Usage-driven Maintenance of Knowledge Organization Systems

Inauguraldissertation
zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften
der Universität Mannheim

vorgelegt von

Kai Eckert
aus Heidelberg

Mannheim, 2012

Dekan: Professor Dr. Heinz Jürgen Müller, Universität Mannheim
Referent: Professor Dr. Heiner Stuckenschmidt, Universität Mannheim
Korreferent: Professor Dr. Marcia Lei Zeng, Kent State University, USA

Tag der mündlichen Prüfung: 15. Juni 2012

Abstract

Knowledge Organization Systems (KOS) are typically used as background knowledge for document indexing in information retrieval. They have to be maintained and adapted constantly to reflect changes in the domain and the terminology. In this thesis, approaches are provided that support the maintenance of hierarchical knowledge organization systems, like thesauri, classifications, or taxonomies, by making information about the usage of KOS concepts available to the maintainer.

The central contribution is the ICE-Map Visualization, a treemap-based visualization on top of a generalized statistical framework that is able to visualize almost arbitrary usage information. The proper selection of an existing KOS for available documents and the evaluation of a KOS for different indexing techniques by means of the ICE-Map Visualization is demonstrated.

For the creation of a new KOS, an approach based on crowdsourcing is presented that uses feedback from Amazon Mechanical Turk to relate terms hierarchically. The extension of an existing KOS with new terms derived from the documents to be indexed is performed with a machine-learning approach that relates the terms to existing concepts in the hierarchy. The features are derived from text snippets in the result list of a web search engine. For the splitting of overpopulated concepts into new subconcepts, an interactive clustering approach is presented that is able to propose names for the new subconcepts.

The implementation of a framework is described that integrates all approaches of this thesis and contains the reference implementation of the ICE-Map Visualization. It is extendable and supports the implementation of evaluation methods that build on other evaluations. Additionally, it supports the visualization of the results and the implementation of new visualizations. An important building block for practical applications is the simple linguistic indexer that is presented as minor contribution. It is knowledge-poor and works without any training.

This thesis applies computer science approaches in the domain of information science. The introduction describes the foundations in information science; in the conclusion, the focus is set on the relevance for practical applications, especially regarding the handling of different qualities of KOSs due to automatic and semi-automatic maintenance.

Zusammenfassung

Wissensorganisationssysteme (Knowledge Organization Systems, KOS) werden üblicherweise als Hintergrundwissen im Information Retrieval zur Indexierung von Dokumenten genutzt. Sie müssen kontinuierlich gepflegt und angepasst werden, um Änderungen im Fachbereich und der verwendeten Terminologie widerzuspiegeln. In dieser Arbeit werden Verfahren vorgestellt, die die Pflege von Wissensorganisationssysteme, wie Thesauri, Klassifikationen oder Taxonomien, unterstützen, indem sie dem Bearbeiter Informationen zur Nutzung der KOS-Konzepte zur Verfügung stellen.

Der Hauptbeitrag ist die ICE-Map Visualization, eine Treemap-basierte Visualisierung, die auf einem generalisierten statistischen Framework aufbaut. Sie kann nahezu beliebige Nutzungsinformationen visualisieren. Die passende Auswahl eines existierenden KOS für vorhandene Dokumente und die Bewertung eines KOS für verschiedene Indexierungstechniken mittels der ICE-Map Visualization wird gezeigt.

Für die Erstellung eines neuen KOS wird ein Crowdsourcing-Ansatz vorgestellt, der Feedback von Amazon Mechanical Turk nutzt, um Terme in eine hierarchische Beziehung zu setzen. Die Erweiterung eines existierenden KOS mit neuen Termen wird mittels eines Maschinenlernverfahrens durchgeführt, das die Terme zu Konzepten im KOS in Bezug setzt. Als Merkmale werden dabei Textfragmente aus der Ergebnisliste einer Websuchmaschine genutzt. Für die Aufteilung überfüllter Konzepte in neue Unterkonzepte wird ein interaktiver Clusteringansatz präsentiert, der auch Namensvorschläge für die neuen Konzepte generiert.

Ferner wird die Implementierung eines Frameworks beschrieben, das alle Ansätze integriert und die Referenzimplementierung der ICE-Map Visualization beinhaltet. Es ist erweiterbar und unterstützt die Implementierung von neuen Evaluationsmethoden, die auf vorhandenen Evaluationen aufbauen. Die Visualisierung der Ergebnisse, sowie die Implementierung neuer Visualisierungen wird unterstützt. Ein wichtiger Baustein für praktische Anwendungen ist der einfache linguistische Indexer, der als Nebenbeitrag vorgestellt wird. Er kommt ohne vorbereitendes Training aus.

Diese Arbeit wendet Ansätze aus der Informatik im Bereich der Informationswissenschaften an. In der Einleitung werden die relevanten Grundlagen aus der Informationswissenschaft vorgestellt und in der abschließenden Zusammenfassung auf die Relevanz für praktische Anwendungen eingegangen, insbesondere, was den Umgang mit verschiedenen Qualitäten von KOSs angeht, die aus automatischer oder halbautomatischer Pflege resultieren.

Contents

1 Introduction **1**
- 1.1 Knowledge Organization Systems . . . 2
- 1.2 The KOS Life Cycle . . . 7
- 1.3 Foundations in Information Science . . . 11
  - 1.3.1 Motivation . . . 11
  - 1.3.2 Creation of a KOS . . . 13
  - 1.3.3 History and new applications of KOSs . . . 17
  - 1.3.4 Information Retrieval in Libraries . . . 21
- 1.4 Research Questions, Contributions, and Limitations . . . 23
- 1.5 Research Data . . . 26

2 Statistical Analysis of Concept Hierarchies **33**
- 2.1 Visual Datamining . . . 34
- 2.2 From Applications to Weight Functions . . . 37
- 2.3 Statistical Framework . . . 39
- 2.4 Visualization . . . 40
- 2.5 Related Work . . . 49
- 2.6 Conclusion . . . 49

3 Selection and Evaluation **51**
- 3.1 KOS Selection based on Topical Overlap . . . 51
  - 3.1.1 Experimental Setup . . . 53
  - 3.1.2 Results . . . 54
  - 3.1.3 Focus on Documents . . . 56
- 3.2 KOS-based Indexing Evaluation . . . 58
  - 3.2.1 Experimental Setup . . . 60
  - 3.2.2 Intellectual Indexing . . . 62
  - 3.2.3 Automatic Indexing . . . 67
  - 3.2.4 Tagging . . . 72
- 3.3 Related Work . . . 76
- 3.4 Conclusion . . . 77

4 Creation and Modification **81**
4.1 Crowdsourcing the Creation Process 81
4.1.1 Method Description 82
4.1.2 Experimental Setup 85
4.1.3 Results 88
4.2 KOS Extension using Web Search Engines 96
4.2.1 Method Description 97
4.2.2 Experimental Setup 101
4.2.3 Results 102
4.3 Concept Splitting and Naming 104
4.3.1 Method Description 105
4.3.2 Experimental Setup 108
4.3.3 Results 109
4.4 Related Work 110
4.5 Conclusion 113

5 Implementation **117**
5.1 Semtinel: An extendable Analysis and Visualization Platform . . . 117
5.1.1 The Semtinel Architecture 119
5.1.2 Data Model 119
5.1.3 Experiment API 122
5.2 LOHAI: A Baseline Indexer 130
5.2.1 Preprocessing 132
5.2.2 Concept assignment with compound term detection 133
5.2.3 Unstemming and word-sense disambiguation 133
5.2.4 Weighting 135
5.2.5 Example 136
5.3 Related Work 138
5.4 Conclusion 139

6 Conclusion **141**
6.1 Summary of Contributions 147
6.2 Future Work 150
6.3 Final remarks 152

References **155**

List of Figures

1.1 Types of KOSs. 3
1.2 The KOS Life Cycle. 8
1.3 Semiotic Triangle. 15
1.4 Excerpt of the STW categories. 27
1.5 Excerpt of the MeSH structure. 29

2.1 Ghost Map. 35
2.2 Calculation of the IC Difference. 40
2.3 Original "slice-and-dice" layout. 42
2.4 Squarified layout. 45
2.5 Treemap of the MeSH concept BODY REGIONS. 46
2.6 Reference implementation of the ICE-Map Visualization. 47
2.7 Concept selection. 47
2.8 Zooming into the hierarchy. 48

3.1 Topical overlap of the TheSoz with EconStor and SSOAR. 55
3.2 Topical overlap of the STW with EconStor and SSOAR. 57
3.3 TheSoz and STW: Zoom on ECONOMY. 58
3.4 Direct comparison of SSOAR and EconStor. 59
3.5 Schematic view of a supervised indexing and retrieval process. . . 61
3.6 STW: Intellectual Indexing vs. IIC. 63
3.7 STW: Zoom on BUSINESS ECONOMICS. 64
3.8 STW: Zoom on ECONOMIC SECTORS. 65
3.9 MeSH: Intellectual Indexing vs. IIC. 66
3.10 MeSH: Zoom on ORGANISMS. 66
3.11 Document example. 68
3.12 STW: Collexis vs. IIC. 69
3.13 STW: Zoom on GENERAL DESCRIPTORS. 70
3.14 STW: Zoom on PRODUCTS. 71
3.15 STW: Collexis vs. Intellectual Indexing. 72
3.16 STW: Tagging vs. Intellectual Indexing. 74

4.1 The presentation of a pair in a HIT. 87

4.2 Histogram of inter-group deviations. 90
4.3 Histogram of deviations from reference group of experts. 90
4.4 Results filtered by working time. 91
4.5 Fragment of WordNet. 97
4.6 KOS extension workflow. 98
4.7 Concept splitting workflow. 106
4.8 Weight calculation. 106

5.1 Semtinel on the Netbeans Platform. 118
5.2 Semtinel architecture. 121
5.3 Semtinel class diagram. 122
5.4 Creation of a new experiment. 122
5.5 Palette 123
5.6 Experiment configuration panel. 126
5.7 Multiple datasets in one register. 126
5.8 Hierarchy of different analyses. 127
5.9 Explanation browser. 129
5.10 Experiment groups. 130
5.11 The indexing pipeline. 131
5.12 Document example. 137

6.1 Extended shell model. 146

List of Tables

1.1 Example of an STW concept. 28
1.2 Example of a MeSH Descriptor. 30

2.1 Mathematical notation. 38

3.1 Example annotations: Intellectual Indexing vs. Collexis. 68

4.1 Distribution of overlapping pairs among InPhO users. 87
4.2 Distribution of the number of HITs among MTurk workers. 88
4.3 Number of users who answered the test pairs correctly. 92
4.4 Conditional probabilities of correct answers for the test pairs. . . . 92
4.5 Effect of different filters on the set of Mechanical Turk workers. . 93
4.6 Financial evaluation. 94
4.7 Accuracy of the classification. 103
4.8 Accuracy of the position. 104
4.9 Results: Term clustering. 109
4.10 Results: Document clustering. 110

5.1 Semtinel modules. 120
5.2 Selected methods of the main interfaces of the Experiment API. . 125
5.3 Properties of an Explanation node. 128
5.4 Example annotations: Intellectual indexing vs. LOHAI. 137

Acknowledgments

First and foremost, I would like to thank my advisers. Heiner Stuckenschmidt initially convinced me to do a doctorate and always helped me to get back on track when I lost it. Marcia Zeng immediately agreed when I asked her to co-supervise my thesis. She took care of the information science perspective and provided invaluable feedback to set things straight in the end.

Second, I would like to thank Magnus Pfeffer who has been a brilliant colleague and co-author. He explained me the mysteries of the library world and piqued my interest in it.

I thank all further co-authors of the publications that form the basis of this thesis: Colin Allen, Cameron Buckner, Christian Hänger, Robert Meusel, Christof Niemann, Mathias Niepert, Dominique Ritze, and Dominik Stork. Furthermore, I thank all my student assistants, especially Alexander Hanschke who participated in the open source release of Semtinel and Jan Gansen for testing and proofreading of the description of the ICE-Map Visualization.

Special thanks to Simone Krug for a very thorough proofreading of the whole thesis. All remaining mistakes and weaknesses are mine.

I thank all colleagues at the KR & KM Research Group (Chair of Artificial Intelligence) for fruitful discussions, a lot of fun, and great coffee breaks, as well as my colleagues at the library who gave me the room to finish this thesis. And great coffee breaks, too.

Last, not least, I want to thank my better half Florence who patiently listened to my complaints and assertions that I am “almost done”, but also pushed me, when I needed it. I thank all my friends who are still there, even though I made myself very scarce in the last years, and finally my whole family who always believed in me and supported me in everything I did.

Chapter 1

Introduction

Wo ich nicht klar sehen, nicht mit Bestimmtheit wirken kann,
da ist ein Kreis, für den ich nicht berufen bin.

Johann Wolfgang von Goethe[1]

Informed decisions need information. In this thesis, we develop approaches to provide practitioners with information that help them to make decisions regarding the creation, utilization, and maintenance of concept hierarchies that are fundamental for supporting subject access to information. Concept hierarchical structures are often represented in a *knowledge organization system* (KOS) such as a thesaurus, a subject headings scheme, or a classification scheme. A typical example is the application of a thesaurus in libraries, information centers, and indexing and abstracting services to describe the content of the holdings in a collection. A thesaurus contains descriptive terms representing concepts that are organized in a hierarchical structure, ordered from general to specific. These terms are assigned to books, articles in journals, scientific reports, etc. to ensure efficient access to relevant information. The assignment is typically performed by librarians and information professionals, but there also exist automatic systems that assign terms based on text analysis. The thesaurus needs to be improved, updated, and maintained constantly to reflect changes in the represented subjects. New subjects arise, others take a back seat. Decisions about maintenance steps are partly based on superordinate design criteria, but also on the actual usage of the concepts in assignments, as well as changes in the terminology found in the literature. The decisions have to be made by a human expert, but the concept usage will drive these decisions. There are many applications for KOSs, e.g., they are used as background knowledge for natural language processing. In this thesis, however, we focus on the main application, for which most KOSs are created in the first place: information retrieval.

[1] "Where I do not see clearly, cannot act with certainty, there is a circle, for which I am not qualified.", cf. Müller and Goethe (1870, p. 130).

Information retrieval (IR) deals with the representation, storage, organization of, and access to information items (Baeza-Yates & Ribeiro-Neto, 1999). An information item can be anything that comprises valuable information for the user of the IR system. In this thesis, we refer to information items as documents in a very broad sense and define accordingly:

Definition 1.1 (Document) *A document is a single information item or resource that usually contains textual information and is made available to the IR system with one or more of the following attributes:*

1. *title,*
2. *abstract,*
3. *full text (structured or plain text),*
4. *other bibliographic information (creator(s), publisher, year, identifiers, ...),*
5. *links to textual representations and/or descriptions of the content.*

Definition 1.1 is a deliberately broad definition of a document.[2] The information that is available for a single document greatly varies, ranging from bibliographic records containing only the title as a subject-related description to electronically available articles and e-books where not only an abstract, but also the full text is available. The definition particularly also covers all kind of web pages (title and structured full text), tagged images, and other resources that are already organized by means of some kind of vocabulary or classification (links to textual descriptions).

1.1 Knowledge Organization Systems

According to Hodge (2000), the term "Knowledge Organization System" was coined by the Networked Knowledge Organization Systems (NKOS) Working Group[3] at the ACM Digital Libraries Conference in 1998. The NKOS working group refers to Tudhope and Koch (2004) who state:

> "Knowledge Organization Systems/Services (KOS), such as classifications, gazetteers, lexical databases, ontologies, taxonomies and thesauri, model the underlying semantic structure of a domain. Embodied as Web-based services, they can facilitate resource discovery and re-

[2] cf. the new ISO (2011) standard for thesauri (ISO 25964-1) that defines information retrieval as "all the techniques and processes used to identify documents relevant to an information need, from a collection or network of information resources."

[3] http://nkos.slis.kent.edu/

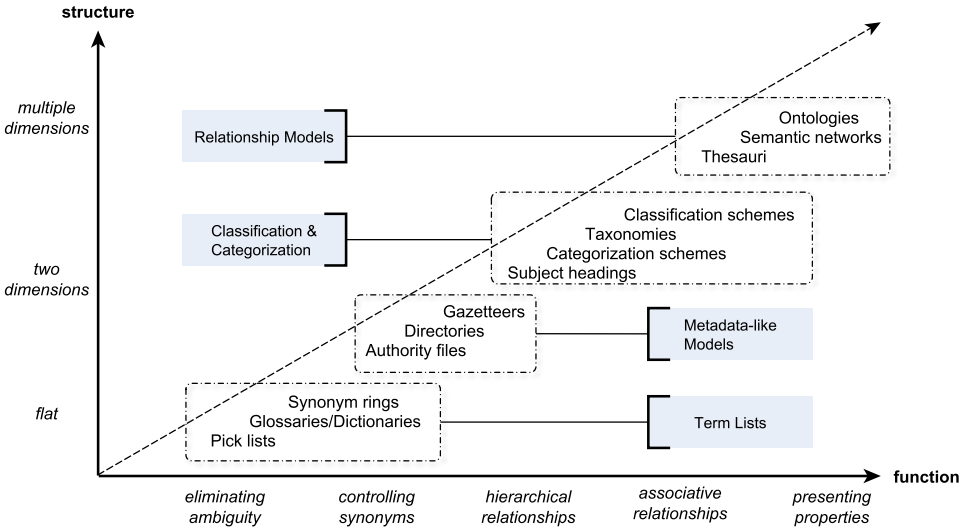


Figure 1.1: Types of KOSs (Adapted from Zeng, 2008).

trieval. They act as semantic road maps and make possible a common orientation by indexers and future users (whether human or machine)."

For this thesis, we accordingly define:

Definition 1.2 (Knowledge Organization System) *A Knowledge Organization System (KOS) is a structured model of concepts that is used to represent and organize knowledge.*

Definition 1.3 (Concept) *A concept represents a real world object and/or abstract entity like a knowledge concept. As a representation, it usually consists of a description that defines the scope of the concept and one or more* labels *to name (and also define) the concept. Labels consist of and are often referred to as* terms.

In the simplest case a KOS is just a list of labeled concepts (controlled vocabulary), but it can also contain information about specific relations between these concepts like hypernym/hyponym[4] relations (thesaurus) and even information about arbitrary relations including logic formulae and constraints that apply to the concepts and relations (ontologies).

The bottom line of Definition 1.2 is that KOS is an umbrella term for various specific knowledge organization systems. An overview about some of them is

[4]Hypernyms are broader, more general concepts, hyponyms are narrower, more specific concepts. From (Fellbaum et al., 2010): **Hypernym:** The generic term used to designate a whole class of specific instances. Y is a hypernym of X if X is a (kind of) Y. **Hyponym:** The specific term used to designate a member of a class. X is a hyponym of Y if X is a (kind of) Y.

given in Figure 1.1. The NKOS taxonomy[5] groups the systems under three general categories:

1. Term lists, which contain lists of words or phrases, often with definitions. Examples include authority files, glossaries, gazetteers, and dictionaries.
2. Classifications and categorization schemes, which emphasize the creation of subject sets. The most notable examples are library classification schemes, taxonomies, and categorization schemes.
3. Relational vocabularies, which emphasize the relationships and connections between terms and the concepts they represent, including lists of subject headings, thesauri, semantic networks, and ontologies.

Often no clear definition for the particular systems exists. Some of them have overlapping areas of use and existing implementations of such systems often combine properties of more than one type of KOS. For example, the Medical Subject Headings – while according to its name a list of subject headings – is often referred to as a thesaurus. The InPhO project, while being an acronym for Indiana Philosophy Ontology, provides access to a browsable taxonomy. Both KOSs are described in detail in Section 1.5.

In this thesis, we focus on the knowledge organization systems that can be represented as a hierarchy, according to the following definition:

Definition 1.4 (Concept Hierarchy) *A concept hierarchy is a knowledge organization system that provides at least one relationship between the concepts that leads to a super- and sub-ordinate hierarchical structure. A common example for such a relationship is "broader than," respectively "narrower than," but there are also more specific ones, such as "is a"/"has subclass" or "part of"/"has part."*

Essential for building a hierarchy is the ***transitivity*** *of the relationship: if A is broader than B and B is broader than C, A has also to be broader than C or the organization in a hierarchy would be counter-intuitive.*

Two types of concept hierarchies are subject of this thesis: classifications and thesauri. In the following, we explain briefly the differences and commonalities and show that the differences can be neglected for most part of this thesis.

Classification: Classifications (sometimes referred as taxonomies) are collections of classes that are used in our context to classify documents according to the major topics that a document is about. This means to assign each document to one of the classes in the classification. The classes are organized in a hierarchy of super- and subclasses. Classes are represented by notations (in classifications)

[5]Taxonomy of Knowledge Organization Sources/Systems, http://nkos.slis.kent.edu/KOS_taxonomy.htm, based on (Hodge, 2000), cf. (Zeng & Chan, 2004).

or category labels and notations (in taxonomies). An example for a taxonomy in a different context is the *Linnaean taxonomy*. In this special scientific taxonomy, Carolus Linnaeus classified the known and still existing concepts belonging into the *imperium naturea* into three different kingdoms: *mineral*, *animal*, and *plant*. All in all his structure included four other classification levels beneath *kingdom*: *class*, *order*, *genus*, and *species*. The work of Linnaeus was the starting point for the botanical and zoological nomenclature as known today and is used to classify all kinds of animals and plants by biologists.

Thesaurus: In comparison to classifications, *thesauri* basically extend their core functionality with additional accepted relations like *synonym*, *antonym*, and *related concept* to improve their ability to describe the subject matter being dealt with in a specific domain or multi- and cross-domains. The concepts in a thesaurus are not required to be disjunct. Thesauri are more focused on the accurate description of the concepts and the concepts are usually combined to describe the content or the topic of resources. An example for a thesaurus is EuroVoc,[6] a multilingual thesaurus in 22 languages covering the activities of the European Union. In the version 4.3, it consists of 6,797 concepts.

Similar to any controlled vocabularies, in a thesaurus, concepts are organized in a scheme where each concept subsumes the synonyms to ensure a unified representation of the concept. It is the hierarchical relationship that distinguishes a thesaurus from a flat controlled vocabulary. As Svenonius (2000, p. 162) indicates: "The most philosophically interesting of the semantic relationships are those that are *hierarchical*. They may be the most effective in furthering the collection and navigation objectives. They are a powerful means for optimizing recall and precision, and at the same time, they are the quintessential means for navigating a knowledge domain." The reason of such efficacy in fulfilling these functions through hierarchical relationships stems from their ability to resolve the retrieval problems caused by the fact that a given object or concept may be referred to at different levels of specificity by users. The creation of the hierarchy follows rules that are also valid for the creation of classification hierarchies. Hierarchical structures in classifications are especially parallel with, or closely match, the structure in knowledge domains, cf. Svenonius (2000, pp. 162ff.).

Kwasnik (1999) provides an overview on classifications for knowledge representation and shows how different relations lead to different knowledge structures and how these relations affect the hierarchical view on the concepts. Khoo and Na (2006) provide in-depth analyses of further semantic relationships.

The earliest example of a KOS that merged a classification and a thesaurus was developed by Aitchison (1970), who called it "Thesaurofacet," based on the notion of a faceted classification. Buchanan (1979) describes the creation of a faceted classification and notes that they can be used as the basis for every kind of structured

[6] http://eurovoc.europa.eu/

and controlled vocabulary. This is also emphasized by Aitchison, Gilchrist, and Bawden (2004, p. 68), who state that "Classification is an essential tool when finding structure and relationships during thesaurus construction." Finally, Broughton (2006) concludes: "It is clear that faceted classification in some form or another now plays an integral part in most methods of information retrieval. It is very well established as a method of construction in classification schemes and thesauri, and has affected the development of even the most conservative of systems in the area of traditional document description and organization."

There are, however, differences between classifications and other KOSs that are neglected by a unified view on them – as for example the representation by means of SKOS, the Simple Knowledge Organization System.[7] Consequently, there are difficulties "to express classifications [in SKOS] without sacrificing a large amount of their semantic richness" (Panzer & Zeng, 2009). These specifics of classifications (and other hierarchical KOSs) can be neglected for this thesis based on the following rationale:

1. The approaches presented in this thesis deal with the computer-aided, usage-driven creation and maintenance of KOSs and are valid and can be used independently of the specifics of the desired KOS. The final creation of the KOS still relies on the human expert who can deal with these specifics.

2. With the reuse and integration of different, existing KOSs for new purposes, it can be expected that there will be a further unification of the different systems and a further blurring of the borders. This results in a need for a unified tool set for the creation and maintenance of such KOSs.

As stated above, there is some overlap between the definitions of different KOSs and often it is not clear which category a certain KOS belongs to. The transitions between them are smooth and the use of the terminology should not be overrated. Moreover, the different terminology also refers to different use cases for very similar data structures. If further understanding about the differences of taxonomies and thesauri is needed, Gilchrist (2003) provides in-depth explanations from an etymological point of view.

Regarding the terminology and its placement, it is interesting that Wersig (1978, author's translation, emphasis of *thesaurus* added) stated about the term thesaurus:

> "Unfortunately, very early an inflationary use of the term [*thesaurus*] started to establish, which will be the result of a fascination, deriving from its external connection to classical models and the simultaneous image of modernity. [...] Although a considerable degree of clarity has been gained within circles of experts [...], almost any collection of descriptive elements [...] is addressed as *thesaurus* especially in

[7] http://www.w3.org/TR/skos-reference/

> adjacent areas of expertise, even though this is often not consistent with the established terminology, nor the original intentions."

More than 30 years later, this statement remains valid, if you replace *thesaurus* with *ontology*. Soergel (1999) even pointed out that ontologies are in some sense a reinvention of classification. So are ontologies only old wine in new skins? Of course not, but facing the current habit to call everything an ontology, it helps to keep in mind that there have been other ways of knowledge organization before. As Vickery (1997, p. 284) stated: "[...] 'ontological engineers' make little or no reference to work in information science. As a consequence, they do not appear to draw at all on the rich experience of constructing knowledge schedules [...]."

1.2 The KOS Life Cycle

KOSs have numerous important applications from improving indexing of document collections to faceted browsing and semantic search applications. The KOS creation and maintenance, however, is cumbersome and time consuming. On the other hand, many KOSs already exist and more and more of them become publicly available, ideally in the form of *linked open data* (cf. Section 1.3.3), and can be reused for different purposes.

If an existing KOS is to be reused, several tasks have to be performed, reaching from the proper *selection* of the source to start with to the adaption for the desired purpose which includes deletion of unnecessary concepts, merging and splitting of concepts and especially the addition of missing concepts. Otherwise, a new KOS has to be created from scratch. After this initial *creation*, the actual life cycle begins, consisting of constant *evaluation* and *modification* of the KOS based on its *use*. This means that a KOS is never finished and always has to be adapted to changes and developments in the reflected domain.

This thesis is organized alongside the KOS life cycle, as illustrated in Figure 1.2. In Chapter 2, we introduce the statistical framework and the visualization technique that we propose to adequately analyze the usage of concepts in a KOS. In Chapter 3, we are concerned with the analysis and evaluation of existing KOSs, i.e., with the selection and the evaluation steps in the life cycle (indicated by the eye 👁 in the diagram). In Chapter 4, we focus on specific approaches for the creation and modification of a KOS (indicated by the hand 👋 in the diagram). In Chapter 5, we explore implementational aspects regarding the integration of these approaches.

In the following, we briefly introduce the components of the life cycle.

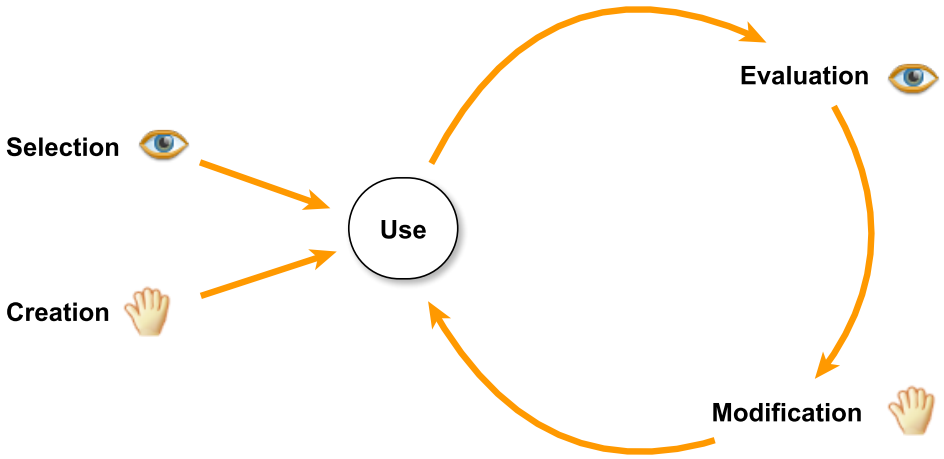


Figure 1.2: The KOS Life Cycle.

Selection. There are mainly two reasons, why one should start with an existing KOS, instead of creating one from scratch:

1. **Costs:** If an existing KOS perfectly meets the requirements, the best way is to just use it. Potential licensing costs will probably always undercut the costs of an own creation; not to mention the maintenance costs, if the existing KOS is properly maintained. And even if the KOS needs refinements for the desired purpose, it is worthwhile to at least start with an existing KOS.

2. **Interoperability:** The reuse of an existing KOS ensures interoperability between systems using it. As with the costs, even if the KOS is further refined and modified, interoperability can be achieved, if the unchanged concepts are mapped or linked to the concepts in the original KOS.

When an existing KOS is selected, the intended users and their prior knowledge of KOSs should be taken into consideration. Hammond (2001) claims "it takes an expert searcher a year to become familiar with a new vocabulary and its use" (quoted in McCulloch, 2004, p. 297). As McCulloch further points out, a mapping is required between all the different KOSs in use and creating a new KOS increases the existing disparity and makes an already difficult problem – the mapping – almost impossible to solve. While the mapping of different KOSs is out of the scope of this thesis, we will introduce a methodology to empirically evaluate existing KOSs with respect to their suitability for the desired domain in Chapter 3.

Creation. If no suitable KOS is available we have to create a new one. The creation process is always iterative; the first iteration that creates a new KOS is followed by subsequent modification steps, as described below. Therefore, we define the creation step as the necessary first step in KOS development, where

basic design decisions have to be made and a first stage is reached that is usable for indexing purposes.

Traditionally, the creation of a KOS is an intellectual, cumbersome work that is done manually by experts. These experts have to be not only KOS experts, but at the same time experts for the domain which is to be represented by the KOS. The general creation process of all kinds of KOSs, be they classifications, taxonomies, thesauri or ontologies, contains on a coarse-grained level the following subtasks:

- collection of significant terms,
- definition of concepts and relationships,
- organization of concepts, and
- testing and validation.

In Chapter 4, we present an approach to identify yet uncovered significant terms in documents (Section 4.2.1) and demonstrate the organization of concepts by means of crowdsourcing. The testing and validation leads to the subsequent iterations following the *use*.

Use. This thesis deals with the use of KOSs for document organization. To organize documents by means of a KOS, concepts of the KOS have to be assigned to the documents, i.e. the documents are annotated with concepts from a given KOS. This task is called *indexing*, which in this thesis is defined as follows:

Definition 1.5 (Indexing) *Indexing is the process of assigning concepts from a given KOS to a document, based on the content of the document. This can be done intellectually by humans or automatically by a computer program.*

We consider three types of indexing in this thesis: intellectual indexing, automatic indexing, and tagging. We restrict the use of the KOS as part of the maintenance cycle on the indexing process (cf. Section 1.4).

Evaluation. In this thesis, we demonstrate, how concept usage in form of indexing results – intellectually or automatically obtained – can be evaluated and how conclusions for the KOS maintenance can be drawn.

Especially for automatic indexing, the evaluation of the KOS and the indexer cannot be told apart. A common problem that is faced by everybody who wants to improve an indexing system is the choice of a quality measure that can be used to quantify the improvements. Generally, the quality of automatic indexers is evaluated by the Precision and Recall measure with manual annotations as gold standard, as shown by Neveol, Rogozan, and Darmoni (2006) or Aronson, Mork, Gay,

Humphrey, and Rogers (2004). Presumably, an advancement of the precision and recall measure is more suitable for this evaluation, as there are some graduations between correct and wrong annotations. These generalization of precision and recall to unsharp measurements has been done by Kekäläinen and Järvelin (2002), Maynard (2005), and Euzenat (2007). While such quality measures may be used to get an overall assessment of the indexing quality, they cannot be used to identify problematic areas that might be responsible for a lack of quality. The ICE-Map Visualization (Chapter 2), one of the main contributions of this thesis, provides a means to visualize the whole result of an indexing process and allows a human expert to depict specific problems.

We argue that the KOS should be the main target of intellectual effort, because it is independent of the indexing process and remains constant for arbitrary amounts of documents. The approaches presented in this thesis are developed along this rationale. We demonstrate the use of the ICE-Map Visualization for KOS based indexing evaluation in Chapter 3.

Modification. The evaluation is not a one-time event. Together with the modification of the KOS based on the evaluation results and subsequent re-evaluation, it is a repeating process that consequently adapts the KOS to changes in the covered domains and languages. Evaluation and modification are both integral parts of KOS maintenance. Typical questions that are asked during KOS maintenance are:

- What concepts are missing?
- At which location do new concepts have to be included?
- Are there concepts that should be removed?

Likewise, literature on thesaurus creation and maintenance mentions a number of tasks that might be necessary including the following taken from (Kuhlen, Seeger, & Strauch, 2004):

- Adaptation of the thesaurus to changes in the vocabulary of the domain of interest by means of adding new terms.
- Splitting, extension, or restriction of extensively used terms.
- Deletion and/or merging of rarely used terms.
- Review of the thesaurus structure to avoid extensive subclassing.

Depending on the intended use of the KOS, there might be more tasks. For example, if the KOS is used for automatic indexing, the following task has to be included:

- Identification of problematic concepts for the indexing software, i.e., concepts that are erroneously assigned or missing.

In Chapter 4, we present two approaches to support a KOS maintainer to modify a KOS based on terminology usage in literature and actual concept usage in the KOS.

1.3 Foundations in Information Science

In this section, we provide the context for this thesis regarding its application in information science. It is not mandatory for the understanding of the approaches in this thesis, but must not be neglected if these approaches are to be applied in a productive setting.

1.3.1 Motivation

The role of KOSs in information retrieval changed during the last decades, if not centuries, just as information retrieval and the role of information providers changed. It is an unavoidable fact that the human knowledge is constantly increasing and that it is increasingly difficult to stay up to date. Long since this is at best possible on a narrow portion, which requires a deeper and deeper specialization and expertise of the people in terms of their profession and – in the case of scholars – their research. A reason for that is the increasingly falling barriers to publish knowledge. The first decisive step was the invention of *book printing*, which led to a massive acceleration of the dissemination of information. Another factor was the progressive development of communication and transportation that brought people together and enlarged the focus of the individual regarding accessibility and incorporability of information. It cannot be denied that the invention and spread of the *Internet* was a similarly fundamental change as the invention of the printing press. Since information is published and distributed via the Internet, the distribution rate has increased to the theoretical and practical maximum: online publications are available world-wide in real time – directly and without delay.

While scientists mostly deal with scientific literature as scientific books or articles in peer-reviewed journals, the distinction between ordinary websites and scientific literature constantly vanishes. On the one hand, there are “black sheep” in the traditional scientific publication market that publish literature based on a very superficial peer-review, usually combined with high prices, either for the author or the buyer of the books. These publishers use new developments like printing on

demand and the economical distribution as e-books to reduce both the costs and the entrepreneurial risk that comes with a low quality publication. On the other hand, almost all established publishers use the Internet now for the distribution of and access to their publications in electronic form and new journals and proceedings arise that are only published online – often with an open access approach – with high quality, peer-reviewed content.

In light of this, the focus of publishers, libraries, and information centers has shifted. They are less important to generate and disseminate information than to review, filter, edit, and present information to the user with the desired quality. Thus, they keep the role as an important building block in the networked knowledge society. Both, however, have to adapt to these fundamental changes introduced by the Internet and closely work together to get under control what is commonly known as *information overload*.

Libraries constantly improved the way they describe and identify the contents: Starting with the use of authority files stored in databases about the authors to identify them in a unified way. Similarly, libraries also use controlled vocabularies to promote the consistent representation of subject matter, thereby avoiding the dispersion of related materials. By using subject headings and thesauri that represent hierarchical and associative relationships between topics, libraries were able to link together terms whose meanings are related paradigmatically or syntagmatically, cf. (Lancaster, 2004, pp. 7-8). Libraries have also developed comprehensive classifications primarily presented in hierarchical structures to organize the documents according to contents, thus to support systematic browsing based on areas of studies and disciplines.

The organizational systems originated in libraries are still relatively rigid. A rapid adaptation to new trends is sacrificed in favor of the precise control of the systems. Additionally, the actual subject analysis and indexing is primarily performed intellectually by information professionals, which on the one hand enables precise content research, yet on the other hand introduces a backlog or makes the complete indexing of all publications almost impossible. Individual articles in journals for example are usually not indexed by libraries,[8] not to mention the publications available only online. They are collected by libraries – in Germany by the German National Library (DNB) – but not indexed intellectually any more. Instead, automatically created search indices and general search engine technology is to be used (Schwens & Wiechmann, 2009).

In return, *web search engines* started to use background knowledge to overcome the lack of a controlled vocabulary. Some search engines now recognize terms and

[8]However, special libraries may provide databases of indexed journal articles, like the Econbiz database of the German National Library of Economics (Deutsche Zentralbibliothek für Wirtschaftswissenschaften, ZBW), http://www.econbiz.de/. Libraries mainly rely on database producers who provide indexing and abstracting services.

can assign a common meaning,[9] others can cluster search results according to different meanings of the search terms.[10] Search engines also started to explore more structured data-based searches, as libraries have been doing, using standardized schemas, for instance provided via Schema.org.[11] Meanwhile, they are taking ontological approaches for more automatic but semantic searches (Kerr, 2012).

These developments suggest that combinations of traditional library techniques with modern search engine approaches are best suitable to control the rising tide of information. This requires on one side the integration of library techniques into the technological infrastructure of the Internet and, on the other side, these techniques must also be considerably more dynamic, to quickly respond to the topical changes of our time at a reasonable speed. One approach for this integration of library techniques is the development of the *Semantic Web*, where the machine-interpretable description of all resources inside and outside of the web becomes an integral part of the infrastructure of the web. The linking and accessing possibilities of this infrastructure are used to access, use and integrate formerly locked-up data and allow completely new usage scenarios for it. Such *Linked Open Data* projects have gained significant attention in the last years within the library domain.[12] The linked data technology allows the immediate use, but also the easy adaption of existing KOSs to describe all kinds of resources. By means of links between these KOSs, while also by links between descriptions of the same resources, indexing results based on one KOS can be transferred to another KOS and the exchange of indexing results created at different locations in the world can be fostered.

