similarity values range from 0 (the root concept is LCS) to $\lim_{P(c)\to 0} IC(c) = \infty$. For our experiment, we normalize 3.12 to

$$IC_{norm}(c) = \begin{cases} \frac{-\log P(c)}{-\log(\frac{1}{\alpha N})} = -\frac{\log P(c)}{\log \alpha N} & freq*(c) > 0 \\ 1 & freq*(c) = 0 \end{cases} \tag{3.18}$$

with α as a weighting parameter for the special case of $freq*(c) = 0$. With $\alpha = 1$, a concept with a $freq*(c)$ of 1 has also an Information Content of 1. With increasing α, the Information Content of these concepts decreases.

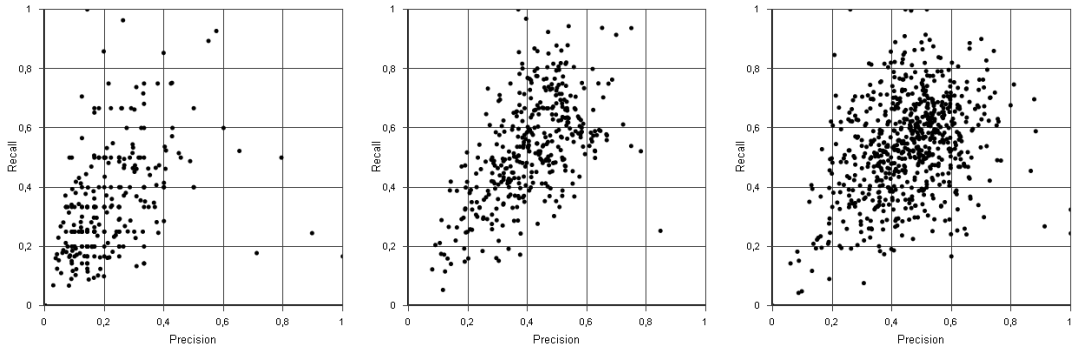
Table 3.8 and Figure 3.4 show the results using the normalized information content with Resnik's measure. As the information content of the least common subsumer is used as similarity value, most even identical concepts have an assigned similarity value far below 1. As the root concept has an information content of 0, at least the flat structure of the broken STW implementation is punished.

Lin. Lin introduced his measure 1998 [Lin98], which builds on Resnik's. It leads to a normalized value between 0 and 1:

$$Similarity_L(c_1, c_2) = \frac{2 \cdot IC(LCS)}{IC(c_1) + IC(c_2)} \tag{3.19}$$

Document Base	# Documents	# Keywords	∑Similarity	Recall
STW (broken)/Elsevier	391	1658	407	0.25
STW/Elsevier	391	1646	883	0.54
MeSH/Medline	706	8143	4347	0.53
		# Concepts	∑Similarity	Precision
STW (broken)/Elsevier	391	2980	445	0.15
STW/Elsevier	391	3377	1380	0.41
MeSH/Medline	706	10041	4578	0.46

Table 3.9: Generalized Precision and Recall (Lin)



(a) STW Lin (b) STW2 Lin (c) MeSH Lin

Figure 3.5: Generalized Precision and Recall (Lin)

Table 3.9 and Figure 3.5 show the results of the Generalized Precision and Recall with this measure. We also have the punishment of the broken STW due to the Information Content of 0 of the root node.

Jiang and Conrath. A very similar approach is used by Jiang and Conrath [JC97]. They introduce not a similarity, but a distance measure. Instead of counting the nodes between two concepts, they sum the link strengths between these nodes. This link strength is defined as the difference of the Information Content of a node and its parent node:

$$Distance_{JC}(c_1, c_2) = IC(c_1) + IC(c_2) - 2 \cdot IC(LCS) \tag{3.20}$$

This distance measure can also be used as similarity measure [CM05]:

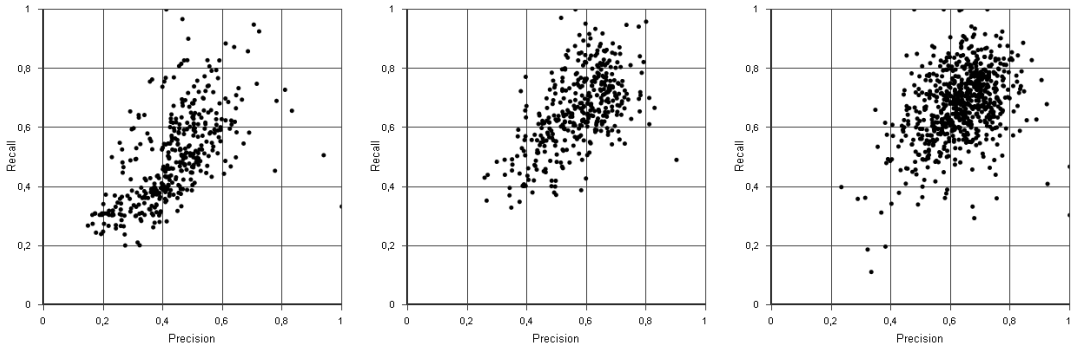
$$Similarity_{JC}(c_1, c_2) = \frac{1}{IC(c_1) + IC(c_2) - 2 \cdot IC(LCS)} \tag{3.21}$$

A linear transformed and normalized version of this measure can be found at [SVH04]:

$$Similarity_{JC}(c_1, c_2) = 1 - \frac{IC(c_1) + IC(c_2) - 2 \cdot IC(LCS)}{2} \tag{3.22}$$

Document Base	# Documents	# Keywords	∑Similarity	Recall
STW (broken)/Elsevier	391	1658	826	0.5
STW/Elsevier	391	1646	1092	0.66
MeSH/Medline	706	8143	5490	0.67
		# Concepts	∑Similarity	Precision
STW (broken)/Elsevier	391	2980	1279	0.43
STW/Elsevier	391	3377	1932	0.57
MeSH/Medline	706	10041	6371	0.63

Table 3.10: Generalized Precision and Recall (Jiang Conrath)



(a) STW Jiang Conrath (b) STW2 Jiang Conrath (c) MeSH Jiang Conrath

Figure 3.6: Generalized Precision and Recall (Jiang Conrath)

With this measure, the root node as LCS does not lead to a 0 result, but the influence is significant. Generally, this measure produces rather high values of Precision and Recall (Table 3.10 and Figure 3.6) compared to Lin.

3.2.3 Intrinsic Information Content

Nuno Seco, Tony Veale and Jer Hayes presented in [SVH04] an approach to determine the information content of a given concept without statistics from an underlying document base. Instead, they only used the thesaurus structure to define a measure for the information content. This so called Intrinsic Information Content is defined as

$$IIC(c) = -\log\left(\frac{hypo(c)+1}{max}\right) \tag{3.23}$$

and can be normalized to

$$IIC_{norm}(c) = \frac{\log\left(\frac{hypo(c)+1}{max}\right)}{\log\left(\frac{1}{max}\right)} = 1 - \frac{\log(hypo(c)+1)}{\log(max)} \tag{3.24}$$

with $hypo(c)$ as the number of hyponyms (i.e. child nodes) of a given concept c and max as the number of concepts in the whole thesaurus.

Document Base	# Documents	# Keywords	∑Similarity	Recall
STW (broken)/Elsevier	391	1658	401	0.24
STW/Elsevier	391	1646	862	0.52
MeSH/Medline	706	8143	4197	0.52
		# Concepts	∑Similarity	Precision
STW (broken)/Elsevier	391	2980	438	0.15
STW/Elsevier	391	3377	1331	0.39
MeSH/Medline	706	10041	4353	0.43

Table 3.11: Generalized Precision and Recall (Lin Intrinsic)

Document Base	# Documents	# Keywords	∑Similarity	Recall
STW (broken)/Elsevier	391	1658	702	0.42
STW/Elsevier	391	1646	999	0.61
MeSH/Medline	706	8143	5081	0.62
		# Concepts	∑Similarity	Precision
STW (broken)/Elsevier	391	2980	888	0.29
STW/Elsevier	391	3377	1621	0.48
MeSH/Medline	706	10041	5642	0.56

Table 3.12: Generalized Precision and Recall (Jiang Conrath Intrinsic)

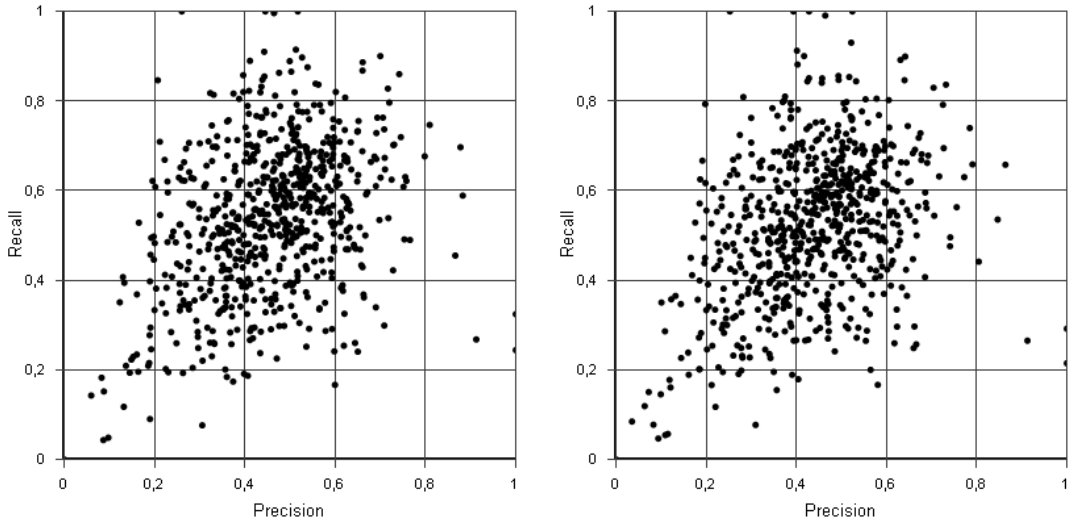
The Intrinsic Information Content can be used as a replacement for Information Content in the above mentioned measures.

The authors used Wordnet ([Fel98]) with very good results (see Table 3.13 in Section 3.2.4) and conclude that further experiments have to be done to see, if the intrinsic metric generalizes to other hierarchical knowledge bases.

We perform the same experiment with Lin and Jiang-Conrath measures and use the Intrinsic Information Content. The results (Table 3.11 and Table 3.12) are indeed comparable to the results with true Information Content. The Intrinsic Jiang Conrath values for the broken STW implementation are worse as with true IC Jiang Conrath. There are a lot of concepts in the broken STW with few or no child nodes. As this leads to a high value for the Intrinsic Information Content, the similarity values are lower than with true Information Content. In this regard, the measures with Intrinsic Information Content are more sensitive to an improper thesaurus structure.

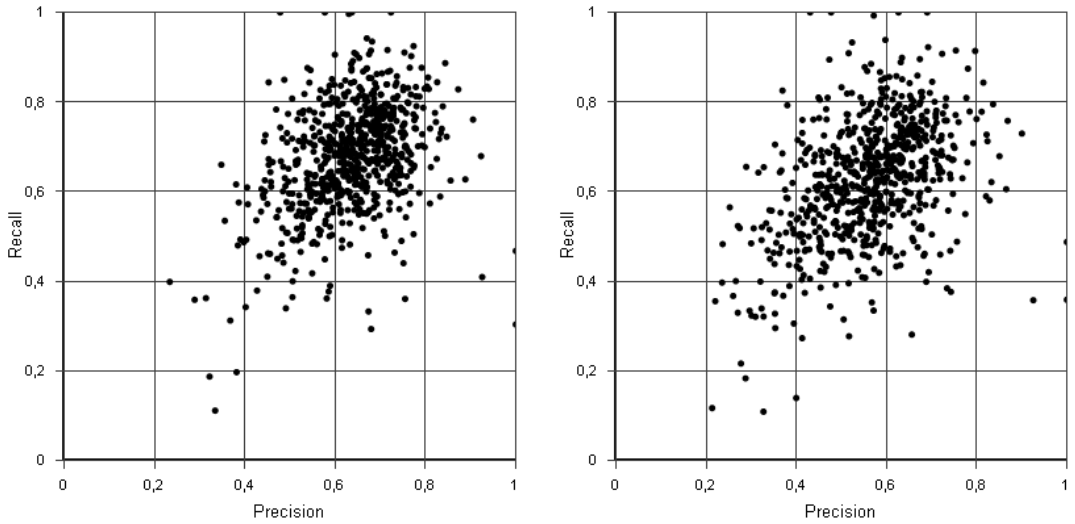
In Figure 3.7 and Figure 3.8 we compare the results of Information Content with Intrinsic Information Content for the MeSH Thesaurus. Indeed there is a high correlation in the two graphs and we can confirm the valuation of Seco et al.

Finally, we examine the graphs of the Jiang Conrath Measure for the broken STW implementation with true IC and IIC (Figure 3.9). The difference seen between the IC values of the broken STW (a) and the IIC values (b) can be an indication for the improper thesaurus structure in the broken STW. This idea is further developed in the next section.



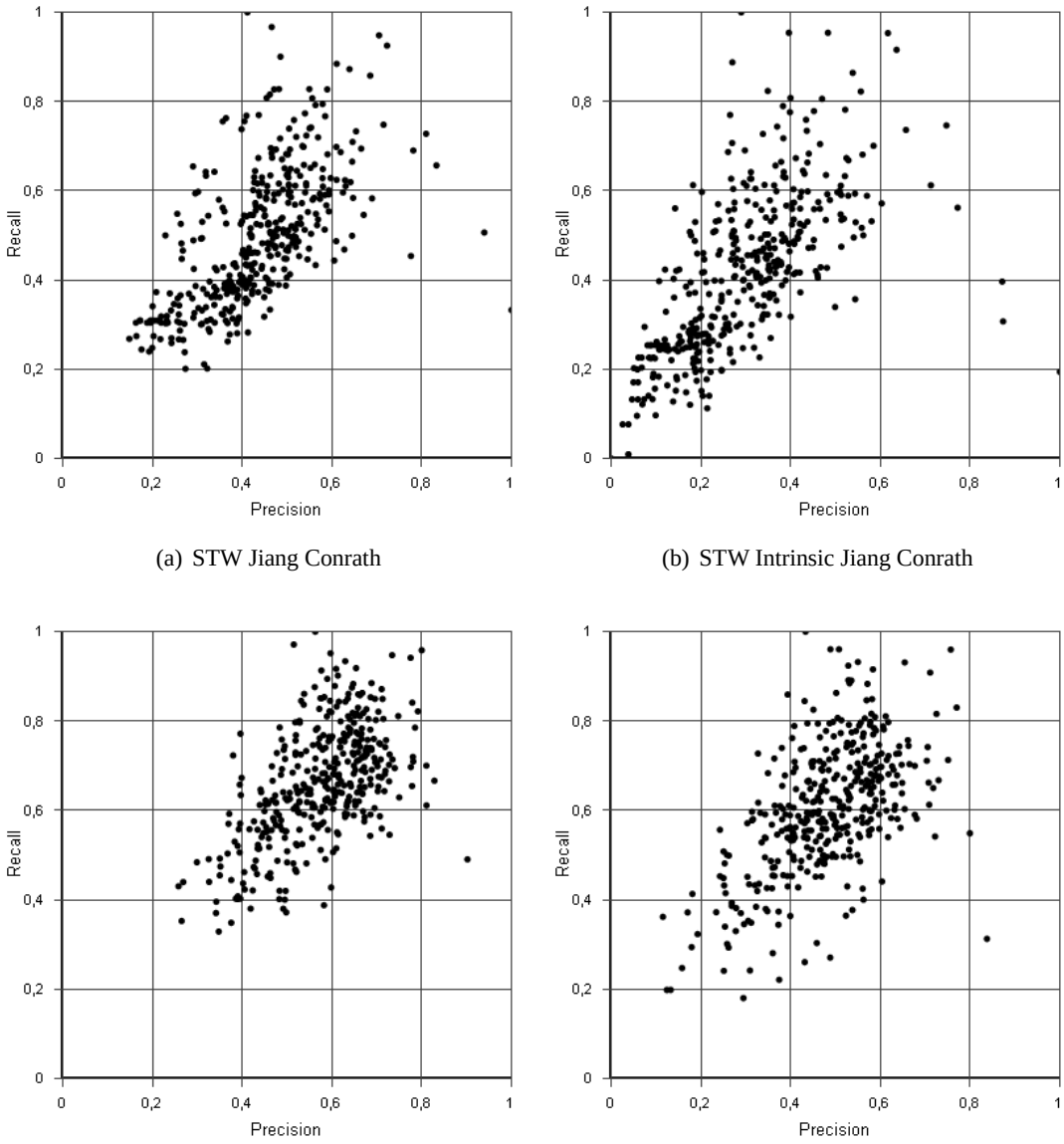
(a) MeSH Lin (b) MeSH Intrinsic Lin

Figure 3.7: Generalized Precision and Recall (MeSH, Lin, IC vs IIC)



(a) MeSH Jiang Conrath (b) MeSH Intrinsic Jiang Conrath

Figure 3.8: Generalized Precision and Recall (MeSH, Jiang Conrath, IC vs IIC)



(a) STW Jiang Conrath

(b) STW Intrinsic Jiang Conrath

(c) STW2 Jiang Conrath

(d) STW2 Intrinsic Jiang Conrath

Figure 3.9: Generalized Precision and Recall (STW, Jiang Conrath, IC vs IIC)

3.2.4 Conclusion

We were interested in a well-defined measure to evaluate our further analysis results. We needed to know if a change of the thesaurus motivated by our analysis results will lead to better indexing results. We argue that this relative comparison of two identical experimental setups (beside the changes) can be done with our evaluation, if an adequate similarity measure is chosen.

Choosing the right similarity measure. We evaluated various different similarity measures and their impact on the results of our Generalized Precision and Recall for the indexing process. The results depend strongly on the similarity measure used, so the preferred measure should be chosen carefully. A measure that does not punish the involvement of the root concept as LCS is not appropriate for our purpose. A similarity value of 0 for all concepts with the root node as LCS corresponds well with the traditional Precision and Recall, because these concepts can be seen as missing, respectively spurious.

In our opinion, a similarity measure is most convenient, if equality is denoted with 1, i.e. the values range between 0 and 1 with the value 1, if and only if a concept is compared to itself. The binary judgement of correct and incorrect matchings fits seamlessly to this kind of measure.

If equality is denoted with 1, it implies that every concept compared with itself should have the same degree of similarity. This equality consistence is an arguable property of a similarity measure; Resnik advanced his opinion about this criterion [Res99, S. 30]: “From a cognitive perspective, however, similarity comparisons involving self-similarity (Robins are similar to robins), as well as subclass relationships (Robins are similar to birds), have themselves been criticized by psychologists as anomalous (Medin, Goldstone, & Gentner, 1993). Moreover, experimental evidence with human judgments suggests that not all identical objects are judged equally similar, consistent with the information-content measure proposed here but contrary to Lin’s measure. For example, objects that are identical and complex, such as twins, can seem more similar to each other than objects that are identical and simple, such as two instances of a simple geometric shape (Goldstone, 1999; Tversky, 1977). It would appear, therefore, that insofar as fidelity to human judgments is relevant, further experimentation is needed to evaluate the competing predictions of alternative similarity measures.”

Table 3.13 shows the correlation of the different approaches to human judgement. These correlation values are taken from [SVH04]. It has to be noted, that the correlation between several humans judging the same contents by far is not 1. As said before, similarity is a very subjective property. Resnik performed a study with human subjects and specified the correlation with 0.88 [Res99, S. 7]. So, this can be seen as an upper bound of what can be reached with a calculated similarity measure.

There are a lot more approaches for measuring semantic similarity. A complete overview goes beyond the scope of this work, but for for example there is a measure using neural networks, proposed by Li, Bandar and Mclean [LBM02]. An extension to the similarity of concept sets or full texts can be found at [BKKB05] and [CM05]. Bernstein et al. suggest, that the choice of a similarity measure depends on the underlying thesaurus. They propose a personalized measure adapted to the thesaurus in [BKBK05].

Algorithm	γ
Leacock Chodorow	0.82
Wu and Palmer	0.74
Resnik	0.77
Lin	0.80
Jiang and Conrath[a]	−0.81
Resnik*[b]	0.77
Lin*	0.81
Jiang and Conrath*	0.84

[a] distance measure
[b] the * denotes the use of the intrinsic information content (Section 3.2.3)

Table 3.13: Correlation between human and machine similarity judgements (Source: [SVH04])

With the above criteria and our results in mind, we prefer the Lin measure to calculate the Generalized Precision and Recall in our context. On small document sets, we tend to the variant with intrinsic information content, but with large sets, the original measure with information content should be used to reduce the dependency on the quality of the thesaurus structure.

The quality of STW. In our case, we did not judge the indexing process itself, but the thesaurus. The MeSH thesaurus is used in real world applications with the Collexis Engine and the Medline publications as document base. We were interested, whether the Precision and Recall of the indexed concepts with respect to manually selected keywords could indicate a different thesaurus with different documents would perform similar.

With only these indexing results a conclusion about the quality should be made very carefully. In the end only the evaluation of search results by human experts can give an answer, whether the application performs well, but that is not the scope of this work. Such evaluations are expensive and time consuming, which is why we try to find techniques to quickly judge a thesaurus with respect to a document base.

Nevertheless, the indexing process performed more or less the same with both thesauri. The significant worse results of the broken STW implementation show that the involved thesaurus indeed makes a difference. Bearing in mind that we indexed only English abstracts and the STW is primarily a German thesaurus, we are very optimistic that the Collexis Engine will perform very well with the STW thesaurus on German/English economic literature.

Measuring improvements. Important for our work is the possibility to use the developed performance measure to evaluate the impact of improvements made due to our further analysis approaches, which are introduced in the next chapter.

4 Thesaurus Analysis

In this chapter, we present a new approach to a semi-automatic thesaurus analysis and develop an analysis and visualization tool to provide human experts a quick overview on a thesaurus and a corresponding document base. We perform different analysises based on this approach and evaluate the impact of our results on the indexing process with the performance measure developed in the last chapter.

4.1 Analysis Approach

Our analysis approach consists of two different steps:

1. the detection of parts of the thesaurus that show an unexpected behavior and
2. a detailed inspection of such parts resulting in a decision whether this particular part has to be revised to better support the indexing process.

Literature on thesaurus creation and maintenance mentions a number of revisions that might be necessary including the following [Bur04]:

- adaptation of the thesaurus to changes in the vocabulary of the domain of interest by means of adding of new terms
- deletion and/or merging of rarely used terms
- splitting, extension or restriction of extensively used terms
- review of the thesaurus structure to avoid extensive subclassing

We add a new aspect to these traditional ones that arises with automatically indexing:

- identification of problematic concepts for the automatic indexer, i.e. concepts that are erroneous assigned due to misleading occurrences in the documents with improper sense

Whereas the first aspect cannot be achieved by our analysis, it supports all other revisions. The principle idea of our approach is to use statistical measures to identify suspicious concepts in the thesaurus. Suspicious concepts are concepts that could potentially be used as origin for a thesaurus improvement.