1.3.2 Creation of a KOS

A KOS is used to describe documents and make them available for later access. Therefore, the first question is how to describe the documents properly for access purpose (other than management purpose). The description process is usually some kind of labeling, which is bounded by the human perception of the world. The human perception of the world and the ability to describe it can be separated into three levels:

- The level of **objects**: This is the "real" world, objects are what they are. For example the table you are sitting at is such an object, it is **the table**, no matter how you describe it. Objects in this sense can also be immaterial or mental constructs. For example, you can study computer science and no matter how you call it – or if you consider it a science – computer science exists, it is **the topic** that computer science students are concerned with for several years.

[9] For example Google, cf. (Baker, 2010)

[10] For example Yippy (http://search.yippy.com/), formerly known as Clusty.

[11] http://www.schema.org/

[12] See the final report of the W3C Library Linked Data Incubator Group, http://www.w3.org/2005/Incubator/lld/.

- The level of **concepts**: This level is the abstract model of the world that we form in our mind to describe the world and distinguish the objects. Objects of the same kind are classified as instantiations of the same concept. Beside the table in front of you, there are other tables and they are common enough that there has to be a concept to describe them. The concept TABLE is the same for all human beings, everybody knows what a table is, no matter that there are different words in different languages to call a table a table.

- The level of **labels**: Labels – or *terms*[13] – are used as placeholders for concepts. They are not abstract anymore, but again objects in the real world – in fact: instantiations of the concept label. We used "table" in this example as the label to describe the concept TABLE. "Desk" or "board" are also labels for the same concept (*synonymy*), as well as labels in other languages like "Tisch," "Tafel," "Schreibtisch," "mesa," "tavolo," or "τραπεζι."

  Labels are not only language dependent, they are also ambiguous and can potentially be used for different concepts (*homonymy*). Whether two labels properly describe the same concept is not always clear. In a strict sense, only synonyms can be used to describe a given concept. But even two synonyms can have subtle differences in the meaning, sometimes dependent on the context, in which they are used. For example, "table" and "desk" are not exactly synonyms, but may be used as such, if the difference between "desk" and "table" is not important for the intended application (*quasi-synonymy*). Which labels are appropriate depends also on the definition of the concept in mind. If, for example, there is only the concept FURNITURE, then beside "table" and "desk," also "chair," "cabinet," "bed," "couch," and "sofa" would be appropriate labels.

The problem of search applications is that they are on an intermediate layer between content creator and content consumer and both sides use a natural language in the first place, either to describe the content (if it is not the content itself, like in a text document) or to search for it. While both hopefully talk about the same concepts, they probably use different labels, which leads to the known problems in search applications: The result contains items that do not meet the searcher's needs (*false positives*) as well as it lacks some items that would be relevant for the searcher (*false negatives*).

Such issues and the notion behind the three levels can be traced back to ancient philosophy as well as modern text used by cross-disciplinary fields including linguistics, philosophy, language, cognitive science and semantics. One of the most well-known conceptualization is Ogden and Richards' (1923) semiotic triangle (Figure 1.3). Their *symbol* conforms to our *label*, the *thought or reference* to

[13] We use *label* and *term* synonymously in this thesis. *Term* is traditionally used in the context of thesauri, as in *preferred term*; SKOS (cf. Section 5.1.2) uses *label*; and ISO 25964-1 uses *term* for the class of terms and *label* for the property assignment, which defines *label* as a role for *term*, a view that certainly has some merit.

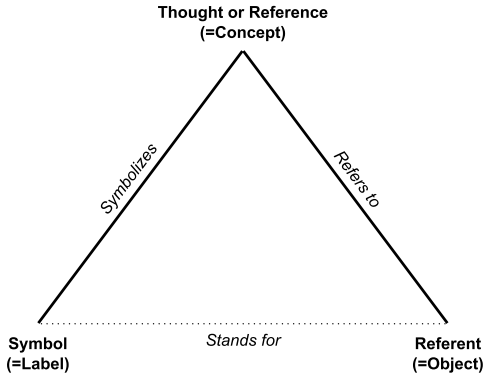


Figure 1.3: Semiotic Triangle (Adapted from Ogden and Richards, 1923).

our *concept* and the *referent* to our *object*. The main point – without going further into details – is the indirect relation between labels and objects via the concepts, as indicated by the dotted line.

For the actual KOS creation, especially for thesauri, detailed descriptions can be found in various guidelines, textbooks, and national and international standards. A list of references for thesaurus construction was composed by Nielsen (2004, pp. 62-63). Regarding the international standards, it has to be noted that there is a new ISO standard for thesauri that is currently under development (ISO, 2011). A brief guidance for the construction of thesauri is provided by McCulloch (2005). Further references about more specific tasks in the thesaurus creation process are provided by Aitchison et al. (2004) and Lancaster (2004). An example for a more detailed division of the creation process into nine stages is presented by Shearer (2004) in form of a practical exercise for the creation of thesauri. The creation of a faceted classification is described by Buchanan (1979), primarily based on the works by Ranganathan (1937, 1945). There is also a German industry standard (DIN 32705) that is concerned with classification systems (DIN, 1987).

Based on the best practices provided by these literature and manuals, the following steps can be identified:

- Collection of significant terms
    1. Collecting the raw terms from the literature and other sources
    2. Controlling synonyms; distinguishing homonyms
- Definition of concepts and associated terms
    3. Grouping terms into broad concepts
    4. Identifying relations between different concepts
- Organization of concepts

5. Ordering the concepts and subconcepts

Generally, there are two different strategies to develop a KOS: top-down (systematic) or bottom-up (pragmatic). While in practice, usually a kind of mixture between these two strategies is applied, it is important to understand the difference and the pros and cons of both strategies:

Top-Down (systematic): With a top-down strategy, the KOS is systematically created to reflect the background knowledge of a whole domain. This requires expert knowledge and is a time-consuming and expensive process. The advantages of this approach – according to Wersig (1978) – are the

- consistent coverage of the whole domain,
- better focus on the adequate hierarchy of the terminology,
- better decisions about equivalences and concept relations, and
- higher generality of the KOS regarding future developments.

It is disadvantageous that

- the KOS probably will consist of many concepts that are not used actually later on,
- the used terminology will probably not reflect the actually used terminology,
- experts often tend to see their domain ideologically and will hardly reach a conclusion about some specific questions. They probably will not understand the pragmatic requirements of a KOS for the specific purpose of the organization of resources for an easy retrieval.

Bottom-Up (pragmatic): The disadvantages of the systematic approach already indicate the advantages of the pragmatic approach, where concepts are identified based on the actual indexing work and later on harmonized and organized into a hierarchy:

- The pragmatic approach is empirical as it introduces concepts that reflect the actual usage in the resources being indexed, independent – or at least more independent than with the systematic approach – of the personal view of the KOS creator.
- There is almost no work to be done before the KOS can be used as it is created on the fly.

The latter advantage is well illustrated by Wolters (1997) for the classification of museum items. Often, the exact description of an item is not even known when it is classified for later retrieval. For example, there might be a thing that looks like a hammer. To the classifier, however, it is not clear,

what kind of hammer it is. While it would be desirable to have a more specific class, it is still necessary to classify it immediately with an appropriate label. If at a later time a specialist for ancient hammers is available, the classification of the thing can be further specialized and narrower subconcepts of HAMMER can be introduced as necessary. The pragmatic purpose of the first classification is not to identify the resource with a perfect and final concept, but to make it retrievable as "another thing that looks like a hammer" when the hammer specialist is around. Of course, the bottom-up strategy also carries some drawbacks:

- The resulting KOS is strongly fixated on the current state of the indexed resources and the actually used terminology. Thus, future trends and developments will be missed and the maintenance of the KOS is more important and probably more expensive than for a systematic approach.
- The indexed literature is usually not a good source for a controlled terminology. The resulting terminology might be overly specific.
- The downstream organization or reorganization is complex, as a proper reindexing of existing resources has to be ensured. So it could happen that the necessary reorganization steps are avoided and the KOS becomes unstructured and cluttered.

Both strategies can benefit from a computer-aided creation process, as we will show in this thesis. A typical practical approach that combines both strategies might look like this:

1. (Top-Down) Create a coarse hierarchy of the target domain based on available expert knowledge. The hierarchy is based on elementary decisions:
   - Is a polyhierarchy desired?
   - What kind of relationships are supported?
   - What is the main relationship that forms the hierarchy?
2. (Bottom-Up) Extract meaningful terms from domain-specific publications and populate your hierarchy with them.
3. (Top-Down) Evaluate the existing KOS and identify areas that are not well populated or overly populated and refine your KOS based on the aforementioned elementary decisions.

1.3.3 History and new applications of KOSs

According to Spärck Jones (1972), the history of KOSs for indexing purposes goes "back to the nineteenth century, subject indexing to Cutter in 1876 and classifications to Dewey, also in 1876, while the best known general thesaurus, that of

Roget, dates from 1852." Spärck Jones also states that there are of course older approaches to organize vocabularies, but the mentioned systems define clear milestones in the history of KOSs that influence the organization of information until today and in the future. Earlier approaches to "conceptualize the world" are listed by McCray (2006).

While classifications like the Dewey Decimal Classification (DDC) and flat subject heading lists belong to the librarians' toolbox since the 19th century, the thesaurus as an information retrieval device became popular only after the Second World War, as the number of (scientific) publications, especially in the non-book sector, massively increased. The introduction of thesauri was a paradigm shift from pre-coordination to post-coordination, i.e. instead of providing classes and subject headings for every thinkable topic of a publication, a thesaurus was a structure of irreducible unit descriptors that could be combined to describe a publication appropriately.

The hierarchy of the thesaurus was mainly needed for the document search: while the most specific terms are generally used to index documents, the hierarchy ensures that specific documents match when general terms are used in search requests (Spärck Jones, 1991). Aitchison and Dextre Clarke (2004) further described the history of thesauri, recent changes due to the use of computer systems to support the creation and use of thesauri, as well as necessary changes in the relevant standards.

With the full text or abstracts of most documents available for indexing, it is questionable if today we need intellectual subject indexing at all. This was examined by Gross and Taylor (2005) for keyword search in document titles. They found that subject headings improve the search results significantly. Over 30% of relevant documents would not be found with keyword search that is limited on the title. Full texts of books are not yet available for retrieval purposes in libraries and it is doubtful if pure full text retrieval would satisfy the users' needs. The intermediation by means of a KOS between the information need of a user and the actual content of relevant documents – and also between different languages – will probably become even more important with the ongoing growth in publications.

In the following, we further elaborate on some aspects of current and possible future usages of KOSs that can especially benefit from the approaches presented in this thesis:

Query Expansion. The traditional use of a KOS, even in the post-coordinated form of a thesaurus, requires the documents to be indexed in the first place. This is not feasible for all kinds of publications and documents that are available today, at least not manually. To overcome the weaknesses of a simple full text search, a thesaurus still can be used: The query of a user can be extended at the time of the search to find as much matching documents as possible. Such a usage of a the-

saurus as a background knowledge for retrieval systems becomes more and more important, but is not trivial. Grefenstette (1994) identified two common reasons that may lead to degraded results for unsupervised query expansion:

- No distinction between modifiers and head nouns in queries. If a noun is used in the query to restrict the results, like "leg" in "leg injuries," the expansion of "leg" will probably not lead to an improvement of the results, while the head noun "injuries" is a suitable candidate for expansion.
- No distinction between the types of semantic relations between the query term and its expansions. The expansion of a term with its antonym is generally not a good idea, for example if the term "female" occurs in a query, it is usually included for a restriction of the results and the expansion with "male" would be counterproductive.

While the latter is avoidable with clear semantics of the concept relationships, the former illustrates just one of the many problems that can be subsumed as problems of *natural language processing* (NLP). A typical approach to avoid the NLP problems is the inclusion of the user. For instance, Nelson (1992) presented the ConQuest text retrieval system that uses preprocessed machine readable dictionaries to enhance the retrieval quality. The query is expanded interactively by the user, the main possibilities include weighting of query terms and word sense disambiguation of ambiguous terms. The query is then extended with related terms for the specified meaning.

Another approach is the attempt to visualize the results in a proper way, as proposed by Korfhage (1991) who proclaimed a new retrieval paradigm, "one that focuses on the organized display of *all* documents, rather than on the linear display of just the 'best'." Grefenstette (1994) mentioned that such a visualization could be adapted to the visualization of related terms for an interactive query expansion that avoids the NLP problems. These approaches have the problem that they describe more or less complex systems that have to be understood by the user. White and Marchionini (2007) examined the effectiveness of real-time query expansion – the suggestion of additional search terms during the formulation of the query – and conclude: "The future of IQE [Interactive Query Expansion] may lie in techniques that couple query expansion more closely with searchers' normal information-seeking behaviors."

KOS and the Web. Based on the current developments in the library domain, it can be expected that KOSs can and will play an important role in the future of information retrieval on the web. More and more libraries have published and are publishing their KOS on the web by means of the RDF[14] and SKOS. In this way, KOSs become a notable and important part of the Semantic Web and can be used to

[14]Resource Description Framework, http://www.w3.org/RDF/

describe unstructured information like web pages with concepts adhering to clear semantics. But they might play an even more important role in the linking of structured information by means of Semantic Web technologies, commonly referred to as Linked Data.

The German National Library has started to publish its authority files (Persons, Corporations and Subject Headings) as Linked Data[15] and it is the declared goal to establish libraries and other cultural institutions as a reliable backbone of the web of data (Altenhöner, Hannemann, & Kett, 2010). Furthermore, authority data is published by the Library of Congress[16] (USA) and the French National Library. All these subject headings are linked.[17] Further examples are the STW Thesaurus for Economics and the TheSoz Thesaurus for the Social Sciences (cf. Section 1.5). This list is by no means exhaustive.

All these developments will have an impact on how KOSs in general are seen and how they will be used. First and foremost the KOSs will be decoupled from their specific purpose for which they were created in the first place. Whenever a KOS is published on the web, it can and will be used in various settings and for various reasons. As a result, it can be expected that the lines between the different types of KOSs, be it thesauri, classifications, or ontologies, will blur even further.

It will be challenging to cope with all these new usages – some of them probably cannot even be foreseen today. It is an interesting question, if today's processes of the creation and maintenance of KOSs – mainly within libraries – will be fit for this future and if and how they might have to change, especially regarding the inevitable disadvantages like high costs and slow adaptions to changes in the domain. The high quality and stability, which are generally considered more important than flexibility, is on the other side the main asset of the KOSs. Only quality and stability can let them become a real backbone.

Reuse in enterprise settings. The general availability of KOSs and their use in data integration scenarios on the web also make them interesting for the use in enterprise settings. They can serve as background knowledge for internal document and knowledge management purposes and retrieval applications. As most documents are available in machine-readable formats today anyway, they can be easily indexed (semi-) automatically.

Maybe even more promising is the possibility to link the internal data – be it documents or other assets and processes – to existing data outside, publicly available or within closed networks or logistics chains. Of course not many enterprises are willing and able to create and maintain their own KOS, due to the high costs and effort. But with the proper tool-support – as presented in this thesis – and the

[15] https://wiki.d-nb.de/display/LDS/

[16] http://id.loc.gov/

[17] http://www.cs.vu.nl/STITCH/rameau/

possibilities of reuse and adaption of existing KOSs, the cost-benefit ratio might change quickly towards the worthwhile use of KOSs in a broad range of applications.

1.3.4 Information Retrieval in Libraries

Until recently, the typical information retrieval system in a library was the Online Public Access Catalog (OPAC). The first OPACs either attempted to emulate the familiar card catalog in its new online form or they adopted the model familiar to online database searchers of commercial search services (Hildreth, 1995). They soon were replaced by so-called second generation OPACs that combined these two approaches and enhanced the search possibilities significantly, but at the same time increased the complexity for the user.

Borgman (1996) performed several studies on the problems of OPAC users between 1986 and 1996. She concludes that only little improvements, if any, were done in this interval to improve the usability of catalog systems. Users still needed assistance to translate their questions in a structured query that can be interpreted by the retrieval system. They were mainly usable by librarian experts, not by the typical library user. The system design should follow the users' search behavior, not the other way round.

Regarding the use of KOSs for information retrieval, the discrepancy between expert searchers and typical library users is evident: Fidel (1991a, 1991b, 1991c) examined in detail the search behavior of professional searchers and found that they heavily relied on thesauri, they "consulted them for 75% of the search keys they selected " (Fidel, 1991a, p. 512). In contrast, studies about the use of the traditional OPACs showed that subject search was often not successful or satisfying, mainly because only few users could take advantage of the controlled subject headings that were available in the library catalog (Sridhar, 2004; Yu & Young, 2004). Greenberg (2004) found in an – admittedly limited – study with typical library users (educationally advanced students pursuing MBA degrees) that the users' thesauri comprehension is extremely limited and that – given a basic thesaurus introduction – users indicate a desire to use thesauri. These studies suggest that the use of KOSs for information retrieval has to be more intuitive for the users.

With the wide-spread use of Internet search engines, a new component came into play: now the users were not only inexperienced with the use of OPACs, they had expectations toward an OPAC that result from their experiences with search engines like Google. Yu and Young (2004) describe this development in depth and suggest that OPACs have to implement search engine features like natural language search with keywords, relevance feedback, spelling corrections, and relevance ranked output. Similar statements are made by Campbell and Fast (2004)

– they see a huge potential for new innovations in the complementary relationship between catalogs and search engines.

Search engines do not only have an impact on usability expectations, today they are an inherent part of any information search: According to Rosa (2006, p. 1-7), "89% of college student information searches begin with a search engine." But what are the differences between OPACs and search engines? In 2002, Eversberg (2002, p. 122) stated that "catalogs and search engines are juxtaposed in a pears vs. apples comparison."[18] But he also admits that "there are, however, widening 'grey' areas: Genuine Internet resources are being cataloged to enrich catalogs. And search engines index files that contain book reviews, abstracts, whole chapters, descriptions, etc."

Since 2002, more and more smooth transitions emerge between catalogs and search engines. In 2004, Google started with two new services: Google Scholar[19] and Google Books – formerly known as Google Print.[20] Both provide access to documents that were only available via library catalogs by then. There are visions of digital libraries, where the user can search and browse the whole inventory and access all documents (and audio files, movies, ...) with one click at any time and any place in the world, for instance the open library project[21] of the Internet Archive.[22]

From the side of the OPACs, the transition towards search engines is highly visible, too. In 2007, the Mannheim University Library introduced Primo, a commercial solution by Ex Libris,[23] that integrates various sources for bibliographic data, not only about the books that are physically available at the library, but also data about single articles in subscribed journals and huge amounts of data for articles and e-books that are available to the library users through other channels. The interface is familiar and intuitive for search engine users and the result lists can easily be sorted and filtered by various aspects (drill down, faceted search), which, in turn, is a feature that Google just recently added to its standard search interface. Like Ex Libris with Primo, other commercial vendors of library solutions have similar products. They are commonly referred to as *Resource Discovery Systems* and employed in an increasing number of libraries world-wide.

Another solution worth to mention is the library resource portal VuFind,[24] which is developed and maintained by the Villanova University, PA, USA and provided free of charge under an open source license. Regarding the features, VuFind is sim-

[18] Quoted from the English version of the author, available at http://www.allegro-c.de/formate/tlcse.htm

[19] http://scholar.google.com/

[20] http://books.google.com/

[21] http://openlibrary.org/

[22] http://www.archive.org/

[23] http://www.exlibrisgroup.com

[24] http://vufind.org/

ilar to Primo and already widely used, i.e., at the time of this writing (April 2012), the website lists[25] 90 libraries that were at least testing VuFind, among them 59 already used it in a production setting.[26]

This development even fulfills the requirement for a more intuitive use of KOSs in the retrieval systems. Recently, Kules, Capra, Banta, and Sierra (2009) showed that subject headings are now used for the faceted search and users seem to literally rediscover them, as one participant commented (p. 320): "The subject thing worked. I don't normally do subject searches."

1.4 Research Questions, Contributions, and Limitations

The classical approach, where KOSs are created and maintained solely at a central location and the indexing based on these vocabularies is done intellectually by experts, does not scale and no longer meets future requirements. Information overload is a phenomenon not exclusive to libraries. Dealing with the organization of information has become an important task in almost all areas, including companies and wherever new information is created and existing information has to be accessed.

On the web, be it "classic" or "semantic," all the traditional procedures are reaching their limits. The success of alternatives such as tagging – the assignment of uncontrolled keywords by all users – shows that such new ways are promising and contribute to the organization of all these information. Tagging, however, lacks control – not necessarily in the sense of quality control, but de facto in the sense of controlled vocabulary. If intellectual indexing by means of librarians is too cumbersome and automatic processes lack quality, then the most promising way is to use all these methods simultaneously, to bring them closer together, to transfer positive properties of one approach to another, and obtain an indexing result that guarantees the best possible quality for retrieval purposes at reasonable costs.

The purpose should never be forgotten: it is not about indexing content especially nice or beautiful or to create the one and only best KOS to describe the world. Ultimately, all this is only a means to an end, to make the information findable for the user.

To achieve wide-spread applicability of KOSs in information retrieval, new tools for efficient KOS evaluation and maintenance are needed.

[25] http://vufind.org/wiki/installation_status

[26] For comparison, in August 2010, 50 libraries used or tested VuFind, among them 27 in a production setting.

Research Questions. In this thesis we develop and investigate suitable approaches and pursue to give answers to the following research questions:

1. **How can the structure and usage of a concept hierarchy be visualized in a way that it provides meaningful information to the practitioner?** This question is motivated by two steps in the KOS life cycle that both invoke analysis: *selection* and *evaluation*. Flexibility is needed as "meaningful" depends on the task that is performed by the practitioner. To select a KOS, the suitability of the KOS for a given set of documents has to be determined. For a proper KOS evaluation, it is necessary to make oneself familiar with complex KOS structures in combination with huge document collection in a short time. Most existing visualizations have problems to deal with the exponential growth of tree structures and fail to provide an overall picture.

2. **To which extent can and should alternative, usage-driven approaches be applied for the creation and maintenance of concept hierarchies?** Many approaches exist that aim to create concept hierarchies from texts. The fully automatic creation of useful concept hierarchies, however, is very difficult – if not impossible, as it requires an understanding of the world that is not available for computers. It might be more promising to start with single subtasks and to provide a good tool support to make the creation and maintenance more efficient and in line with the actual usage. Furthermore, we take into account alternatives like crowdsourcing.

3. **What are the characteristics of different indexing processes regarding concept usage and how does that affect KOS maintenance?** Automatic and alternative indexing approaches play an important role, but how do they perform in comparison to traditional intellectual indexing? We think that analyses like precision and recall that produce a mere number are not a suitable means to give answers to practitioners. If we use these alternative indexing processes, which are not under our full control, we need to evaluate and constantly monitor the results. Weaknesses in the vocabulary lead to weak indexing quality, just as a weak indexing process. Can these weaknesses be identified and subsequently be settled?

Contributions. Alongside these questions, we make the following contributions in this thesis:

- **ICE-Map Visualization (Chapter 2):** From a theoretical perspective, the main contribution is the development of the ICE-Map Visualization, which is used throughout this thesis to support the usage-driven KOS maintenance.

- **Application of the ICE-Map Visualization for the proper selection of a KOS (Chapter 3):** With the increasing availability of concept hierarchies online and their use in a (Semantic) Web context, the proper evaluation and

choice of one of them becomes more important. We show that the ICE-Map Visualization combined with a simple indexing approach is a suitable means to help the user in this decision process. In the same way, the ICE-Map Visualization can be used to explore new document collections, e.g., to decide whether to purchase a new database in a library.

- **Comprehensive evaluation of indexing processes (Chapter 3):** Indexing results are usually evaluated only by comparison with a reference in terms of precision, recall, and F-measure. We show that the ICE-Map Visualization clearly improves such evaluations and helps the practitioner to understand the indexing results, as this visualization provides a far more intuitive and rich insight in the characteristics of the results, than the single figures of precision or recall. We use the ICE-Map Visualization to show and explain the characteristics of three different indexing approaches: traditional intellectual indexing, automatic indexing, and tagging.
- **Bootstrapping a KOS by means of crowdsourcing(Chapter 4):** In this thesis, we assume that there is no satisfying approach for the full automation of a creation process for concept hierarchies. To reach a suitable starting point for further refinements of a KOS during the life cycle, we examine the use of crowdsourcing to bootstrap a hierarchy from scratch.
- **Support of KOS modifications as part of maintenance (Chapter 4):** We focus on the support of domain experts for the necessary refining modifications of a KOS. We develop an algorithm that proposes new terms as candidates for synonyms or new concepts, combined with a guess for their appropriate locations in an existing hierarchy; as well as an algorithm that proposes new subconcepts for overpopulated concepts by means of document clustering.
- **Semtinel and LOHAI (Chapter 5):** The main practical contribution of this thesis is the definition and implementation of a framework that supports the development and use of the approaches as presented in this thesis. It is extendable and supports developers and users to follow the principles that are established in this thesis: put the human in the loop and focus your effort on the concept hierarchy. The framework is called Semtinel[27] and is published under an open source license. Not least, we develop a simple automatic indexer called LOHAI for the KOS selection approach that is integrated in Semtinel.

Limitations. This thesis is subjected to the following limitations: The approaches for KOS creation and modification are very specific and function as examples how a KOS can be created from scratch or how common maintenance steps can be sup-

[27] http://www.semtinel.org/

ported keeping the maintainer in the loop. We argue that fully automatic creation of KOSs lacks quality, however, automatic approaches could serve well for bootstrapping.

We limit ourselves to hierarchical KOSs, particularly thesauri and classifications. At the same time, we abstract from specific relations between concepts; we only assume that the relations are suitable to form a hierarchy. The approaches are not applicable for flat structures like simple controlled vocabularies and only with restrictions applicable for expressive ontologies that provide many – usually non-hierarchical – relations. Although the involved KOSs (see next section) in the experiments represent common structures of thesauri and classifications, there could be other types or variations that are not represented by those involved.

Furthermore, we limit ourselves to indexing regarding the usage of a KOS. This means we solely evaluate a KOS based on indexed documents, not based on query results as part of information retrieval. This makes the approaches presented in this thesis applicable in almost every setting. In contrast, the evaluation of query results requires access to a retrieval system with a significant number of users and queries. Moreover, user feedback has to be obtained, either explicitly or by means of information extraction from query logs. An extension of our approach – e.g., the visualization of queries mapped to the KOS – would be interesting future work.

We aim for a proper evaluation of our approaches, wherever this is possible. Regarding the ICE-Map Visualization, we demonstrate the usage and give – in our opinion – convincing examples that indicate the usefulness of our approach. Additionally, we demonstrated and showcased the visualization at various occasions (listed in the text and the acknowledgements of Chapter 2 and Chapter 3) and gathered general feedback from practitioners who tested it. A thorough evaluation in a productive setting, i.e., the actual application of our approaches for KOS maintenance, however, is missing. This is mainly due to the fact that such an evaluation needs significant time, also on the side of the KOS maintainers; despite the general willingness of some maintainers, we were not able to set up such a large-scale evaluation with reasonable effort.

1.5 Research Data

For reference, we describe briefly the Knowledge Organization Systems that we use in this thesis. For two KOSs, we provide examples for concept representations. Additionally, we provide further information and links about the research data and implementations that have been created for the experiments. We aimed at using publicly available resources as much as possible to make our experiments transparent and reproducible.

- N Related subject areas
  - N.01 Philosophy, Philosophy of Science, and Religion
  - N.02 History
  - N.03 Demography and Population
  - N.04 Politics and Political Science
    - N.04.01 Political Theory, Political Philosophies, and Ideologies
    - N.04.02 Political System
    - N.04.03 Areas of Politics
    - N.04.04 International Relations and Foreign Policy

Figure 1.4: Excerpt of the STW categories.

Knowledge Organization Systems

Standard Thesaurus for Economics (STW)

The Standard Thesaurus Wirtschaft[28] (STW) is a thesaurus for economics and related areas. It is maintained by the German National Library of Economics - Leibniz Centre for Economics (ZBW). In its version 8.08, it consists of about 6,600 concepts and about 25,000 terms in German and English. In our experiments, we used two versions, 8.03 and 8.08. The STW is available in RDF (Neubert, 2009).

The thesaurus is divided in two parts, the hierarchically related descriptors and a top-level hierarchy that is not used for indexing. The top-level hierarchy consists of the following subtrees:

1. General descriptors (Allgemeinwörter, A)
2. Business economics (Betriebswirtschaft, B)
3. Geographic names (Geographische Begriffe, G)
4. Related subject areas (Nachbarwissenschaften, N)
5. Commodities (Produktteil, P)
6. Economics (Volkswirtschaft, V)
7. Economic sectors (Wirtschaftszweiglehre, W)

Figure 1.4 shows an excerpt of the top-level hierarchy.

The concepts are polyhierarchically ordered and can be assigned to several of the top-level categories. For our experiments, we built a single thesaurus and treated the categories as additional concepts on top of the STW concepts. Table 1.1 shows an example for an STW concept (German terms excluded).

[28] http://zbw.eu/stw/

Preferred Label	Globalization
Categories	B.00 Business Economics N.04.04 International Relations and Foreign Policy V.07 International Economics
Used for	Globalisation Internationalization
Related Terms	Economic Integration Fair trade Free trade Geo economics ...
Broader Terms	International relations
Persistent Identifier	http://zbw.eu/stw/descriptor/19071-3

Table 1.1: Example of an STW concept.

Thesaurus for the Social Sciences (TheSoz)

TheSoz[29] is the Thesaurus for the Social Sciences maintained by GESIS Leibniz Institute for the Social Sciences. It serves as a crucial instrument for indexing documents and research information in the social sciences and covers all topics and sub-disciplines of the social sciences. Additionally terms from associated and related disciplines are included to support an accurate and adequate indexing process of interdisciplinary, practical-oriented, and multi-cultural documents. The TheSoz comprises about 8,100 concepts, described by German, English, and French terms. We use the RDF implementation (Zapilko & Sure, 2009) in version 0.86. In size and structure, the TheSoz is comparable to the STW. Additionally, it partly covers the same topics, which makes it highly interesting for our work on KOS selection in Chapter 3.

Medical Subject Headings (MeSH)

The MeSH thesaurus is a well-established polyhierarchic thesaurus from the medical domain that is extensively used to annotate large collections of medical documents. It is produced by the National Library of Medicine (NLM) and continuously updated since 1960. It is used for cataloging the documents and related media and as an index to search these documents in a database. MeSH 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 is available online.[30]

[29] http://lod.gesis.org/thesoz/

[30] http://www.nlm.nih.gov/mesh/

- 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

Figure 1.5: Excerpt of the MeSH structure.

Similar to the STW, MeSH concepts – called descriptors – are not directly ordered by means of broader/narrower relations. Instead, they are assigned to hierarchically ordered categories, which form the tree structure. On top, there are 16 categories – category A for anatomic terms, category B for organisms, C for diseases, etc. Each is further divided into subcategories. These categories form a tree with up to eleven levels which is primarily used to provide an access to the terms and is not meant to be an exhaustive classification.[31]

In its current 2011 version, MeSH contains 26,142 concepts and over 177,000 terms. These descriptors are assigned to one or more categories in the tree structures. For each appearance of a descriptor, a number is assigned (Figure 1.5). These numbers are used to locate the descriptors in each tree and to alphabetize those at a given tree level. They 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 1.2 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.

Indiana Philosphy Ontology (InPhO)

The InPhO project [32] maintains a taxonomy of about 1,000 philosophical concepts (called ideas) extracted from the Stanford Encyclopedia of Philosophy. This taxonomy is special as it is created automatically based on information provided by the users of the project. This is not only a productive example of applied crowdsourcing, it is also the ideal basis for our evaluation of the Amazon Mechanical Turk as an example for a paid crowdsourcing solution in Chapter 4.

[31] cf. http://www.nlm.nih.gov/mesh/intro_trees.html

[32] https://inpho.cogs.indiana.edu/

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 1.2: Example of a MeSH Descriptor.

20 Newsgroups

20 Newsgroups[33] is actually not a KOS, it is a collection of about 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups. The 20 newsgroups collection has become a popular data set for experiments in text applications of machine learning techniques, such as text classification and text clustering. The newsgroups are organized in a shallow hierarchy:

- comp
    - comp.graphics
    - comp.os.ms-windows.misc
    - comp.sys
        - comp.sys.ibm.pc.hardware
        - comp.sys.mac.hardware
    - comp.windows.x
- rec
    - rec.autos
    - rec.motorcycles
    - rec.sport
        - rec.sport.baseball
        - rec.sport.hockey
- sci

[33] http://people.csail.mit.edu/jrennie/20Newsgroups/

        - sci.crypt
        - sci.electronics
        - sci.med
        - sci.space
    - misc.forsale
    - talk
        - talk.politics
            - talk.politics.misc
            - talk.politics.guns
            - talk.politics.mideast
        - talk.religion.misc
    - alt.atheism
    - soc.religion.christian

In Chapter 4, we use this dataset as a testbed for our concept splitting approach, mainly due to its popularity for clustering and its public availability. For the specific task of splitting a concept into subconcepts based on assigned documents, no further hierarchy is needed.

WordNet

WordNet[34] is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations. The nouns and the provided hypernym/hyponym relationships form a KOS following the definition in this thesis consisting of about 80,000 concepts and 118,000 terms.

WordNet is more than a thesaurus, not only because it also provides verbs, adjectives and adverbs: WordNet disambiguates the different senses of terms and provides semantic relations among them. Not least due to these fine-grained semantic relations on the term-level, WordNet is widely used for various research activities. We use it in Chapter 4 to evaluate our approach for KOS extension using web search engines.

Further Data and Implementations

For this thesis, we used Semtinel in the version of the commit b4c49d6 in the Git repository on Sourceforge.[35] The compiled version of Semtinel[36] and the complete

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

[35] SHA1: b4c49d67830af5cafd07812dd4a1a1892a847ac8, http://semtinel.git.sourceforge.net/git/gitweb.cgi?p=semtinel/semtinel;a=commit;h=b4c49d6

[36] SHA1: ec51a1a4e3e2d1bc448551eddeb36720eabde51c

database[37] working with this version containing all KOSs, document sets and annotation sets for the experiments in Chapter 3 is available online.[38] The hierarchy produced in Chapter 4 by means of Amazon Mechanical Turk is available online.[39] The implementation of the KOS extension in the same chapter is available as plugin for Semtinel and can be found in the Git repository as well, under *plugins/Thencer*. The implementation of the concept splitting is available online.[40]

[37] SHA1: 2e28a413da94e14d40b4951b3113d995a494a8e5
[38] http://www.kaiec.org/2012/dissertation/semtinel
[39] http://www.kaiec.org/2012/dissertation/amt-inpho
[40] http://www.kaiec.org/2012/dissertation/concept-splitting

Chapter 2

Statistical Analysis of Concept Hierarchies

The creation, maintenance, and the actual use of a KOS requires a sound knowledge of its structure. Therefore, in a dynamic setting, where new KOSs have to be chosen and deployed, as well as during the maintenance of existing ones, a proper tool support is required.

As a motivating example, consider the use of an automatic indexing system. Naturally, one is interested in the evaluation of the indexing results, but the evaluation is not trivial for KOSs with thousands of concepts and thousands of documents to be indexed. At the very beginning of the work leading to this thesis, we faced the problem to evaluate and judge such a KOS-based automatic indexing system. It turned out that this is not an easy task. In the light of thousands of concepts and even more documents, manual checking is not feasible.

The standard approach to evaluate such a system is the comparison of the results to the results obtained intellectually by a domain expert. The result of this comparison is a number, called F-Measure. We calculated, we got an F-Measure of 56%. Is 56% good, is it bad? Can we improve it? Where are the problems? Based on one number, you cannot make an informed decision.

As Goethe in our opening quotation, we wanted to see it clearly. We wanted to see the KOS and we wanted to see the indexing result. We wanted to see the forest for the trees. The ICE-Map Visualization is developed for exactly this scenario: The concept use is visualized to allow for a proper and intuitive evaluation of the indexing result. The evaluation and analysis of a concept hierarchy, however, is not only needed when the hierarchy is actually used. We will show that the ICE-Map Visualization directly supports the creation and maintenance of a KOS. As described in Section 1.2, we assume five typical tasks during the maintenance of a concept hierarchy:

1. Adaptation of the concept hierarchy to changes in the vocabulary of the domain of interest by means of adding of new terms or concepts,
2. splitting, extension or restriction of extensively used concepts,
3. deletion and/or merging of rarely used concepts,
4. review of the hierarchical structure to avoid extensive subclassing and
5. identification of problematic concepts for the indexing software, i.e. concepts that are erroneous assigned or missing (introduced in this thesis).

The first task is separately addressed in this thesis (cf. Section 4.2). The remaining four tasks can be supported by means of the ICE-Map Visualization, which makes it a very universal and powerful tool.

In order to enable a domain expert to carry out these actions, we analyze the KOS and detect unbalanced hierarchy structures as well as terms that are more often or less often used in indexing than we would expect. We support this step using a statistical framework together with a proper visualization that makes it easy for the user to spot potential problems.

2.1 Visual Datamining

In 1854, there was a severe Cholera outbreak in London in the Soho district. At this time, people generally believed that Cholera was caused by polluted air (miasma theory). John Snow, a physicist, questioned this theory and tried to find evidence for another source for the Cholera, particularly the drinking water. So he investigated the Cholera cases carefully and gathered a lot of data about them. He drew a map of the affected area and marked every fatal case with black bars (Figure 2.1).

On this map – later called the "Ghost Map"– it can be seen that the cases are scattered around the Broad Street and based on the distribution, John Snow had the suspicion that the water pump in the Broad Street could be the source. He convinced the district council to disable the pump and subsequently, the Cholera cases decreased.

This is the very condensed version of the story that is often called in various slightly modified versions as the "invention" of visual data mining, i.e. the analysis of data and the identification of correlations by means of a proper visualization. In this sense, it became a myth[1] (McLeod, 2000).

[1] In fact, John Snow was not the first one to use maps for data visualization and the drawing of the map was actually only one means for his investigations (Koch, 2004) – see also the original report (J. Snow, 1855). Nevertheless, the "Ghost Map" is a very nice example for the power of a proper visualization. The story about John Snow and the Cholera outbreak is the central theme of a novel called "The Ghost Map" (S. Johnson, 2006).