First of all, we can search for concepts which appear very often or very rarely in the document base. A very frequent concept could be split into more specialized concepts, if it is too common. Or there are too many synonyms describing the concept. It is also possible that the concepts should be removed completely from the thesaurus, because they have no significance in the concrete use case.

On the other hand, a rare concept could be joined with other concepts to a more common concept. Or it lacks synonyms which are used in the document base. And again, it could be advisable to remove a concept completely, if it is not significant.

Limits of the analysis. Every such analysis is always an analysis of a thesaurus in conjunction with a document base and perhaps a special use case. So we do not expect a result stating that a thesaurus is good or bad, but a result assessing the suitability of a thesaurus for a given document base. However, if the thesaurus is used as an established standard, the same analysis can give an overview on the (unknown) document-base and reveal focuses and spreading of the contents.

To decide on the concrete improvement that should be performed, almost always a domain expert is needed. In most cases the thesaurus, the domain, the document base and strategic considerations are involved and need to be respected. It is easier to find the cause and reasonable solutions without in-depth domain knowledge in the case of concepts that are especially problematic for the automatic indexing.

4.1.1 Thesaurus Suitability

In this section, we introduce our measure to evaluate the suitability of a thesaurus for a given document base. We already discussed the frequency of a concept and said that extraordinary frequent or rare concepts could indicate a problem with the thesaurus.

The decision whether a concept is used more often for indexing as expected depends on our expectation about the frequency it should occur. This in turn depends on the level in the thesaurus hierarchy it is situated. Generally, more common concepts should have a higher frequency than special concepts. To take this into account, we do not operate on the frequency directly, instead we propose the difference of Information Contents as a distance

$$D_{IC}(c) = IC(c) - IIC(c) \tag{4.1}$$

where $IC(c) = -\log P(c)$ is the information content of a concept c with respect to a given document set, as introduced in Section 3.2.2.

The IIC of a concept c denoted as $IIC(c)$ is defined as

$$IIC(c) = 1 - \log\left(\frac{hypo(c)+1}{max}\right)$$

with $hypo(c)$ as the number of hyponyms (i.e. child nodes) of a given concept c and max as the number of concepts in the whole thesaurus (see Section 3.2.3).

We use the Intrinsic Information Content as reference and compare it to the Information Content based on the concept frequency in the document base. We refer to this measure as *IC Difference Analysis.*

Equation 4.1 is related to the Kullback-Leibler Divergence used in information theory as a measure of the differences between two probability distributions p and q. It is defined as

$$D_{KL}(p||q) = \sum_i p(i) \log \frac{p(i)}{q(i)} \tag{4.2}$$

and can be seen as the expected value of D_{IC}: $D_{KL}(p||q) = \sum_i p(i) D_{IC}(i)$. The Kullback-Leibler Divergence is used to get an overall measure of the thesaurus suitability, instead of evaluating a single concept [CD04b, CD04a].

4.1.2 Thesaurus Structure

Additionally, we experimented with analysis techniques to evaluate the thesaurus structure. We assume that a well-engineered thesaurus has a more or less balanced structure. The depth of a concept should indicate how abstract it is and the concepts on the same depth level should have a similar number of child nodes. We introduce several measures to get information about the balance of a thesaurus:

Variance of Intrinsic Information Content. The Intrinsic Information Content introduced in Section 3.2.3 should be uniformly distributed in a balanced thesaurus. For a concept c, we can calculate the variance of the Intrinsic Information Content of its children:

$$\sigma^2_{IIC}(c) = \begin{cases} \frac{\sum_{i \in children} (\mu_{IIC}(c) - IIC(i))^2}{|children|} & N > 0 \\ 0 & N = 0 \end{cases} \tag{4.3}$$

with

$$\mu_{IIC}(c) = \sum_{i \in children} \frac{IIC(i)}{|children|} \tag{4.4}$$

and $children$ representing all direct child nodes of a given concept c.

Deviation of Intrinsic Information Content. To evaluate a certain concept, we can use its deviation of the Intrinsic Information Content:

$$\Delta_{IIC}(c) = \begin{cases} IIC(c) - \mu_{IIC}(parents) & parents \neq \emptyset \\ 0 & parents = \emptyset \end{cases} \tag{4.5}$$

with $parents$ representing the parents of the given concept. Note that there can be more than one parent in a polyhierarchic thesaurus (see Section 2.5.3). So the set of children consists of the union of all children of all parents.

Variance and Deviation of Height. Beside the Information Content, the height $H(c)$ of a subtree in the thesaurus hierarchy with concept c as root should correspond to its degree of abstraction. In a balanced thesaurus, we would expect that all subtrees represented by the children of a concept have more or less the same height. So along the lines of Equation 4.3 and Equation 4.5 we define $\sigma^2_H(c)$ and $\Delta_H(c)$.

4.1.3 Analysis on concept level

As far as we presented our analysis approach, we introduced measures to get some information about the quality of a given thesaurus in conjunction with a document base. Our approach is not the first one that points in this direction. Usually, these measures are used to quantify a special quality aspect for the whole thesaurus. Such an aggregation of information is useful to get a first idea about the suitability of the thesaurus.

But in an environment of a semi-automatic thesaurus improvement, we not only need an evaluation of the thesaurus as a whole, we need an analysis that points us to potential problems and weaknesses even more.

With this in mind, we transferred all of the above measures to the concept level. For example, the balance measure of the root concept evaluates the whole thesaurus and if we calculate the balance of all concepts in the thesaurus, we can find the most unbalanced concepts. This leads to the next problem: How to visualize such results in a user-friendly way.

4.2 Thesaurus Visualization

A major challenge in supporting thesaurus maintenance is to provide adequate tools that guides the user to potential problems in a thesaurus based on the measures described above. In particular we have to find a way to provide the user with a view on the thesaurus that encodes the overall structure as well as parts of the thesaurus and the suitability values for the different concepts in the thesaurus.

So we had to find a sophisticated method to visualize and browse a thesaurus. To be precise, our requirements are:

1. Visualization of the whole thesaurus to get an overview.
2. Ability to zoom into parts of the thesaurus to see the details.
3. Presentation of all relevant information to a selected concept. This information can be for example the preferred term, all terms, the depth of the concept or various analysis results.
4. Ability to find remarkable results by an intuitive presentation.

(a) Tree view component (b) Additional information via a tooltip

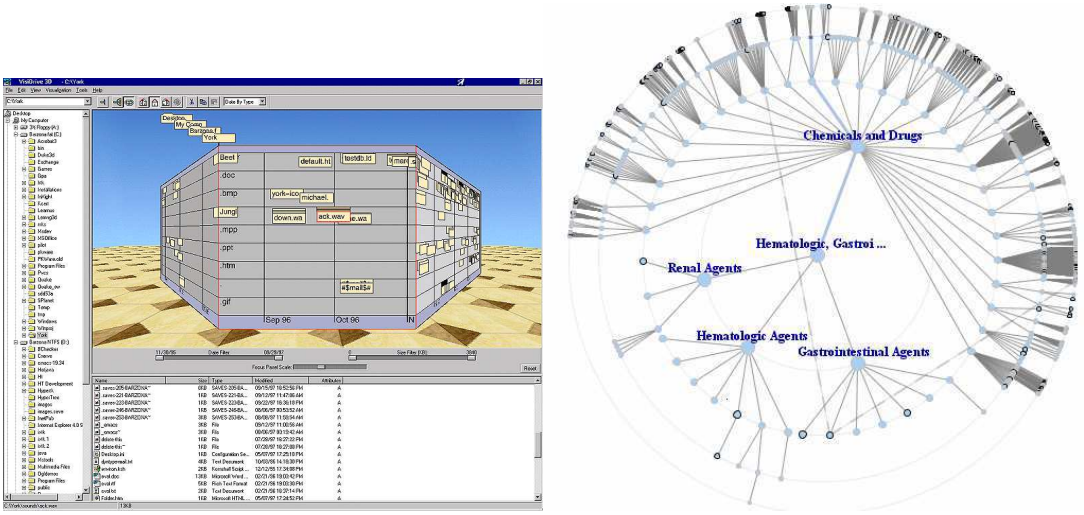
Figure 4.1: Tree View of the STW Thesaurus

4.2.1 Tree View and Lists

The obvious approach is a tree browsing component like in Figure 4.1 (a). In this case of a polyhierarchic thesaurus, the tree structure can be reached via an artificial root node with all of the multiple thesaurus roots as child nodes.

Such tree view components are very convenient and familiar for the user. Like in a file browser, one can expand and collapse the nodes and is able to get a quick overview over the thesaurus. Additional information can be given via tool-tips like in Figure 4.1 (b) or in an extra text field or table view. The drawback of this approach is that it is not easy to get an overview of this additional information. It is possible to put the information in the tree view (like the id in front of the preferred term in the example), but that is not very satisfying and a complete overview involves a lot of scrolling and expanding/collapsing of nodes.

To guide the user to the tree, some lists could be provided with the results of the thesaurus analysis like the most unbalanced or most frequent concepts. We experimented with these techniques, but we found them not really satisfying. Even if the results are quantified, it is hard to get a feeling of how serious the result of the analysis is. One can provide lots of information and the user still does not get the full understanding of the whole thesaurus.



(a) Perspective Wall (Source: unknown)

(b) Radial Tree (Source: [SC03])

Figure 4.2: Visualization Techniques 1

4.2.2 Graphical Visualization

These problems are addressed by many visualization approaches. They try to show local information in detail and at the same time the global context of the currently focused aspect. Figure 4.2(a) shows such a visualization, called Perspective Wall. It is not suitable for tree visualization, but illustrates the principle that the user gets the details while the global context is shown in a compressed and aggregated way.

A true tree visualization is shown by Figure 4.2(b). The user can navigate through the tree and the node of interest is always centered in the circle. The environment is arranged on concentric circles around the centered node.

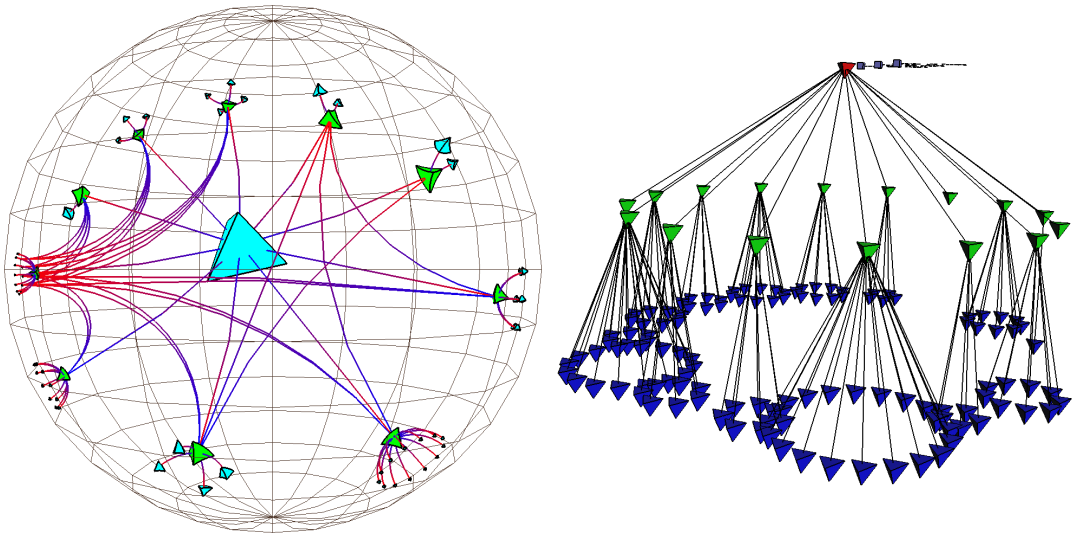
Figure 4.3(a) uses the same principle and extends the visualization to the third dimension. The sphere can be rotated by the user.