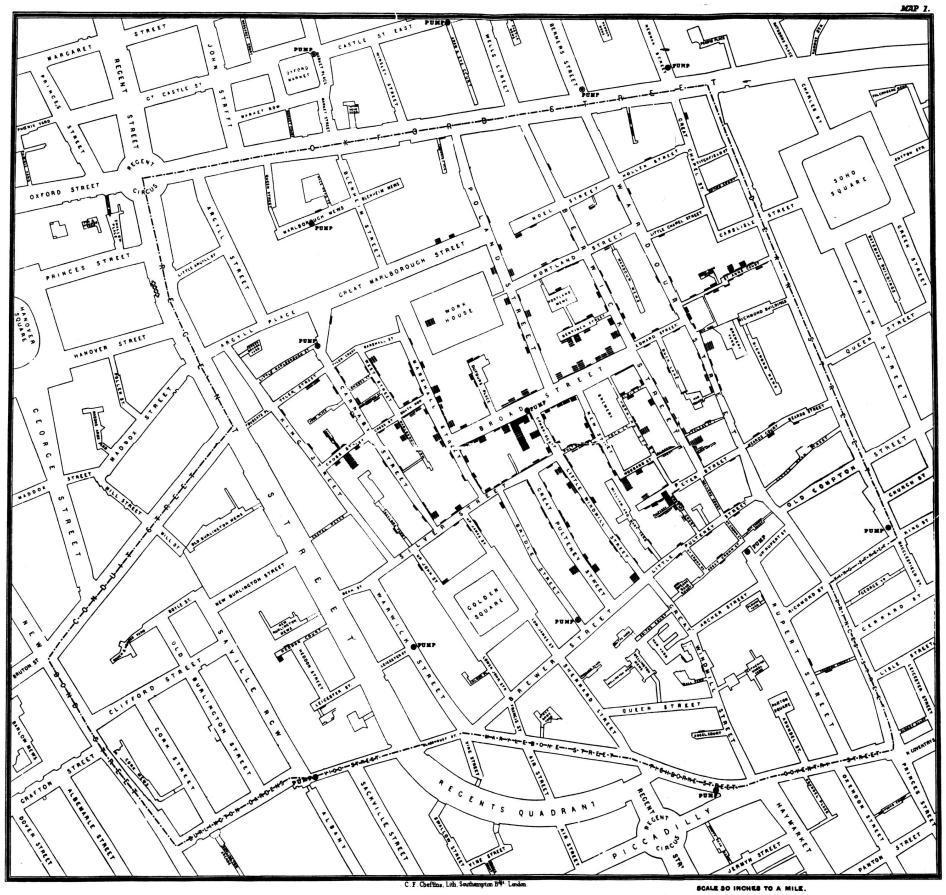


Figure 2.1: Ghost Map. Original map made by John Snow in 1854; cholera cases are highlighted in black.

Visual data mining (VDM) is not just the visualization of data; it usually refers to a whole process that involves the user to interpret the visualization of the data and adapt it interactively to discover interesting correlations or facts that are otherwise not visible and hard to recognize. Applications of visual data mining range from the search for specific information – also in retrieval settings to fulfill a specific need – to the browsing of huge amounts of data to find interesting aspects. The interaction with the data and the visualization plays an important role; the visualization system has to help the user to navigate through the data. This combination of visualization and interaction is solely possible by means of modern computers with high resolution graphical displays.

VDM approaches can be found in various systems and for various purposes. For instance, Fluit, Sabou, and Harmelen (2005) present three different applications that use the Cluster Map technology, an interactive visualization for overlapping clusters. DaCosta and Venturini (2006) use the concept of points of interest for the purpose of VDM on numeric or symbolic data. G. Smith et al. (2006) introduce FacetMap, an interactive visualization, primarily as a means to organize and retrieve data from heterogeneous sources.

Shneiderman (1996, p. 337) introduces a mantra for this kind of VDM: "Overview first, zoom and filter, then details-on-demand."

Overall, he identified seven high-level tasks for VDM:

- Overview: Gain an overview of the entire collection.
- Zoom: Zoom in on items of interest.
- Filter: filter out uninteresting items.
- Details-on-demand: Select an item or group and get details when needed.
- Relate: View relationships among items.
- History: Keep a history of actions to support undo, replay, and progressive refinement.
- Extract: Allow extraction of sub-collections and of the query parameters.

The ICE-Map Visualization is a VDM approach that follows the mantra of Shneiderman and is specifically designed to support the maintenance and use of concept hierarchies. It focuses on the usage of a concept by means of annotations, according to the following definition:

Definition 2.1 (Annotation) *An annotation is the assignment of a concept to a document (Definitions 1.1 and 1.3) for information retrieval purposes, i.e. the result of an indexing process. For each annotation, a weight may be assigned during the indexing process.*

Particularly, the ICE-Map Visualization supports at least the following applications:

Comparison of indexing techniques. Different usage frequencies of concepts based on different indexing techniques are visualized. This way, a set of annotations can directly be compared to a reference set of – usually intellectually created – annotations. This measure shows for example deviations between manually and automatically assigned concepts and therefore directly points to potential problems in the automatic indexing process.

Evaluation of indexing results. By employing a heuristic that calculates an expected frequency, an indexing system can be monitored without the need of a reference set.

Evaluation of document sets and/or a KOS. The same approach can be used to visualize the distribution of assigned concepts in a document set over the KOS. This can be used to gain an overview on the focus of a document base, as well as to gain an understanding of the underlying KOS.

Comparison of document sets. Two different indexed document sets can be compared, e.g., to visualize the different foci of two libraries.

Visualization of document distributions over a KOS. This is an example for an application that is not demonstrated in this thesis, as it is not related to the maintenance of KOSs: Any subsets of indexed documents can be visualized by this technique, e.g., the result of a search query in an information retrieval system.

2.2 From Applications to Weight Functions

All applications are based on the comparison of two values that are assigned to a concept, be they based on the usage frequency of a concept with respect to two different indexing techniques or the usage frequency of a concept compared to some heuristic. The statistical framework underlying the ICE-Map Visualization uses the general notion of weight functions $w(c)$ that lead to different values for each concept in the hierarchy. Table 2.1 introduces the notation based on the definitions so far that is used in the remainder of this chapter. In this thesis, we use the following three simple weight functions:

1. A weight based on the usage frequency of the concept, i.e., how often a concept was assigned to a document in a given document set:

$$w_f(c) = |\mathcal{A}(c)| \tag{2.1}$$

Symbol	Explanation
c	A concept according to Definition 1.3.
$\mathcal{C}(c)$	The direct child concepts (narrower concepts) of c.
$\mathcal{C}^+(c)$	All recursive child concepts (narrower concepts) of c.
$\mathcal{P}(c)$	The direct parent concepts (broader concepts) of c. That can be more than one in the case of a polyhierarchy.
$\mathcal{S}(c)$	The sibling concepts of c. In case of multiple parents, the corresponding parent has to be denoted, but we skip this here for simplicity.
$\mathcal{A}(c)$	The set of annotations (Definition 2.1) related to concept c.
$\gamma(a)$	The weight of a single annotation a.
H	A concept hierarchy according to Definition 1.4. H is a partially ordered set of concepts based on the broader/narrower relationship and forms a (polyhierarchic) tree.
$\tau(H)$	The root concept of H, i.e. the only concept c in H for which holds that $\mathcal{P}(c) = \emptyset$. Note that we require H to have a single root concept. Otherwise, we introduce an artificial single root concept that becomes the parent of all former root concepts.
$\tau(c)$	The root concept of the concept hierarchy H where c belongs to.
Lower case denotes single elements, while upper case denotes sets. Accordingly, functions returning single elements are written lower case, functions returning sets are written upper case.	

Table 2.1: Mathematical notation.

2. A weight based on the *weighted* frequency of the concept. For instance, we can calculate the weighted frequency based on weights, ranks or confidence values that are produced by automatic indexing systems:

$$w_w(c) = \sum_{a \in \mathcal{A}(c)} \gamma(a) \tag{2.2}$$

3. A weight based on the *expected* frequency of the concept. In this thesis, we use a simple heuristic that calculates the expected frequency based on the number of child concepts:[2]

$$w_e(c) = |\mathcal{C}(c)| \tag{2.3}$$

[2]This is not intuitive at first sight, however, the statistical framework in which the weight functions are employed, has a recursive component that turns this simple weight function into a heuristic that follows the idea that the usage of concepts is evenly distributed. An almost equivalent weight function is $w'_e(c) = 1$, which simply increases the values of $w^+(c)$ by 1 for all c (Equation 2.4) with no significant difference for the resulting evaluation. With Equation 2.3, the resulting calculation of the information content exactly corresponds to the notion of the Intrinsic Information Content, as we explicate after the introduction of Equation 2.7.

2.3 Statistical Framework

As stated above, the usage of a concept c is determined by a weight function $w(c) \in \mathbb{R}_0^+$ that assigns a non-negative, real weight to it. Based on this weight function, we further define:

$$w^+(c) = w(c) + \sum_{c' \in \mathcal{C}(c)} w^+(c') \tag{2.4}$$

$w^+(c)$ is a monotonic function on the partial order of the concept hierarchy H, i.e. the value never increases while walking down the hierarchy. This gives the value of the root node a special role as the maximum value of w^+, which we denote as $\hat{w}^+$:

$$\hat{w}^+(c) = w^+(\tau(c)) = \max_H w^+(c) \tag{2.5}$$

The next step if directly motivated by information theory. If we use the number of annotations (i.e., Equation 2.1) made for a given concept as the weight function $w(c)$, we can calculate the likelihood that a concept is assigned to a random document as follows:

$$L(c) = \frac{w^+(c) + 1}{\hat{w}^+(c) + 1} \qquad L(c) \in (0, 1] \tag{2.6}$$

The addition of 1 is necessary to allow a value of 0 for $w(c)$. Otherwise, the logarithm of $L(c)$ (cf. Equation 2.7) would not be defined for $w(c) = 0$.

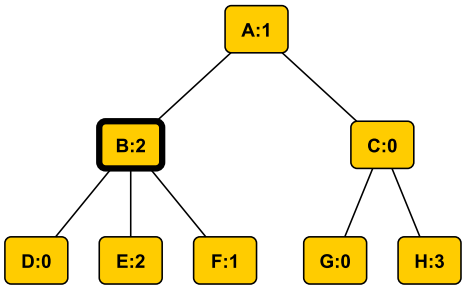
In information theory, the Information Content or Self-information of an event x is defined as $-\log L(x)$, i.e., the information content of an event is the higher, the more unlikely the event is. Together with a normalizing factor, we get the following definition for the Information Content $IC(c) \in [0, 1]$ of a concept c:

$$IC(c) = \frac{-\log L(c)}{\log(\hat{w}^+(c) + 1)} \qquad \hat{w}^+(c) \neq 0 \tag{2.7}$$

This is again a monotonic function on the partial order of H and assigns 0 to the root concept and 1 to concepts with $w(c) = 0$.

Notably, Equation 2.7 is a generalization of different implementations that are also based on the information content: If we use the number of annotations for a given concept ($w_f(c)$, Equation 2.1), Equation 2.7 corresponds to the information content of a concept, as introduced by Resnik (1995).[3] Therefore, by employing

[3] Beside the normalization and the addition of 1 to deal with zero values.



Example: The numbers denote the number of annotations for each concept. Weights for concept B:

	w	w^+	$\hat{w}^+$	L	IC
w_f	2	5	9	0.6	0.22
w_e	3	3	7	0.5	0.33

Result: $D(B) = -0.11$

Figure 2.2: Calculation of the IC Difference.

the weighted frequency of annotations ($w_w(c)$, Equation 2.2), we generalize the information content of Resnik for weighted annotations, as often provided by automatic indexing systems. Finally, the use of the heuristic ($w_e(c)$, Equation 2.3) leads to the Intrinsic Information Content (IIC), i.e. an information content that is determined only by means of the KOS structure itself, as introduced by Seco, Veale, and Hayes (2004).

The ICE-Map Visualization always compares two data sets based on the difference of the information content. Therefore we originally referred to it as IC Difference Analysis, but that way the underlying statistics could be confused with the overall VDM approach. Nevertheless, the basis of the ICE-Map Visualization is the difference of two information content calculations $IC_1(c)$ and $IC_2(c)$ by means of two different weight functions or a weight function applied to two different data sets, e.g. two sets of annotations from two different indexing processes. Accordingly, we define the IC Difference $D(c) \in [-1, 1]$:

$$D(c) = IC_1(c) - IC_2(c) \tag{2.8}$$

Figure 2.2 illustrates the calculation of the IC Difference. The result for concept B is -0.11, which means that concept B has a *lower* information content than expected based on the heuristic, i.e., there is a relatively *high* number of annotations for the subtree of concept B, compared to the rest of the concept hierarchy. This is an example, where the heuristic weight function is used. The power of the ICE-Map Visualization lies in the possibility to choose arbitrary weight functions for $IC_1(c)$ and $IC_2(c)$. All applications mentioned above are supported by different combinations of the weight functions introduced in Section 2.2.

2.4 Visualization

The statistical framework is only one half of the ICE-Map Visualization. While it can be used independently of the visualization to calculate the IC Difference for

one concept, the main purpose is to provide the user with the big picture of a full analysis of a concept hierarchy.

A major challenge in supporting KOS maintenance is to provide adequate tool support that guides the user to potential problems in a KOS based on the measures described above. In particular we have to find a way to provide the user with a view on the concept hierarchy that encodes the overall structure of the KOS or selected parts of it and the evaluation results for the different concepts in the KOS.

The ICE-Map Visualization uses a treemap to visualize the concept hierarchy together with the results of the analysis. The treemap visualization was developed by Shneiderman (1992) in the early 1990s, originally with the purpose to get an overview of disc usage of a particular hard drive. Shneiderman needed a compact representation of its directory structure, showing additional information like file size and file type in one view. According to Shneiderman, treemaps are a representation designed for human visualization of complex traditional tree structures: arbitrary trees are shown with a 2-d space-filling representation.

Treemaps belong to the implicit graph visualizations, as the graph structure is only implicitly reflected by the visualization of the nodes as nested rectangles. Treemaps belong together with icicle plots (Kruskal & Landwehr, 1983) to the most prevalent implicit visualization techniques (Schulz, Hadlak, & Schumann, 2011). Alternatives include radial visualizations like polar treemaps (B. S. Johnson, 1993), or sunburst (Stasko & Zhang, 2000). All visualizations can be extended into three dimensions. We prefer a 2D visualization, which have several advantages, including that "they are suitable for static media (e.g., printouts) [...] and they perform better than 3D techniques for comparison tasks on node attributes, as areas can perceptually be better compared than volumes" (Schulz et al., 2011).

We decided for the treemap particularly because of its very good space-filling property (McGuffin & Robert, 2010) and the ability to visualize two features of a concept at the same time by means of color and size of a rectangle. Drawbacks of treemaps are user disorientation, especially, if they are not familiar with treemaps (Turo & Johnson, 1992). Particularly the structure of the hierarchy is not easy to figure out (Bruls, Huizing, & Wijk, 2000). For instance, Barlow and Neville (2001) compared several graph visualization techniques including treemaps and found that the "treemap was uniformly disliked by the participants and their performance while using it was worse than with the other three views." This conclusion is disputable (e.g., the authors first tested the perception of the hierarchy and then excluded the treemap from further experiments due to the bad performance), however, it makes clear that a treemap visualization needs a proper support to help the user in the understanding of the hierarchy. An example for a treemap extension in that direction is presented by Zhao, McGuffin, and Chignell (2005): They combine the treemap visualization with node-link diagrams. We use a similar technique with a much simpler and in our opinion more intuitive implementation: we combine the treemap visualization with a traditional, explorer-like treeview that

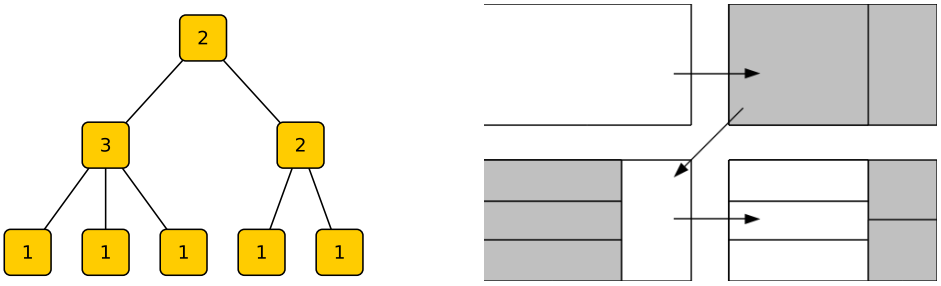


Figure 2.3: Original "slice-and-dice" layout.

is synchronized with the treemap and always shows the user the position in the hierarchy that is currently selected in the treemap.

Treemaps

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). Examples for such a size metric are the number of direct child nodes for each node or the size of the subtree represented by that node.[4] For each child c_i of the root node r, a partition of the space along the x-axis is calculated. For the first partition, this reads as

$$x_3 = x_1 + \left(\frac{|c_1|}{|r|} \right) (x_2 - x_1) \tag{2.9}$$

with $|c_1|$ as the size of child node 1 and $|r|$ 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 (Figure 2.3). Shneiderman called this approach the "slice-and-dice" algorithm. Since then, a lot of different implementations and optimizations have been presented, e.g. by Shneiderman and Wattenberg (2001) or Bederson, Shneiderman, and Wattenberg (2002).

Squarified Layout

The ICE-Map Visualization uses the squarified layout, as presented by Bruls et al. (2000). In this section, we describe in detail our reference implementation.

As a starting point, we want to layout the children $c \in C$ of a given parent concept p. Therefore, we want to determine the dimensions (width, height) and

[4] Or any other result of an analysis performed on the concept hierarchy. We recommend to use a metric that is based on the hierarchical structure and stays stable for different analyses. The result of the IC difference analysis is visualized using the color of the rectangles.

the position (x,y) of the rectangle that is occupied by each concept. We denote them with c_w, c_h, c_x, and c_y, respectively. Note that we regard c and p as compound objects, containing the dimension and position information denoted by the subscript.

We know the dimensions of the area that can be used to layout the child concepts, p_w and p_h. For each concept c, we can calculate its area c_a based on a weight function $w(c) \neq 0$ as a fraction of the area of the parent concept p:

$$c_a = p_w \cdot p_h \cdot \frac{w(c)}{w(c) + \sum_{i \in \mathcal{S}(c)} w(i)} \tag{2.10}$$

The general idea of the squarified layout is to split the children into several rows that are laid out one after the other. Each row is placed in the lower, left corner of the remaining area and the rectangles of the concepts are assembled horizontally, if the remaining area is higher than wide, and vertically, if the remaining area is wider than high. In the first case, a row uses the full width of the remaining area, in the latter the full height.

We can calculate the width and height of every concept c in a row R, as well as its position, i.e. the coordinates of its lower left corner, based on the rectangle s of the remaining free area. First, we introduce the calculation of c_w and c_h under the assumption that the row is laid out horizontally with a given $width$ and with relative positioning, i.e. the lower left corner of the row is $(0, 0)$.

CALCULATE-ROW$(R, width)$

```
1  area = Σ_{c∈R} c_a                    // cf. Equation 2.10.
2  height = area/width
3  x = 0
4  for c ∈ R
5      c_w = width · (c_a/area)
6      c_h = height
7      c_x = x
8      x = x + c_w
9      c_y = 0
10 return height
```

Note that CALCULATE-ROW returns the $height$ of the calculated row. This is used in the following procedure, where a row R is actually placed within a free area s. PLACE-ROW adheres to the above mentioned strategy and returns the remaining free area after the placement of the new row.

PLACE-ROW(R, s)
1 $width = \min(s_w, s_h)$
2 $height =$ CALCULATE-ROW$(R, width)$
3 **if** $s_w > s_h$ // Distinction between horizontal and
4 ROTATE-ROW(R) // vertical layout, see text above.
5 $s'_w = s_w - height$
6 $s'_h = width$
7 $s'_x = s_x + height$
8 $s'_y = s_y$
9 **else** $s'_w = s_w$
10 $s'_h = s_h - height$
11 $s'_x = s_x$
12 $s'_y = s_y + height$
13 SHIFT-ROW(R, s)
14 **return** s'

The rotation[5] – if the row has to be layed out vertically – and shift of the row are implemented as follows, using simple vector arithmetic:

ROTATE-ROW(R)
1 **for** $c \in R$
2 SWAP(c_w, c_h)
3 SWAP(c_x, c_y)
4 **return**

SHIFT-ROW(R, s)
1 **for** $c \in R$
2 $c_x = c_x + s_x$
3 $c_y = c_y + s_y$
4 **return**

The remaining question is: How should the children be distributed to the single rows? The heuristic used in this case is as follows: Sort the children by their size in descending order and then start adding them to a row R. Then calculate the "badness" of the row based on the worst aspect ratio of the concepts in the row:

$$\text{badness}(R) = \begin{cases} \max_{c \in R} |c_w / c_h - 1| & R \neq \emptyset \\ \infty & R = \emptyset \end{cases} \tag{2.11}$$

If the addition of a concept would increase the badness, do not add it and instead start a new row. This leads to the following procedure for a parent concept p and its children C:

[5] The rotation is that simple because the position of the concept is only relative at the time of the invocation, i.e. it is a rotation around $(0, 0)$.

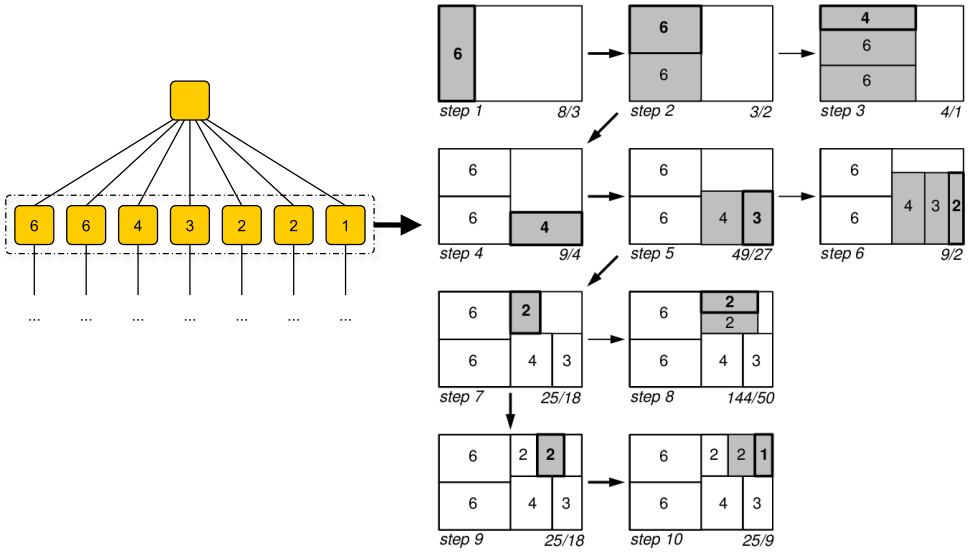


Figure 2.4: Squarified layout (Source: Bruls, Huizing, and Wijk, 2000).

LAYOUT-CHILDREN(C, p)

```
1  Sort C by c_a decreasing          // cf. Equation 2.10.
2  R = T = ∅                         // Initialize (temporary) rows.
3  s = COPY(p)                       // Start with the area of p.
4  for c ∈ C
5      ADD(T, c)
6      CALCULATE-ROW(T, min(s_w, s_h))
7      if badness(T) > badness(R)    // Check, if badness is increased.
8          s = PLACE-ROW(R, s)       // Place row and
9          R = T = ∅                 // start a new one.
10         ADD(T, c)                 // Prepare T for next row.
11     ADD(R, c)                     // Extend the row and continue.
12 if R ≠ ∅
13     PLACE-ROW(R, s)               // Place remaining concepts.
14 return
```

Figure 2.4 (Bruls et al., 2000) illustrates the algorithm for one concept with child concepts having the weights $(6, 6, 4, 3, 2, 2, 1)$.

With LAYOUT-CHILDREN, we can now recursively layout the whole treemap. The drawing of the treemap gives us two degrees of freedom that can be used to visualize information beside the hierarchical structure. One is represented by the size of the concepts, the other by its color.

Further Aspects of the Implementation

We experimented with different combinations of metrics to determine the size and color weights of a concept. It turned out that the size should usually not be used to visualize aspects other than the hierarchy, because otherwise we would not get a stable visualization of the hierarchy that does not change its layout if another analysis on the concept usage is performed.

The most convenient weight function for the size is based on the number of children of a concept, either only the direct children (Equation 2.3) or with all subchildren (Equation 2.4 with Equation 2.3 as internal weight function). Usually the latter is to be preferred, as this way the space is evenly distributed between all the concepts of the hierarchy and thus uses the space optimally to view as much concepts as possible. In any case, some positive value has to be added to the weight function to prevent zero values for concepts without children.

The color is determined by the result of the analysis that is performed on the concept hierarchy. The ICE-Map Visualization uses the IC difference with arbitrary weight functions. In the default setup, the weight between -1 and 1 is mapped to a color range from red (-1) over white (0) to blue (1). The lower the information content of a concept is, the higher is the underlying weight function. This way, the treemap can be interpreted as a temperature map, with red areas indicating "hot" areas regarding the usage (or whatever is used as weight function) and blue ones "cold" areas, compared to the chosen reference.[6]

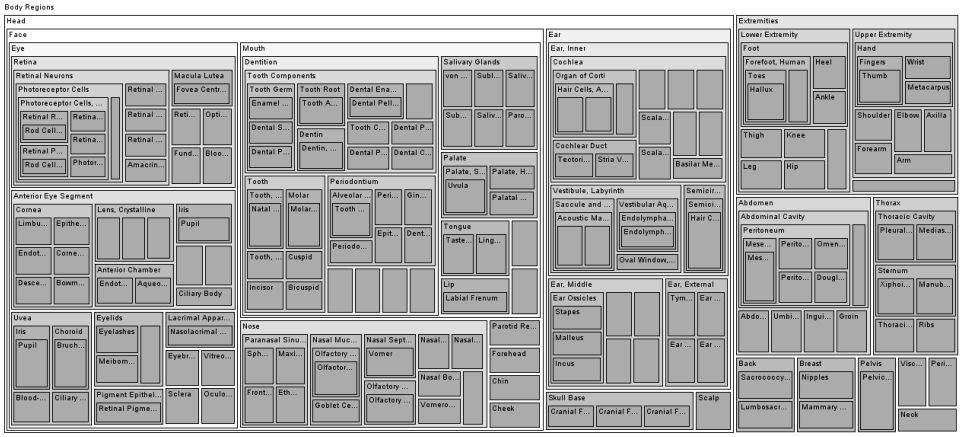


Figure 2.5: Treemap of the MeSH concept BODY REGIONS.

Figure 2.5 shows the treemap of a part of the MeSH thesaurus (Body Regions), where each area represents a concept in the thesaurus. As Bruls et al. (2000) point

[6]The colors are inverse to the colors used in (Eckert, Stuckenschmidt, & Pfeffer, 2007, 2008), as we found that users interpret the map as a kind of temperature map with red indicating higher usage of a concept, not a higher information content due to less usage.

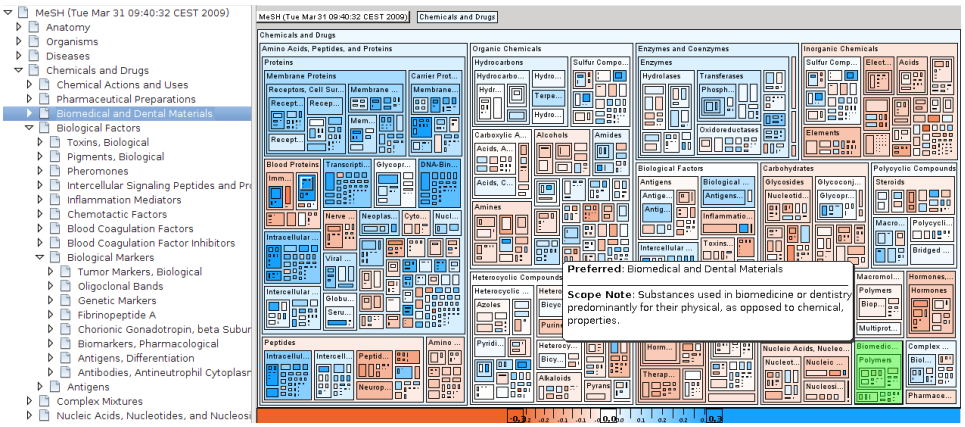


Figure 2.6: Reference implementation of the ICE-Map Visualization.

out, a drawback of the treemap visualization in general and especially the squarified layout is that it is not easy to recognize the underlying hierarchy. They propose the use of a profiled border, in combination with a cushion visualization (Wijk & Wetering, 1999). Our reference implementation of the ICE-Map Visualization uses nested areas with line borders and a written title on top of each concept – provided there is enough space; otherwise, the title is omitted. In our experiments, we found this very convenient and usually there is no problem to see and understand the nested structure of the underlying hierarchy.

Nevertheless, the treemap visualization requires some time for the user to get familiar with. Thus, the reference implementation introduces further means to improve the usability, following the above mentioned mantra: "Overview first, zoom and filter, then details-on-demand."

First of all, the treemap visualization itself is highly interactive. By double-clicking on a concept in the treemap the user can zoom into the hierarchy. A double-click on the top concept zooms out again. A major drawback of treemaps is the possibility for the user to lose the orientation in the hierarchy as the visualization cannot provide information about the environment of the currently selected top concept, when zooming in.

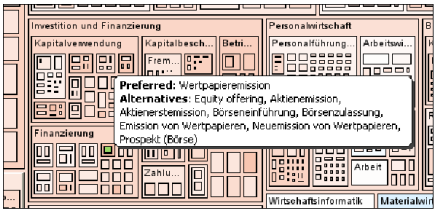


Figure 2.7: Concept selection.

We deal with this problem in two ways (Figure 2.6): First, we provide a root-line above the treemap visualization that shows the path from the top of the hierarchy to the currently shown concept. The concepts in the root-line are colored accordingly. A click on a concept in the root-line directly zooms out to the con-

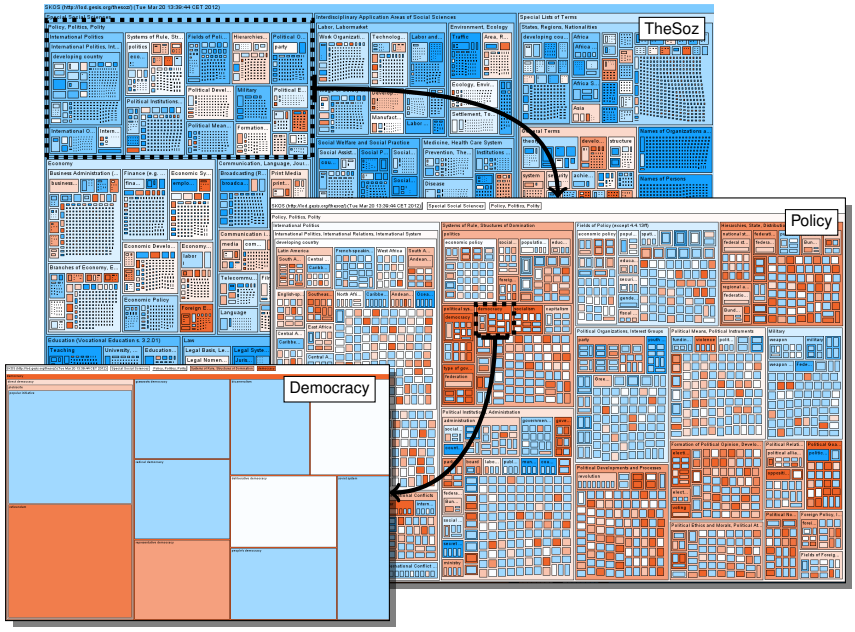


Figure 2.8: Zooming into the hierarchy.

cept. Second, the treemap is combined with a hierarchical common treeview. This allows interactive navigation through the hierarchy without losing the orientation. The selection of a concept in the treeview leads to a selection of the concept in the treemap and vice versa. For a selected concept, additional information about the concept is provided in a pop-up box (Figure 2.7).

The colors of the visualization can be adjusted by the slider below the treemap. This way, the contrast can be improved by narrowing the color range to smaller values of the analysis result. Additionally, the balance between red and blue can be adjusted. The latter can be used to set the color of the top concept to white and thus visualize the subconcepts as if the current top concept would be the root of the hierarchy.

To illustrate the zooming, Figure 2.8 shows the ICE-Map Visualization of the TheSoz. The colors can be neglected for now; they reflect the analysis results of one of the experiments conducted in the next chapter. On the first level, the whole KOS is visible. To be able to visualize the complete KOS is clearly a strength of the treemap visualization. Details in deeper levels of the hierarchy are not visible in the structure. The analysis result, however, “radiates” to the parent concepts and makes the visualization usable on the first level. For detailed information about the single concepts, the user can zoom into the hierarchy by simply double-clicking a concept, in this case into the subtree of the concept POLICY.

Above the visualization, the root path is visible that prevents the user from losing overview of the relations between the individual concepts. The deeper the

user browses, the more specific the concepts. The example finally zooms into the concept DEMOCRACY, where all 10 subconcepts are visible. The user has reached the bottom of the hierarchy.

2.5 Related Work

To the best of our knowledge, no one ever used such a combination of statistical analysis and the treemap visualization to perform visual datamining on concept hierarchies. There are, however, several aspects of our work where related approaches exist.

The treemap visualization itself is widely used, especially to visualize large hierarchical datasets. For example, M. Smith and Fiore (2001) employed it to visualize Usenet newsgroups.

Calmet and Daemi (2004a, 2004b) evaluate ontologies using the Kullback-Leibler Divergence, which is widely used in information theory and defined as follows:

$$D_{KL}(p||q) = \sum_i p(i) \log \frac{p(i)}{q(i)} \tag{2.12}$$

This is a measure of the differences between two probability distributions p and q and as such related to Equation 2.8. The authors use the Kullback-Leibler Divergence to get an overall measure of the thesaurus suitability, instead of evaluating a single concept.

Rayson and Garside (2000) use a log-likelihood approach to compare text corpora and show that it can be used to determine key terms in a corpus which distinguishes it from the reference corpus.

2.6 Conclusion

In this chapter, we presented the ICE-Map Visualization, one of the main contributions of this thesis. The ICE-Map Visualization consists of two parts: the underlying statistical framework based on information theory and a proper visualization that creates the big picture of the KOS for the user. The power of the statistical framework lies in its ability to make use of almost arbitrary weight functions. The weight functions can be seen as adaptors that turn characteristics of data into visualizable numbers. For instance, we can define a weight function like $w_l(c) = \sum_{a \in \mathcal{A}(c)} |\mathcal{L}(a)|$ with $\mathcal{L}(a)$ being the set of loan transactions (or electronic accesses) for the document that is associated to concept c by annotation a. Using this weight function visualizes the actual topical interests of the users of a

library. By replacing $|\mathcal{L}(a)|$ with the aquisition costs of the document, a topical map is created that shows the distribution of money over the topics reflected by the KOS. The possibilities are endless.

We proposed the treemap as suitable visualization in this chapter. Other visualizations would work as well, as long as they have the ability to visualize the analysis result properly – which typically involves colors. For interactive applications, a 3D visualization could be promising, despite of the above mentioned drawbacks. In our experiments with other visualizations like hyperbolic trees, polar treemaps and icicle plots, we did not find a convincing alternative to the treemap. We did not perform a comprehensive user study, but a rationale for the choice of the treemap is that it actually draws a map of the hierarchy. Due to the recursive definition of $w^+(c)$, analysis values "radiate" to the parent concepts which has a smoothing effect. This further emphasizes the analogy to a map with different regions. This analogy is lost immediately with explicit graph visualizations like hyperbolic trees and arguably less convincing for 3D or radial visualizations. In the next chapter, we will demonstrate the application of the ICE-Map Visualization for various use cases. The reference implementation of the ICE-Map Visualization is integrated in Semtinel, which is further described in detail in Chapter 5.

Acknowledgements: Parts of this chapter have been published before. The ICE-Map Visualization was first presented at the Fourth International Conference on Knowledge Capture (K-CAP 2007) as IC Difference Analysis where it was granted the best paper award (Eckert et al., 2007). In (Eckert et al., 2008), we published an extended description of the methodology on the use case of the evaluation of automatic indexing results. In (Pfeffer, Eckert, & Stuckenschmidt, 2008), the methodology was adapted to classification systems and automatic classification. In (Eckert, Hänger, & Niemann, 2009), we evaluated tagging results and compared them to intellectual indexing and automatic indexing using the ICE-Map Visualization. A draft of this chapter has been published as Technical Report TR-2011-003 of the Department of Computer Science, University of Mannheim (Eckert, 2011a).

The ICE-Map Visualization was solely developed by me. The pseudocode of the squarified algorithm is my own implementation which differs from the implementation of Bruls et al., albeit the algorithm is the same.

Chapter 3

Selection and Evaluation

Whenever a KOS is employed, its suitability for the desired application has to be evaluated. If possible, the reuse or modification of an existing KOS is preferable over the creation of a new one (cf. Section 1.2). In any case, a KOS needs maintenance throughout its lifetime. It is never perfect nor finished. This makes KOS evaluation crucial for every application.

In this chapter, we show how the ICE-Map Visualization supports the evaluation and selection of an existing KOS. With the weight functions introduced in Chapter 2, the ICE-Map Visualization always visualizes a KOS and the usage of its concepts for some kind of indexing or classification of documents. Depending on the interest of the user, there are different perspectives on the result:

1. With a good knowledge of the **indexing process**, the suitability of the KOS for these documents can be evaluated. We therefore introduce the notion of the topical overlap that is visualized by the ICE-Map Visualization. This use case is examined in Section 3.1.
2. With a good knowledge of the **documents and the KOS**, the visualization can be used to evaluate an indexing process. One option to improve an indexing process is the modification of the underlying KOS, i.e. its maintenance for the given application. This is demonstrated in Section 3.2.

3.1 KOS Selection based on Topical Overlap

The decision for a KOS cannot be determined by just one factor. Questions that have to be answered include the following:

- What is the intended goal that is pursued by the employment of a KOS? For example a classification can be used to organize books in a library. A thesaurus can be used to improve the quality in an information retrieval system.

But both KOSs can also be used to relate resources to others that are organized in the same way. Here, interoperability might be the overall goal and the KOS is primarily used as a common "language" between systems.

- Are there existing KOSs that are already used? Reindexing of resources with a new KOS is expensive. Not only the resources have to be reindexed, the editors and users have to make themselves familiar with the new KOS. If an existing KOS is to be replaced, what kind of problems have been identified?