A more intuitive visualization of trees are the so called Cone Trees (Figure 4.3(b)). They use the traditional visualization of a tree and just add the third dimension to get more space. The child nodes are arranged on a circle and the whole tree has to be rotated to get the whole overview.

These visualizations are very good to get an overview of the tree structure of a thesaurus. But they still do not provide additional information in an intuitive way. Their strong point is the visualization of the environment of a single node. There is another way of visualization often mentioned along with the above visualization: the Treemap.

4.2.3 The Treemap

To get an overview of disc usage of a particular hard drive and its directory structure, Ben Shneiderman invented the Treemap algorithm in the early 1990s, published in [Shn92]:



(a) Hyperbolic Tree (Source: [MB95])

(b) Cone Tree (Source: [MB95])

Figure 4.3: Visualization Techniques 2

According to Shneiderman, Tree-maps are a representation designed for human visualization of complex traditional tree structures: arbitrary trees are shown with a 2-d space-filling representation.

Consider a tree with weight or size information attached to each node and a 2-d space with corners (x_1, y_1) and (x_2, y_2). Figure 4.4 illustrates the 2-d space. For each child of the root node, a partition of the space along the x-axis is calculated. For the first partition, this reads as

$$x_3 = x_1 + \left(\frac{|c_1|}{|r_1|}\right)(x_2 - x_1) \tag{4.6}$$

with $|c_1|$ as the size of child node 1 and $|r_1|$ as the size of the root node. For the next level, the corresponding partition is partitioned again along the y-axis, then again on the x-axis and so on. Shneiderman called this approach the “slice-and-dice” algorithm.

Since then, a lot of different implementations and optimizations were made by several people. One such optimization are the squarified treemaps, which try to avoid the long and thin rectangles resulting from small nodes. More sophisticated approaches are presented in [SW01] and [BSW02].

Marc Smith and Andrew Fiore used the treemap algorithm to visualize the Usenet newsgroups [SF01]. Their treemap component is used in our evaluation tool for thesaurus visualization.

With treemaps, information about two additional aspects can be displayed beside the tree structure. One is used to calculate the size of the partitions, the other is used to determine its color.

Figure 4.5 shows the treemap of the MeSH thesaurus, where each area represents a concept in the thesaurus. The hierarchy is visualized through the nesting of areas. The color of the

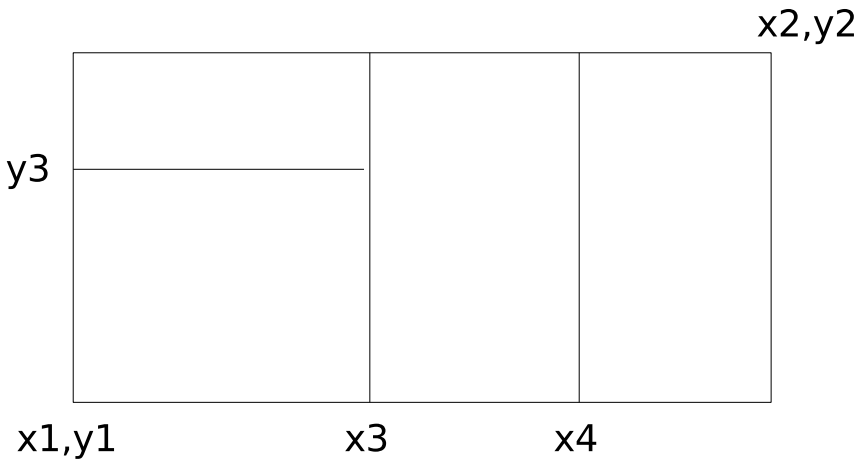


Figure 4.4: Treemap Algorithm

different areas is used to represent the result of the different measures introduced above. In Figure 4.5 the color corresponds to the intrinsic information contents of a concept. While the color of concepts that are low in the hierarchy and therefore have a intrinsic high information content lean towards a red color whereas concepts with a low intrinsic information content lean towards blue. The use of this representation and in particular the color coding for representing different measures is discussed in the following section.

Treemaps have proven their applicability in real life. In [Shn06a], Shneiderman demonstrates some usage examples of treemaps for business intelligence applications. He points out, that some training and experience is necessary to derive its maximum benefit. A complete history of treemaps and further readings can be found at [Shn06b].

With respect to our requirements, the treemaps perform very well. The only drawback is that you lose the concept's context, if you zoom into the thesaurus. But that is only a minor drawback and could be circumvented by providing an additional visualization, if this context is needed. In our implementation, we used a split view with the treemap for result visualization and a simple tree view for the context. Additionally, a textfield was used to provide some information to a selected concept.

Use in an interactive tool. The tool (Figure 4.6) that we developed for our experiments combines a hierarchical common treeview with the treemap visualization. This allows interactive navigation through the thesaurus hierarchy without losing the orientation. With the buttons in the upper right corner, all the different analysis techniques can be used to colorize the treemap.

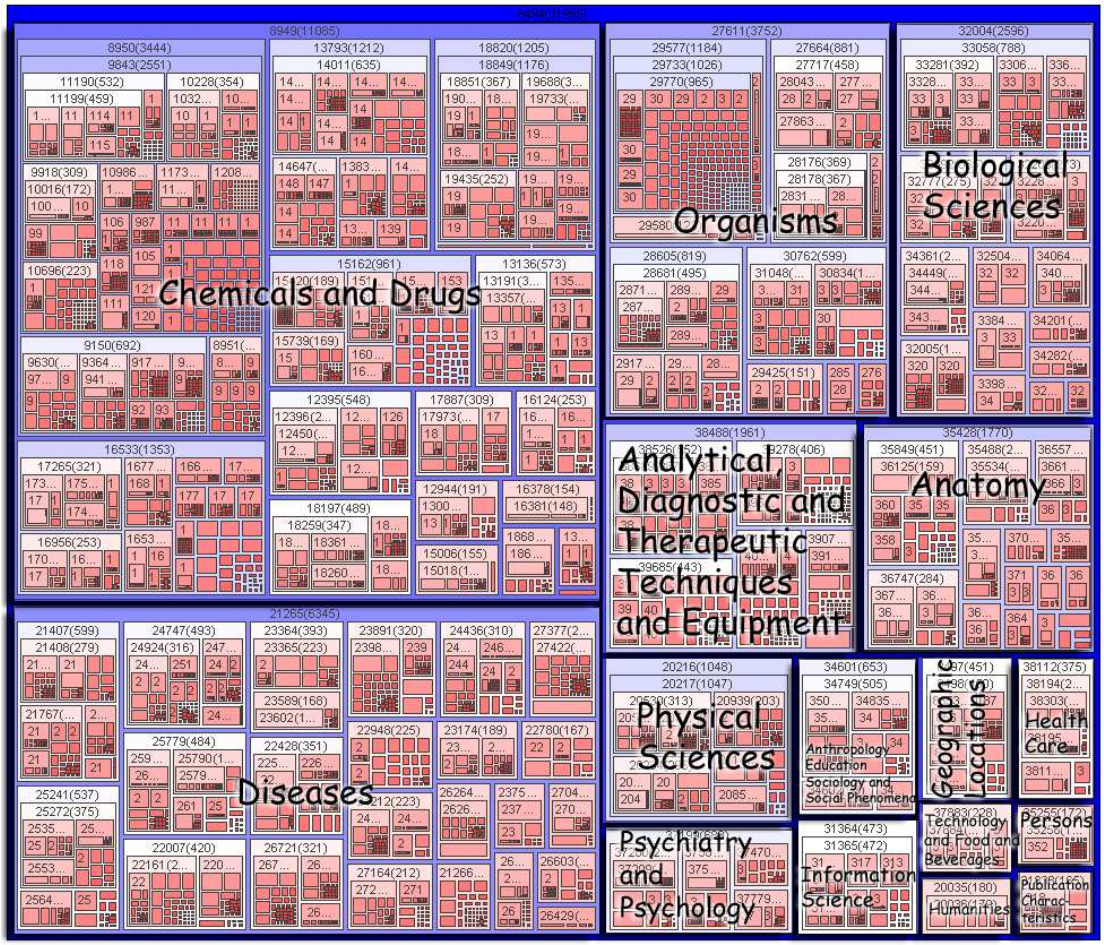


Figure 4.5: Treemap of the MeSH thesaurus

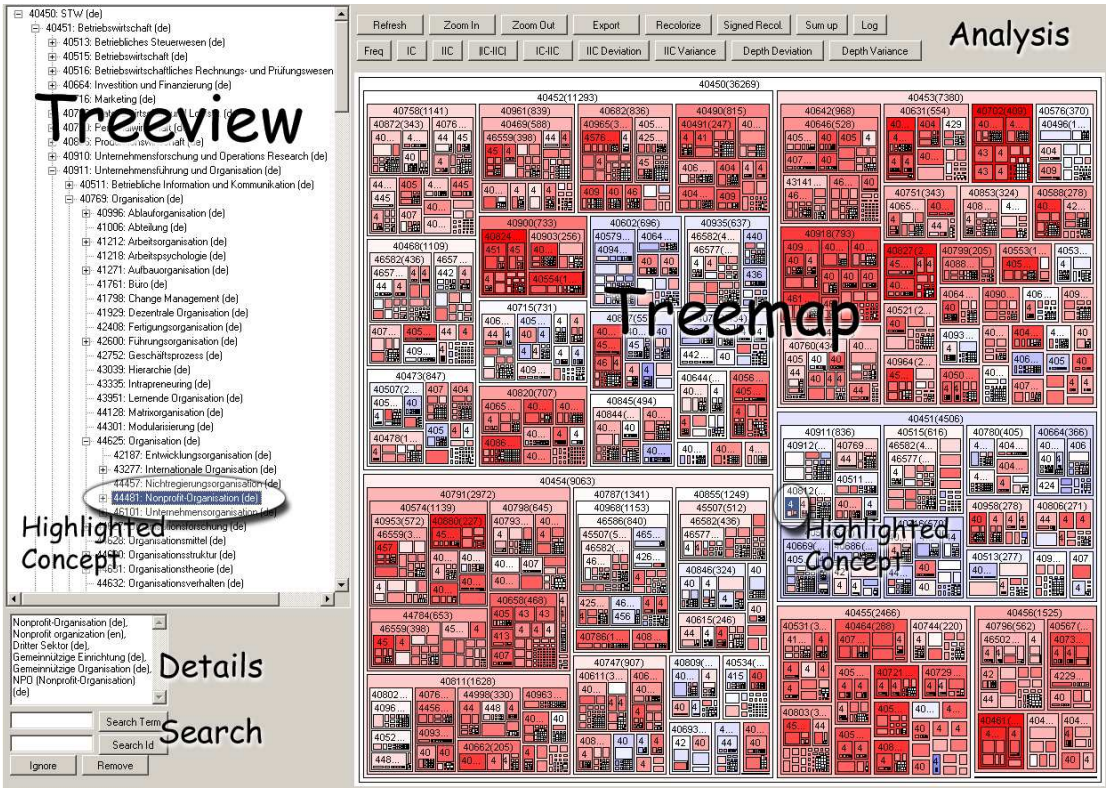


Figure 4.6: Interactive tool

4.3 Results

Using our tool, we analyzed the suitability of the MeSH and the STW thesaurus providing a basis for automatically indexing the document sets described in Section 3.1.1. In the following, we present the results of this analysis.

We show that the visualization of the IC Difference Analysis naturally leads us to parts of a thesaurus that cause trouble in the indexing process. In particular, we discuss a number of severe indexing errors that we detected using our tool. We also show that we can improve the indexing result by fixing the detected problems. The evaluation is by no means exhaustive. Our goal is to show that the methods proposed in this thesis work in principle.

4.3.1 Thesaurus Structure and Balance

First of all, we examine the thesaurus structure and balance. The treemap visualization is able to show the whole thesaurus at once. Some experience is necessary to interpret the results.

MeSH. Figure 4.5 shows the structure of the MeSH thesaurus. The intrinsic information content is used to colorize the concepts. As we generally use the sum of children for the size of the concepts, the colors correspond directly to the size, ranging from blue (most children, root node) over white to red (no children, leafs).

The 16 top concepts (MeSH Categories) are highlighted. It can be seen that they have different sizes, so the thesaurus is not equally balanced. The two largest categories are “Chemicals and Drugs” and “Diseases”. This is obvious as there has to be a fine grained distinction between all the available substances and their impact on the various known diseases.

STW. As described in Section 2.5.2, we at first used a broken implementation that lacks the top hierarchy above the vast number of subthesauri. Compared to Figure 4.5, the IIC visualization for the broken STW thesaurus (Figure 4.7) reveals a completely different image. A huge number of top concepts can be seen, some of them with a tree of subconcepts, as expected, but lots of them only with a small number of child nodes or even no child nodes at all.

Figure 4.8 is colorized by the height deviation $\Delta_H(c)$ (Equation 4.5) of every concept. This highlights the leaf nodes directly under the top node and indicates the problem of unbalance of the many isolated concepts.

We developed the structure measures (Section 4.1.2) to identify such weaknesses of a thesaurus and our experiments show that these measures work. For example, you could use them to calculate a list of the most unbalanced parts of the thesaurus. But with the treemap visualization, the defective structure is already visible without any colorization.

Next, we repeat the above experiment with the fixed implementation of the STW thesaurus. Figure 4.9 shows a by far more balanced view with the top concepts representing the first level of the subthesauri: *Political Economics* (Volkswirtschaft), *Related Disciplines* (Nachbarwissenschaften), *Industries* (Wirtschaftszweiglehre), *Business Economics* (Betriebswirtschaft),

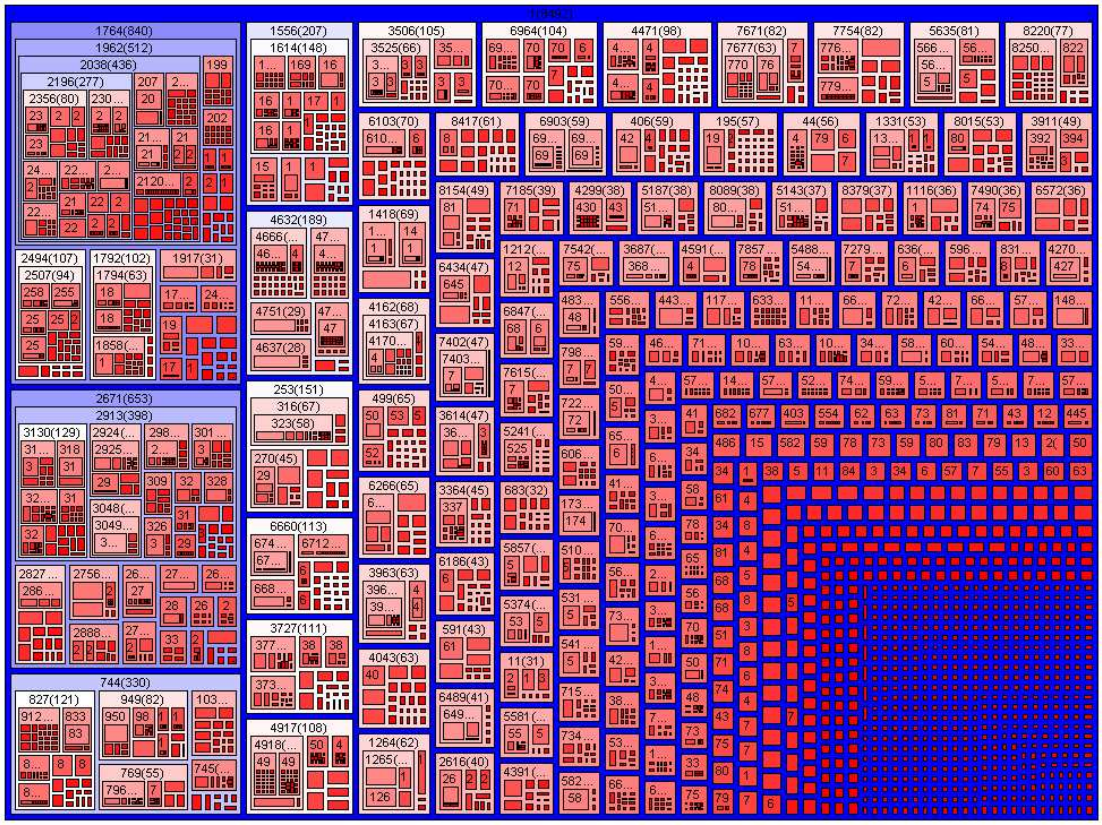


Figure 4.7: Broken STW: Intrinsic Information Content

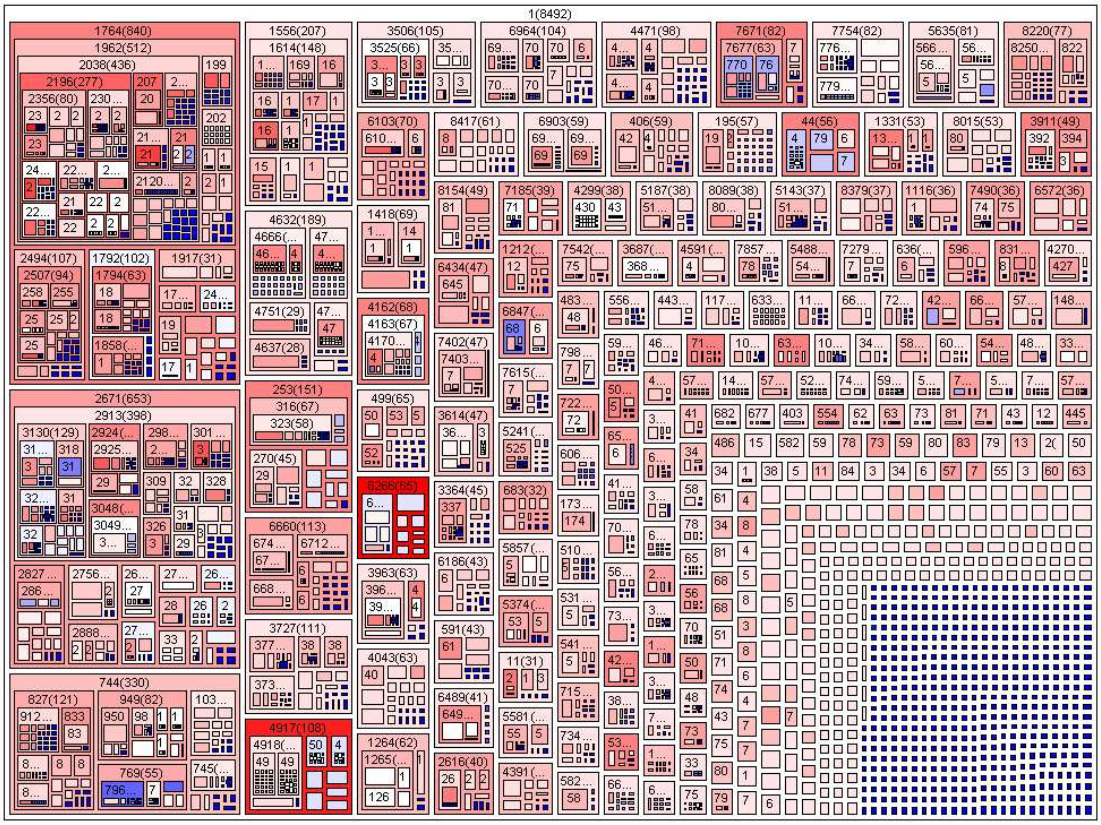


Figure 4.8: Broken STW: Height Deviation

Figure 4.9: STW: Intrinsic Information Content

Products (Produktteil), *Geographics* (Geographische Begriffe) and *Common Terms* (Allgemeinwörter). Note that the Common Terms only consists of 29 concepts and thus are hardly visible in the bottom right corner.

Figure 4.10 augments this visual impression with the statistical height deviation. The thesaurus is now better balanced, especially the top concepts show by their white color only minimal deviations in the height of the subtrees represented by them.

4.3.2 STW Analysis

We continue with an analysis of both the STW and the MeSH thesaurus (in the next section). These sections should convey an idea of the possibilities of our various analysis techniques, especially of the IC Difference Analysis.

Frequency Analysis. With a frequency analysis, we use the absolute frequency of documents containing the concept as color metric. For this analysis, the values should be logarithmized to get the most illustrative picture. Our tool provides the possibility to logarithmize all analysis results.