- Can an existing KOS be used as a basis and get modified and adapted for the desired purpose? This reduces the effort of the KOS creation; when indexed documents already exist, it might not be necessary to reindex them – or at least the reindexing can be reduced to documents indexed with concepts that are actually changed.

- Are there reasons to use a special KOS? This will mostly be interoperability reasons, but interoperability can also be more subtle, like the choice of the Dewey Decimal Classification, because it is expected that the users are familiar with it and would profit from its use.

- Which KOS is suitable for the given resources? Interoperability can also be achieved by translating one KOS to another. The KOS has to be able to describe the documents appropriately in the first place.

For the last question, the ICE-Map Visualization provides support. The only prerequisite is the existence of annotations, i.e., the documents have to be indexed first. Intellectual indexing just for the selection of a KOS is obviously not feasible. We propose to use a simple automatic indexer to provide the necessary annotations. We then use the ICE-Map Visualization to visualize the *topical overlap* of the documents and the KOS in question. It is reasonable that the topical overlap is crucial for the suitability of a KOS. A KOS that contains a significant number of irrelevant concepts with respect to the documents is not optimal. Especially if topics of the documents are missing in the KOS, it is not possible to describe them adequately.

The presented application leads to special requirements for the automatic indexer: no training step must be involved, as the creation of a training set is not feasible as well. And the indexing result hat to be comprehensible, as the user must be able to judge the visualization. Following these requirements, a simple indexing system called LOHAI has been developed that is employed for our experiments. LOHAI is described in detail in Chapter 5.

With LOHAI and the ICE-Map Visualization, we have everything that we need to calculate and visualize the topical overlap of a KOS and a document set or to compare two document sets based on a KOS. For this approach, we use the weight function that takes the *tf-idf* weighting into account (Equation 2.2):

$$w_w(c) = \sum_{a \in \mathcal{A}(c)} \gamma(a)$$

with $\gamma(a)$ denoting the weight of a single annotation a as calculated by LOHAI[1] and $\mathcal{A}(c)$ being the set of annotations that are assigned to a concept c.

3.1.1 Experimental Setup

To demonstrate the usefulness of the ICE-Map Visualization in the context of this chapter, a convincing data set is needed. We do not only need two KOSs that have a significant overlap – it would not be convincing, if we show that we are able to tell apart the medical thesaurus MeSH and the Getty Art & Architecture Thesaurus – we also need at least one document set for each KOS where we can assume that it fits to the KOS. Furthermore, we prefer to use well-established KOSs that are freely available. They need to have a significant size and at least one language in common matching the language of the document sets – we do not want to introduce the complexity of multilingual indexing here. We need an overlap, but both KOSs need a different overall topic that allows us to see if the topical focus of the documents is properly reflected in the visualization. With the *STW* and the *TheSoz*, we fortunately have two KOSs that fulfill all these requirements (see Section 1.5 for detailed descriptions). For the following experiments, we use the version 8.08 of the STW and the TheSoz RDF implementation (Zapilko & Sure, 2009) in version 0.86. For both, we identified matching document sets, maintained by the same institutions that provide the thesauri:

Document Set 1: SSOAR. The Social Science Open Access Repository[2] is an open-access server maintained by the GESIS, which provides full texts like articles or theses together with the according metadata. It focuses on documents of the social sciences and related disciplines. From SSOAR, we used all available 2,718 (as of December 1st 2011) documents with an English[3] abstract.

Document Set 2: EconStor. Like SSOAR, EconStor[4] is an open-access server maintained by the ZBW. Its focus is on economy and all related sciences. Similar to SSOAR, articles, theses, or working papers can be found. Altogether, EconStor

[1] Strictly speaking, from an information-theoretic perspective, this function interprets the *tf-idf* weight of the annotation as the likeliness of being an annotation for the document. This interpretation is not correct, as *tf-idf* is no probability value.

[2] http://www.ssoar.info/

[3] We decided to use only English abstracts and the English terms of the KOSs to keep the setup simple and reproducible for everyone.

[4] http://www.econstor.eu/

provides 23,866 (as of December 1st 2011) documents with an English abstract. The different sizes are no problem, as the ICE-Map Visualization is designed to be independent from the size of the analyzed document sets.

Social sciences and economics have large overlaps and are at the same time clearly distinguishable. Both maintaining institutions fulfill similar functions in their respective area in the German scholarly system; and they provide similar services. With these two KOSs and document sets, we have a data set as required.

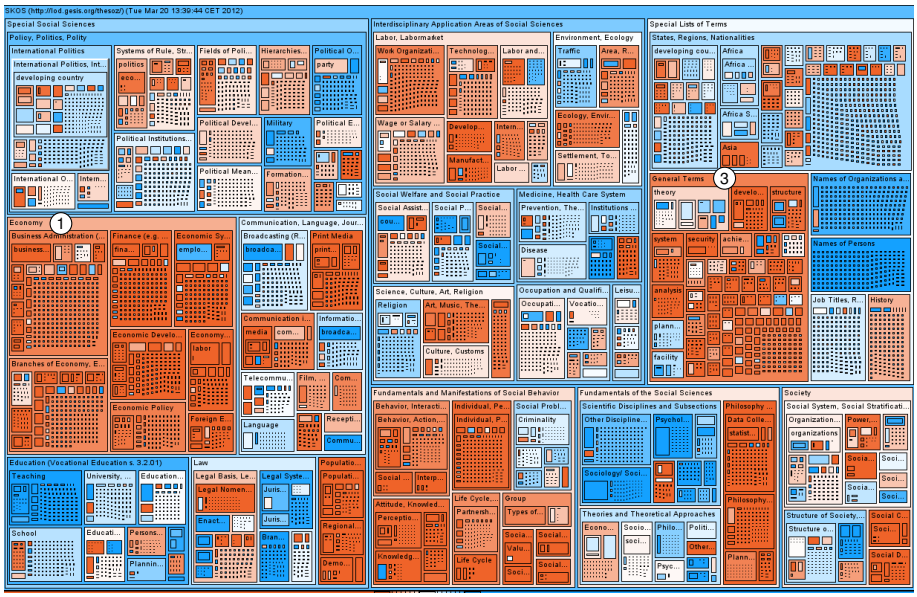
To prepare the actual experiments, we used LOHAI to index both document sets with both KOSs. The indexing results are represented as annotation sets. We denote them in the following by the name of the document set, followed by the name of the KOS for which indexing results are available in the annotation set. For instance, EconStor/TheSoz is the annotation set that contains annotations of EconStor documents with concepts from TheSoz.

For the following experiments, it is assumed that a document set (SSOAR or EconStor) is available and a suitable KOS is to be identified. Two candidates are available: TheSoz and STW. We know that TheSoz and SSOAR match, as well as STW and EconStor. Furthermore, we know that there is a substantial topical overlap between both document sets and KOSs respectively. The question is: will we be able to visualize this by means of LOHAI and the ICE-Map Visualization.

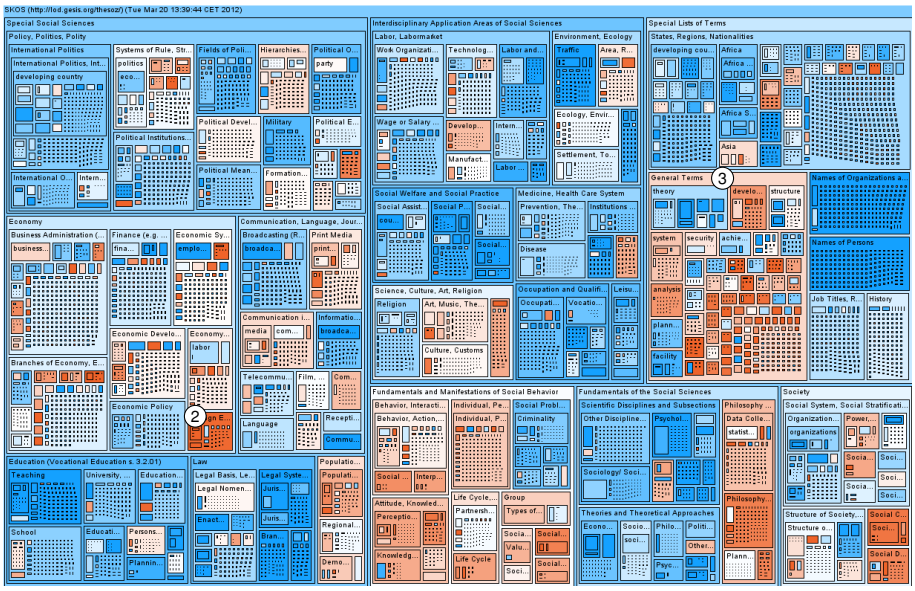
3.1.2 Results

Figure 3.1 shows EconStor/TheSoz and SSOAR/TheSoz. This is the social sciences thesaurus. We would like to see how SSOAR is distributed over the thesaurus as well as the economical bias of EconStor. This bias can clearly be seen, most terms which are used in the documents are narrower terms of ECONOMY ①. In contrast, the results of SSOAR/TheSoz do not point out such a clear focus on one specific field. In the Economy subtree are well used areas as well, an indicator that both sciences indeed have an overlap reflected in the document sets. The deep red intensity of the FOREIGN ECONOMIC RELATIONS concept ②, however, is partly the result of an indexing mistake; IMPORT is mistakenly assigned for the occurrence of "important." This is quickly figured out by examining the documents that are associated with the concept. A second concept in the same area, EXPORT, is assigned correctly. The weighting provided by LOHAI ensures that the effect of such errors on the visualization is minimized. In this case, each assigned concept EXPORT has a weight of about 0.6, compared to about 0.08 for the concept IMPORT." An expected result is the usage of the GENERAL TERMS section ③ that is relatively high in both cases.

Figure 3.2 shows the cross-check for STW. This time, the topics of the documents in EconStor spread almost the whole thesaurus, with relatively low usage only in some special areas. In contrast, SSOAR documents use terms which are



(a) EconStor/TheSoz



(b) SSOAR/TheSoz

Figure 3.1: Topical overlap of the TheSoz with EconStor and SSOAR.

narrower ones of RELATED SUBJECT AREAS and especially of POLITICAL SYSTEM ①, SOCIOLOGY ②, DEMOGRAPHICS ③, and HUMANITIES ④. The red areas in ECONOMICS reflect the topical bias of the documents, too: STATISTICAL METHODS ⑤ or LABOUR ⑥. Sections that are used well by both document sets are again general sections like GEOGRAPHIC NAMES ⑦ and GENERAL DESCRIPTORS ⑧.

The suitability of a KOS cannot only be determined by means of the pure coverage of concepts by the documents that have to be organized. If we know which parts of a KOS are actually used, we have to ask next, if the KOS is elaborated enough in this area. By zooming in, we can compare the STW concept ECONOMICS and the TheSoz concept ECONOMY (Figure 3.3).

Evidently and as expected, the STW is much more elaborated in this area than the TheSoz. We can zoom deeper into STW, but for TheSoz the bottom of the term hierarchy is already reached. Moreover, the TheSoz subsumes in this concept ECONOMY not only ECONOMICS, but also topics like BUSINESS ADMINISTRATION or FINANCE. Both topics have own subtrees in STW outside the here visualized concept ECONOMICS. Even without any knowledge about the topic of the thesaurus, we directly get to know that STW provides much more detail in this area. Again, we can learn more detailed about the document set. For example we see that the section ENVIRONMENTAL AND RESOURCE ECONOMICS ① is relatively rarely used. Therefore, EconStor might not be interesting for an institution that has a strong focus on environmental economics.

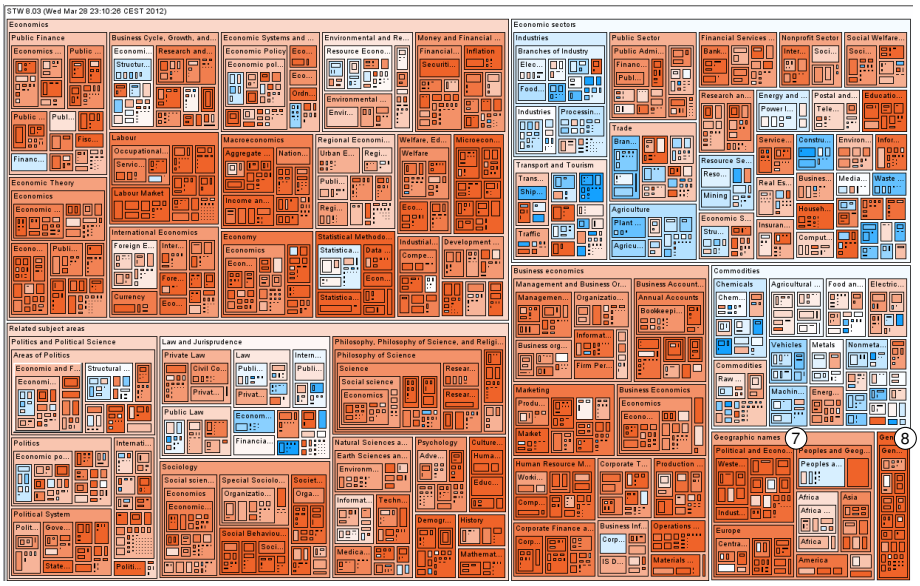
3.1.3 Focus on Documents

The topical overlap is not only important, if a new KOS has to be chosen. It can also be used to evaluate a document set based on a given KOS. A possible use case would be a library which has to decide if a new document set should be licensed. The library needs information, how the subjects of the documents are distributed regarding a given KOS. Provided that at least the abstracts are available for indexing,[5] our approach would work, as the setup is technically the same as for the KOS selection.

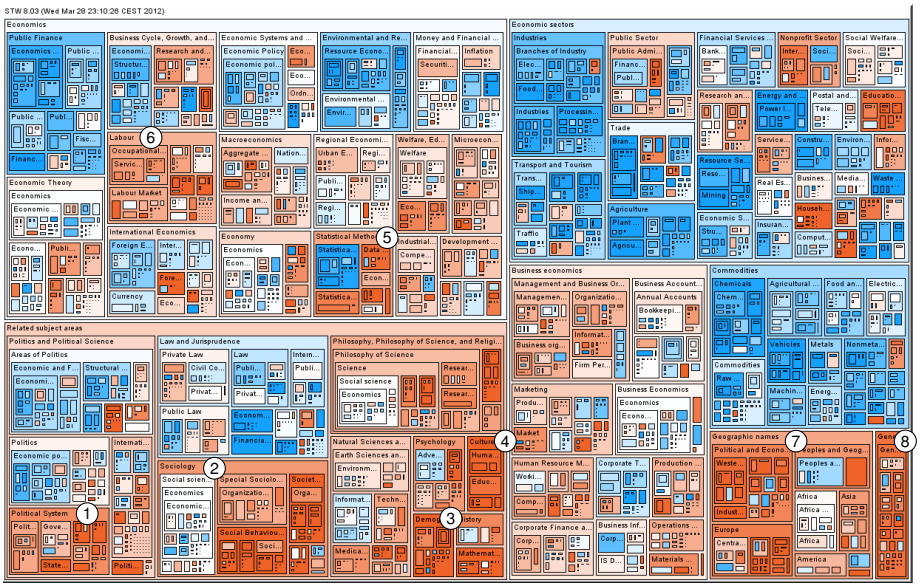
Another document-related use case would be a library which would like to compare the contents of a document set with another one to decide, whether the subjects of the documents match the current holdings. Similarly, it can be interesting to compare two libraries, e.g. to identify the individual focus of each library or as part of a SWOT analysis.[6]

[5] This will rarely be the case for commercial database providers, nevertheless libraries should start to ask for that in order to properly judge the contents.

[6] SWOT analysis is used in strategic business planning to identify and evaluate Strengths, Weaknesses, Opportunities, and Threats.

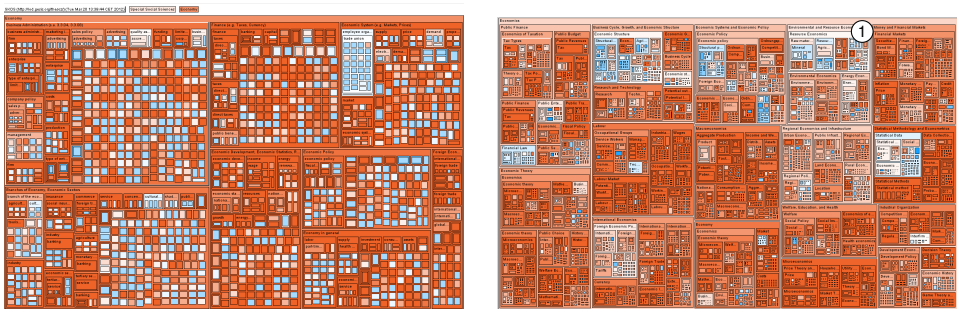


(a) EconStor/STW



(b) SSOAR/STW

Figure 3.2: Topical overlap of the STW with EconStor and SSOAR.



(a) EconStor/TheSoz (b) EconStor/STW

Figure 3.3: TheSoz and STW: Zoom on ECONOMY.

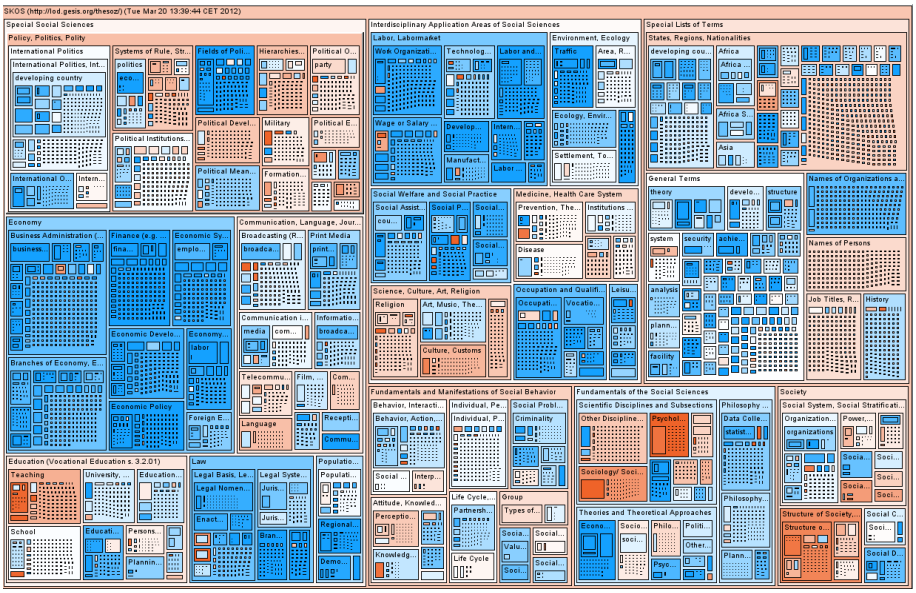
This use case employs two different sets of annotations to compare two document sets directly. We will expand on this use case, although it is not directly concerned with KOS maintenance. However, it is such an interesting application of our approach that we do not want to suppress it. Moreover, it completes the picture, as the direct comparison of two annotation sets is a key feature of the ICE-Map Visualization.

We use the ability of the ICE-Map Visualization to compare two annotation sets instead of using the heuristic weight function as reference. In this case, the interpretation of the results changes a bit. One set needs to be defined as the base set. Every comparison is then relative to this set. The coloring now indicates, if the concept is used more often in the analysis set (indicated in red) or in the base set (indicated in blue). Figure 3.4 shows the results of the direct comparison SSOAR against EconStor as base.

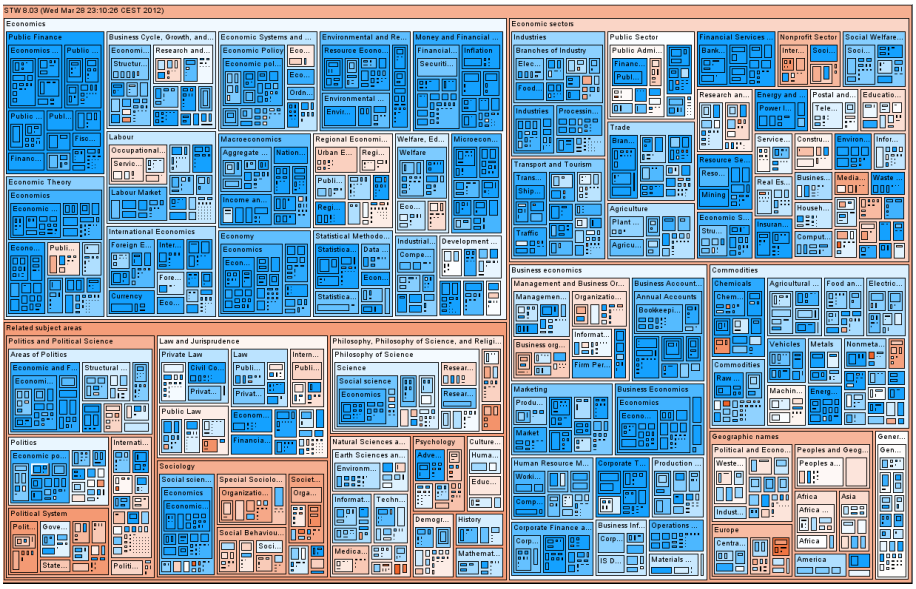
The results more or less speak for themselves. The difference between both document sets is clearly visible. A side-effect of the direct comparison is that possible indexing errors are faded out, as they typically occur independent of the document set. All in all, these experiments show the power of our approach. Without any further information, we evaluated two document sets and two KOSs and were able to develop a deeper understanding of them by just browsing through the ICE-Map Visualization.

3.2 KOS-based Indexing Evaluation

After a KOS is selected and employed in an application, it has to be evaluated constantly. Actually, the whole application has to be evaluated and monitored constantly. We will see that both cannot be told apart. Similarly as for KOS selection, the ICE-Map Visualization can be used to visualize the KOS usage in any application and help the maintainer to identify possible weaknesses and starting points



(a) TheSoz



(b) STW

Figure 3.4: Direct comparison of SSOAR (red) and EconStor (blue).

for improvements. On the basis of different indexing approaches, this is demonstrated in the following, as well as the challenges that arise from the employment of the ICE-Map Visualization. Generally, we want to keep the human expert in the loop of otherwise unsupervised alternative indexing systems and use the expert knowledge in the most efficient way to ensure the quality of the whole information retrieval system.

The benefits of using a KOS for document indexing comes at the price of the effort needed for annotating large document sets. Traditionally, this is done intellectually by specialists that read a document and decide which of the preferred terms in a KOS best describe its content. While books always have been and still are indexed that way, other information resources like single articles, contributions to conferences and web pages are not or at least not completely indexed. Nevertheless, these resources play a more and more important role. Since the overwhelming majority of information searches start with a web search engine (Rosa, 2006, cf. Section 1.3.4), libraries should follow and integrate these resources into their own search systems and thus make them available. Search facilities have to enable the user to access at least all resources that are available for the user by means of the library. Not only books, but also articles in subscribed journals, databases, and freely available open access journals. Today there is no coherent indexing information available for these resources; therefore alternative indexing approaches are needed. Two projects were conducted at the University of Mannheim to investigate alternative indexing approaches; in both projects the ICE-Map Visualization was used for the evaluation of the results:

1. *Automatic indexing* by means of a commercial search engine was investigated in the project "Verbesserung der Fachrecherche in großen Volltextsammlungen mit Methoden des Semantic Webs," funded by the German Research Foundation/Deutsche Forschungsgemeinschaft (DFG), 2007-2009.

2. *Tagging*, i.e. a crowdsourcing approach (cf. Section 4.1) that employs the user who can freely add tags to information resources, e.g. during the search in the search engine or as part of the later organization of the resources in a reference management system, was investigated in the project "Collaborative Tagging als neuer Service von Hochschulbibliotheken," funded by the DFG, 2008-2010.

3.2.1 Experimental Setup

Figure 3.5 presents the workflow in a typical KOS-based semantic search system. Such a system consists mainly of two parts: The indexing of new documents and the retrieval of relevant documents based on a user query and the indexed documents. The ICE-Map Visualization is used to evaluate the former which relies on

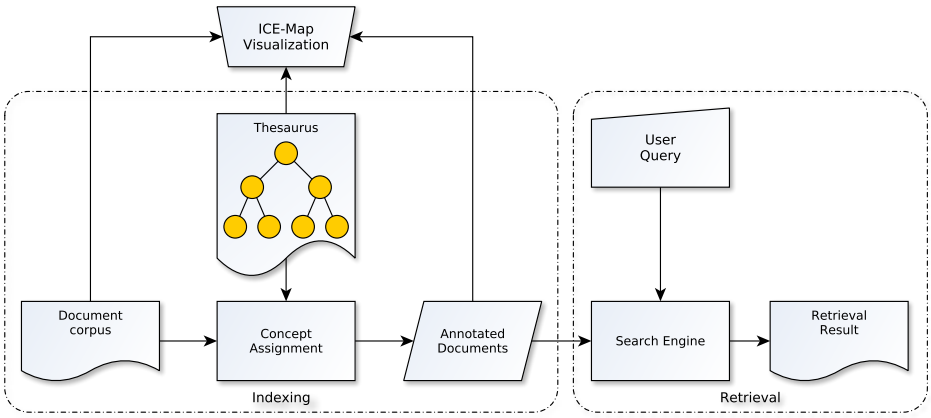


Figure 3.5: Schematic view of a supervised indexing and retrieval process.

three different components: the document set to be indexed, the KOS that provides concepts to index with, and the indexing method involved.

The ICE-Map Visualization uses all these components as input to allow the maintainer to evaluate the indexing process. In the following, we analyze the suitability of a KOS as a basis for alternative indexing and show that the ICE-Map Visualization naturally leads us to parts of the KOS that causes trouble in the indexing process. At the same time, the different characteristics of the employed methods can be seen.

The different characteristics of the indexing approaches are evaluated with the *STW* and a set of documents for which we have indexing results for all three approaches available. For intellectual indexing, we additionally demonstrate the ICE-Map Visualization for the *MeSH* (see Section 1.5 for detailed descriptions) with a set of Medline abstracts.

Document Set 1: Elsevier. The first set that is used with the STW consists of 371 articles included in three economic journals published by Elsevier, namely the *Journal of Financial Economics* (ISSN: 0304-405X), the *Journal of Accounting and Economics* (ISSN: 0165-4101), and the *Journal of Health Economics* (ISSN: 0167-6296). Every article in the dataset is described by the name(s) of the author(s), the title of the article as well as the abstract. It is intellectually indexed[7] by librarians at the ZBW. Furthermore, we have annotations for this set created by a layman that we use as basis for our tagging experiment. This document set is small, but it is the only set that we have that provides indexing results for all three indexing techniques. On the other hand, it demonstrates that our approach works

[7]Extracted from the Econis Database (http://www.econis.eu/).

with a small number of documents; the applicability for large document sets and the ability to deal with different sizes of document sets has been demonstrated in Section 3.1.

Document Set 2: Medline. For our experiments with MeSH, we use a document set containing 822 randomly selected Medline abstracts from 2008.[8] Only abstracts that are intellectually indexed with MeSH concepts by the NLM were used.

The information content of all concepts is calculated based on the annotations resulting from the different indexing approaches. The results of the intellectual indexing have a special role. We first investigate their own characteristics in the next section, but they are also used as a reference in the following sections. As the ICE-Map Visualization always compares two information content values, we can directly compare the characteristics of the different approaches. As before, we use the intrinsic information content as a heuristic to evaluate all approaches without the requirement of reference annotations. We use the weight function based on the number of annotations (Equation 2.1):

$$w_f(c) = |\mathcal{A}(c)|$$

Note that for the automatic indexing system, we could use the weights provided by the indexer, as in Section 3.1. In this section, however, we want to examine the characteristics of the indexing approaches and the characteristic is determined by the decision of a system to assign a specific concept. A weighting of the assignment makes the recognition of such a characteristic more difficult – hence its employment for the KOS selection where we wanted to reduce the impact of the indexing process as much as possible. Moreover, we want to keep all approaches comparable; neither the librarians nor the tagging student have the possibility to assign a weight to their assignments.

3.2.2 Intellectual Indexing

Currently, thesauri are primarily designed and used for intellectual document indexing. So there are many bibliographic databases or document sets that have been meticulously annotated by domain experts. By assessing these annotations, the ICE-Map Visualization can help finding problems with the thesaurus structure and its concepts. In the context of these experiments, it is at least equally important to analyze the intellectual indexing results to get an overview of the overall focus of the documents regarding the topics that are represented by the thesaurus, as well as the characteristics of the annotations.

[8] Medline abstracts are available via PubMed, http://www.ncbi.nlm.nih.gov/pubmed

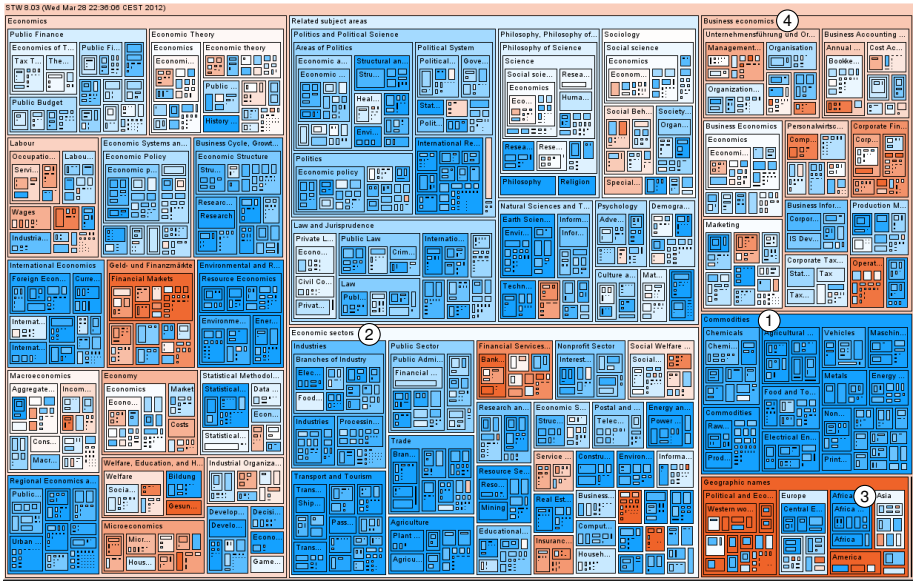


Figure 3.6: STW: Intellectual Indexing vs. IIC.

The information content of the concepts based on the intellectual annotations is compared to their intrinsic one. We are especially interested in concepts that are used with a very high or low frequency compared to the heuristic. Both cases can be a cause for concern, as the thesaurus is not only used for indexing, but to facilitate searching the document base using its concepts. Usually, the query is refined by specializing and generalizing its concepts, until a satisfactory result set is achieved. For queries containing *red* concepts without children, the result set cannot be minimized in this way. The opposite holds true for *blue* concepts without parents. This illustrates that the two tasks of understanding the reference set and analyzing it together with the thesaurus cannot easily be told apart. The strength of the ICE-Map Visualization lies in the visualization of the whole data set, the interpretation rests with the human expert.

STW, Elsevier: Figure 3.6 shows the visualization of the manual assignments against the intrinsic information content. Striking at first glance are the blue area in the right part (COMMODITIES ①), the heterogeneous impression of the area at the bottom (ECONOMIC SECTORS ②) and the blue isolated areas in the otherwise well-used GEOGRAPHIC NAMES section (AFRICA ③, etc.).

The blue color of the COMMODITIES indicates that the concept and its whole subtree are underrepresented. This is hardly surprising given the nature of the journals comprising our dataset, where no commodities in the sense of the thesaurus (like textiles, chemicals etc.) are mentioned.

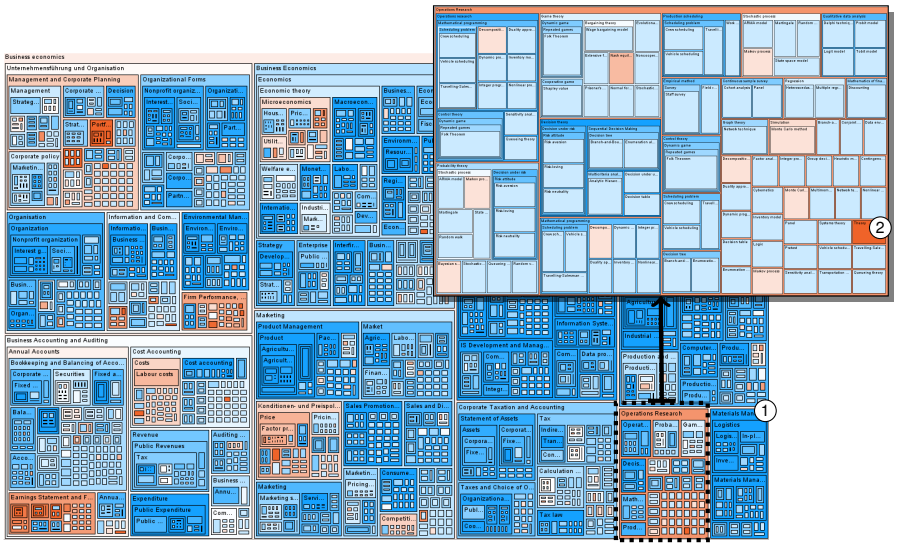


Figure 3.7: STW: Zoom on BUSINESS ECONOMICS.

The concept BUSINESS ECONOMICS ④ shows red and blue areas, therefore we want to have a close look (Figure 3.7). As expected and corresponding to the COMMODITIES section, concepts like MATERIALS MANAGEMENT AND LOGISTICS ① are underrepresented, while the adjacent area on the left is colored in deep red. By zooming into that area (OPERATIONS RESEARCH), we see that the color mainly results from the high usage frequency of one concept. It is the general concept THEORY ②, which is used by the librarians to annotate theoretical approaches in the given articles and which in our dataset sums up to 171 articles (about 46%).

When we demonstrated the software and the results at the ZBW,[9] we learned that this use of the concept THEORY is unsatisfying – and of course known to the maintainers who conceded that it is at least "at the wrong place." That confirms the visual evidence gathered by means of the ICE-Map Visualization. In the current version of the STW, the concept is relocated under GENERAL DESCRIPTORS.

The area in the thesaurus showing the general thematic bias of the documents best is the concept ECONOMIC SECTORS (Figure 3.8). The heterogeneous picture in the overall view shows that the subconcepts in their sum of annotations perfectly fit the expectation. Nevertheless, a closer look reveals that the distribution within the economy branches is not well balanced and reflects exactly the thematic foci of the journals comprising our dataset: HEALTH CARE SYSTEM ①, FINANCIAL SERVICES AND BANKING ②, INSURANCE INDUSTRY ③, and STOCK MARKETS ④ are dominant while most other areas are practically nonexistent.

[9] Workshop on Thesaurus Maintenance, March 10th 2011, ZBW, Hamburg, non-public workshop with participants from ZBW, German National Library, SUB Hamburg and Mannheim University Library.

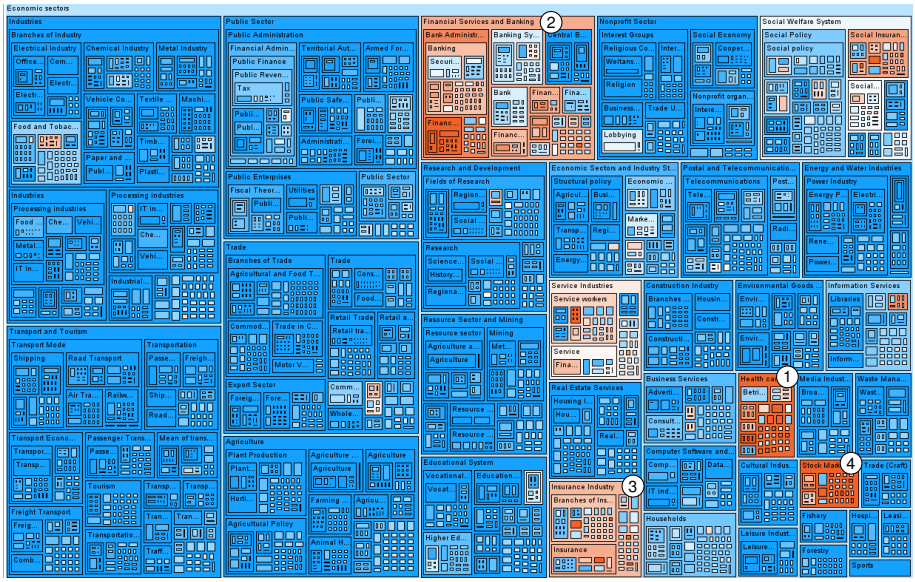


Figure 3.8: STW: Zoom on ECONOMIC SECTORS.

MeSH, Medline: Figure 3.9 visualizes the intellectual indexing result for the MeSH thesaurus and the corresponding Medline document set. As the documents are selected randomly, we will not examine in detail how the documents are distributed. But it can be seen that BIOLOGICAL SCIENCES ①, HEALTH CARE ②, and NATURAL SCIENCES ③ are dominant.

To demonstrate the ICE-Map Visualization, the concept ORGANISMS is more interesting (Figure 3.10). On first glance, the node representing the concept ANGIOSPERMS ① with its subconcepts is structurally visibly different. While such a structure can indicate a problem with the thesaurus, in this case, it 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."[10] While the structural irregularity of the thesaurus correctly reflects the nature of the domain, the blue color indicates that the individual concepts are used infrequently.

In contrast, the node representing the concept MAMMALS is dark red. The zoomed picture reveals that HUMANS ② is used often, which is not surprising as most Medline articles are concerned with the treatment of human patients. Others are MICE ③ and RATS ④, which gives us a direct insight on the favorite subjects of animal testing for drug discovery. Additionally, all these concepts are among "check tags" in the Medline database that are explicitly reviewed for every article and so have a much higher frequency in our sample document base.