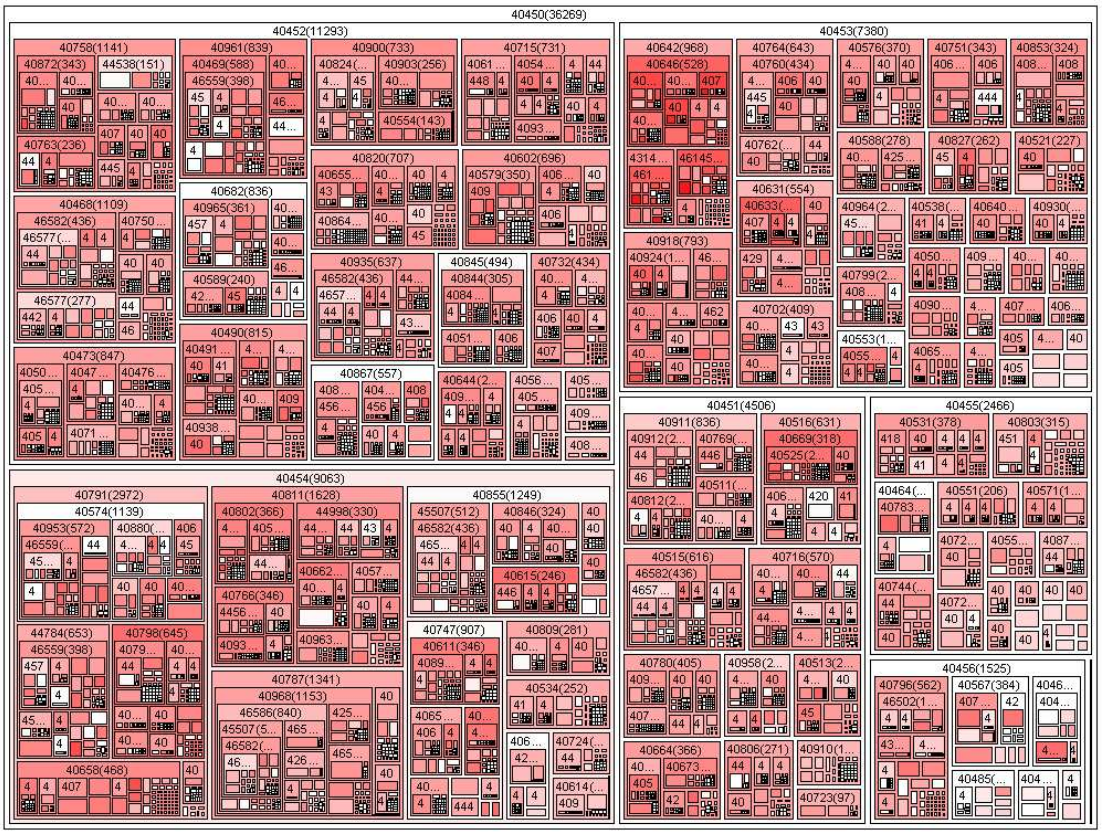


Figure 4.10: STW: Height Deviation

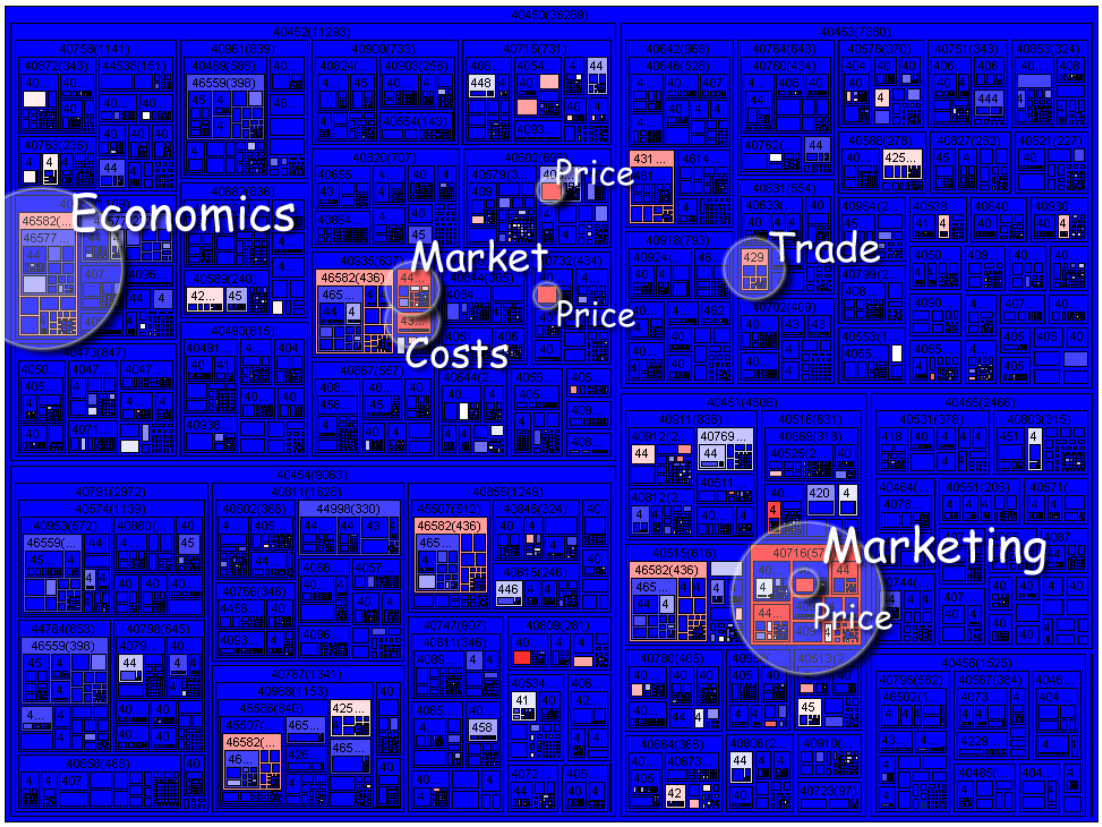


Figure 4.11: STW Frequency Analysis

The visualization of the frequency analysis of the STW thesaurus can be seen in Figure 4.11. We have highlighted several remarkable concepts to illustrate some of the aspects that can be addressed with such an analysis.

The color represents the frequency of documents containing the specified concepts ranging from 0 to 151. The majority of concepts is never or very rarely used in the documents. The more often used concepts are common concepts like *Economics*, *Marketing*, *Market*, *Costs* and *Price*.

We can see that the hypernyms are generally used more frequently than the contained hyponyms. With the most used concepts, we have a first idea about the main focus of the document base.

Concepts like *Price* are contained multiple times in the thesaurus. As the context of the concept is not disambiguated, *Price* (especially as a common term) has a high frequency. As an improvement, the used engine could try to disambiguate the context, which a concept is used in and provide information about the corresponding hypernym in the given context.

IC Diff Analysis. Compared to the frequency analysis, the IC difference analysis provides a more diversified picture of the thesaurus (Figure 4.12).

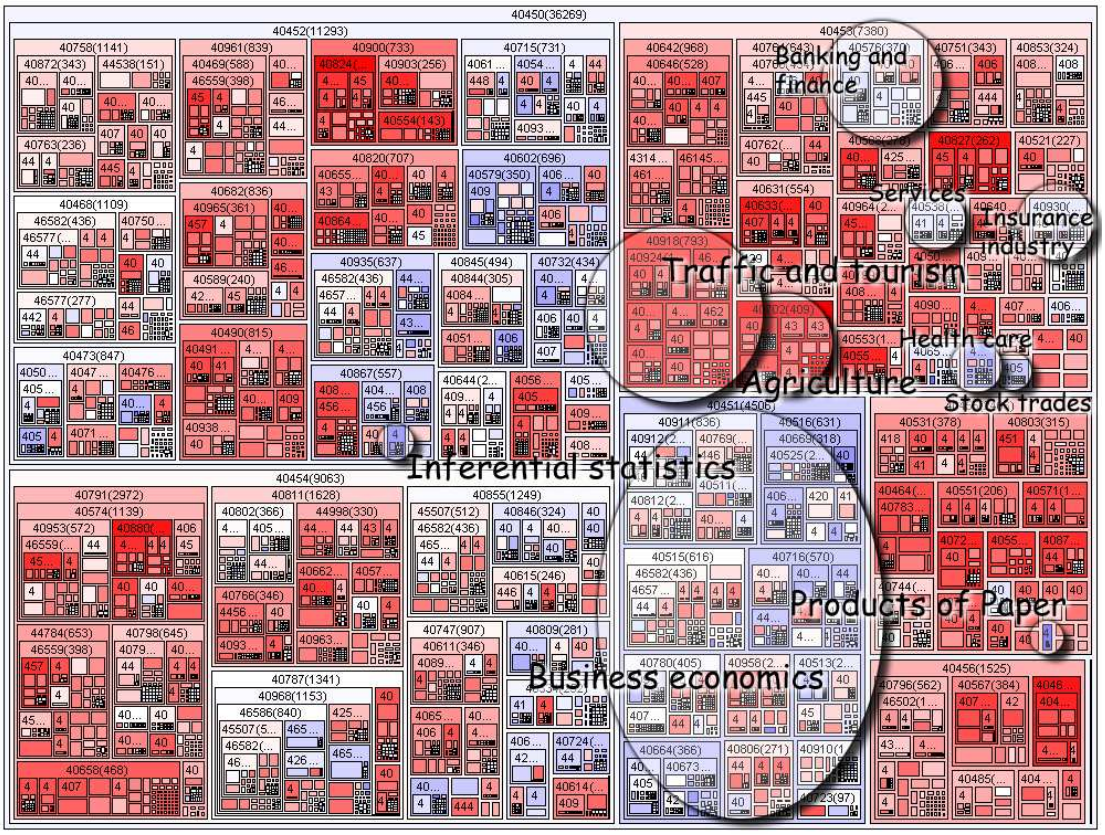


Figure 4.12: STW IC Difference Analysis

The colors represent the difference of the Information Content IC (Equation 3.12) and the Intrinsic Information Content IIC (Equation 3.24). The IIC is seen as the expected information content, the IC is the actual Information Content. Red denotes concepts with a higher IC than expected, blue vice versa. Remember, that a high IC is a result of a low frequency of the concept in the document base.

The IC Diff Analysis provides us with the information which parts of the thesaurus are more often used to annotate documents than one would expect. Figure 4.12 shows that this is the case for the subpart of the thesaurus rooted at the concept *Business Economics*, which is not very surprising. In this part of the thesaurus, there are also a lot of common concepts, like *Ratio, Capital, Price, Market*. Other areas that are well represented in the document base are health care as well as finance and banking. This focus can be explained easily by looking at the journals the articles were taken from (Journal of Health Economics, Journal of Accounting and Economics and Journal of Financial Economics).

In the case of *Business Economics* the high difference between expected and real information contents is not an indicator for a problem in the thesaurus but merely a result of the topics covered in the document base.

There are also cases where the difference analysis identifies problems. A good example is the concept *products of paper* shown in figure Figure 4.13. This concept originally refers to the branch of economics concerned with the production of paper products. In the indexing process, the concept paper was often assigned to documents that are not concerned with this intended meaning because of phrases that contain references to scientific publications ("in a recent paper ... suggest").

The solution is to use a disambiguation step during indexing in order to disambiguate between the two meanings of *paper*.

On the other hand, IC Diff analysis also shows parts of the thesaurus that are less often used to annotate documents than one would expect. An example would be the concept of *raw materials*. A closer look with the interactive tool reveals that the commonly used synonym *commodity* is missing in the thesaurus node. In our document base, the term *commodity* is used twice as often as *raw materials* and the terms never co-occur.

4.3.3 MeSH Analysis

Finally, we want to point out some interesting results from the analysis of the MeSH Thesaurus. Figure 4.14 shows the Difference of Information content for the whole thesaurus. The MeSH Category “Organisms” is highlighted.