[10] http://tolweb.org/Angiosperms/20646/2005.06.03 in The Tree of Life Web Project, http://tolweb.org/

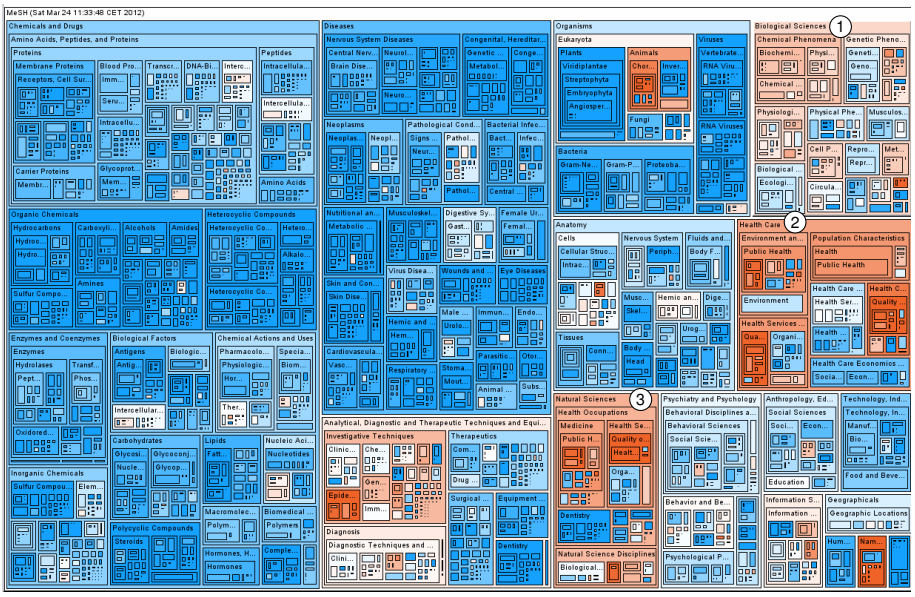


Figure 3.9: MeSH: Intellectual Indexing vs. IIC.

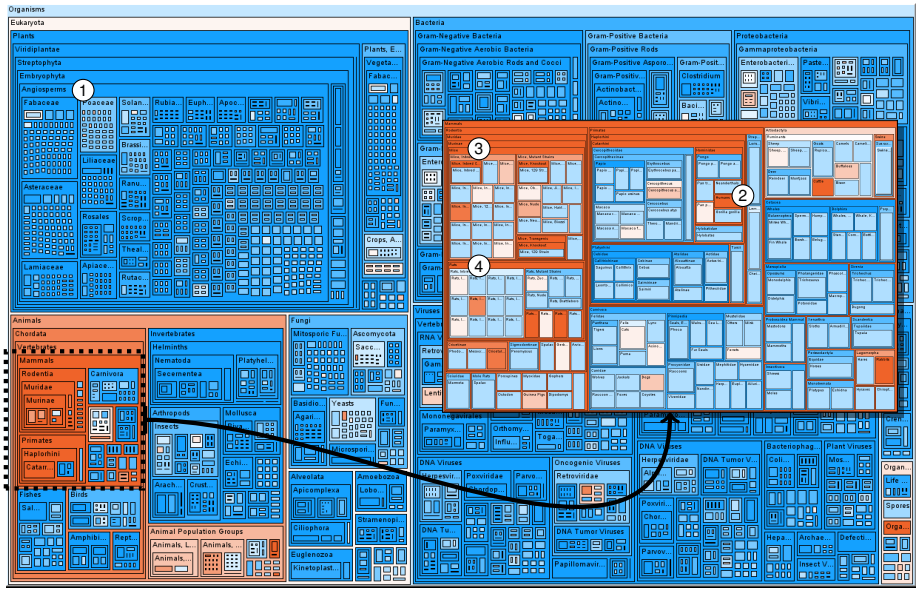


Figure 3.10: MeSH: Zoom on ORGANISMS.

These examples show that the ICE-Map Visualization is able to visualize indexed document sets appropriately. Indexing characteristics like special concepts that are used to categorize the documents can be identified and taken into account for possible reorganizations in the hierarchy. As such, the ICE-Map Visualization is valuable to KOS maintainers, librarians that perform the actual indexing, as well as software developers who need an understanding of both the structural characteristics of a KOS and the annotation conventions of the document when designing specific search interfaces.

3.2.3 Automatic Indexing

Automatic indexing in the context of this thesis is the automatic assignment of KOS concepts to a document based on its content. Automatic indexing approaches are widely introduced to close the gap between the subset of publications that are traditionally indexed intellectually – books in libraries, but also selected journal articles in mostly commercial databases – and the subset of unindexed literature. For instance, the German National Library decided that web publications, while being collected, will not be indexed intellectually, but only by means of automatic processes and search engine technology (Schwens & Wiechmann, 2009).

Mainly two types of approaches can be distinguished: linguistic and statistical approaches. Linguistic approaches use techniques from natural language processing (NLP) to process the texts and extract meaningful concepts; statistical approaches use machine learning techniques to assign concepts based on a manually created training set. There are, however, smooth transitions between both. A recent workshop on automatic indexing[11] showed that there is currently a preference for statistical approaches in German libraries, although the reported quality of the results varies (Schöning-Walter, 2011). A mentioned problem was the bias that is introduced by the training set. For example, for recent news articles, the indexer learned that the occurrence of "Nuclear power plant" should lead to JAPAN as a concept to be assigned.[12] More general is the observation that the indexing quality relies on the homogeneity of the documents to be indexed. If they vary largely regarding content, style or even length, the quality of the indexing result is affected.

For our experiments, we use the Collexis Engine, a state of the art system for linguistic concept-based document indexing and retrieval provided by Collexis.[13] The engine has already been applied successfully in the medical area (Mulligen,

[11] Workshop on Automatic Indexing in the context of the PETRUS project, March 21/22 2011 at the German National Library, Frankfurt am Main, Germany. Semtinel and the ICE-Map Visualization were also presented on request of the German National Library at this workshop: http://www.dnb.de/EN/Wir/Projekte/Abgeschlossen/petrus_workshop.html

[12] Due to the Fukushima nuclear disaster in Japan, following an earthquake with a tsunami on Mach 11th 2011, which dominated world-wide news for many weeks.

[13] http://www.collexis.com

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

Figure 3.11: Document example.

Pricing behavior of firms
Oligopoly
Effects of taxation
Tobacco tax
Cigarette industry

(a) Intellectual Indexing

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

(b) Collexis

Table 3.1: Example annotations: Intellectual Indexing vs. Collexis.

Eijk, Kors, Schijvenaars, & Mons, 2002; Stuckenschmidt et al., 2004) and therefore provides an adequate basis for our investigations. Collexis uses a pure linguistic approach to assign concepts to documents: first, stop words are identified and removed, then the text is normalized and concepts are selected by comparison with the labels of the concepts in the underlying KOS. At last, the assigned concepts are weighted based on several algorithms. The whole process is comparable to the indexing pipeline implemented by LOHAI, our simple baseline indexer used in Section 3.1.

Figure 3.11 provides an example article from document set 1. Table 3.1 lists the concepts that were assigned to the article by a librarian and contrasts them to the concepts found by the Collexis Engine. As we can see, there are significant differences: there is only one concept (OLIGOPOLY) that is assigned both manually and automatically. Other concepts are obviously related like PRICING BEHAVIOR OF

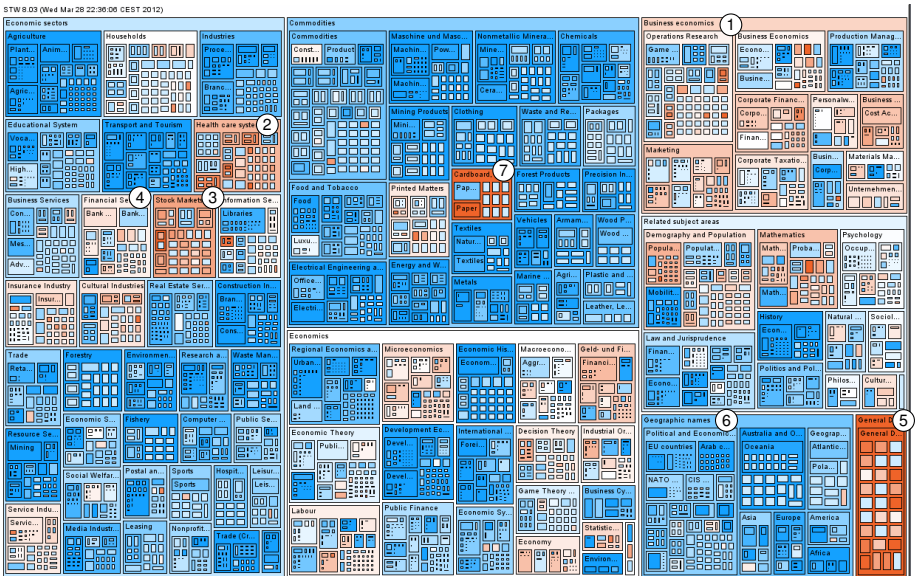


Figure 3.12: STW: Collexis vs. IIC.

FIRMS to PRICE and RETAIL PRICE. The same holds for the tax related concepts. It seems to be that the automatic indexer prefers shorter, more common terms, in contrast to the librarian who assigned more specific and abstract terms. This is understandable if we consider, how the Collexis Engine works. At last, a concept – i.e. one of its labels – has to occur in the text literally, which is unlikely for a label like "Pricing behavior of firms."

It becomes clear that the Collexis Engine follows a different strategy compared to a librarian, as usually PRICE would not be assigned together with the more specific concept RETAIL PRICE. This of course could be filtered out, if the goal would be to resemble an intellectual indexer as close as possible, which is declaredly not the case for the Collexis Engine. Which brings us back to our goal: to learn about the indexing characteristics of the Collexis Engine by means of the ICE-Map Visualization.

For the STW and document set 1, Figure 3.12 shows the visualization of the Collexis indexing results. We see that there are many assignments of concepts in the subtree of BUSINESS ECONOMICS ①, which is not very surprising. Beside the general economic focus of the documents, there are also a lot of common concepts in this subtree, like RATIO, CAPITAL, PRICE, and MARKET. Other areas that are well represented in the annotations are HEALTH CARE SYSTEM ②, STOCK MARKETS ③, as well as FINANCIAL SERVICES AND BANKING ④. This focus can easily be explained by looking at the journals the articles were taken from (e.g. the Journal of Health Economics) and corresponds with our findings in Section 3.2.2. 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. A characteristic of the automatic indexing system can be seen in the high usage of GENERAL DESCRIPTORS ⑤ and the low usage of GEOGRAPHIC NAMES ⑥. While the former contain often occurring terms like EXPERIMENT and RISK (Figure 3.13) that are rarely assigned by a librarian, the latter are often not mentioned directly in the abstracts and therefore hard to identify by the automatic indexer.

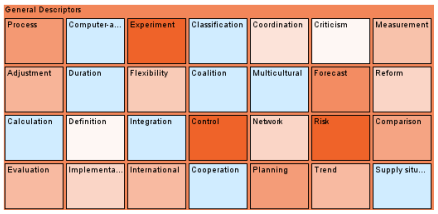


Figure 3.13: GENERAL DESCRIPTORS.

The difference analysis also identifies actual errors. The red color of CARDBOARD, PAPER, AND PAPER PRODUCTS ⑦ leads us to a good example. This concept refers to the branch of economics concerned with the production of paper. In Figure 3.14, we zoom into COMMODITIES. It can be seen that PAPER ① is the concept responsible for the red color. The indexing process often assigned this concept to documents that are not concerned with its intended meaning because of phrases that contain references to scientific publications ("In a recent paper ... suggest," "This paper is about ..."). A solution is to add additional information to the concept to help the indexer with the disambiguation between the two meanings of paper. Similarly wrong assignments can be found in the subtree of PRINTED MATTERS ②, due to the concepts BOOK and JOURNAL. Furthermore, we identified COMPONENT ③ as a problematic concept. Here, a more specific translation would help. The preferred term in German is unambiguously "Baufertigteil," which means a prefabricated component for building. Finally, we see the concept DRUG, as red as COMPONENT, but in this case the assignments are correct. This emphasizes that the ICE-Map Visualization is a great tool for a human maintainer to discover the characteristics of an automatic indexer, but the decision if frequent assignments indicate a problem cannot be made automatically.

Comparing Automatic to Manual Annotations So far, we only used the automatic indexing results to gain insights into its characteristics. This way, automatic indexers can be monitored constantly and suspicious assignments can be spotted easily. There is, however, a drawback that should not be concealed: while it is easy to identify *wrong* assignments when they happen systematically, it is much harder to identify *missing* assignments. Furthermore, errors and biases with regards to content have to be told apart.

When developing automatic indexing processes, usually a document set annotated by a human expert is used to compare the annotations against the automatic indexing results. When we use such a reference set in the ICE-Map Visualization one of its main strengths becomes visible: where traditionally only numbers like

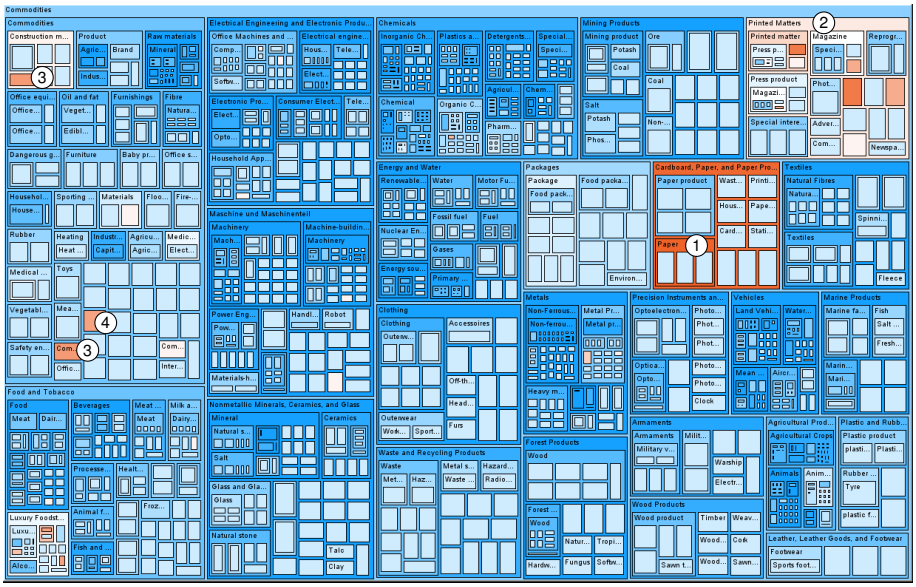


Figure 3.14: STW: Zoom on PRODUCTS.

precision, recall and f-measure are calculated to compare two indexing results, now a meaningful visualization of the differences is provided.

In this experiment, two information content values based on both annotation sets are used. Deeply tinted tiles still can indicate problematic concepts, but in this analysis, the effect of the topical focus is diminished and the different characteristics of the indexing processes can be seen directly, i.e. we will directly compare the results of the automatic indexing system to the annotations made by the librarians. Figure 3.15 shows the resulting visualization.

Our previous findings are confirmed that the automatic indexer often assigns GENERAL DESCRIPTORS ① and misses GEOGRAPHIC NAMES ②. The reason for the latter is that geographical terms rarely appear verbally in the abstracts used for the automatic indexing process. In related articles, mainly written for the domestic market, there seems to be no necessity to mention the name of the country explicitly. However, considering annotations assigned for foreign users combined with a conscientious librarian, geographical information will surely be part of the keyword chain.

Besides the already identified PAPER other concepts in the commodities section are problematic as well. An example where the word sense disambiguation failed is HIDES AND SKINS in the section AGRICULTURAL PRODUCTS ③, where the synonym “Fell” leads to wrong assignments. On the other side, the often used concept THEORY (cf. Section 3.2.2) is missed by the indexer, while other concepts in the area OPERATIONS RESEARCH ④ are assigned more frequently, like PANEL. Similarly, in the area of CULTURAL INDUSTRIES ⑤, the concept LITERATURE –

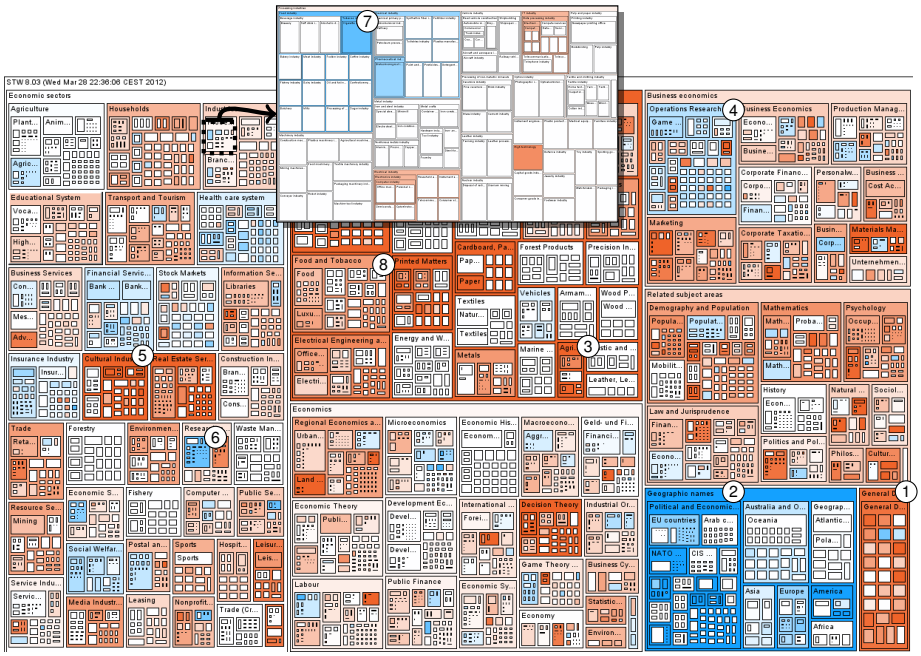


Figure 3.15: STW: Collexis vs. Intellectual Indexing.

meaning the art – is often incorrectly assigned for occurrences of the term "literature." The list could be continued, as such ambivalent concepts causing problems in an automatic indexing process can easily be identified using the ICE-Map Visualization.

Instead, we investigate a different characteristic that can be recognized from the missing concepts: the automatic indexer has problems with compound terms and abstract concepts. For instance, it misses several times the concept STOCKHOLDING BEHAVIOUR (in FIELD OF RESEARCH ⑥). For the documents in question, it assigns, however, concepts like STOCK or STRATEGY. Similarly, we can see that CIGARETTE INDUSTRY ⑦ is underrepresented, whereas CIGARETTE and other concepts in FOOD AND TOBACCO ⑧ are overrepresented.

We can conclude that associative knowledge such as similarities of a concept to certain theoretical edifices or to more general concepts can hardly be found by automatic indexing. It becomes clear at this point that the counting of words and/or the comparison of strings cannot produce any additional knowledge beyond the identification of similarities.

3.2.4 Tagging

A completely different possibility to index documents with alternative means is *collaborative tagging* (also known as folksonomy, social classification, social in-

dexing, and by other names), which indicates the practice and method of collaboratively creating and managing tags to annotate and categorize content. In contrast to traditional subject indexing, annotations are not only generated by experts but also by creators and consumers of the content itself. Usually, freely chosen terms are applied instead of a controlled vocabulary. Among the most popular applications based on collaborative tagging are Flickr for storing photos or Del.icio.us for collecting links to websites. CiteULike, Connotea and BibSonomy are bookmarking services for academic purposes organizing individual and common access to scientific information.

Tagging is generally seen as one of the cornerstones of the social web, or *Web 2.0*. As an appealing new concept, a lot of researchers investigated the phenomenon of tagging and its advantages and disadvantages for various purposes. As for the advantages, the immediacy of tagging is probably the first and foremost: When a new concept like WEB 2.0 evolves, librarians have to integrate this new term in the existing KOSs. This process is often handled in a very conservative manner as the indexers wait to see whether a new term will gain more importance or not. Their aim is to keep all parts of the system in balance regarding their size and relevance. For instance, our example WEB 2.0 has not yet been included in the Regensburg Union Classification.[14]

In contrast to that, with tagging there is no delay between the publishing of a document and its annotation because a controlled vocabulary is neither necessary nor used (Mai, 2006). In addition, KOSs often represent the scientific paradigms of their date of origin. For example the classification of the Library of the University of Bielefeld was created in the late 1960s; its main feature is a strong focus on economic and social aspects within the historical classes - an approach typical for the research interests of historians at that time. User generated annotations do not have this problem because they represent current perspectives as well as the thematic landscape of publications at a given moment. They can follow the change of interest within the subjects dynamically (Quintarelli, 2005).

On the other hand, the lack of controlled vocabularies is also the biggest disadvantage of tagging. Indexing with free vocabulary will result in ambiguous terms using synonyms or homonyms in different contexts. Take for example a search for the computer language Python, which will also yield hits including the snake or the ancient potter. Abandoning the librarian indexing will have negative consequences for the quality of the information retrieval when using library search tools (Guy & Tonkin, 2006; Gordon-Murnane, 2006). The organization, validation, and integration of the collected data falls well short of professional standards regarding structural depth and reasoning capabilities.

[14]Regensburger Verbundklassifikation, RVK. The RVK has been developed by librarians of the University of Regensburg and is utilized by about 20 other university libraries

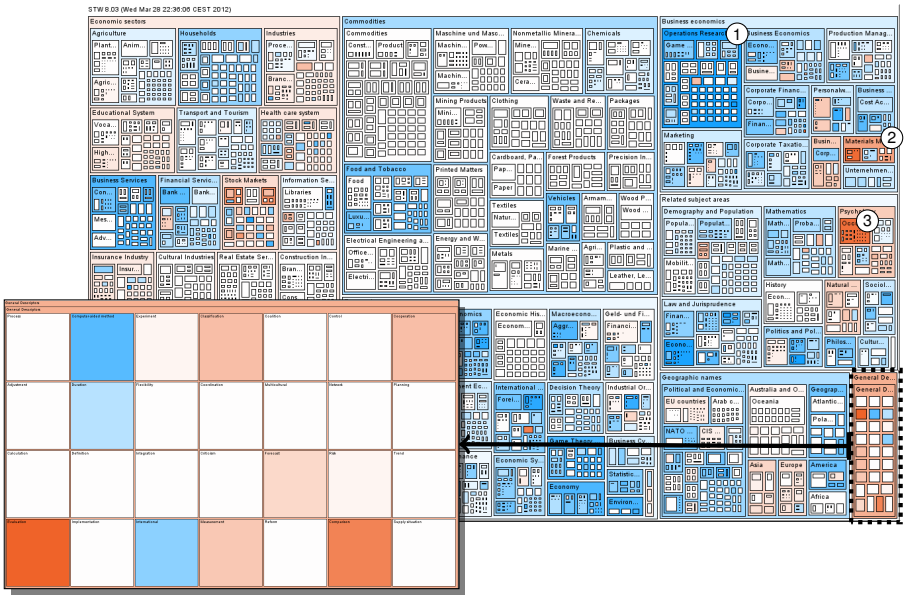


Figure 3.16: STW: Tagging vs. Intellectual Indexing.

In this experiment, we want to examine the behavioral characteristic of a tagger with the ICE-Map Visualization. To employ the ICE-Map Visualization, we have to postulate the existence of a hierarchy and restrict the user to it: for our experiment, concepts of the STW are proposed for tagging. We asked an undergraduate to assign adequate STW concepts to our documents without preparatory training. The student was instructed to think about possible tags to describe the given text and then choose suitable concepts in the thesaurus. This is comparable to the recommendations of tags to users based on the overall collection of tags that were already used in a given system. Of course, the results cannot be generalized from one person, however, we assume that the student behaved like a typical layman user. Moreover, in the context of this thesis we want to demonstrate how this behavior can be examined by means of the ICE-Map Visualization. We are not focused on the in-depth assessment of tagging as an alternative indexing method.

We use the ICE-Map Visualization to compare the tagging results with the results of the intellectual indexing performed by a librarian – as examined in Section 3.2.2. For the whole document set 1, we got 579 annotations by tagging, roughly one third of the annotations made by the librarians. This leads to the first hypothesis that some details are missing. Figure 3.16 shows the overall view of the ICE-Map Visualization, as anticipated mostly colored in blue, showing that most areas are used less frequently.

The area with the biggest negative difference is the subtree below the concept OPERATIONS RESEARCH ①. The primarily responsible concept for this finding is THEORY. Although THEORY is used very often by librarians, it was never as-

signed by the undergraduate. The reason can be found in considering the training of the librarians, who usually evaluate a document according to its practical or theoretical focusing. In the abstracts of the document set, this particular aspect is not often mentioned explicitly and thus was completely ignored by the undergraduate. Additionally, the position of the concept might play a role, as it is not recognizable as a general categorizing concept (cf. Section 3.2.2).

In contrast, the area with the biggest positive difference is the one with the GENERAL TERMS (highlighted). Whereas the COMPUTER-AIDED METHODS are underrepresented, similar to the results of the automatic annotations, concepts like COOPERATION or EVALUATION are used more often by the undergraduate. A closer look on the documents involved reveals two reasons: First, the librarians tend to use more specialized concepts in the thesaurus where available. For example, they assign BUSINESS COOPERATION instead of COOPERATION and CORPORATE ASSESSMENT instead of EVALUATION. Second, at several times the undergraduate used only one of the GENERAL TERMS to describe a document. We assume that he failed to find adequate concepts in these cases and thereupon switched to a general one like COMPARISON.

The concepts ORDER (in MATERIALS MANAGEMENT AND LOGISTICS ②) and PERFORMANCE INCENTIVE (in OCCUPATIONAL AND ORGANIZATIONAL PSYCHOLOGY ③) are used frequently. They are not assigned wrongly, but especially ORDER is not descriptive for the documents in question, even if the term appears in the abstract. The same holds for PERFORMANCE INCENTIVE, in all cases, the term literally appeared in the abstract.

Generally, it can be stated that the undergraduate in our example was able to avoid obvious mistakes made by the automatic indexing system (like assigning PAPER to every occurrence of the term). However, the results turn out to be similar in the sense that concepts preferably are assigned when occurring in the text explicitly. The user does not have the long-time experience and the specialized training of a librarian and thus does not possess the same ability (or motivation) to read between the lines. Nevertheless, the assignments of the user showed no severe mistakes regardless of a somewhat imprecise usage of terms. This could presumably further improved if the access to all concepts is facilitated by a more intuitive and easy-to-use method.

A last point has to be made regarding the tagging approach: We compared the tagging results of only one user to the annotations of professional librarians. The general success of tagging in the Internet strongly depends on the "wisdom of crowds," the collective intelligence of a large quantity of users. It can be expected that at least some of the weaknesses concerning the lack of appropriate annotations can be resolved simply by taking more users into account, which is among others confirmed by Suchanek, Vojnovic, and Gunawardena (2008).

3.3 Related Work

Visualization techniques for documents in a large set have been explored by several groups. The approaches differ in the methods for grouping and navigation as well as the neccessary characteristics of the document set. Granitzer, Kienreich, Sabol, Andrews, and Klieber (2004) describe the *InfoSky* visualization, which is applied on hierarchical document sets and utilizes bounding polygons using a modified weighted Voronoi diagram combining it with a seamless zooming interface for navigation. The metaphor employed is "Documents of similar content are placed close to each other and displayed as stars, while sets of documents at a particular level in the hierarchy are visualized as bounding polygons." The authors note that during user tests on prototype implementations, users sometimes had difficulties with this metaphor. Therefore, instead of the inherent hierarchy of a KOS, the file system hierarchy of the visualized repository is used. Lehmann, Schwanecke, and Dörner (2010) explore ideas for visualization of very large document spaces, using Wikipedia as a set. The *Wivi* visualization relies on links between the documents to allow for grouping and interactive navigation between individual objects and cannot be applied to unlinked documents. Telles, Minghim, and Paulovich (2007) use Kolmogorov complexity approximations to detect similarities between documents. Based on these similarities, they create map visualizations of a document set based on fast distance multi-dimensional projections. This approach differs, as the map is not based on an underlying KOS, but only on the contents of the documents.

Various researchers investigated the characteristics of indexing approaches, as well as different human indexers. The (in)consistency between different human experts regarding the indexing results is discussed since the introduction of thesauri and electronic cataloging in the 1960s, e.g., by Rolling (1981). Olson and Wolfram (2006) provide a comprehensive overview on further studies. Recent approaches to measure the consistency of indexers incorporate the structure of the underlying KOS and use the vector space model from information retrieval (Medelyan & Witten, 2006a; Wolfram & Olson, 2007). Generally, it has to be stated that the consistency is not very high. This does not mean that the results have a poor quality. Two indexers can assign different concepts to a document and both sets can be considered correct. This reduces the usability of intellectual indexing results as a gold standard for evaluation purposes and increases the need to deal with and incorporate different indexing sources to reach a high coverage of index terms that are suitable to describe a document. This is in line with our findings, where the tagging student significantly differs from the librarians, without actually assigning wrong concepts.

The characteristic of an automatic indexing system depends on the underlying approach. In this chapter, we evaluated one specific linguistic indexer. However, various other approaches exist to automatically index documents based on a given

KOS, commercial systems as well as technological studies that have not yet left the research labs (cf. Section 5.3). All of them have in common that they rely on the quality of the underlying KOS and face various problems that generally belong to aspects of natural language processing.

Tagging as an alternative indexing approach has been evaluated by many other scientists. Shirky (2005); Wal (2005) found that usually only few tags are chosen to describe a given article by many users. A graph containing the number of the tags annotated to a resource on the x-axis and the rank of a tag on the y-axis results in a so-called long tail. This is confirmed by our experiment. Wolff, Heckner, and Mühlbacher (2008) present an empirical study on tagging behavior in the scientific annotation system Connotea and selected 500 tagged articles covering information and computer technology. They set up a model for linguistic and functional aspects of tag usage and the relationship between tags and a documents full text. Their results describe the typical tag as a single-order noun, taken from the title of the article and directly related to the subject. Razikin, Goh, Chua, and Lee (2008) investigated the effectiveness of tags as resource descriptors determined through the use of text categorization using Support Vector Machines. For this, they randomly collected 100 tags and 20,210 documents. Their results were ambivalent: some tags were found to be good descriptors while others were not. They state: "Given that tags are created for a variety of purposes, the use of tags to search for relevant documents must therefore be treated with care."

3.4 Conclusion

In this chapter, we demonstrated the application of the ICE-Map Visualization for the evaluation of KOSs considering different use cases. First, we dealt with the proper selection of an existing KOS for a given document set. We combined the ICE-Map Visualization and a simple automatic indexer and performed several experiments with two KOSs and two document sets to prove our assumption that the ICE-Map Visualization is suitable to visualize the topical overlap, i.e., that it is possible to identify whether KOS and document set topically fit together. We showed that the assumption holds even for relatively small document sets yet benefits from larger ones. The ICE-Map Visualization gives not only a broad overview, it can also be used to see in detail on which topics the documents focus. The choice of a suitable KOS or the maintenance of an already used KOS is strongly simplified.

We explained that the same approach can be employed to see if a document set fits to a given KOS. By providing own, intellectually obtained annotations, the ICE-Map Visualization was used that way already by the Library of the University of Leipzig to visualize the whole stock and the distribution over the employed classification system (Regensburg Union Classification). Additionally, we directly compared two different document sets, in which case the ICE-Map Visualization

reveals exactly how they differ in the topical focus with respect to the employed KOS.

In future, the approach could be improved in several ways. First of all, the indexer as a main component can be improved. Our simple indexer works for two reasons: First, the ICE-Map Visualization is stable and not immediately influenced by some indexing errors. Second, if the indexing errors influence the visualization significantly, this can be easily spotted, as Semtinel provides the functionality to see the documents associated with a concept as well as the location in the original abstract that led to the assignment of the concept. We used this to make sure that everything we see in the visualization makes sense and is not pure coincidence. However, a better indexer would not hurt. Another aspect is the employed weight function. We used a simple measure by just summing up *tf-idf* measures. This is reasonable, but other approaches that take more information into account – e.g. additional information provided by the indexer, like an indication if word-sense disambiguation was performed – might lead to significant improvements.

In the second part of this chapter, we demonstrated the use of the ICE-Map Visualization for the evaluation of indexing results, be they intellectually created by a librarian, automatically created or created in a tagging-like environment by a layman. The focus of this chapter is not on the actual indexing results, but on the general demonstration which tasks can be performed by means of the ICE-Map Visualization and how it can contribute to the evaluation and maintenance of indexing systems, if a KOS is to be employed. Nevertheless, the experiments also showed that alternative indexing processes can be used, certainly with some compromises regarding the quality compared to the quality of intellectual indexing by a librarian. Especially when the concept annotations for an article are missing – be it a matter of time (if the article in question was published recently) or a matter of granularity (if the article will not be annotated in the usual process of a library) – the additional sources can come in handy to fill this gap. Despite their lower quality they can improve the search experience by this means.

With the ongoing growth of scientific publications, it is not questionable that fast, informal and ad-hoc mechanisms like automation and tagging are needed to keep up with the increasing amount of new publications. But such sources for annotations have to be supervised, especially automatic indexers have to be maintained and problematic concepts have to be detected in an efficient way. Structural irregularities can easily be spotted with the ICE-Map Visualization, and deeply tinted tiles represent possibly problematic concepts. They can be caused by problems in the KOS or just reflect a bias in the documents regarding the content. But they also can indicate a problem of the indexer – or a combination of all. We identified the following typical problems:

Context Dependence Concepts are sometimes homonyms of commonly used terms in a text. This preferably happens in highly specialized domains where special terminology is used. In this case there are two options. Either, advanced

mechanisms for context detection can be used or, in cases where these methods are too expensive, the corresponding term can be deleted from the KOS to avoid false annotations. This kind of problem normally causes a relatively low information content with respect to the automatic annotations (many occurrences, colored red).

Missing Definitions Sometimes, concepts are not detected in documents because a certain synonym used in the text is not included in the KOS. In this case, the definition of the concept has to be refined adding the corresponding synonyms. This problem normally causes a relatively high information content with respect to the automatic annotations (few occurrences, colored blue).

Normalization Errors In cases where linguistic tools are used for preprocessing, the meaning of terms can be lost as ambiguity is introduced in the normalization step. In this case we either have to use more advanced preprocessing methods that are capable of eliminating the ambiguity introduced for instance by first detecting noun phrases and only using them as a basis for indexing. As above, if this approach is too expensive, we can also eliminate the corresponding terms from the KOS to avoid wrong annotations. This kind of problem normally causes a relatively low information content with respect to the automatic annotations (many occurrences, colored red).

Indexing Preferences Human annotators sometimes show certain preferences in selecting index terms that cannot be reproduced in an automatic indexing process. A typical example is the use of check-tags, predefined lists of index terms that can more easily be assigned by selecting a check-box. These terms will be over-represented in manual annotations. These terms should be treated separately in the indexing process and special strategies need to be developed for this purpose. This problem normally causes a relatively high information content with respect to manual annotations (few occurrences, colored blue).

We showed that the with the ICE-Map Visualization it is easy to identify potentially problematic parts of a KOS and that a manual inspection of these problematic parts often reveals problems that appeared in the indexing process. Thus we conclude that the ICE-Map Visualization is a suitable means to improve the results of alternative document indexing. By focusing on the KOS, the general analysis approach is scalable with increasing amounts of documents. The concept hierarchy is the constant factor in the process, represents the subject domain of the documents and is presumably well known by the human expert performing the evaluation. Identifying necessary changes to the concept hierarchy is essential and thus it is worthwhile to focus the effort of the human expert on it. The complete replacement of intellectual indexing by automatic annotation systems is neither likely nor wanted, instead, the combination of various indexing approaches is promising for the best possible document retrieval. The ICE-Map Visualization can be an important part of such an integrated approach.

Acknowledgements: Parts of this chapter have been published before. The application of the ICE-Map Visualization for the selection of a KOS and for the evaluation of unknown document sets is published in (Eckert, Ritze, & Pfeffer, 2012). In (Eckert et al., 2007), we introduced the ICE-Map Visualization first, for the assessment of automatic indexing processes (Best-Paper-Award). An extended version was published in (Eckert et al., 2008). The investigation of tagging as an alternative means for indexing is published in (Eckert et al., 2009).

Chapter 4

Creation and Modification

To achieve a high interoperability between knowledge-based applications, generally the new creation of a KOS should be avoided, if an existing one can be reused. In Chapter 3, we have shown, how existing KOSs can be evaluated for a given purpose. Sometimes, however, a KOS has to be created from scratch and almost always an existing one has to be adapted and modified to fit perfectly to the resources to be described.

Creating and maintaining a KOS is a costly and cumbersome task. It is normally performed by a group of specialists skilled in both the domain of interest and the relevant methods of formal knowledge representation. As such experts are expensive and in short supply, the discovery of alternative methods of creating and maintaining concept hierarchies would be a major benefit. Significant work has been performed on the automatic creation of concept hierarchies from texts, but these methods often fail to correctly capture semantic relations between topics. In particular, automatic methods are often weak on the task of determining the type of relation that holds between two terms. Therefore, we investigate different approaches that do not create or modify a KOS fully automatically, but put the human in the loop in one way or another.

4.1 Crowdsourcing the Creation Process

The above mentioned problems have inspired researchers to search for alternative sources of information to support the construction and validation of concept hierarchies. The Indiana Philosophy Ontology (InPhO) project (Niepert, Buckner, & Allen, 2007) routinely solicits small amounts of information from a concept hierarchies' users while they are engaged in the process of using and maintaining it. The system is based on the involvement of the user community that consists of a relatively small number of domain experts whose expertise is gathered and

combined to dynamically generate a taxonomy of philosophical ideas. The general applicability of this approach, however, is hampered by the fact that it relies on the existence and commitment of expert volunteer users.

In this section, we investigate whether similar results can be achieved in a setting where no group of experts is available and instead a much larger number of non-experts provide the input. This approach, often referred to as "wisdom of crowds" (Brabham, 2008), has become very popular recently in the context of Web 2.0 applications. Typical applications are the tagging of resources (cf. Section 3.2.4) by users, usually to organize them for personal purposes. Other examples are collaborative projects like Wikipedia where people contribute, partly for fun, partly because they get credits from the community. Then there are applications that produce information as a byproduct, like reCaptcha (Ahn, Maurer, McMillen, Abraham, & Blum, 2008) where people have to enter text from images on webpages to prove that they are human. The images are taken from book scans and the user input helps transcribing them. Even games can be created that produce valuable information, so-called *games with a purpose*. It has been demonstrated that good results can be achieved for tasks such as annotating unlabeled images (Ahn & Dabbish, 2004).

All these approaches have in common that there is an incentive for the crowd to participate, be it a personal advantage to organize resources, credits from a community, access to a webpage or simply fun. A special form of incentive is to pay for the wisdom of the crowd. This form of value creation is usually referred to as crowdsourcing, a term coined by Jeff Howe in a 2006 *Wired* article (Howe, 2006). A number of platforms have emerged to provide a framework for crowdsourcing of a variety of general purpose tasks. Probably the best known is *Amazon Mechanical Turk* (MTurk).[1] With MTurk, Amazon offers extensive options for creating customized questionnaires. Results can easily be processed as they are made available in standard formats. Due to its relatively high publicity (roughly 250,000 tasks available at the time of this writing), it attracts a lot of users and consequently seems most suitable for our purpose to answer the following question: Is it possible to use the wisdom of crowds to create high quality concept hierarchies in a challenging and abstract domain like philosophy?