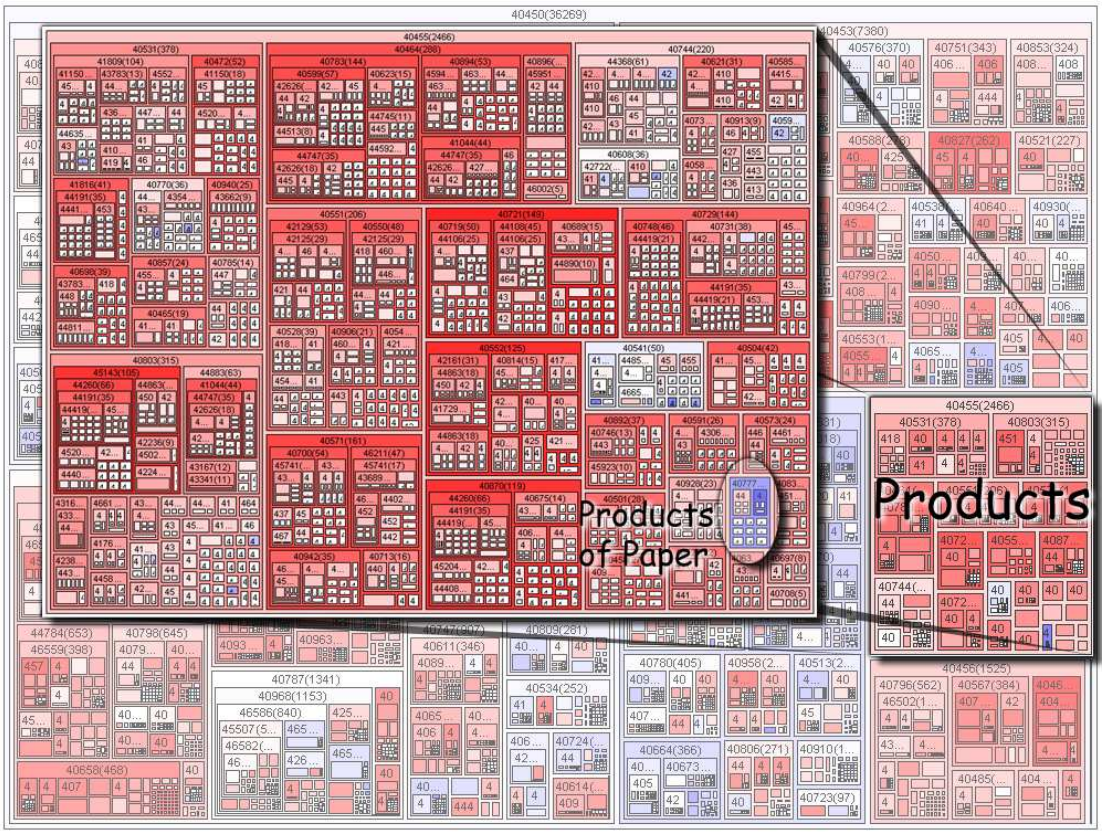


Figure 4.13: STW IC Diff Analysis: Zoom on Paper

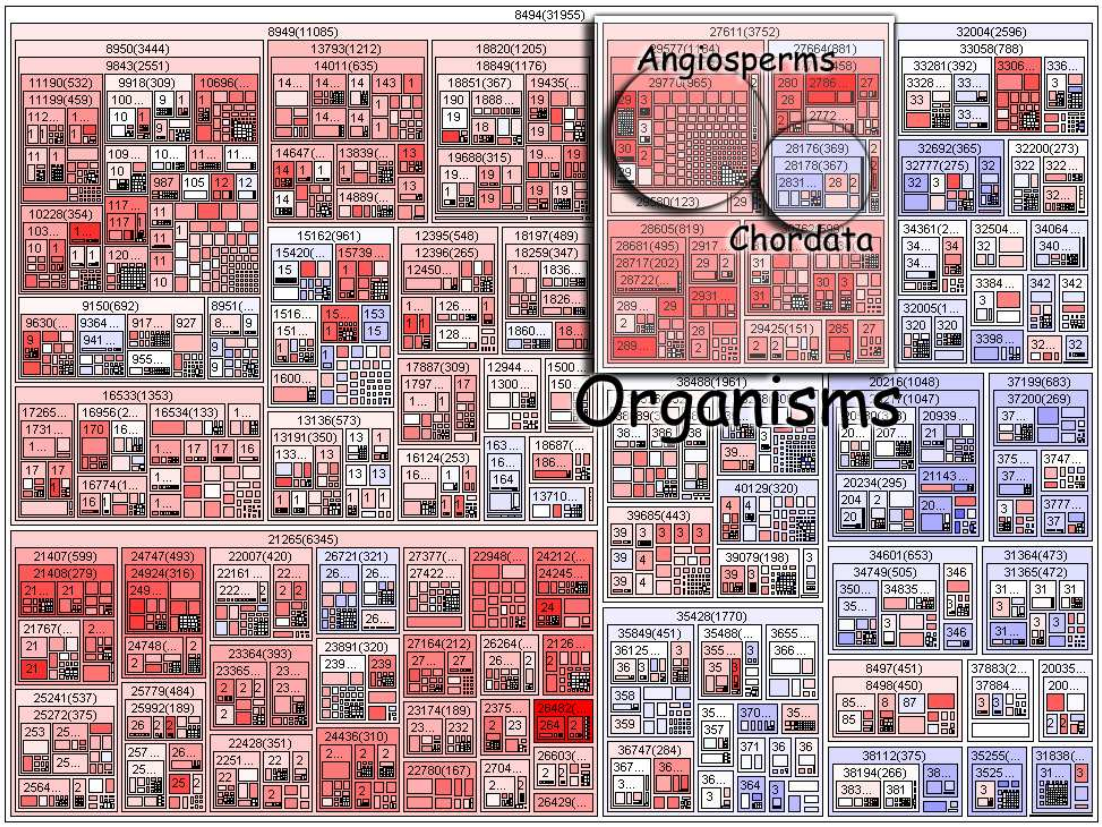


Figure 4.14: MeSH IC Difference Analysis

First of all, you see the “Angiosperms” concept with lots of subconcepts. The structure is visibly different to the other concepts. Whereas such a structure could indicate a problem with the thesaurus, in this case, it just reflects the fact that “the angiosperms, or flowering plants, are one of the major groups of extant seed plants and arguably the most diverse major extant plant group on the planet, with at least 260,000 living species classified in 453 families” [SSE05]. This clearly shows that the structural irregularity in the thesaurus is not a design mistake but correctly reflects the nature of the domain.

When looking further at the differences in the information content another interesting concept sticks out: The Chordata (a group of animals including the vertebrates and some closely related invertebrates). It shows a significant lower Information Content as a result of some concepts with a unexpected high frequency compared to the other concepts in this group.

Figure 4.15 zooms into the *Animals* concept for further analysis. Now, the upper right concept is the “Chordata” concept.

We see several blue areas and two dark blue concepts. One of them are the “Equidae”, also known as horse-like animals. The result shows a by far too low information content for this concept, thus it has a very high frequency in the document base. As the document base is not dedicated to horse diseases, this indicates a problem: In the MeSH Thesaurus, we find the following terms for the concept *Equidae*: Asses, Donkeys, Mules, Zebras, Ass, Equus asinus

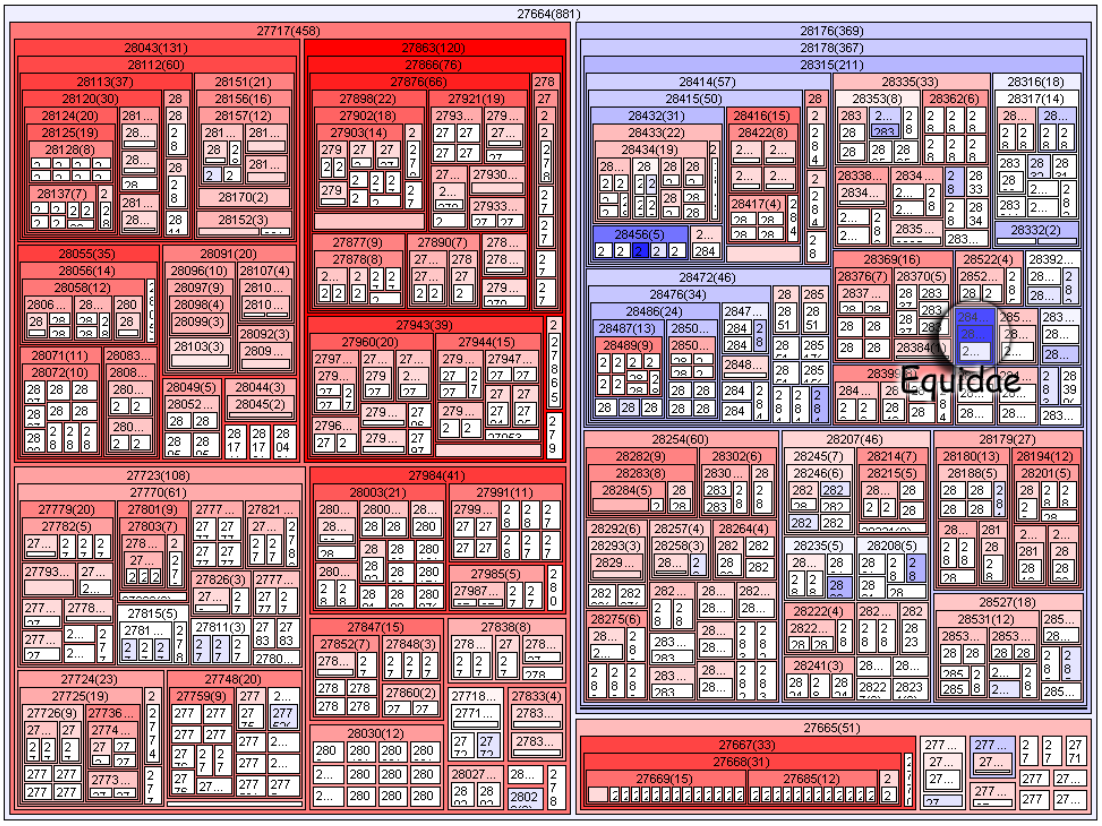


Figure 4.15: MeSH IC Difference Analysis Animals

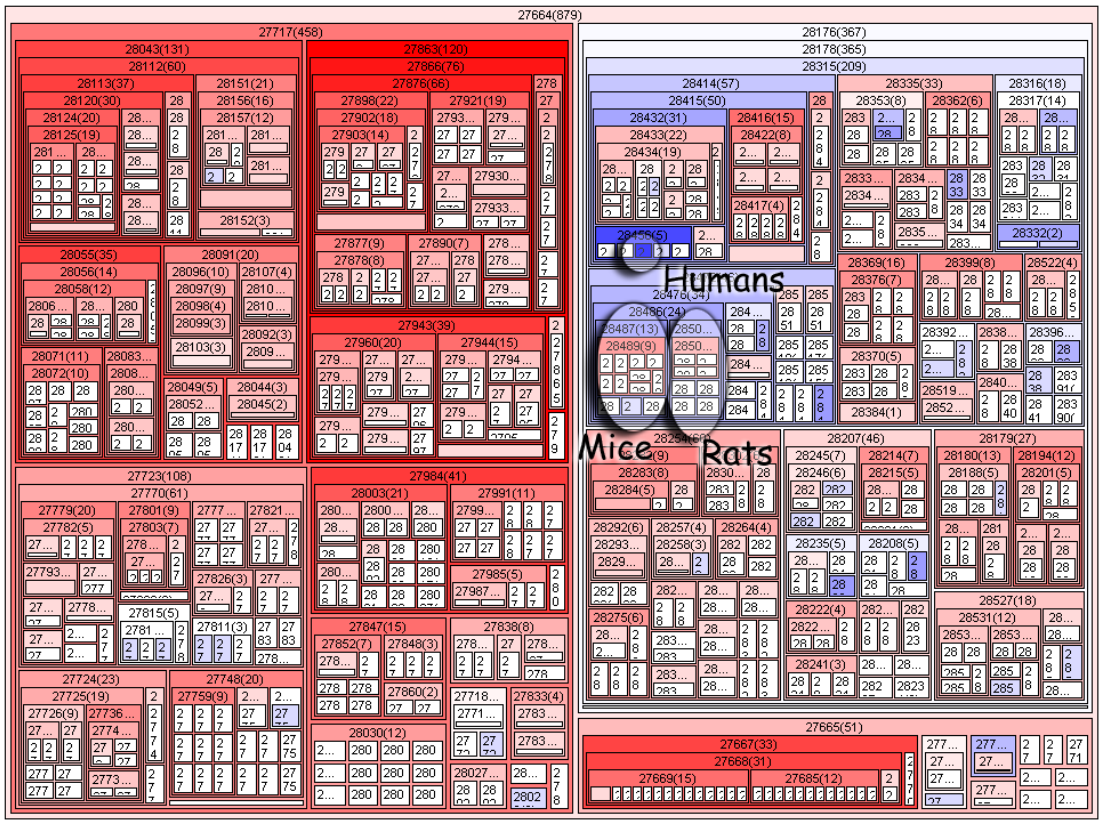


Figure 4.16: MeSH IC Difference Analysis Animals without Equidae

(the horses itself are contained in a subconcept of *Equidae*). Analysis shows that the problem lies in the term “Ass”. The language normalizer interpreted every occurrence of the word “as” as the singular form of “Ass”.