4.1.1 Method Description

In previous works, Niepert et al. (2007); Niepert, Buckner, and Allen (2008, 2009) presented the InPhO project as one of the first to maintain a dynamically growing knowledge representation of the discipline of philosophy. The system is primarily developed to create and maintain a formal ontology for a well-established, open-access reference work, the Stanford Encyclopedia of Philosophy (SEP). Three fea-

[1] http://www.mturk.com

tures of the SEP make it an ideal environment for developing and testing digital tools to learn and manage ontologies:

First, it is substantial and complex: over 1,150 entries (>14 million words) of sophisticated humanities content that is beyond the comprehension of any one individual. Second, the SEP is dynamic: new and revised entries are published online each month. Finally, it is expert-driven: more than 1,400 professional philosophers serve as its editors and authors.

Many online reference works are well-positioned to address the mentioned challenges by making use of their most valuable informational resource: the domain experts who serve as their editors and contributors. Carefully obtained expert feedback can be used to approve the recommendations of automated methods without presuming knowledge of ontology design or placing undue demands on the contributors' time. The InPhO project successfully maintains a dynamically growing concept hierarchy of philosophical ideas by leveraging feedback facts provided by a user community consisting of users ranging from interested amateurs to domain experts including the SEP authors. The concepts in the InPhO hierarchy are related over *is-a* relations. Each of these concepts (e.g., *rationalism*) is referred to by a term in InPhO's controlled vocabulary. The problem of determining hierarchical relationships between concepts can be reduced to that of finding hierarchical relationships between terms, i.e., extracting hypernym and hyponym relations from text.

There are two necessary conditions for a term t_1 to be a hypernym of term t_2: it has to be (a) semantically similar to t_2 and (b) more general than t_2 in the context of the subject area the terms are used in. Conversely, for a term t_1 to be a hyponym of term t_2 it has to be (a) semantically similar to t_2 and (b) more specific than t_2. A large number of measures exist for the semantic similarity between terms. Such measures of similarity and generality have been combined to provide, for any given term, a ranking of possible hyponyms and hypernyms, respectively (Niepert et al., 2007). The ranking is then presented to InPhO's users to approve or falsify the estimates of semantic relatedness and relative generality of pairs of terms. The relatedness is scored on a five-point scale from highly related to unrelated, and the generality question has four options: same level of generality, $idea1$ is more general than $idea2$, $idea1$ is more specific than $idea2$, and the two are incomparable. The generality of two ideas is deemed incomparable if they are entirely unrelated or if one idea can be both more *and* less general than the other, depending on the context. In this manner, expert feedback can be obtained to *confirm* or *disconfirm* hypotheses about semantic relationships between terms without presuming any knowledge of ontology design.

This use of expert feedback, however, raises three additional challenges. For one, while expert feedback may be the highest quality feedback available for the domain, it is hardly infallible, and experts will often be biased in predictable ways, for example by privileging their own preferred area of specialty over oth-

ers. Niepert et al. (2007, 2008, 2009) have tried to remedy the issue of expert bias by collecting redundant feedback from multiple experts and by looking for inconsistencies, either direct (e.g. expert 1 says A is more general than B whereas expert 2 says B is more general than A) or implied (e.g. inferred through the transitivity of taxonomic relations). Secondly, the presence of inconsistencies raises a further challenge of finding rational strategies to cope with these forms of expert disagreement, preferably in a way which mitigates expert bias. Thirdly, feedback is collected asynchronously, either as volunteers evaluate pairs or as feedback is solicited during routine tasks, such as during the process of adding and updating encyclopedia articles. To address all of these issues, the authors have recommended a *dynamic* approach to ontology population and design, on which ontologies are built and populated continuously as feedback is received using a non-monotonic answer set program with stable model semantics. Expert feedback is translated into first-order facts as they come in, and an answer set program is run on these facts daily to flexibly re-construct the global populated ontology.

On this scheme, several methods to address the problem of inconsistent expert feedback are used by Niepert et al. (2008). First, each user self-reports a level of expertise (1=amateur, 2=undergrad course, 3=grad course, 4=publication in area) in up to two areas of philosophy. Inconsistencies of the answers belonging to users of the same level of expertise (intra-level inconsistencies) and inconsistencies between these levels (inter-level inconsistencies) are dealt with separately. Intra-level inconsistencies are settled before feedback facts are passed to the answer program by using a pre-processing *voting* filter, which takes a *majority rules* vote at each strata of expertise. For example, if at expert level 2 are 4 users asserting that A is more general than B and 2 users asserting that B is more general than A, only the majority opinion at level 2 passes through the filter (where ties are settled by returning to the statistical estimates of generality and similarity). This way "eccentric" expert judgments can get screened out without being paralyzed by inconsistencies.

A further challenge, however, is to flexibly and rationally integrate inter-level inconsistency while making good use of the insight that not all user feedback is created equal. In short, when resolving inter-level inconsistencies, it should be possible to privilege expertise without throwing away possibly useful information contained in responses provided by non-experts. The current solution to this problem involves a second round of filtering within the answer set program. Candidate taxonomic facts are asserted in the final ontology only when there is evidence for them and no evidence against them. Only when two facts – whether directly asserted by users or inferred from user feedback by the answer set program – are inconsistent, the fact at the *lower* level of expertise is said to have strong evidence against it and is discarded. In addition, trust and reliability scores are automatically computed for all users to evaluate their reliability. These further sources of provenance information can be used in future inconsistency-resolution schemes. The final inferred ontology is thus a mosaic continuously constructed through the

flexible integration and cross-validation of partial and overlapping feedback provided by a number of users of varying levels of expertise. The growing knowledge base can be browsed online.[2]

While the InPhO project is fortunate enough to continuously collect feedback facts from its volunteer users, the existence of a motivated user community is an exception. The question we mainly address here is whether the InPhO approach can be applied in scenarios where a user community is absent. Instead of relying on volunteers can we, for instance, pay MTurk workers to acquire feedback facts? And what is the quality of these feedback facts?

4.1.2 Experimental Setup

The objective of our experiments is twofold. First, we want to compare the quality of feedback provided by the InPhO community with the feedback provided by the MTurk workers. Thus, for the first time, we directly compare the quality of contributions provided by a typical Web 2.0 community of volunteer users with those provided by MTurk's workers. Second, considering that many real-world scenarios lack de-facto gold standards such as InPhO's set of expert evaluations, we describe and compare different strategies to filter users according to their feedback quality. We believe that these strategies are not only applicable for extending and populating taxonomies but also in other knowledge management scenarios.

At the time of our experiments, the InPhO system had 114 registered users, 45 of which provided one or more of the 4,883 feedback facts. Table 4.1 shows, for each $1 \leq i \leq 5$, the number of pairs that were evaluated by at least i different users. Among the 114 users, 43 reported the highest level of expertise, meaning that they had published in their respective area; 45 had finished a graduate class in philosophy. Based on the existing evaluations from the InPhO community we created the dataset of pairs that were given to the MTurk workers for evaluation. As a significant overlap with InPhO's users is needed to compare the results, we selected only concept pairs that were evaluated by at least two distinct InPhO users, resulting in 1,154 pairs of concepts.

The experience we gained from preliminary small-scale experiments indicated that a rigorous evaluation of the results is impossible if most MTurk workers evaluate only a small number of concept pairs. To avoid this data sparseness problem, we created single tasks – referred to as Human Intelligence Tasks, HITs – that consisted of 12 distinct pairs of philosophical concepts. This way we obtained at least 12 different evaluations from each MTurk worker. For each set of 12 concept pairs we created 5 HITs to obtain at least 5 evaluations by 5 different workers for each distinct pair of concepts. This resulted in 8,640 pairs that were presented to MTurk workers in 720 distinct HITs each consisting of 12 concept pairs. A finished HIT

[2] http://inpho.cogs.indiana.edu/

was awarded 0.16 USD and the maximum work time for each HIT was set to 20 minutes. The HITs were presented to the MTurk workers in the same form as they are presented to InPhO users ensuring equivalent conditions and comparability of the results. Figure 4.1 depicts the MTurk interface with the concept pair VIRTUE EPISTEMOLOGY and EEPISTEMOLOGY.

Measuring Agreement

A large set of measures is available to assess the deviation of two statistical variables. In our setting, we are interested in quantifying the agreement of groups of users. Therefore, in our experiments, we always compute the degree of deviation between the feedback facts obtained by different sets of users. As described by Niepert et al. (2009), this can be used to determine the disagreement between a user and other users in the same group. In the following, we define the evaluation deviation framework in a more general way, so that it may also be used to compute the evaluation deviation of groups of users. Let U and U' be two sets of users, let A and B be two sets of concepts in the ontology, and let L be the set of possible labels that can be assigned to elements in $A \times B$. Let the label distance $dist : L \times L \to \mathbb{R}^+$ be a function that assigns to each pair of labels a non-negative real number. Let $E = \{(a, b, l, u) \mid a \in A, b \in B, l \in L, u \in U\}$ be the set of 4-tuples representing the evaluations of users in set U and correspondingly let $E' = \{(a, b, l', u') \mid a \in A, b \in B, l' \in L, u' \in U'\}$ represent the evaluations of users in set U'. Note that here user evaluations are assignments of labels in L to elements in $A \times B$ by the users in U and U'.

We define the evaluation deviation measure $Dev : U \to \mathbb{R}^+$ as

$$Dev(u) = \frac{1}{|N(u)|} \sum_{(a,b,l,u) \in E} \sum_{(a,b,l',u') \in E', u \neq u'} dist(l, l') \tag{4.1}$$

with overlap $N(u) = \{(a, b, l', u') \in E' \mid \exists (a, b, l, u) \in E \text{ with } u' \neq u\}$. To also measure the quality of the evaluations of groups of users we use the mean of the deviations of the individual users in the groups. The *group deviation* is defined as:

$$Dev(U) = \frac{1}{|U|} \sum_{u \in U} Dev(u) \tag{4.2}$$

with U being the group of users to be compared against the reference group U'. We prefer this measure over standard correlation approaches since it more intuitively reflects the relative degree of disagreement among groups of users and since it is more easily adaptable to different distance measures.

Minimum overlap	$i = 1$	2	3	4	5
Number of Pairs	3,237	1,154	370	187	92

Table 4.1: Number of distinct pairs that were evaluated by at least i InPhO users.

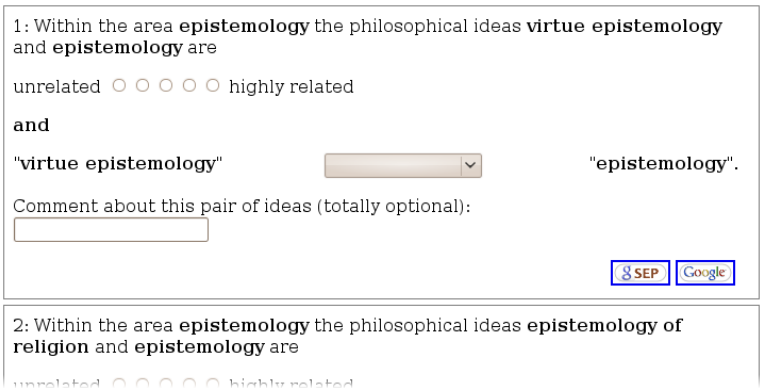


Figure 4.1: The presentation of a pair in a HIT.

Telling the Good from the Bad

In this section we describe some strategies that support the assessment of a worker's response quality when no or only a small set of gold standard pairs is available. Possible factors influencing the feedback quality are (a) the time a worker has spent on a specific task and (b) the quality of the worker's feedback on a small set of gold standard pairs included in each HIT. Each of the presented strategies is evaluated by comparing the *group deviation* between 13 selected InPhO experts and MTurk workers selected through the application of certain filters.

Working Time. Here, the underlying idea is that the more time the workers spent on average on the tasks the higher the quality of their feedback. We hypothesized that there exists a group of workers who provide quick random responses in order to maximize their monetary gain while risking the potential disapproval of their HITs. To test this hypothesis, we filtered the set of MTurk workers according to the average time needed for completing a HIT.

Hidden Gold Standard. We placed a small set of diagnostic pairs in each HIT and used the worker's performance on those pairs to assess the quality of the worker's responses. To ensure comparability between all obtained responses, we decided to include the same four concept pairs in every HIT. This means that users who answered more than one set encountered these pairs repeatedly in each set.

Minimum HITs	80	30	10	5	2
Number of users	3	6	13	24	41

Table 4.2: The number of Mechanical Turk users who completed at least a certain number of HITs.

To disguise this we inserted the pairs in each set at random positions. The chosen concept pairs and the corresponding correct responses are:

SOCIAL EPISTEMOLOGY - EPISTEMOLOGY (P_1): Related concepts; social epistemology is more specific than epistemology.

COMPUTER ETHICS - ETHICS (P_2): Related concepts; computer ethics is more specific than ethics.

CHINESE ROOM ARGUMENT - CHINESE PHILOSOPHY (P_3): Unrelated concepts.

DUALISM - PHILOSOPHY OF MIND (P_4): Strongly related; dualism is more specific than philosophy of mind within the area philosophy of mind.

The rationale behind choosing these specific pairs is that the first two pairs should be answerable by everyone without any knowledge about philosophy, only using common sense. The idea is that workers who get one or both of these questions wrong are likely unreliable. The third concept pair presents a more challenging task as some degree of philosophical knowledge is necessary to correctly evaluate this pair. In addition, this is an example where superficial lexical parsing (both concepts contain the term "Chinese") will lead to an incorrect conclusion. Whereas the first two concept pairs evaluate the *lexical performance* of a worker, the third concept pair presupposes *semantic knowledge*. The same is true for the fourth pair. Both concepts are highly related, though the relative generality between these two terms is not obvious. Since we ask users to evaluate the pair *relative to the philosophical area philosophy of mind*, the correct response is that dualism is more specific than philosophy of mind.

4.1.3 Results

The 720 HITs were completed in 19.7 hours. The average time that a Mechanical Turk user needed for one HIT (12 pairs) was 178 seconds. This means an average hourly rate of 3.25 US$, which is above the average remuneration on MTurk of just under 2 US$ (Ross, Irani, Silberman, Zaldivar, & Tomlinson, 2010). There were 87 distinct users that completed on average 8.3 HITs. Table 4.2 shows the distribution, how many users completed at least a given number of HITs. There was only one user who completed all 144 HITs.

Measuring Agreement

As described above, we required users to evaluate a given pair regarding two different aspects: The relatedness of the terms (from unrelated to strongly related) and their relative generality (more specific than, more general than, same generality, incomparable/either).

For the labels used to describe the relatedness we define the distance function as

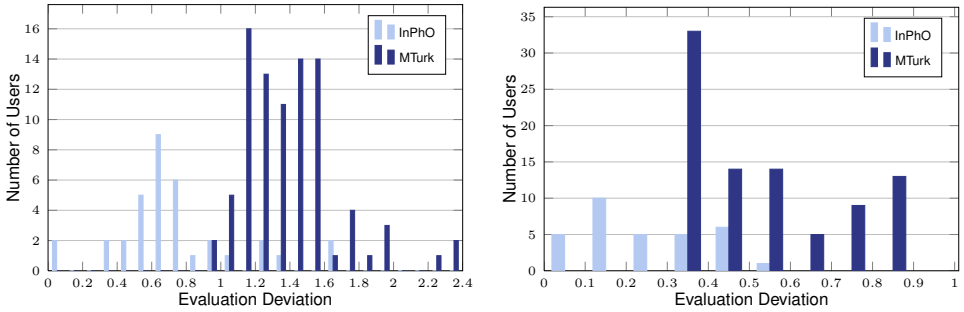
$$dist(l, l') = |l - l'|,$$

where the labels l, l' range from 0 (unrelated) to 4 (strongly related). For stability reasons, we only calculated the deviation for users with an overlap size $|N(u)| \geq 10$. For the relative generality evaluations, we have a set of four independent Labels $L = 0, 1, 2, 3$ with 0="more specific," 1="more general," 2="same generality," and 3="incomparable/either more or less general." For relative generality evaluations, we define the distance function as

$$dist(l, l') = \begin{cases} 0 & l = l' \\ 1 & l \neq l'. \end{cases}$$

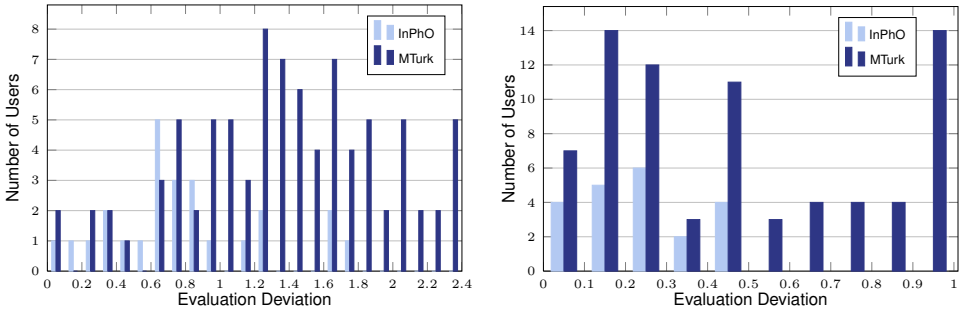
Inter-group Agreement. We use Equation 4.1 with $U = U'$ to compute the inter-group deviation once for the InPhO users and once for the Mechanical Turk users. Figure 4.2 shows the results for the deviation on the relatedness and relative generality evaluations. Since we required an overlap of at least 10 concept pairs, we compared 35 InPhO users with each other. The number of 87 MTurk workers did not need to be adjusted, as we ensured in the experimental setup an overlap of every user with at least four other users over 12 pairs ($|N(u)| \geq 48$). The result shows that the MTurk workers perform considerably worse than the InPhO users regarding their internal agreement on the correct answers for the given pairs. This implies that the answers were not as consistent as the answers given by the InPhO community, possibly indicating that the MTurk responses are of highly-variable quality.

Comparison with Experts. Measuring the quality of answers is not an easy task, as the relation of terms and the perception of relatedness is very subjective and even human experts only agree up to a point on the correct answer. In our setting, we have the experts of the InPhO system and we can use their feedback as a de-facto gold standard. We singled out a set of 13 experts, all of whom have published in their area of philosophy, and used this set as the gold standard for all subsequent evaluations. Figure 4.3 shows the histograms of evaluation deviations, this time with the experts forming the reference set. Of course, these expert users were removed from the InPhO users set. It can be seen that the deviation from the



(a) Relatedness evaluation (b) Relative generality evaluation

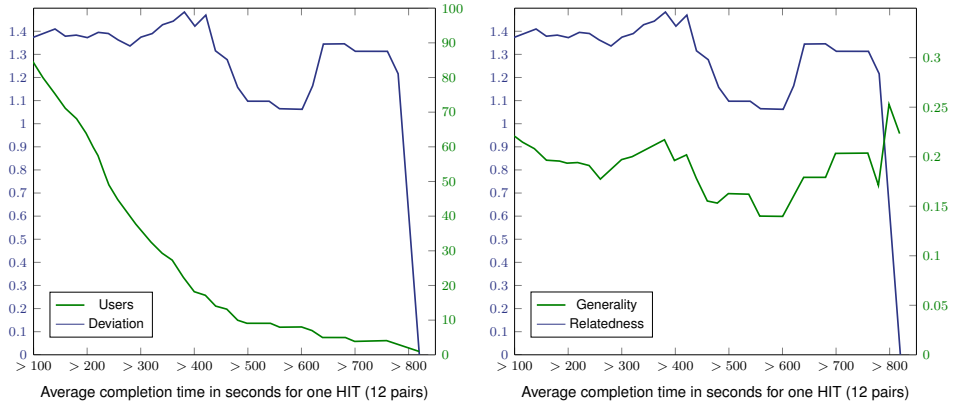
Figure 4.2: Histogram of inter-group deviations.



(a) Relatedness evaluation (b) Relative generality evaluation

Figure 4.3: Histogram of deviations from reference group of experts.

experts' answers, and thus the quality of the answers, is much more variable within the group of MTurk workers than within the group of InPhO users. This is not surprising, as the InPhO users already showed a higher consistency in their answers when the experts' answers still were included. It is promising, however, that there is a large number of Mechanical Turk users who perform in the same deviation range as the InPhO users. The results for the relative generality (Figure 4.3b) look even better. Probably due to the categorical "right or wrong" definition of the distance, the histogram curve is not as smooth as for relatedness. Instead we have a clear distinction between a set of users who performed well and a large set of poorly performing ones, as a deviation of 1 means there is complete disagreement with all the expert users' feedback facts.



(a) Number of users and the resulting deviation for the relatedness evaluation.

(b) Deviation results for relatedness and relative generality evaluation.

Figure 4.4: Results filtered by working time.

Telling the Good from the Bad

The results above indicate that we could achieve a high quality if we were able to distinguish between "good" and "bad" MTurk workers. In the following, we test the two proposed strategies of Section 4.1.2.

Working Time. The first approach is based on the hypothesis that unreliable users take less time to think about their answers. Thus we try to filter out users based on their average completion time for a single HIT. Figure 4.4a shows the impact of this filter on the number of excluded workers and the resulting variations in group deviation values compared to the expert reference set. The results show that the completion time is not a good feature for assessing user feedback quality.[3] Both relatedness and relative generality (Figure 4.4b) stay roughly at the same level. The graph also demonstrates that the quality of the responses for relatedness and relative generality are correlated (Kendall τ : 0.36, Spearman ρ: 0.52).

Hidden Gold Standard. The most straight-forward way to distinguish between reliable and unreliable performers is by comparing the workers' responses to a set of gold standard concept pairs for which we know the relatedness and relative generality. To facilitate this test, we included four concept pairs ($P_1 \ldots P_4$) into each HIT, as explained in Section 4.1.2. For these pairs, there exist correct answers on which all InPhO experts agreed. We considered the following answers as correct:

[3] Simultaneously with our publication in (Eckert et al., 2010), Downs, Holbrook, Sheng, and Cranor (2010) published results of a screening of MTurk workers and came to the same conclusion regarding the exploitability of time stamps for user evaluation.

	InPhO Users	MTurk Workers
P_1	7/10 (0.70)	52/87 (0.60)
P_2	2/3 (0.67)	50/87 (0.57)
P_3	2/2 (1.00)	20/87 (0.23)
P_4	5/6 (0.83)	32/87 (0.37)

Table 4.3: Number of users who answered the test pairs correctly.

	P_1	P_2	P_3	P_4
P_1	1.00	0.77	0.21	0.50
P_2	0.80	1.00	0.22	0.56
P_3	0.55	0.55	1.00	0.40
P_4	0.81	0.86	0.25	1.00

Table 4.4: Conditional probabilities of correct answers for the test pairs.

P_1: Relatedness ≥ 3 (the two highest levels of relatedness) and relation "more specific than."

P_2: Relatedness ≥ 3 and relation "more specific than."

P_3: Relatedness $= 0$ (unrelated).

P_4: Relatedness ≥ 3 and relation "more specific than."

Table 4.3 lists the number of users who evaluated the given pair correctly, as well as the overall number of users who answered it. We received several answers for each pair for the MTurk workers who completed more than one HIT as the pairs were repeatedly included. To maintain comparability, we only used the worker's response for the first HIT.

It is notable that the InPhO community seems to have more problems with the comparatively easy pairs P_1 and P_2 than with P_3 and P_4. This is probably due to the low number of cases. The Mechanical Turk users perform best for these pairs with roughly 60% of them providing the correct responses. MTurk workers had the most problems with the pair P_3 (CHINESE ROOM ARGUMENT - CHINESE PHILOSOPHY), but performed better on the evaluation of P_4 (DUALISM - PHILOSOPHY OF MIND). To get a better understanding of the dependencies between the four pairs of questions, we calculated the conditional probabilities for the correctness of a pair, given that another pair was answered correctly (Table 4.4).

There is a high probability (around 80%) that, if P_1 is answered correctly then P_2 is also answered correctly and vice versa. As both pairs can be answered correctly by just using some common "lexical" sense, we consider their correct evaluation as a minimum requirement that a user has to fulfill. The probabilities for the hardest pair P_3 are surprising, as answering it correctly does not seem to be a good indicator for the correct response on other pairs (about 50% for each). Pair

Filter	Users	$Dev(U)$	Range $Dev(u)$
$P_1 \wedge P_2 \wedge P_3 \wedge P_4$	7	0.60	0.00 – 1.00
$P_1 \wedge P_2 \wedge P_3$	10	0.87	0.00 – 1.78
$P_1 \wedge P_2 \wedge P_4$	23	0.84	0.00 – 1.41
$P_1 \wedge P_2$	40	1.11	0.00 – 1.96
All MTurk workers	87	1.39	0.00 – 2.96
InPhO users	25	0.77	0.00 – 1.75
Random	—	1.80	—

(a) Relatedness evaluation.

Filter	Users	$Dev(U)$	Range $Dev(u)$
$P_1 \wedge P_2 \wedge P_3 \wedge P_4$	7(5)	0.12	0.00 – 0.22
$P_1 \wedge P_2 \wedge P_3$	10(8)	0.14	0.00 – 0.27
$P_1 \wedge P_2 \wedge P_4$	23(20)	0.15	0.00 – 0.45
$P_1 \wedge P_2$	40(35)	0.21	0.00 – 0.59
All MTurk workers	87(78)	0.45	0.00 – 1.00
InPhO users	21	0.23	0.00 – 0.47
Random	—	0.75	—

(b) Relative generality evaluation.

Table 4.5: Effect of different filters on the set of Mechanical Turk workers.

P_4 possesses a better predictive property, since workers who answered it correctly also answered P_1 and P_2 correctly with a probability of over 80%. Using these findings, we defined several configurations to filter the workers, based on their answer on $P_1 \dots P_4$. Table 4.5 summarizes the results of these experiments. The filter criterion is defined in a Boolean way, with P_i indicating that the response for P_i has to be correct for the worker to pass the filter. We compared the resulting groups both with the performance of the InPhO community and with the performance of a worker who responds at random. Note that there is no evaluation on the relative generality if a user rates a pair as unrelated. Thus the number of workers for whom a deviation can actually be computed is reduced and given in parentheses. The results show that, with the most restrictive filter setting, it is possible to achieve a higher agreement with the experts than the InPhO community. Of course, this comes at the price of sacrificing a lot of the evaluations. Asking for this level of quality would require many more completed HITs to collect the needed number of responses. The simple filter $P_1 \wedge P_2$ significantly improves the quality of the results compared to the whole set of MTurk workers. It is still worse than the InPhO community for the relatedness evaluation but outperforms it slightly for the relative generality (0.21 compared to 0.23). Adding P_3 confirms our hypothesis, based on the conditional probabilities that the users had problems with this pair. Evaluating it correctly, however, does not imply a generally high response quality. With this filter the number of users is reduced to only 10. The result for the filter

Filter	Pairs	Evaluations	C_{Pair}	C_{Eval}
—	1138	5690	0.111	0.022
$P_1 \wedge P_2$	1074	1909	0.117	0.066
$P_1 \wedge P_2 \wedge P_3$	215	215	0.586	0.586
$P_1 \wedge P_2 \wedge P_4$	1018	1558	0.124	0.081
$P_1 \wedge P_2 \wedge P_3 \wedge P_4$	183	183	0.689	0.689

Table 4.6: The number of unique pairs and single evaluations we gather from different sets of users, as well as the costs in US-Dollar per pair and per evaluation.

$P_1 \wedge P_2 \wedge P_4$ shows that this configuration performs even better, while leaving a much bigger set of 23 users.

Financial Considerations

Using MTurk within an approach means obviously that money is involved. Thus, for a full evaluation of the results we not only have to focus on the feedback quality we can reach but also the financial price we have to pay for it. Table 4.6 lists some figures that illustrates the relationship between different filter settings and the number of obtained concept pairs.

For our whole experiment we paid 126 US-Dollar. This is 0.11 USD per concept pair and 0.02 USD per evaluation. With the lowest filtering we still obtained 1,074 pairs barely increasing the price to 0.12 USD per pair, but the number of usable evaluations was reduced to 1,909. If we would like to have the same amount of redundancy that the experiment was designed for, we would had have to pay about 376 USD. For the highest quality of feedback ($P_1 \wedge P_2 \wedge P_3 \wedge P_4$), the costs for 1,138 pairs are estimated at 784 USD, for 5,690 evaluations we estimate 3,920 USD. Of course, these sums are only estimations, based on the assumption that the coverage of pairs would scale across our whole set of pairs with a proportional increase of HITs.

Constructing the Concept Hierarchy

To apply our answer set program (Niepert et al., 2008) to the data gathered from the MTurk workers, we have to determine, for each of the workers, an expertise level between 0 (no expertise) and 3 (high expertise). The answer set program we have developed for this task considers these expert levels when resolving conflicting feedback facts, as described in Section 4.1.1. We decided to exclude all workers who evaluated all of the gold standard questions incorrectly.

Then, we again used the filter configurations described above to determine the expertise level:

- Users who answered all test pairs correctly ($P_1 \wedge P_2 \wedge P_3 \wedge P_4$) were considered as level 3.
- Users who answered pair 1, 2 and 4 ($P_1 \wedge P_2 \wedge P_4$) correctly were considered as level 2.
- Users who answered only pair 1 and 2 ($P_1 \wedge P_2$) correctly were considered as level 1.

To conclude, with our approach we achieved a feedback quality comparable to that of the InPhO community. The resulting concept hierarchy can be browsed online.[4]

Ethical Considerations

Besides considering the statistical results and hard financial facts, we should remember that the Mechanical Turk is no computer, no algorithm or approach like others developed in computer science. Even if it provides an API that allows a seamless integration into computer systems, the actual work is done by real human beings. Online piece work like MTurk has been criticized as possibly leading to "digital sweatshops," in which the inexpensive labor of citizens from developing countries is exploited to complete menial tasks that others are unwilling to do (Zittrain, 2009). While it is beyond the scope of this thesis to provide a detailed analysis of the social or ethical implications of the use of services like Mechanical Turk, a few preliminary comments seem appropriate. The ethical factors involved in MTurk use for such a project can be organized into two groups: user-level considerations (e.g. pertaining to the well-being of workers) and systemic considerations (e.g. whether MTurk itself encourages unjust or unethical practices). We discuss each in turn, with an eye towards practical advice for other projects.

Let us first consider the welfare of the workers completing the HITs. First, it is not clear that the demographics of MTurk workers support the digital sweatshop narrative. While HIT providers are forbidden by MTurk terms of service from asking demographic questions, a study conducted by Ross, Zaldivar, Irani, and Tomlinson (2009); Ross et al. (2010) has found that a significant majority of MTurk workers reside in the U.S. and many have relatively high household incomes. A minority of MTurk workers are citizens of developing nations and an even smaller minority depend upon MTurk for a significant portion of their income. Many users reported that they complete MTurk tasks as a diversion, suggesting that the tasks themselves are not as onerous as one might suppose. Secondly, for the minority of users who *do* live in developing countries and depend on MTurk for primary income, one might compare the wages and conditions of MTurk tasks to other employment opportunities locally available to these users. No forms of coercion

[4] http://www.kaiec.org/2012/dissertation/amt-inpho

other than payment are directed towards the workers and workers can freely choose their work hours and conditions.

A study conducted by Horton (2011) found that MTurk workers reported finding MTurk employers as fair as or more fair than local employers (though there are serious issues with the sample in this study, given that MTurk was itself used to conduct the experiment). One persistent worry related to worker exploitation is that employers can, at their discretion, opt to reject HITs and not pay users (while possibly still making use of the data), and users have no ability to appeal this decision. Users can, however, see HIT provider rejection rates before accepting a task and web sites have sprung up to evaluate HIT providers – so MTurk workers may peruse reviews of tasks completed by other users before choosing to participate. Where applicable, we recommend that employers warn users that some form of quality control will be used to evaluate HIT responses before compensation will be provided (though providing specific information about the controls would of course erode their utility).

Another more systemic concern is that users do not know what their work will be used for and some providers have used MTurk for nefarious ends such as writing fraudulent product reviews. While much of this should be settled by better screening of HITs by Amazon, we recommend that employers give users some idea as to the ends to which their responses will be put.

4.2 KOS Extension using Web Search Engines

The life cycle of a KOS does not end with the creation. There are constantly new concepts evolving that have to be included and existing ones have to be refined or extended to reflect the terminology that is actually used in the literature. Like with the creation, the manual maintenance of comprehensive KOSs is hardly feasible in fast changing domains and especially outside the libraries. Especially automatic indexing approaches are reliant on the existence of proper synonyms to identify concept occurrences in texts. For instance, the outbreak of the H1N1 pandemic has recently sparked numerous media and research reports about the *swine flu*. At that point the term "swine flu" was not included in any of the major medical thesauri because it was only recently coined by the media. The current version of the MeSH thesaurus lists the term "Swine-Origin Influenza A H1N1 Virus" as a synonym for INFLUENZA A VIRUS, H1N1 SUBTYPE but not the more commonly used term "swine flu."

In this section we describe a possible approach to the problem of identifying important terms in text documents and semi-automatically extending thesauri with novel concepts. The proposed system consists of three basic parts each of which we will briefly motivate by means of the swine flu example.

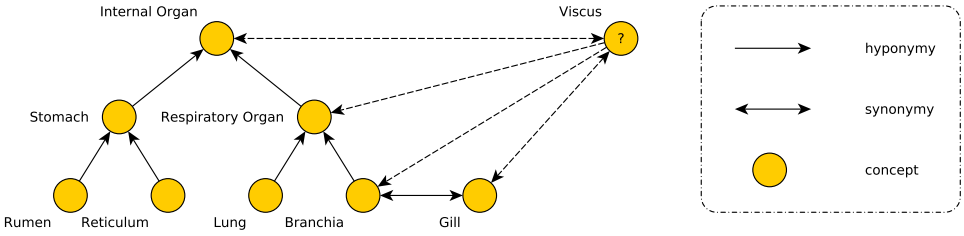


Figure 4.5: Fragment of WordNet.

1. In a first step, we identify candidate terms to be included in the thesaurus. In our example this is the case for swine flu as many existing documents discuss the different aspects of swine flu, including its origin, treatment, and impact on the economy.

2. Once we decide that the term "swine flu" should be included in the thesaurus, we have to identify a location that is most appropriate. This step requires a deeper understanding of the concept SWINE FLU since we want to place it in the disease branch and not the animal branch of the thesaurus. In particular, the term should be classified next to the concept INFLUENZA A VIRUS, H1N1 SUBTYPE.

3. After deciding to place "swine flu" close to INFLUENZA A VIRUS, H1N1 SUBTYPE one still needs to determine the relation of the two concepts. In particular, we have to decide whether the new term should be regarded as a synonym or whether it should be included as a concept of its own - either as hyponym or hypernym or whether the similarity of the two terms was incidental.

4.2.1 Method Description

Let us assume we are given a KOS that needs to be extended with novel concepts. The process of this extension can be divided in two major phases. First, concept candidates have to be extracted from document collections and other textual content. To achieve satisfying results it is necessary that the text corpora under consideration are semantically related to the concepts in the KOS. For instance, if we want to extend a KOS of medical terms we would have to choose a document collection covering medical topics. Given a set of candidate terms, the second step of KOS extension involves the classification of these candidates as either synonyms or hyponyms of already existing concepts.

Figure 4.5 depicts a typical instance of the KOS extension problem. The candidate term "Viscus," which has been extracted from a text corpus, needs to be positioned in the existing hierarchy. Our approach provides a small ranked list of potential new terms together with suitable positions in the KOS.

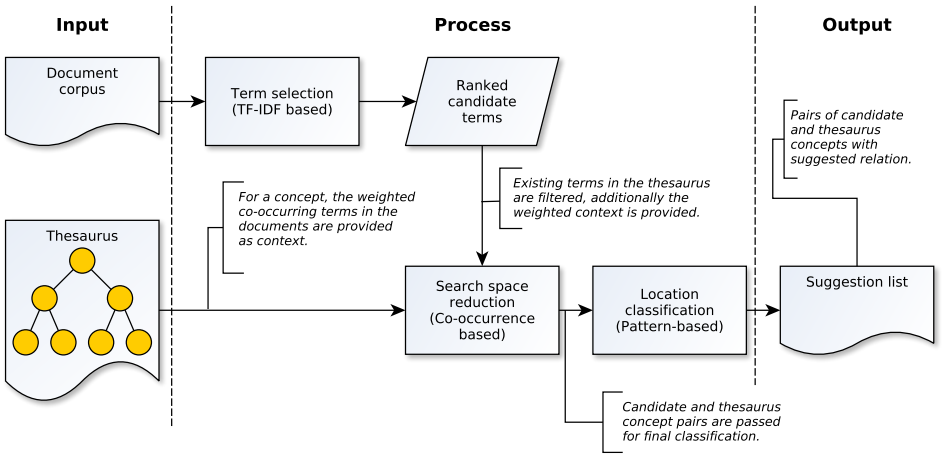


Figure 4.6: KOS extension workflow.

We propose a method supporting the KOS modeler during both of these phases by (a) extracting terms from text corpora using a novel extraction method based on the *tf-idf* measure (Salton & Yang, 1973) and (b) by generating, for each of the extracted candidates, a reasonable sized set of suggestions for its position in the hierarchy. For the latter, we distinguish between synonymy and hyponymy relationships. Figure 4.6 depicts a workflow of the proposed method.

Term Selection

Term selection is the process of extracting terms that are candidates for new concepts in the hierarchy or additional synonyms for existing concepts. To quantify the importance of a term t in a corpus D we first compute the *tf-idf* value $w_{t,d}$ of term t in document d:

$$w_{t,d} = \begin{cases} (1 + \log tf_{t,d}) \cdot \log \frac{|D|}{df_t} & tf_{t,d} > 0 \\ 0 & tf_{t,d} = 0 \end{cases} \tag{4.3}$$

with term frequency $tf_{t,d}$ denoting the number of occurrences of term t in document d and document frequency df_t denoting the number of documents in D that contain term t.

To compensate for different lengths of the documents, the weights are cosine normalized:

$$w_{t,d}^{norm} = \frac{w_{t,d}}{\sqrt{\sum_{t' \in d} (w_{t',d})^2}} \tag{4.4}$$

Since we want to assess the importance of a term t not only for a single document but the entire corpus, we compute the mean $\overline{w}_t$ of the *tf-idf* values over all documents in which term t occurs at least once:

$$\overline{w}_t = \frac{\sum_{d \in D} w_{t,d}^{norm}}{df_t} \tag{4.5}$$

We finally assign the importance weight $\hat{w}_t$ to term t by multiplying the squared value $\overline{w}_t$ with the logarithm of the document frequency df_t.

$$\hat{w}_t = \overline{w}_t^2 \cdot \log(df_t + 1) \tag{4.6}$$

The intuition behind this approach is that terms that occur in more documents are more likely to be concept candidates for a KOS covering these documents. The presented importance measure $\hat{w}_t$, therefore, combines the average importance of a term relative to each document in the corpus with the importance of the term relative to the entire corpus.