This example nicely illustrates the problems that can arise in the course of automatic indexing and it also shows that our method is an adequate means to identify such potential problems.

We fixed this issue and rerun our analysis, the result is shown by Figure 4.16. The *Equidae* are now as rare as they should be. But there are still several blue concepts. The most frequent concept now is the concept *Humans*, which is not very surprising as most articles are concerned with the treatment of human patients. Other high frequent concepts are *Mice* and *Rats*, which gives us an direct insight on the favorite subjects of animal testing for drug discovery.

We conclude that these derivations of the information content do not point to a problem in the thesaurus or the indexing process as they are a result of the nature of the domain of interest. This shows that the final interpretation is up to a human user, who alone can decide if such remarkable results are due to a problem or just an embodiment of the underlying document base.

4.3.4 Improvements

As stated before, the goal of the thesaurus analysis is an improvement of the indexing results in terms of increased precision and recall. In this thesis, we only performed some initial experiments in improving the indexing results based on the results of the thesaurus analysis. In particular, we improved concepts that have been identified as problematic on the basis of a high difference between expected and real information content from the index terms and recalculated precision and recall. In the case of the STW thesaurus, we removed the ambiguous synonym *paper* from the concept. This led to an increase of the precision by 4.8%.

In the case of the MeSH example, improving the normalizer led to an increase of the precision by 1.2% while the recall remained unchanged. These improvements are not very impressive at first sight, but we have to keep in mind that the increase is the result of just fixing a problem with a single concept of one out of about 32000 concepts.

5 Conclusion

5.1 Summary

In this thesis we presented a method for analyzing the suitability of a thesaurus with respect to providing the basis for automatically indexing a given document set.

With Chapter 2, we introduced some basic concepts of thesaurus-based search, the Collexis Engine used for automatic indexing in our experiments as well as the involved thesauri.

Then, we developed a well-defined performance measure to evaluate the results of the automatic indexing in Chapter 3. As this measure is based and dependent on semantic similarity, we introduced several approaches for semantic similarity and evaluated their impact our performance measure. We concluded with a first optimistic prediction about the quality of usability of the STW thesaurus for automatic indexing and decided for the Lin similarity measure as basis for our further performance evaluations.

The main part of this thesis is the proposition of our general thesaurus analysis approach in Chapter 4. We found the treemap visualization the best match to our approach of analyzing on concept level, instead of analyzing the thesaurus as a whole. Our experiments with different techniques led to the development of the promising IC Difference Analysis.

We showed that it is easy to identify potentially problematic parts of a thesaurus with our analysis approach for the suitability of a thesaurus and that a manual inspection of these problematic parts often reveals problems that were induced by the automatic indexing process. We also presented results of initial experiments in improving the process based on the result of the analysis in which fixing a single problem leads to a significantly increased precision of the annotation.

We conclude that interactive thesaurus assessment is a suitable means to improve the results of automatic document annotation. Especially for relatively small document sets, identifying indexing errors can have a significant impact on the quality of the annotation.

5.2 Outlook

There are a number of issues that require further investigation. One open question is the impact of the choice of a particular distance measure as the basis for generalized precision and recall has on the results of our analysis. We believe that a comprehensive theory of generalized precision and recall is needed to answer that question.

Another important observation was that many of the problems found by our method can actually be seen as problems of the indexing algorithm rather than problems of the thesaurus.

These problems have to be faced, as automatic indexing of documents becomes more and more important.

There are several directions for future work that are implied by this thesis. The major direction of research are methods for improving annotation results based on the results of the analysis. So far, we have only considered the improvement of over-represented concepts in order to improve the precision of annotations. In order to improve recall, we need to identify terms that are under-represented in the annotations and find out why this is the case. A possible problem is the lack of proper synonyms. Finding such missing synonyms is a possible way to improve recall that should be investigated in the future.

Another challenging direction is the development of further analysis techniques like the IC Diff Analysis. So far, we used only the smallest subset of available information. We see our general approach with the treemap visualization as a framework that may be used to perform more sophisticated analysises.

Bibliography

[BKBK05] A. Bernstein, E. Kaufmann, C. Burki, and M. Klein. How similar is it? towards personalized similarity measures in ontologies. In *7. Int. Tagung Wirtschaftsinformatik*. Bamberg, Germany, 2005.

[BKKB05] A. Bernstein, E. Kaufmann, C. Kiefer, and C. Burki. Simpack: A generic java library for similarity measures in ontologies. Technical report, University of Zurich, Department of Informatics, 2005.

[BSW02] Benjamin B. Bederson, Ben Shneiderman, and Martin Wattenberg. Ordered and quantum treemaps: Making effective use of 2d space to display hierarchies. *ACM Trans. Graph.*, 21(4):833–854, 2002.

[Bur04] Margarete Burkart. *Grundlagen der praktischen Dokumentation und Information, Band 1*, chapter Thesaurus, pages 151–153. Saur, 2004.

[BYRN99] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.

[CD04a] Jacques Calmet and Anusch Daemi. Assessing conflicts in ontologies. Technical report, IAKS Calmet, University Karlsruhe (TH), Germany, 2004.

[CD04b] Jacques Calmet and Anusch Daemi. From entropy to ontology. Technical report, Institute for Algorithms and Cognitive Systems (IAKS), University of Karlsruhe (TH), Germany, 2004.

[CLR90] Thomas H. Corman, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. MIT Press, 1990.

[CM05] Courtney Corley and Rada Mihalcea. Measuring the semantic similarity of texts. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, pages 13–18. ACL, 2005.

[Els07] Elsevier B.V. Elsevier. Online, http://www.elsevier.com, visited 2007-01-29, 2007.

[Euz07] Jérôme Euzenat. Semantic precision and recall for ontology alignment evaluation. In *IJCAI*, pages 348–353, 2007.

[Fel98] Christiane Fellbaum. *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press, 1998.

[Fer03] Reginald Ferber. *Information Retrieval*. dpunkt Verlag, Heidelberg, 2003.

[GF04] David A. Grossmann and Ophir Frieder. *Information Retrieval - Algorithms and Heuristics*. Springer, Dordrecht, Netherlands, second edition edition, 2004.

[HS98] Udo Hahn and Klemens Schnattinger. Towards text knowledge engineering. In *AAAI/IAAI*, pages 524–531, 1998.

[IV98] Nancy Ide and Jean Véronis. Word sense disambiguation: The state of the art, 1998.

[JC97] Jay J. Jiang and David W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Natural Language Engineering - Proceedings of International Conference Research on Computational Linguistics (ROCLING X)*, volume 11, pages 81–111. Tapei, Taiwan, 1997.

[KJ02] Jaana Kekäläinen and Kalervo Järvelin. Using graded relevance assessments in ir evaluation. *J. Am. Soc. Inf. Sci. Technol.*, 53(13):1120–1129, 2002.

[LBM02] Yuhua Li, Zuhair Bandar, and David McLean. Measuring semantic similarity between words using lexical knowledge and neural networks. In *IDEAL 2002: Third International Conference*, page 111. Springer Berlin/Heidelberg, 2002.

[LC98] C. Leacock and M. Chodorow. Combining local context and wordnet similarity for word sense identification. In C. Fellbaum, editor, *WordNet: An Electronic Lexical Database*, chapter 11, pages 265–283. MIT Press, 1998.

[Lin98] Dekang Lin. An information-theoretic definition of similarity. In *Proc. 15th International Conf. on Machine Learning*, pages 296–304. Morgan Kaufmann, San Francisco, CA, 1998.

[May05] Diana Maynard. Benchmarking ontology-based annotation tools for the semantic web. In *UK e-Science Programme All Hands Meeting (AHM2005) Workshop "Text Mining, e-Research and Grid-enabled Language Technology"*. Nottingham, UK, 2005.

[MB95] Tamara Munzner and Paul Burchard. Visualizing the structure of the World Wide Web in 3D hyperbolic space. In *Proc. 1st Symp. The VRML Modelling Language: Special issue of Computer Graphics*, pages 33–38. ACM Press, 14–15 1995.

[MPL06] D. Maynard, W. Peters, and Y. Li. Metrics for evaluation of ontology-based information extraction. In *WWW 2006 Workshop on "Evaluation of Ontologies for the Web" (EON)*, Edinburgh, Scotland, 2006.

[NLM07a] NLM. Medical subject headings. Online, http://www.nlm.nih.gov/mesh/, visited 2007-01-29, 2007.

[NLM07b] NLM. National library of medicine: Entrez pubmed. Online, http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?DB=pubmed, visited 2007-01-29, 2007.

[Res95] Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*, 1995.

[Res99] Philip Resnik. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, 11:95–130, 1999.

[SC03] Nihar Sheth and Qin Cai. Visualizing mesh dataset using radial tree layout. Technical report, Indiana University, Bloomington, 2003.

[SF01] Marc Smith and Andrew Fiore. Visualization components for persistent conversations. In *Proceedings of the SIG-CHI on Human factors in computing systems*, pages 136–143, 2001.

[Shn92] Ben Shneiderman. Tree visualization with tree-maps: 2-d space-filling approach. *ACM Trans. Graph.*, 11(1):92–99, 1992.

[Shn06a] Ben Shneiderman. Discovering business intelligence using treemap visualizations. Online, http://www.b-eye-network.com/view/2673, visited 2007-02-18, April 2006.

[Shn06b] Ben Shneiderman. Treemaps for space-constrained visualization of hierarchies. Online, http://www.cs.umd.edu/hcil/treemap-history/, visited 2007-02-18, April 2006.

[Sol07] SyynX Solutions. Syynx mediator. Online, http://www.mediator.syynx.de, visited 2007-01-29, 2007.

[SSE05] Pam Soltis, Doug Soltis, and Christine Edwards. Angiosperms. flowering plants. http://tolweb.org/Angiosperms/20646/2005.06.03 in The Tree of Life Web Project, http://tolweb.org/, visited 2007-03-10, June 2005. Version 03.

[STW07] STW. Standard thesaurus wirtschaft. Online, http://www.gbi.de/thesaurus/, visited 2007-01-29, 2007.

[SVH04] Nuno Seco, Tony Veale, and Jer Hayes. An intrinsic information content metric for semantic similarity in wordnet. In *Proceedings of the 16th European Conference on Artificial Intelligence*, pages 1089–1090. Valencia, Spain, 2004.

[SW01] Ben Shneiderman and Martin Wattenberg. Ordered treemap layouts. Online, ftp://ftp.cs.umd.edu/pub/hcil/Reports-Abstracts-Bibliography/2001-06html/2001-06.htm, visited 2007-02-19, 06 2001.

[SWY75] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, November 1975.

[WP94] Zhibiao Wu and Martha Palmer. Verb semantics and lexical selection. In *32nd. Annual Meeting of the Association for Computational Linguistics*, pages 133–138, New Mexico State University, Las Cruces, New Mexico, 1994.

[ZBW07] ZBW. Deutsche zentralbibliothek für wirtschaftswissenschaften, datenbank econis. Online, http://zbwopc4.zbw.ifw-kiel.de:8080/DB=1/LNG=DU/, visited 2007-01-29, 2007.

Ehrenwörtliche Erklärung

Ich versichere, dass ich die vorliegende Diplomarbeit selbstständig verfasst und nur mit den angegebenen Quellen und Hilfsmitteln gearbeitet habe. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mannheim, den 15. Mai 2007,