Search space reduction

The principle idea of our approach is the classification of candidate terms as synonyms or hyponyms of existing concepts. As most KOSs are comprised of a large number of concepts and, for every candidate term, we would have to send a query to a web search engine for every of the existing concepts, we have to reduce the amount of potential positions for any given candidate term.

To achieve such a search space reduction we compute for every candidate term that needs to be classified its similarity to each of the concepts using the weighted Jaccard similarity – cf. Grefenstette (1994, p. 47) – based on the terms in the context according to the following definition:

Definition 4.1 (Context of a term) *Let D_t be a subset of the corpus D that contains documents d that are associated with a term t. Let a document context $C_{t,d}$ be a set of terms that appear close (e.g. within a windows of 100 words) to term t in $d \in D_t$. We then define the context C_t of term t as the union of all document contexts: $C_t = \bigcup_d C_{t,d}$. The terms forming the context are ranked by $\hat{w}_t$ (Equation 4.6).*

The weighted Jaccard similarity s of two terms a and b is then calculated as follows:

$$s = \frac{\sum_{C_a \cap C_b} \min(\hat{w}_a, \hat{w}_b)}{\sum_{C_a \cup C_b} \max(\hat{w}_a, \hat{w}_b)} \tag{4.7}$$

As we deal only with abstracts in our experiments for this approach, we deliberately define $C_{t,d}$ as all terms that occur in the same abstract, i.e. $C_{t,d} = \{t : t \in d\}$. D_t is determined as (1) all documents that contain t for the candidate terms and as (2) all documents that are associated with a concept for the existing concepts.

Then, for each candidate term, only the top-k most similar concepts are considered for further classification, as described in the next section.

Location classification

In a second step, the previously extracted concept candidates are *classified* in the existing KOS. Classification is the process of finding concepts that are potential hypernyms and synonyms, respectively, for each of the candidate concepts.

We apply established machine learning approaches to learn lexico-syntactic patterns from search engine results. Typical patterns for concepts c_1 and c_2 are, for instance, [c_1 *is a* c_2] for hyponymy and [c_1 *is also a* c_2] for synonymy relationships. Instead of only using a predefined set of patterns (Hearst, 1992), we learn these patterns from text snippets of search engines using existing thesauri as training data. The learned patterns are then used as features for the classification of the relationship between each concept candidate and existing concepts. Since we are mainly interested in hyponymy and synonymy relationships, we need to train at least two different binary classifiers. Fortunately, the classifiers can be trained with concept pairs contained in the existing KOS. If the KOS is not yet elaborated enough, concept pairs of any other – possibly related – KOS can be used as well.

The pattern extraction approach of the proposed system is based on the method presented by Bollegala, Matsuo, and Ishizuka (2007). Instead of retrieving lexico-syntactic patterns to assess the semantic similarity of term pairs, however, we also extract the patterns to classify relationships as either synonymy or hyponymy. For each pair of concepts (c_1, c_2) of which we know the relationship because it is contained in a training KOS, we send the query "c_1" +"c_2" to a web search engine. The returned text snippet is processed to extract all n-grams ($2 \leq n \leq 6$) that match the pattern "$c_1 X c_2$," where X can be any combination of up to four space-separated word or punctuation tokens.

For instance, assume the training KOS contains the concepts CAR and VEHICLE with car being a hyponym of vehicle. The method queries a search engine with the string "car" + "vehicle." Let us assume that one of the returned text snippet is "every car is a vehicle." In this case, the method extracts the pattern "car is a vehicle." This pattern is added to the list of potential hyponymy patterns with "car" and "vehicle" substituted with matching placeholders. The set of patterns extracted this way is too large to be used directly for machine learning algorithms. Therefore, we rank the patterns according to their ability to distinguish between the types of relationships we are interested in.

For both the synonymy and hyponymy relationship we rank the extracted patterns according to the chi-square statistic (Bollegala et al., 2007). For every pattern v we determine its frequency p_v in snippets for hyponymous (synonymous) word pairs and its frequency n_v in snippets for non-hyponymous (non-synonymous) word pairs. Let P denote the total frequency of all patterns in snippets for hyponymous (synonymous) word pairs and N the total frequency of all patterns in snippets for non-hyponymous (non-synonymous) word pairs. We calculate the chi-square value for each pattern v as follows (Manning & Schuetze, 1999):

$$\chi^2_v = \frac{(P+N)(p_v(N-n_v) - n_v(P-p_v))^2}{PN(p_v+n_v)(P+N-p_v-n_v)} \tag{4.8}$$

Based on this ranking, we choose the highest ranked patterns – in our experiments, between 60 and 80 turned out to be reasonable good number of patterns. For each concept pair, we build feature vectors consisting of the normalized frequencies for these top-ranked patterns. A machine-learner is trained on the existing concepts in the KOS to be extended that is finally used to classify the candidate terms and their respective candidate positions. Based on this classification, the final list of suggestions is compiled.

4.2.2 Experimental Setup

KOSs and Document Sets. To evaluate and test our methods we use a KOS extracted from the 2008 Medical Subject Headings (MeSH, cf. Section 1.5). The KOS was created by combining all concepts located under the top-level concept *anatomy* with all concepts located under the top-level concept *humanity* and contains 1,797 concepts. For each concept in this KOS we retrieved the most relevant documents from PubMed[5] of the years between 2005 and 2008. The final document set includes 14,860 documents. Additionally, we conducted our experiments with WordNet 3.0 as an example for a broader, less populated thesaurus.

Pattern extraction. We used the Yahoo search engine API,[6] mainly because it has low restrictions on the allowed number of queries per day. A single query with the API took up to three seconds. From the initially extracted set of patterns we kept only the 60 highest ranked patterns extracted with MeSH as training thesaurus and the 80 highest ranked patterns with WordNet.

For each of the three classes "synonymy," "hyponymy," and "neither synonymy nor hyponymy" we sampled 300 pairs of concepts belonging to the respective class. For the MeSH training set, these pairs were randomly sampled from the MeSH

[5] http://www.ncbi.nlm.nih.gov/pubmed/

[6] http://developer.yahoo.com/

thesaurus excluding the previously constructed anatomy/humanity sub-thesaurus. Similarly, to create the WordNet training set, we randomly sampled 300 negative and positive training pairs for each class from WordNet.

Machine Learning. The experiments are conducted with a decision tree learner (J48). Results for both a linear support vector machine (SVM) and a SVM with radial basis functions are reported by Meusel, Niepert, Eckert, and Stuckenschmidt (2010). Without further tuning of the SVM parameters, however, the decision tree performs better and is preferred here due to its simpler approach and better understandability. We conducted four different classifications for each pair of concepts:

synonym vs. no synonym: Binary classifier to determine if c_1 is a synonym of c_2.

hyponym vs. no hyponym: Binary classifier to determine if c_1 is a hyponym of c_2.

synonym vs. hyponym: Binary classifier to distinguish between synonym and hyponym.

synonym vs. hyponym vs. none: Tertiary classifier to assign in one step the appropriate class.

4.2.3 Results

To evaluate our approach, we isolated 100 concepts each from the anatomy/humanity sub-thesaurus and from WordNet. These concepts serve as candidate concepts and the goal is to evaluate whether our approach can identify their correct positions.

For both the 100 MeSH and WordNet candidate concepts we determined the top 100 most similar concepts in the MeSH and WordNet thesaurus, respectively, by applying the co-occurrence similarity measure.

Search space reduction. On average, 97 percent of the correct positions for each candidate concept were included in this set for WordNet and 90 percent for the MeSH thesaurus. This indicates that the Jaccard similarity measure is able to exclude the majority of all concept positions while retaining most of the correct positional concepts.

Location classification. For each of the 100 concept candidates, we applied the trained classifier on the set of the previously ranked 100 most similar concepts, resulting in 10000 classification instances for each classification task.

The accuracy values – $\frac{\text{true positives + true negatives}}{\text{all instances}}$ – of the classification results are shown in Table 4.7. Evidently, the accuracy of the classifiers is strongly influenced

Thesaurus	Classification task	Accuracy
WordNet	synonym vs. no synonym	98 %
	hyponym vs. no hyponym	82 %
	synonym vs. hyponym	71 %
	synonym vs. hyponym vs. none	70 %
MeSH	synonym vs. no synonym	85 %
	hyponym vs. no hyponym	87 %
	synonym vs. hyponym	68 %
	synonym vs. hyponym vs. none	68 %

Table 4.7: Accuracy of the classification.

by the properties of the thesauri. For instance, for the synonymy classification task, we achieved an accuracy of 98 percent for WordNet but only an accuracy of 85 percent for the MeSH thesaurus. Not surprisingly, the three-class classification problem is more difficult and the approach is not as accurate as for the binary classification tasks. An additional observation is that the classification results for the hyponymy vs. synonymy problem are rather poor given the semantic similarity of the synonymy and hyponymy relations. However, the distinction between synonyms and hyponyms in a KOS is often left to the maintainer who deliberately might choose to add hyponyms as quasi-synonyms to a concept to avoid extensive subclassing.

The main application of the support system is to locate the correct position of the candidate concepts in the hierarchy. For instance, consider we have to determine the position of the concept candidate “tummy” in the KOS fragment depicted in Figure 4.5. Two pieces of information will lead us to the correct location: “tummy” is a hyponym of “internal organ”; and “tummy” is a synonym of “stomach.”

We evaluated for each concept candidate, in how many cases we were able to determine the correct position in the target thesaurus. Hence, for each concept candidate, we looked at the set of concepts in the thesaurus which the pattern-based approach classified as either synonyms or hyponyms and checked whether at least on of these concepts led us to the correct position. The size of this set was 14 on average, meaning that, on average, the number of choices was reduced from 100 to 14, a number that is digestible for a KOS maintainer. Table 4.8 lists the percentage of cases for which we could determine a position for the candidate within a certain distance from the correct one, where the graph distance 1 represents direct synonymy or hyponymy relations. This means that a distance of 1 is the best possible result, as one edge is always needed to relate the candidate to the correct concept. The suggested position was at most 4 edges away from the correct one. All in all, this confirms that the approach presented in this section indeed can support the task of identifying and positioning new terms to enhance existing KOSs.

Graph distance	Correct
1	85%
2	95%
3	99%
4	100%

Table 4.8: Accuracy of the position. Fraction of candidate concepts for which the correct position in the thesaurus could be inferred using the pattern-based classification results; considering a graph distance of $1 \leq n \leq 4$.

While a positioning at or near the original location indicates that the approach works, it cannot be concluded that a remote position is wrong. The following example, taken from (Grefenstette, 1994, Section 4.3) illustrates this problem: A KOS creation process identified a (correct) similarity between *administration* and *injection* in a medical corpus. Merriam-Webster's[7] dictionary lists the following definitions, which have no words in common:

> ADMINISTRATION: 1. performance of executive duties: management; 2. the act or process of administering; 3. the execution of public affairs as distinguished from policy-making; 4. a) a body of persons who administer, b) often capitalized: a group constituting the political executive in a presidential government, c) a governmental agency or board; 5. the term of office of an administrative officer or body.
>
> INJECTION: 1. a) an act or instance of injecting, b) the placing of an artificial satellite or a spacecraft into an orbit or on a trajectory; also: the time or place at which injection occurs; 2. something (as a medication) that is injected; 3. a mathematical function that is a one-to-one mapping – compare bijection, surjection.

Grefenstette further shows that both Roget's thesaurus (from 1911) and another thesaurus from the University of Macquarie in Australia list concepts for *Injection* and *Administration* under distinct topic headings. This is of course not a mistake of the mentioned sources, but shows that an existing KOS cannot easily be used as a gold standard to evaluate another KOS. There simply is not only one correct position for a concept. Only the maintainer can decide if a suggested location is appropriate or not.

4.3 Concept Splitting and Naming

The introduction of new concepts has not only to be triggered by new terms that are found in the documents. Sometimes, concepts should be split because too many

[7] http://www.merriam-webster.com/

documents are described by them and a more precise description is indicated. In this section, we are concerned with such a splitting of concepts into new subconcepts. Especially for classifications, where the classes are usually distinct, the problem of splitting a concept into useful subconcepts is akin to the problem of clustering a set of documents into useful clusters and finding a suitable name for each cluster to define the new subconcept. We again rely on the human maintainer of the classification and include two interaction steps in the process that allow the maintainer to influence the result and subsequently get better recommendations for new subconcepts.

In short, we use a straight-forward clustering algorithm to cluster the documents based on their content. The following naming step employs simple *tf-idf* weighting. The contribution of our approach is the combination of these steps with a first preparatory step that extracts meaningful terms from the documents to be clustered which are used to "push" the clustering in the desired direction. This can be seen as a variant of the *description-comes-first* (DCF) paradigm (Osinski, Stefanowski, & Weiss, 2004) that states that it might be preferable to find descriptive cluster labels before the documents are clustered, i.e., assigned to these labels. We believe that this approach in its pure form misses the opportunity to use the strength of clustering approaches to find similarities between documents even if synonymous terms are used for the same concepts. Thus, we use the best of both worlds.

4.3.1 Method Description

Our approach comprises three steps (Figure 4.7): First, we let the maintainer determine the desired number of new subconcepts by means of term-based suggestions (Step I). Next, the documents are clustered based on their contents, yet biased by the predetermined subconcepts (Step II). At last, the clusters are presented to the maintainer as suggestions for new subconcepts together with name suggestions based on the predetermined term clusters (Step III).

Term-based Cluster Preselection (Step I)

The motivation for Step I is twofold: On the one hand, prior experiments showed that generally cluster algorithms using an a-priori defined number of target clusters perform better for our purpose (Stork, 2010). So we need this step to determine the desired number of clusters. On the other hand, we that way incorporate the DCF paradigm, which has an additional advantage: For the maintainer, it is easier to evaluate and select possible clusters based on a limited set of terms than on the actual content of the documents in the clusters.

To identify meaningful terms within the documents to be split, we use a weighting scheme based on *tf-idf*. The modification solely lies in the definitions of the

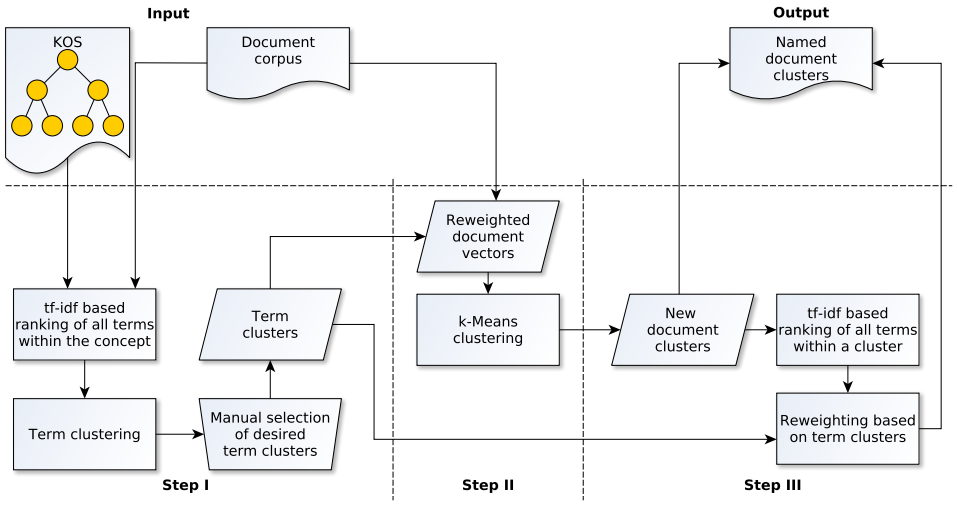


Figure 4.7: Concept splitting workflow.

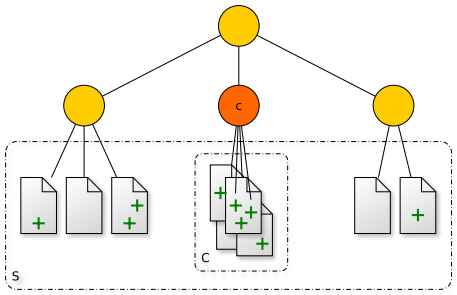


Figure 4.8: Weight calculation: $|S| = 6$, $df_{t,S} = 4$, $tf_{t,C} = 5$.

document sets employed. First of all, we merge all documents to be split that belong to concept c into one large document, denoted as C. We further define S as the set of all documents that belong to sibling concepts, plus the artificial document C (Figure 4.8). The weight for each occurring term t in C is then calculated straight-forward:

$$w_{t,C} = tf_{t,C} \cdot \left(1 + \log \frac{|S|}{df_{t,S}}\right) \tag{4.9}$$

with term frequency $tf_{t,C}$ denoting the number of occurrences of term t in C and document frequency $df_{t,S}$ denoting the number of documents inside S that contain term t.

With this approach, we identify meaningful terms describing the broad, overall topic of the concept, as well as terms that are meaningful, but not representative

for the whole concept. The latter are the interesting terms that can be used to discriminate between clusters and possibly account for new subconcepts.

To support the maintainer in a proper term cluster selection, the top n highest weighted terms have to be pre-clustered. n is configurable and depends on the setting; we used $n = 50$, because we expect at most 5 new subconcepts and do not want to deal with more than 10 terms per cluster. The clusters are created simply based on co-occurrence of the terms in the documents. Therefore, we create a term-relationship matrix T where each element contains the document overlap between two terms, D_t being the set of documents containing term t:

$$T_{i,j} = \frac{|D_i \cap D_j|}{\min(|D_i|, |D_j|)} \tag{4.10}$$

T is transformed into a binary matrix using a configurable threshold where 1 indicates that both terms belong to the same cluster. We used 0.5 as a starting point, the adjustment of the threshold directly affects the number of the resulting term clusters and is an intuitive means for the maintainer to influence the clustering result.

The resulting term clusters are finally presented to the maintainer, who may merge obviously related clusters, remove terms which are out of place in a cluster, or disregard entire clusters as desired.

Content-based Clustering (Step II)

The result of Step I that forms the basis for Step II are k term clusters containing a total of m terms. From the term clusters, k initial document clusters are built, where each cluster is comprised solely of documents containing at least one term of a term cluster and no term of any other cluster. The remaining documents form an additional cluster. The following content-based clustering is performed on these $k + 1$ initial clusters.

The documents are generally represented by term vectors with standard *tf-idf* weighting. As we consider the preselected terms to be more important, we increase their value by setting $df = 2$ – this is the lowest occurring document frequency, as terms that occur only in a single document do not affect the clustering and are therefore removed.

The actual clustering is performed with k-Means, despite two requirements of this algorithms that often cast its application into doubt: the number of target clusters has to be defined beforehand and the result depends on the choice of initial seed points. Both requirements are met in our case using the results from Step I: we use the number of selected term clusters – plus one outlier cluster – as the specified number of output clusters; instead of single seed points we use pre-initialized

clusters, based on documents solely containing terms from one term cluster. After the maintainer's curation of term clusters, these initial clusters can reasonably be expected to be thematically homogeneous.

With these provisions and the increased weight of the m preselected terms, we ensure that the clustering result is in line with the input of the maintainer, while we still harness the benefits of a content-based clustering approach.

Cluster Naming (Step III)

In the last step, the term-based clusters and the content-based clusters have to be combined to generate the final suggestions for the maintainer. First, we calculate the most meaningful terms in each *cluster*, using the same approach as described in Step I (Equation 4.9). The *tf-idf* values calculated in this manner are sorted in descending order and a new term list is built for each cluster. The number of considered terms depends on the number of terms in the corresponding term cluster, which was used to initialize the cluster in question. This list of new terms is presented to the maintainer in combination with the list of original terms in form of a *diff* visualization, i.e. both new and dropped terms are highlighted and displacements are marked.

With this visualization, it is easy to evaluate the final clusters and the maintainer is able to judge, whether or not the content-based clusters are created as expected. From these clusters, the resulting subconcepts can directly be created, using one or more labels from the proposed term list, or the maintainer provides a better-suited, possibly superordinate term. The documents of the cluster are assigned to the new subconcept directly.

4.3.2 Experimental Setup

Due to the lack of publicly available classifications with full texts, we used the *20 newsgroups* collection, which is a popular dataset for the evaluation and testing of clustering algorithms (cf. Section 1.5). Our experiments are based on the version of Jason Rennie[8] with duplicates and most headers removed. The newsgroups are organized in a hierarchy, creating a classification where each newsgroup forms a concept.

For this test, we create an artificially broad concept by merging the newsgroup messages (i.e., our documents) of all groups below SCIENCE (sci.*): SPACE, MEDICAL, CRYPTOGRAPHY, and ELECTRONICS, amounting to $2,373$ documents. The remaining 16 groups ($8,941$ documents) in the collection found above are viewed

[8] http://people.csail.mit.edu/jrennie/20Newsgroups/

Cluster 1	Cluster 2	Cluster 3	Cluster 4
clipper, encryption, key, chip, crypto, privacy, data, security, information, keys, des, algorithm, system, cryptography, escrow, public, ripem, government, available, secret, pgp, nsa, rsa, people, announcement, wiretap, secure, encrypted	tapped, code	space, nasa, launch, shuttle, spacecraft	moon, lunar
858	34	187	43

Table 4.9: Term clusters and number of associated documents, as presented to the maintainer.

as sibling concepts. The task is to use our method to cluster the documents belonging to the artificial SCIENCE concept. The original classification based on the four subgroups is used as gold standard to evaluate the results.

4.3.3 Results

The first result presented to the maintainer are the term clusters of the 50 highest weighted terms from the artificial SCIENCE concept, together with the number of associated documents (Table 4.9). The remaining 12 terms ("db," "don," "health," "medical," "orbit," "patients," "program," "research," "sci," "science," "technology," and "time") are not related to any other term, they are presented to the maintainer as additional terms. In this test, we expect the maintainer to conduct the following refinements to define the result of Step I:

- remove "announcement," "public," "people," "system," "available," "information," and "data" from Cluster 1, which is concerned with cryptography;
- merge Cluster 4 into Cluster 3, as both are dealing with space;
- discard Cluster 2; and
- add term "orbit" (amongst the remaining 12 terms) to Cluster 3.

The remaining term clusters 1 and 3 are used to create the initial clusters as input for the content-based clustering (Step II). Based on the k-Means clustering, the next result is presented to the maintainer: the document clusters, together with meaningful terms that can be used for the final naming of the desired clusters. As the term clusters already contained terms selected by the maintainer, these final terms are presented for an efficient review: terms that already belonged to the first term cluster are marked in bold, the number in brackets indicates the difference in position in the sorted terms; new terms are in plain text; and terms that appear in

Cluster	# of documents	Precision	Recall	F1-measure
1	563	**0.961**	0.909	0.934
3	452	**0.969**	0.739	0.838

Table 4.10: Evaluation of the resulting clusters.

the first term cluster but not in the new list are striked-through.

Cluster 1: **clipper**($\pm$0), **encryption**($\pm$0), **key**($\pm$0), **chip**($\pm$0), **security**(+2), **keys**(+2), **privacy**(-1), **des**(+1), **crypto**(-4), **cryptography**(+1), **algorithm**(-1), **nsa**(+5), ~~rsa~~, ~~wiretap~~, **secret**(+2), **escrow**(-2), **government**(+3), **ripem**(-3), tapped, **secure**(+2), code, **pgp**(-4), announcement, ~~encrypted~~

Cluster 3: **space**($\pm$0), **launch**(+3), **moon**(+1), **lunar**(+2), **nasa**(-3), **orbit**(-3), **shuttle**($\pm$0), **spacecraft**($\pm$0)

Both term lists exhibit a great degree of overlap to the original term clusters, indicating that the content-based clustering was performed according to the curated term clusters. In Table 4.10, we list a general evaluation for these clusters according to the gold-standard, without further curation by the maintainer. While we failed to identify the two other topics contained in the SCIENCE concept, namely ELECTRONICS and MEDICAL, the two subconcepts CRYPTOGRAPHY and SPACE were correctly identified by the term clusters created in Step I. For the latter topics, the created subconcepts exhibit a very high precision, which is in line with our goal.

It is worth noting that executing our approach after new subconcepts have been introduced will generate new term clusters and possibly aid in detecting topics that were hidden by dominating topics during earlier executions. Based on the cluster presentation, the maintainer can easily select an appropriate name for the new subconcepts. With these two simple steps, the maintainer created two new subconcepts containing about 1000 documents at an average precision of 96.5%.

4.4 Related Work

The suitability of crowdsourcing in general, as well as paid services such as MTurk in particular, has been evaluated for various tasks. Similar to our scenario, R. Snow, O'Connor, Jurafsky, and Ng (2008) evaluated MTurk for natural language tasks, including word similarity and word sense disambiguation. They conclude that by means of redundancy, an expert-quality result can be achieved and that for this purpose on average four non-expert answers are needed, a result confirmed by our

experiments. Likewise, Sheng, Provost, and Ipeirotis (2008) show that redundant crowdsourcing can be used to significantly improve the quality in the context of data mining.

Effects of the task format on the resulting quality were evaluated by Kittur, Chi, and Suh (2008) who state that validation tests and a good design of the task are useful to filter suspicious answers. According to Hsueh, Melville, and Sindhwani (2009), the quality of crowdsourcing results for sentiment classification can be improved by eliminating noisy annotators and ambiguous examples. The authors demonstrate that quality measures in this context are useful for selecting annotations that also lead to more accurate classification models. Alonso, Rose, and Stewart (2008) describe in detail, how MTurk can be used to evaluate the relevance of information retrieval systems.

Sorokin and Forsyth (2008) used the MTurk platform to label images, in particular body parts and shapes on photographs containing people. They experimented with different kinds of annotation tools and described the differences in the results. We already mentioned the *games with a purpose* as an alternative approach to attract a community and get it to complete the desired task. Ahn and Dabbish (2004) and Ahn, Kedia, and Blum (2006) let users play games, and, in the process of playing, label images, locate labeled objects in images, or gather common-sense knowledge.

Heymann and Garcia-Molina (2006) discovered a simple but effective algorithm for converting a large corpus of tags (annotating objects in a tagging system) into a navigable hierarchy of tags. The algorithm leverages notions of similarity and generality that are present in the user generated content. Based on the similarity to certain nodes the tags are placed within the hierarchical system.

KOS extension is also a field that is worked on by many researchers. Comparable to our approach, Nguyen, Matsuo, and Ishizuka (2007) used lexico-syntactic patterns mined from the online encyclopedia *wikipedia.org* to extract relations between terms. Gillam, Tariq, and Ahmad (2005) describe a combination of term extraction, co-occurrence-based measures and predefined linguistic patterns to construct a thesaurus structure from domain-specific collections of texts. Another combination of these techniques using hidden Markov random fields is presented by Kaji and Kitsuregawa (2008). Cimiano, Pivk, Schmidt-Thieme, and Staab (2004) use different sources of evidence – including Hearst patterns – as input for machine learning to derive taxonomic relations.

Witschel (2005) employs a decision tree algorithm to insert novel concepts into a taxonomy. Kermanidis, Thanopoulos, Maragoudakis, and Fakotakis (2008) present a system called Eksairesis for ontology building from unstructured text adaptable to different domains and languages. For the process of term extraction they use two corpora, a domain-specific corpus and a balanced, unspecific corpus. The semantic relations are learned from syntactic schemata, an approach that is

applicable to corpora written in languages without strict sentence word ordering such as modern Greek.

A further example is the IKEM platform as presented by Vervenne (1999): An automatic indexing system with integrated thesaurus maintenance. The maintenance is for example supported by providing the human expert with terms from the indexed documents that might be meaningful, but could not be assigned to existing concepts. Advanced methods that follow the same approach of identification of meaningful terms based on Latent Semantic Indexing and Multidimensional Scaling are presented by Weiner (2005).

Other methods focus only on the extraction of synonyms from text corpora: Turney (2001) computes the similarity between synonym candidates leveraging the number of hits returned for different combinations of search terms. Matsuo, Sakaki, Uchiyama, and Ishizuka (2006) apply co-occurrence measures on search engine results to cluster words. Curran (2002) combines several methods for synonym extraction and shows that the combination outperforms each of the single methods, including Grefenstette's (1994) approach. In some cases, special resources such as bilingual corpora or dictionaries are available to support specialized methods for automatic thesaurus construction. Wu and Zhou (2003) describe a combination of such methods to extract synonyms. Other techniques using multilingual corpora are described by Plas and Tiedemann (2006) and Kageura, Tsuji, and Aizawa (2000).

Our work on concept splitting, was mainly motivated by Brank, Grobelnik, and Mladenic (2008) who use machine learning to predict the additions of new concepts in a classification. They list the assignments of documents by means of clustering and especially the naming of the new concepts based on these clusters as possible extensions.

Clustering of documents is a common task and many other approaches have been developed, e.g. Suffix Tree Clustering (Zamir & Etzioni, 1998) (STC), an incremental algorithm which creates clusters on the basis of common phrases between documents. That way, descriptions for each cluster can directly be taken from these common phrases. With STC, however, only documents containing common phrases are grouped together, neglecting thematic overlap using varying terms. STC focuses on isolated document sets (or text snippets) and is suitable to extract key phrases to be used for further exploration of the documents. As STC allows overlapping clusters, it cannot be used in a classification context. Moreover, as it favors longer cluster labels, STC tends to produce a high number of rather small clusters.

Some of these drawbacks are addressed by *SHOC* (Semantic, Hierarchical Online Clustering) (Zhang & Dong, 2004), an extension to STC designed to cluster a set of web search results with meaningful cluster labels. It is based on *Latent Semantic Indexing* (Deerwester, Dumais, Landauer, Furnas, & Harshman, 1990), an

indexing and retrieval method that employs *Singular Value Decomposition* to discover hidden concepts contained in a body of text. By identifying these semantic concepts in a document corpus, the shortcomings of a clustering algorithm solely depending on lexical matches can be mitigated. SHOC introduces the notion of *complete phrases* to identify possible cluster label candidates with the help of suffix arrays. In our scenario, SHOC has the disadvantage that it behaves like a black box. As the discovery of cluster labels is performed after the clusters have been created, it is not possible to let the maintainer support the process in an intuitive way.

The *description-comes-first* paradigm is employed by Lingo (Osinski et al., 2004), an algorithm inspired by SHOC. In contrast to SHOC, the clustering step is executed after the discovery of the cluster labels, which are used to assign documents to clusters. Similar to STC, Lingo's preference for longer cluster labels leads to a large number of clusters representing topics at a higher granularity than desired for our purpose.

Another supposedly DCF-based approach is Descriptive k-Means (Stefanowski & Weiss, 2007) (DKM) that at first sight looks similar to our approach. The authors extract cluster labels with two different approaches, frequent phrase extraction (implemented with suffix trees) and simple linguistic processing (noun phrase extraction with a trained statistical chunker). The k-Means clustering is performed independently, subsequently the cluster labels are assigned to clusters based on their similarity to the cluster centroids and the contents of the documents in the cluster. Clusters without an assigned label are discarded. The number of target clusters has to be selected beforehand, the initial seeding points are created from randomly selected documents in a way that the most diverse documents of this subset are used. It can be questioned, if DKM follows the DCF paradigm, as the descriptions are not used to influence the clustering, instead they are used to filter the clusters. Nevertheless, the approach follows the same motivation as ours: the identification of labeled clusters favoring a high precision.

4.5 Conclusion

In this chapter, we presented three different approaches regarding the creation and modification of a KOS. All had in common that they rely on human input in one way or another. Furthermore, they all take usage-information into account by means of terminology extraction from documents. For the efficient creation of a new KOS, we proposed to use crowdsourcing by means of the Amazon Mechanical Turk. For KOS modification, we developed two methods that support the maintainer in the identification and positioning of new terms relevant for the KOS, as well as the splitting and subsequent naming of overpopulated concepts in a KOS.

Regarding the **crowdsourcing of the creation process** we made the following contributions: (1) We designed an experiment for acquiring concept hierarchies from arbitrary web users using Amazon Mechanical Turk. (2) We compared the results provided by non-experts with the results reported in (Niepert et al., 2009). (3) We proposed effective methods for filtering non-expert feedback based on quality-diagnosing questions. (4) We showed that the "wisdom of the crowd" performs well when applied in the right way.

We examined the prospects of a paid crowdsourcing service, namely Amazon Mechanical Turk, to complement the role of a community project in the context of learning and populating a taxonomy for the discipline of philosophy. The experiments are promising but several important aspects have to be taken into account. Generally, the answers of MTurk workers are of varying quality, particularly if they are directly compared to a community of experts and motivated laypersons.

The comparison also revealed that there are MTurk workers who perform very well and simple filtering rules can sometimes be used to identify them. In line with findings of other authors (Alonso et al., 2008; Hsueh et al., 2009; Kittur et al., 2008; R. Snow et al., 2008), we showed that it is possible to achieve high quality results, even outperforming the community. In particular, we accomplished this with the following steps:

1. Every pair was evaluated 5 times by 5 different users to ensure the necessary redundancy;

2. We included a small set of concept pairs for which we could objectively determine a correct answer; and

3. Based on the responses to our test pairs, we filtered the users to improve the overall quality of the answers.

With these steps and a moderate filtering policy we achieved a feedback quality comparable to that of the InPhO community. The remaining high quality users still covered $1,018$ of the original $1,138$ (89%) concept pairs that we collected for our experiments. The completion of the MTurk task cost less than 1 day and came to a total amount of 126 US-Dollars. Based on the workers' feedback we were able to create a concept hierarchy which can be browsed online at http://www.kaiec.org/2012/dissertation/amt-inpho.

An important property of the method presented in this section is that it does not rely on any existing data, gold standards or training data provided by experts. Possible next steps include further refinement of the KOS learning process using Amazon Mechanical Turk and a transfer of this approach from the domain of philosophy to other domains. Another promising avenue of future work is the employment of more sophisticated algorithms such as support vector machines to classify MTurk workers according to their feedback quality.

For the **KOS extension using web search engines**, our contributions are the following: (1) We developed a method to identify relevant terms in a set of documents and to provide suggestions for locations in the KOS hierarchy. (2) We identified and adapted existing approaches to carry out the necessary steps. (3) We presented a large-scale experiment applying these methods to extend parts of the MeSH thesaurus with new terms extracted from documents. (4) We presented detailed results on the use of web search engines as a means for generating feature sets for learning the correct relation of new and existing terms.

The results of the experimental evaluation demonstrate that the presented approach has the potential to support and speed-up the laborious task of KOS construction and maintenance. The concept candidate ranking based on the adapted *tf-idf* relevance measure (see Equation 4.6) could identify most of the significant terms of a text corpus. The combination of co-occurrence guided search space reduction and pattern-based position extraction results in accurate classification results, leaving a drastically reduced number of choices to the knowledge modeler. Furthermore, the experiments indicate that web search engine snippets contain enough information to also learn lexico-syntactic patterns for the problem of hyponymy extraction. The combination of synonymy and hyponymy classification allows us to locate, for each extracted candidate concept, the appropriate position in the KOS. We believe that only slight modifications are necessary to adapt the system to several important real-world use cases including KOS maintenance for digital libraries and information retrieval systems. Both of these use cases are important to businesses as well as university libraries. A bottleneck of the pattern based approach is the time it takes to query the web search engine. Therefore, we reduced the number of pairs by using a co-occurrence similarity measure.

For the **splitting and naming of concepts** we made the following contributions: (1) We proposed a new implementation of the description-comes-first paradigm, preserving its advantages without putting the burden on the user to actually name the new concepts first. (2) We integrated this approach with straight-forward content-based k-Means clustering to ensure that the splitting of the concept best reflects the actual content of the documents. (3) We evaluated the method under laboratory conditions. (4) We showed that the maintainer is able to create new subconcepts with matching documents assigned at an average precision of 96.5%.

We presented a workflow in three steps to create recommendations for new subconcepts in a hierarchical classification system. The creation is mainly performed by clustering documents associated to the concepts to be split (Step II). We improved the result by incorporating the human maintainer of the classification: once before the clustering takes place, when the maintainer selects term clusters to influence the clustering; once afterwards, when the actual subconcepts are created based on the recommendations. We have shown that our approach works with promising results under laboratory conditions and are confident that it can be used in a productive setting. The strength of our approach lies in the transparency for

the user who can influence the result easily based on comprehensible term clusters, while the actual recommendations are still created on the document contents and not just on a term basis.

Acknowledgements: Parts of this chapter have been published before. The creation of a KOS by means of crowdsourcing is published in (Eckert et al., 2010). The approach builds on prior works by Niepert et al. (2007, 2008, 2009). The extension of a KOS using web search engines is presented in (Meusel et al., 2010). In (Eckert, Meusel, & Stuckenschmidt, 2011), we described both approaches together with the visual analysis of the results by means of the ICE-Map Visualization. The semiautomatic clustering of documents to generate new subconcepts was presented at the Joint Conference of the German Classification Society (GfKl) and the German Association for Pattern Recognition (DAGM) 2011 (Stork, Eckert, & Stuckenschmidt, 2011).

Chapter 5

Implementation

Approaches and visualizations as presented in this thesis can only be part of a solution and need to be integrated in an environment that enables the user to work with them, but also with other analyses, tools, and assisting approaches in an interactive and intuitive way.

In this chapter, we introduce Semtinel as a platform that allows the integration of new approaches together with a presentation to the user in an intuitive way, based on common concepts of approaches for KOS maintenance. In the second part of this chapter, we describe the implementation of LOHAI, the automatic indexer that was developed for the KOS selection use case in Chapter 3.

5.1 Semtinel: An extendable Analysis and Visualization Platform

To be functional as a platform, Semtinel has to provide typical platform functionality, i.e. in the first place it has to be extendable by means of a plug-in or module system. The whole platform should use a component model that allows the replacement of an existing component by a new implementation. The core components should provide an API and further infrastructure to allow communications between the components. From a user-perspective, the application has to be highly configurable and adaptable to individual needs. It should provide all the basic functionality that the user is familiar with from other applications and additionally define common guidelines how to interact with the specific features that are unique to Semtinel.

In the Java[1] world, there are only two platforms for the creation of rich client applications that are widely adopted: Eclipse RCP[2] and Netbeans Platform.[3] Both platforms fulfill all requirements for a rich client platform (Tödter, 2007b, 2007a). Eclipse is based on OSGi[4] since version 3.0 (Gruber, Hargrave, McAffer, Rapicault, & Watson, 2005), which makes it more open to the rest of the world, but Netbeans recently started to support OSGi as well.[5]

We decided to develop Semtinel on top of the Netbeans platform. The main reasons were the better GUI builder that comes with the Netbeans IDE and the fact that it uses pure Java Swing components without any platform-specific code. As a Netbeans platform application, Semtinel provides all functionalities of the platform. From the user's point of view, this is mainly the windowing system that allows the almost arbitrary arrangement of all components via drag and drop (Figure 5.1).

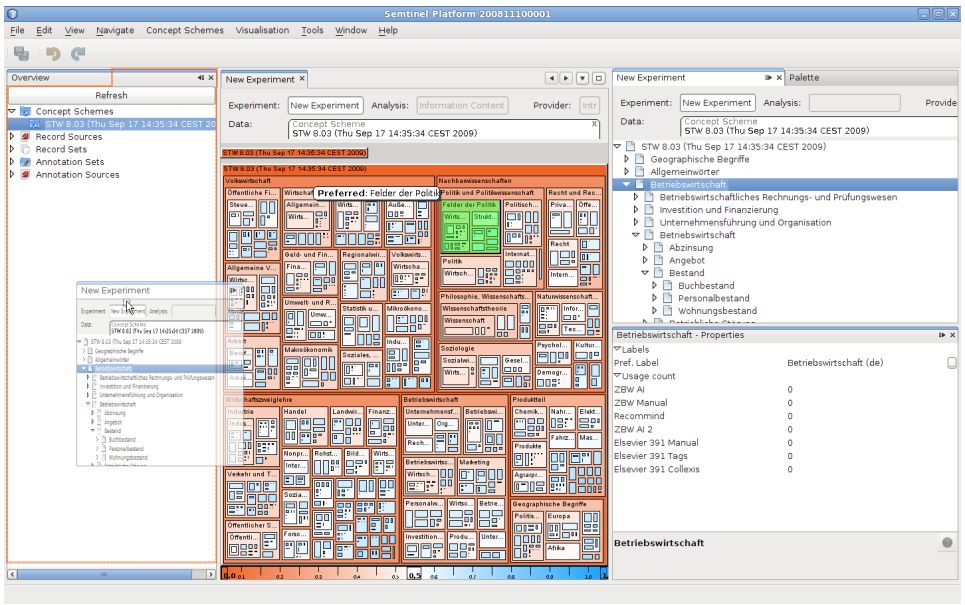


Figure 5.1: Semtinel on the Netbeans Platform.

From a developer's perspective, the main aspect is the component system, where components are called modules within the Netbeans platform. Modules provide well defined functionalities and may or may not depend on the availability of other modules. Modules can communicate with each other by means of the *Lookup*

[1] The first prototypes were implemented in Microsoft .Net, but we migrated to Java, mainly due to the availability on all major operating systems and the higher acceptance in the scientific community.

[2] http://www.eclipse.org/

[3] http://netbeans.org/features/platform/index.html

[4] http://osgi.org

[5] http://netbeans.org/features/java/osgi.html

mechanism. Besides this main infrastructure, the Netbeans platform provides lots of additional mechanisms and libraries that release the developer from the need to constantly reinvent the wheel: Support for background threads with progress bars, the implementation of wizards, toolbars, and menus or the central provision of a properties view for arbitrary contents, just to mention some.

Gast (2009) describes the implementation of a user interface for a theorem prover on top of the Netbeans platform. He details the implementation by means of the specific Netbeans features. In contrast, we concentrate on the interface concepts and Semtinel specific APIs that were developed as part of this thesis.

5.1.1 The Semtinel Architecture

Figure 5.2 illustrates the architecture of Semtinel. There are three layers based on the dependencies of the modules that are involved. On *layer one*, there is the whole Netbeans platform as a basis, together with the object relational mapping framework Hibernate[6] that provides the access to the data. The data itself is stored in a relational database.[7]

Layer two is the actual *Semtinel Platform*. It consists of modules that can be grouped into *Data Model, I/O, and Core Services*, *Experiment API*, and *Utilities*. The most interesting part is the *Experiment API*, which will be described in detail in Section 5.1.3.

Finally, on *layer three*, we have modules that make use of the underlying platform and provide importers for the datasets that we have used in our experiments and implement among others the approaches that are discussed in this thesis, especially the *Treemap Visualization* and the *IC Difference Analysis*, which build the ICE-Map Visualization. Table 5.1 gives an overview on all modules.

5.1.2 Data Model

The core data model of Semtinel is focused on the following three classes (cf. Figure 3.5) which serve in the default implementation as possible input values for experiments:

Concept Scheme A thesaurus or concept hierarchy that is the main subject of the analysis.

Record Set A set of records describing a document corpus. The currently implemented analyses only use it indirectly via the annotation set, but for example

[6] http://www.hibernate.org

[7] Any database that is supported by Hibernate can be used. By default, H2 (http://www.h2database.com) is used in embedded mode, which means that there is no need for a running database server.

Module	Description
Datamodel, I/O, Core Services	
Database	Central API to the datasets and access point for all other modules
Overview	Loaded Datasets (Concept Schemes, Records and Annotations)
Property Sheet Extender	A pluggable and extendable system for the presentation of properties
Extended Properties	Additional information about the loaded datasets
Search	Concept search
Recordset Viewer	Simple viewer for records
Experiment API	
Experiment Registry	Experiment API and implementation, including the Explanation API
Register Manager	Registers are widely used throughout Semtinel to enable drag and drop selection of datasets
Utilities	
Cockpit	System information, memory usage, API version, ...
Plugins	Central module distribution
Semtinel DB Util	A helper module to import data from other databases
Semtinel Logging	Configuration of the logging facilities at runtime
Semtinel Utilities	Various general purpose utility classes
Application Modules	
Treemap Visualization	Treemap visualization of the ICE-Map Visualization
Treeview Visualization	Treeview visualization (an explorer-like view)
IC Diff Analysis	Statistical analysis implementation of the ICE-Map Visualization
Concept Scheme Importer	Several importers for concept schemes
CSV Center	CSV Import for documents and annotations
Pubmed Access	Direct access to Pubmed to retrieve new abstracts and import them as record sources

Table 5.1: Semtinel modules.

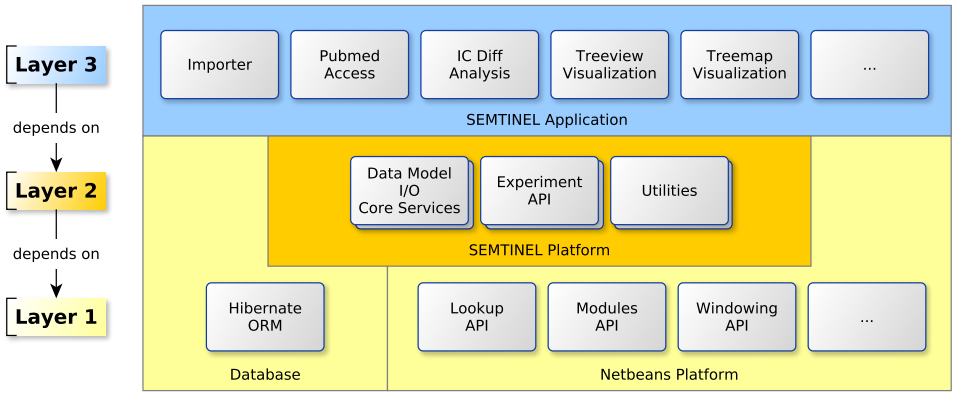


Figure 5.2: Semtinel architecture.

a new analysis could directly work on the record set and provide results like word counts.

Annotation Set A set of annotations where an annotation is the link between a record and a concept, optionally with an additional weight, if the annotation was created automatically and the creation process provides such a weight.

The data model uses SKOS for the concept schemes.[8] This ensures that Semtinel is widely usable as SKOS is the W3C recommendation regarding the representation of KOSs in the web. Additionally, this means that KOSs conforming to the new ISO standard (ISO, 2011) are generally supported, even though not with all details.

The record and annotation sets are very flexible; they can be created and modified during runtime – e.g. by splitting or merging other sets – and thus allow a very fine-grained choice and definition of the datasets that are used in an experiment. To avoid losing the overview and especially the provenance of a record or an annotation, each of them is tightly related to a record or annotation source, respectively. The sources are generated during the import of new data and contain information like the date of the import and other information regarding the provenance of the data. Figure 5.3 illustrates these classes and their relationships. This object model is extendable, i.e. new Semtinel modules can define additional classes and relationships that can be provided to the user via the overview.

The Database module provides access points to the loaded data, as well as an API that can be used by import modules to load data in arbitrary formats. The Overview module displays the loaded datasets and allows the user to manipulate the data and to select it for other modules, especially for the use in experiments.

[8] http://www.w3.org/2009/08/skos-reference/skos.html

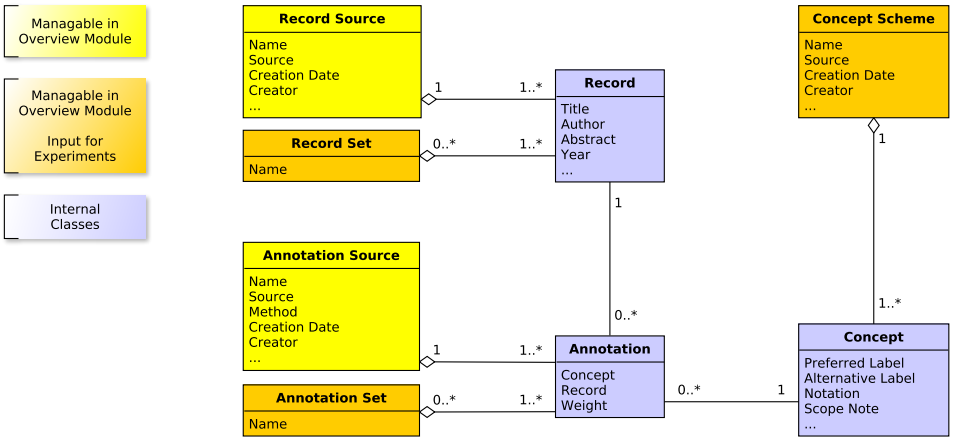


Figure 5.3: Semtinel class diagram.

5.1.3 Experiment API

The Experiment API defines the central and unique operational concept of Semtinel. Figure 5.4 shows the typical view that is used to create a new *Experiment*. An experiment is the visualization of analysis results based on one or more input datasets. The result of the visualization is shown in the output area. The available datasets are provided by the Overview module, the Palette module contains all loaded visualization and analyses. The experiment is set up in the configuration area by dragging and dropping the desired components and input datasets.

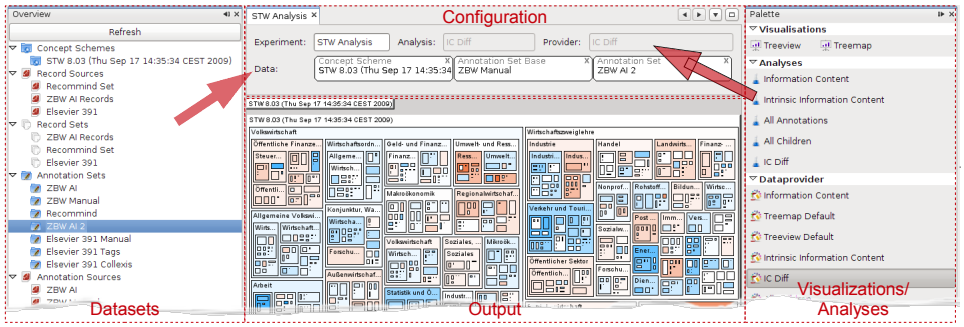


Figure 5.4: Creation of a new experiment.

The configuration panel constantly adapts to the already selected components, i.e. it provides the adequate number and types of drop fields, depending on the requirements of the visualization and the analysis. For example, if the user just drags and drops the treemap visualization on the configuration panel, only one drop field for a concept scheme appears, because the treemap visualization needs a

concept scheme to work, but by default does not require any other datasets. In this case, the color feature of the treemap is not used.

The coloring of the treemap reflects the result of an analysis. If the user chooses the intrinsic information content (cf. Chapter 2) as analysis, we still have only one drop field, as the intrinsic information content only depends on the hierarchical structure of the concept scheme. On the other hand, if the user chooses IC Diff as analysis, as in Figure 5.4, three drop fields appear, because IC Diff needs at least one annotation set as input to calculate the information content of the concepts. The field in the middle (Annotation Set Base) is optional and can be left empty; in this case, the intrinsic information content is used as base value.

Visualizations/Analyses

All visualizations and analyses are provided as modules. This means that new and customized implementations both for visualizations and for analyses can be developed by the user and loaded during runtime. The infrastructure for the management and distribution of these modules is provided by the Netbeans platform. Semtinel contains an *Experiment Registry* where new visualizations and analyses are registered automatically. The registry collects information like a display name, features and requirements of the modules that are used to build the palette of available visualizations and analyses (Figure 5.5). It also enables the communication between these modules.

The interfaces of these modules are deliberately simple to avoid unnecessary restrictions for own developments. Table 5.2 lists their most important methods. For each visualization and analysis exists a factory implementation that describes it and is used to register it in Semtinel. Notable is the method getRegisterDefinitions() that returns an array of register definitions, i.e. the definition of the input parameters that are needed.

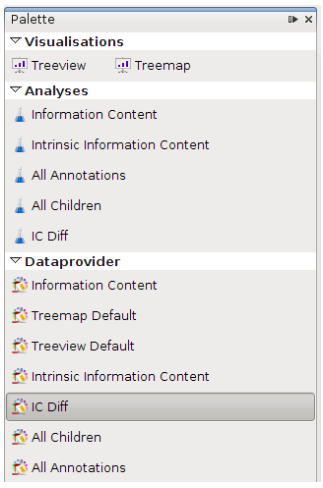


Figure 5.5: Palette

A visualization in the simplest case just provides access to a java.awt.Component that is shown in the output area of the experiment window. Lookups are optional and usually used to broadcast the highlighted concept, but can be used for other purposes, too. Lookups are part of the Netbeans platform and a powerful and flexible mechanism to communicate between loosely coupled modules. Besides that, the interface contains basic methods for the selection management, i.e. another module can read and manipulate the selection of a concept, if such a selection is supported

by the visualization. This is used by the experiment groups, as described in Section 5.1.3.

An analysis in the simplest case just provides a method to check if all requirements regarding the input data are satisfied and another method that returns the actual value produced based on the inputs. Semtinel provides two additional mechanisms that support the development and use of new analyses: *Population* and *Explanation*.

Population means the precalculation of some values that would be too time-consuming to calculate during runtime. The development of such a data population is very easy for the developer and Semtinel provides everything that is needed to manage and store these data values.

Explanation is a mechanism that enables the user to get detailed information on how an analysis value is actually calculated. The support of explanations is optional for the developer, but strongly encouraged. Explanations are described in detail in Section 5.1.3.

Dataprovider. Besides visualizations and analyses, there is a third type of modules available in Semtinel and visible in the palette: The dataproviders. A dataprovider connects an analysis to a visualization, i.e. it translates the analysis result to an input value that can be displayed. This can be as simple as providing a normalized value for an analysis that returns a value with no upper bound. To avoid the necessity to provide a dataprovider for every combination of visualizations and analyses, the visualizations and analyses characterize themselves by means of *flavors*, for example the fact that an analysis produces a real value between 0 and 1.

No restrictions exist regarding the flavors, for instance you can create an analysis that returns geographic coordinates as results and a dataprovider that translates the coordinates to a real value by calculating the distance towards a predefined location. A dataprovider can add its own input registers, as in this example the reference location.

Currently, the user has to select the dataprovider manually. While Semtinel checks the compatibility of the selected visualization, analysis and dataprovider, it is planned to hide the dataproviders from the user and select an appropriate dataprovider automatically. Alternatively, one could let the user choose only if several suitable dataproviders are available. This is not yet implemented.

Configuration and the Register Set

Figure 5.6 shows the configuration panel that is used to configure a new experiment. The user can assign a name to the experiment and a combination of visu-

Method	Description
Factory.java (Visualization/Analysis/Dataprovider)	
String getId()	Unique Id
String getDisplayName()	The display name
Image getIcon(int)	Icons for the palette
Flavor getFlavor()	Provided flavor (see text)
RegisterDefinition[] getRegisterDefinitions()	Required input data
Object getInstance()	Creates new instances
Visualization.java	
Lookup[] getLookups()	Enables inter-module communication
Component getOutputComponent()	Arbitrary component that is displayed in the output area of the experiment window
Selection Management	Listener and Getter/Setter to manipulate the selection of the visualization
Analysis.java	
boolean hasRequiredData(RegisterSet)	Checks, if all required input data is available
Object[a] getValue(Concept, RegisterSet)	Performs the actual analysis and returns the result
Population Management	Support for the precalculation of values to improve speed
Explanation Management	Explanation support (see text)

[a] In the current implementation the analyses are restricted to float values, but this should be changed in future releases.

Table 5.2: Selected methods of the main interfaces of the Experiment API.

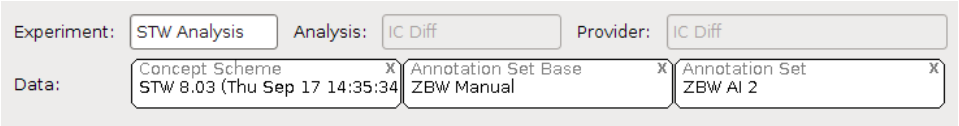


Figure 5.6: Experiment configuration panel.

alization, analysis and a dataprovider is chosen by means of drag-and-drop. The visualization is visible in the output area, the currently selected analysis and dataprovider are indicated in the configuration panel. But the most interesting part is the lower half of the panel: the selection of the input data.

As described above, all modules (especially the analysis modules) can provide register definitions to specify their input parameters. An input parameter can be used by two components. For example, the treemap visualization requires at least one concept scheme to display something meaningful. The IC difference analysis needs at least one concept scheme and one annotation set, optionally a second one, if a base set is desired as reference. Thus, in sum we need three input parameters to visualize the results of the IC difference analysis in a treemap: A concept scheme and two annotation sets (where one can be empty). For each input parameter, the user has to choose a dataset. In Semtinel, the input parameters are managed via registers. A register specifies a class that represents the valid values (e.g. ConceptScheme) and an identifier that is used globally to characterize the register.

Semtinel defines several default registers that can directly be used by the modules to specify their desired inputs, these default registers are available for concept schemes and annotation sets. It is possible to define a register as representing multiple values. In this case, the user can drag and drop more than one dataset on the register; with a click on the register, the single values can be ordered and managed (Figure 5.7).

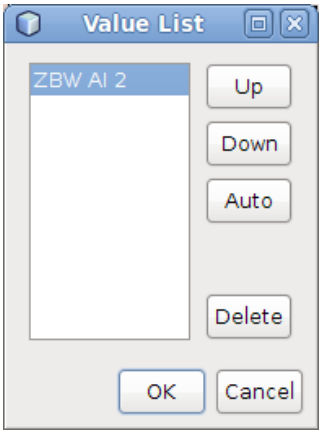


Figure 5.7: Multiple datasets in one register.

If the predefined registers are not suitable, arbitrary custom registers can be defined by the module developers. For instance, this is needed, if a module introduces new data classes that are shown in the overview and should be selectable by the user for a new analysis.

The dynamic set of registers that is generated by the combination of requirements of the involved modules is called register set. It defines the actual input for the current experiment and the RegisterSet class is the central interface for all modules to access the input data.

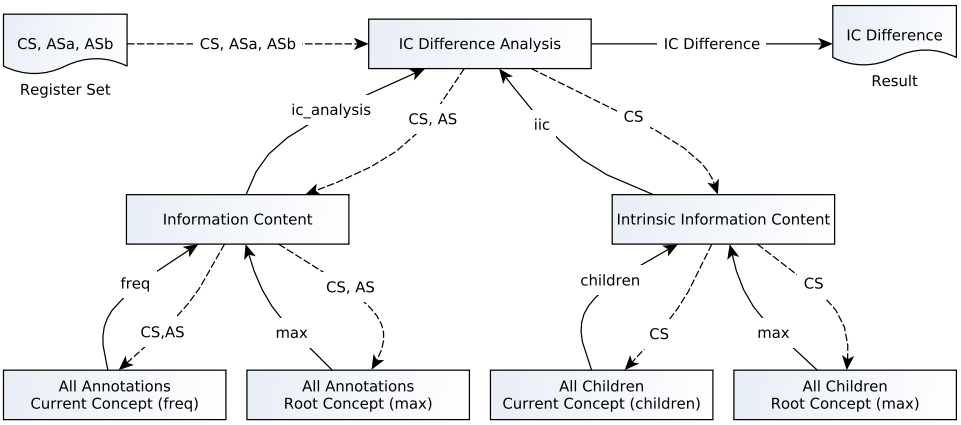


Figure 5.8: Hierarchy of different analyses.

Hierarchical Analyses

A main purpose of Semtinel is the provision of a framework to develop new analyses for the maintenance of concept hierarchies. The core APIs make the access of the loaded data sets very simple for the developer and the Experiment GUI directly makes the new analysis available to the user. But there is another mechanism that facilitates the development of new analyses: A new analysis can be based on existing ones. Such an extension of existing analyses is not the exception, it is the rule.

Consider for example the IC difference analysis. In Semtinel, it is built upon two other analyses: *Information Content* and *Intrinsic Information Content*. The information content is calculated by means of the analysis *All Annotations*. This is a very simple analysis that just returns the number of annotations of a concept and all its subconcepts. On the other hand, *All Children* is the basis for the intrinsic information content and it returns simply the number of child concepts of a concept and all its subconcepts. These analyses exactly reflect the statistical framework and the weight functions of the ICE-Map Visualization (cf. Chapter 2).

To facilitate such a hierarchy of analyses, the framework has to ensure that an analysis works independently of the context in which it is used, i.e. it cannot directly access the register set of the current experiment, because these registers are determined only by the top analysis that is chosen by the user. This is realized by creating distinct register sets for every analysis that is used in the hierarchy. By default, the original register set is just passed along, but it is possible to remap single registers – this is supported directly via a helper class – or to create new register sets that contain the values that are needed for the subsequent analysis.

Figure 5.8 illustrates the calculation of the IC difference analysis with only one selected annotation set (ASa), i.e. the intrinsic information content is used

Property	Description
String label	A label that identifies the analysis
String text	A describing text that explains what the analysis does (e.g. with a formula)
Map inputMap	All registers with their values
Map extensionMap	Extensions (intermediate results) that were calculated for the final result (optional)
String result	The final result that was returned
List subnodes	Links to explanation nodes that were created by analyses that were used inside this analysis

Table 5.3: Properties of an Explanation node.

as reference. The register set that is passed to the Information Content contains the concept scheme (CS) and one annotation set (AS) which was remapped from ASa. The Intrinsic Information Content gets a register set that only contains CS. In both cases, the register sets then are passed through to the lower analyses. These analyses are executed twice, once to calculate the actual value for the selected concept and once to calculate the value for the root concept for normalization.

Explanations

While the reuse of analyses to create new ones makes the development faster, it has one major disadvantage: The debugging of a new analysis can be harder, as existing analyses are used like a black box. And the more complex the analyses become, the higher the need of the user to obtain information, how a result actually was calculated. For this reason, we developed the Explanation API together with the Explanation Browser. The idea is simple: Every analysis creates an explanation node (Table 5.3) that contains all the information that is needed to understand what the analysis did.

The explanation node can be filled arbitrarily by the analysis developer, as long as it is suitable to "explain" the user what happens inside. Figure 5.9 shows the Explanation Browser in action; it explains the calculation of the IC difference analysis for the concept BUSINESS ECONOMICS. The explanation corresponds to the data flow illustrated in Figure 5.8. The browser just lists the properties in the order of Table 5.3, with indented subnodes. A more appealing presentation is planned.

Experiment Groups

Already in the very first prototypes of Semtinel (Eckert et al., 2008), we regarded it to be important to combine two visualizations (treemap and treeview) to get the best of both worlds. In the meantime, this approach was generalized: experiments –

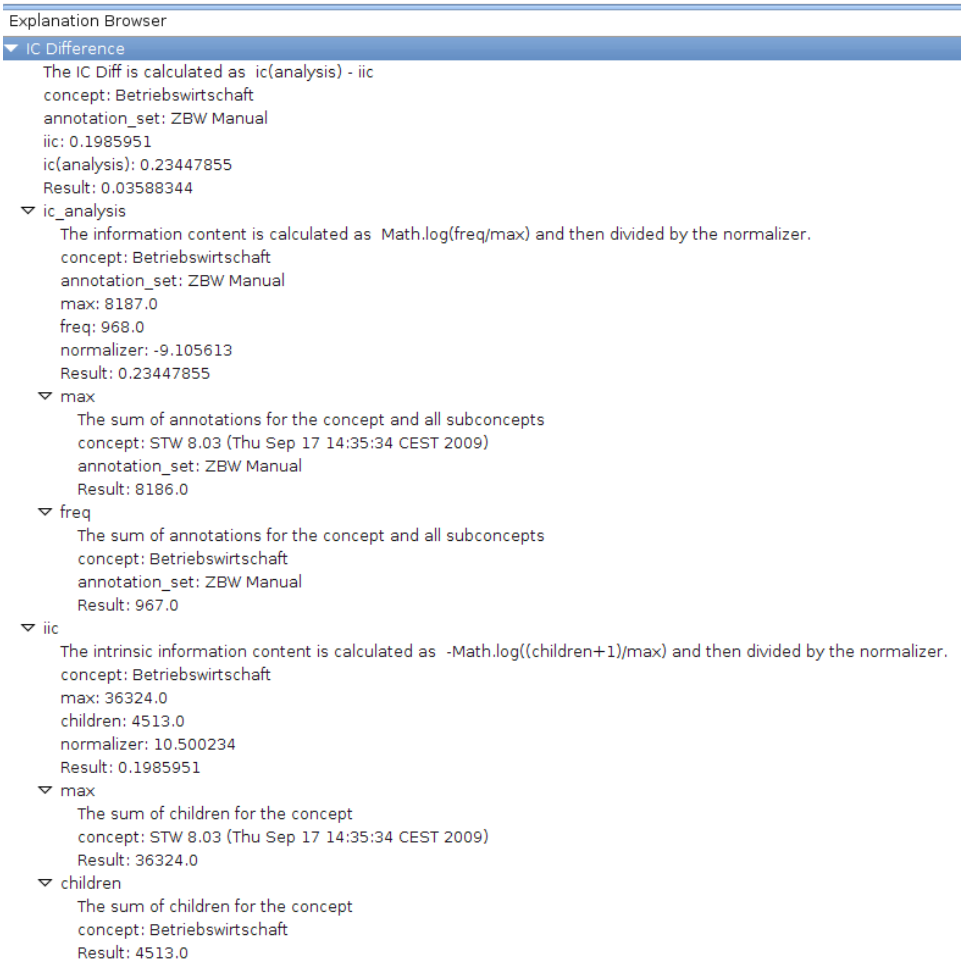


Figure 5.9: Explanation browser.

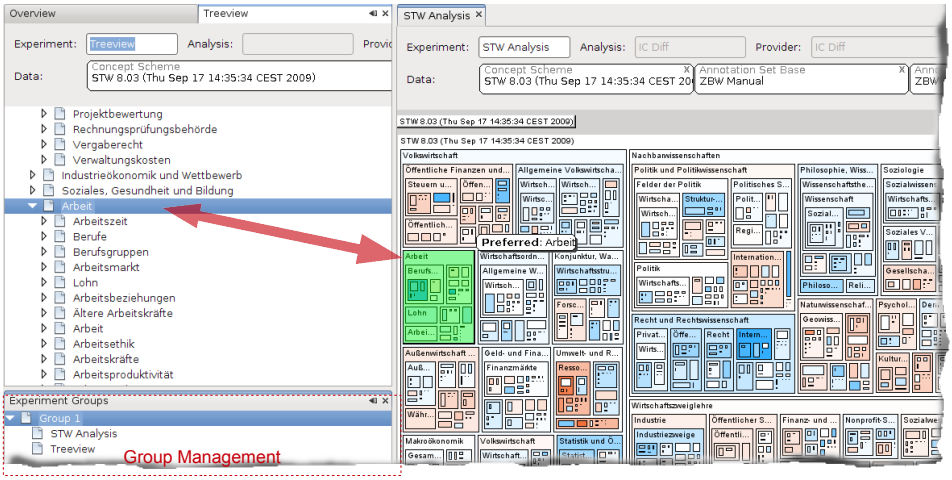


Figure 5.10: Experiment groups.

i.e. different visualizations of analysis results – can be grouped together. Grouped experiments are synchronized regarding their selection, if the user selects a concept in one experiment, it also gets selected in the other experiments of the group (Figure 5.10).

While typically a treeview is combined with a treemap, the grouping mechanism can also be used to synchronize different treemaps, for example to view different analyses at the same time, e.g. the ICE-Map Visualization with two different reference sets.

5.2 LOHAI: A Baseline Indexer

In Chapter 3, we presented a use case where we want to evaluate a KOS and a document set based on their topical overlap. Evidently, this is only possible with a set of annotations that relate the documents to the concepts in the KOS. When no such annotation set is available and its creation not feasible, an automatic indexing system is needed. There are two general approaches for automatic indexing: statistical ones, i.e. the use of machine learning to assign concepts to documents based on assignments in a preindexed training set, and linguistic ones, i.e. the use of natural language processing techniques to identify meaningful terms in the text and assign concepts based on these terms. With Maui (Medelyan, 2009), there exists a statistical indexer that incorporates a lot of NLP techniques. To the best of our knowledge, however, there is no free and open source implementation for a strictly linguistic indexer that can be used without any training data on arbitrary documents. Nevertheless, we need such a pure linguistic indexer for our purpose, as we cannot provide a preindexed training set.

Therefore, we introduce our own implementation called LOHAI,[9] a strictly linguistic indexer that uses mainly all techniques that are state-of-the-art in information retrieval.

The development of LOHAI is led by the following motivational thoughts:

Knowledge-poor and without any training: To be usable for arbitrary KOS and documents, the indexer cannot rely on any *additional* knowledge sources. Only the KOS itself can and will be used. The indexer must not employ a training step, as in this setting, usually no preindexed documents are available and the creation of a training set would be too cumbersome for the user.

Simplicity over quality: On the one hand, the ICE-Map Visualization is tolerant to indexing errors, when it comes to an overall assessment of a KOS and a document set. On the other hand, indexing errors can be spotted easily, if they affect the visualization result and thus do not affect the interpretation of the KOS suitability by the user. While every single step could be improved or replaced by a more sophisticated technique that is already developed and published somewhere, we tried to develop everything as simple as possible. Everything should be easy to use, easy to understand and easy to improve if needed. The understandability of the indexing results is especially important in our case, as we use LOHAI for a specific purpose: We want to see, if the documents and the KOS fit together.

With these prerequisites in mind, we compose the indexer as a pipeline with several components, as illustrated in Figure 5.11.

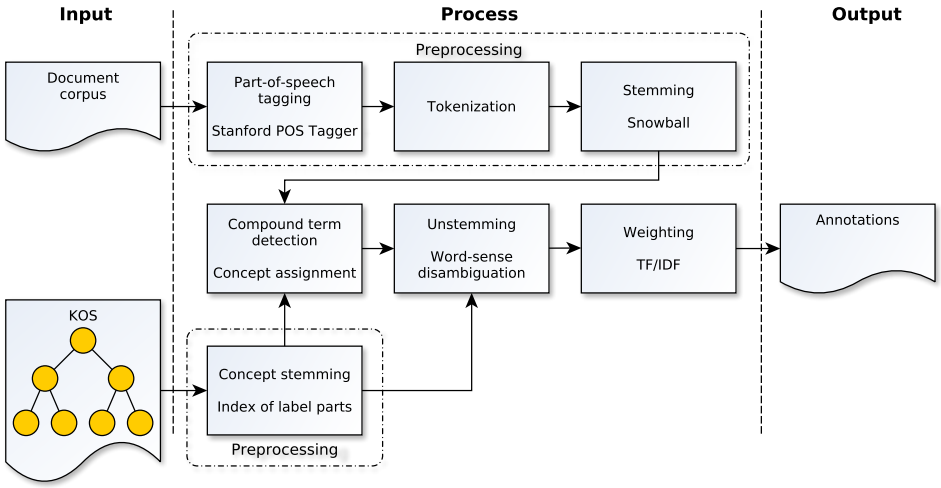


Figure 5.11: The indexing pipeline.

[9] LOHAI is pronounced like Low-High and means something like LOw HAnging Fruits Automatic Indexer, which gives a brief summary about the development process: we used Semtinel to evaluate the indexer and added components and improved them until we reached satisfying results.

5.2.1 Preprocessing

Before the indexer actually assigns concepts to term occurrences in the text, the KOS and the text to be indexed is preprocessed. This preprocessing consists of the following steps:

1. **Part-of-speech tagging:** We use the Stanford Log-linear Part-Of-Speech Tagger,[10] as described by Toutanova, Klein, Manning, and Singer (2003). Part-of-speech (POS) tagging simply means the identification of nouns, verbs, adjectives and other word-types in a text. To avoid wrong concept assignments like the assignment of the concept NEED (as noun in the sense of requirement) whenever the verb "to need" is used, we only consider nouns (NN,[11] NNP, NPS, NNS), adjectives (JJ, JJR, JJS), foreign words (FW) and unknown words (untagged).

2. **Tokenization:** Tokenization splits the text into single terms and is performed together with the POS tagging. The result is a list of terms that are further investigated for proper concept assignments. The tokenization step also includes term cleaning: everything is truncated that is not a letter, a hyphen or a space. Note that numbers are truncated, too, as they usually contain no meaning and are generally highly ambiguous. In some domains, this would not be desired, consider for example history or chemistry.

3. **Stemming:** Finally, the single terms are stemmed, i.e. they are reduced to their stem. That way, same terms can be matched, even if they use different grammatical forms, like "banks" and "bank." We use the English (Porter2) stemming algorithm for the Snowball stemmer (Porter, 2001).

4. **KOS preparation:** This is only performed once per KOS. All concept labels are stemmed by means of the same stemmer that is employed on the document texts. An index is created that maps the single stems to the corresponding concepts. Additionally, an index of stemmed label parts is created that is used for the identification of compound terms – for instance, "insurance market" is stemmed to "insur market," mapping to the corresponding concept, additionally, both stem parts are indexed and mapped to the stemmed compound term.

The preprocessing uses only freely available standard approaches. The POS tagging and the stemming are language dependent; both algorithms employed are implemented for various languages, including English and German. We assume that both the KOS and the documents are in the same language and that only one language is used in the document, so that the appropriate implementations can be used. If the KOS is multilingual and the documents use different languages, an additional language detection step has to be employed.

[10] http://nlp.stanford.edu/software/tagger.shtml

[11] Tag definitions according to the Penn Treebank tag set (Santorini, 1990).

After the preprocessing steps, the actual concept assignment and weighting takes place, as described in the next section.

5.2.2 Concept assignment with compound term detection

The general assignment strategy is a pure string based matching: If a stem that belongs to a concept in the KOS appears in the stems extracted from the text, the concept is assigned. In this step, we consider every concept a matching concept that contains a label that has the same stem or contains the same stem in the case of a compound term. Under this assumption, we have to deal with three possibilities that can lead to wrong assignments:

1. A stem can belong to several concepts, including compound term concepts, e.g. "insur" that belongs among others to INSURANCE and INSURANCE MARKET.
2. A stem can belong to several concepts that have different labels with the same stem (overstemming), like NATIONALISM, NATIONALITY, and NATION.
3. A stem can belong to several concepts that have the same labels with the same stem (homonyms), like "bank" (the financial institution) and "bank" (a raised portion of seabed or sloping ground along the edge of a stream, river, or lake).

Approaches to handle the latter two variants are described in the next section, the first variant is dealt with directly in the assignment phase: The basic assumption is that we want to assign the most specific concept, i.e. in the above example, we would like to assign INSURANCE MARKET, but not MARKET.

We implement this as follows: Whenever a stem is recognized as a potential part of a compound term, the stem is temporarily stored in a list. When a stem is found that cannot be part of a compound, the list is analyzed for contained concepts. In this step, the algorithm simply checks every chain of stems for every starting stem if it corresponds to a compound concept. The algorithm starts with the longest possible chain and stops if a compound is found, thus avoiding the assignment of additional concepts contained in the compound. With this approach, the algorithm has generally a linear runtime with respect to the words contained in the text. Only the parts that potentially contain compounds have to be further analyzed with a runtime of $O(n^2)$ with n denoting the number of words within such a part.

5.2.3 Unstemming and word-sense disambiguation

Whenever one or more stems could be assigned to more than one concept, we would like to identify a single concept as the correct one in the given context. This

task is generally denoted as word-sense disambiguation (WSD). We use two different approaches for WSD, the first being a specific check that tackles the problem of overstemming mentioned above. If this step is not able to disambiguate the potential concepts, the actual WSD is performed.

Unstemming. Overstemming – the reduction of two different terms to the same stem – leads to ambiguous stems that have to be disambiguated during indexing. Consequently, we first unstem the stem, i.e. we go back to the original, unstemmed form of the term, as found in the text. If the unstemmed term corresponds directly to an unstemmed label of a concept, we assign this concept. If there is only one such concept, we finish the WSD step. Otherwise, we continue with the actual WSD, as described in the following.

KOS based word-sense disambiguation. Word-sense disambiguation is a broad field in the area of natural language processing. Leaving the technical issues of overstemming aside, it generally consists of the task to determine the correct sense of a word that appears in a particular context. The variety of possible senses is often based on some background-knowledge, as a thesaurus or other types of KOS. As Manning and Schuetze (1999, pp. 229 f.) point out, this can be unsatisfactory from a scientific or philosophical point of view, as the definitions in the background knowledge are often quite arbitrary and possibly not sufficient to describe the actual sense of a word in a given context. Our goal, however, is not the perfect assignment of a sense to a word, our goal is the assignment of the best fitting concept in the KOS.

WSD approaches can be divided in supervised and unsupervised approaches, additionally in knowledge-rich and knowledge-poor approaches (Navigli, 2009). In our setting, we need an approach that is unsupervised – as it has to work without any previously tagged texts – and knowledge-rich – as we have a KOS at hand and of course want to use it to improve the disambiguation quality.

A supervised, knowledge-rich approach is the adaptive thesaurus-based disambiguation, as presented by Yarowsky (1992), where a Bayes classifier is trained on a large document set and thus probabilities for the occurrence of specific words in the context of a specific sense are determined.

Yarowsky (1995) also proposed an (almost) unsupervised approach that makes use of two assumptions:

One sense per collocation. We assume that words collocated with the word to be disambiguated are unique for the correct sense and would not be collocated with the word for other senses. This basically is the rationale to use the context of a word – usually a window of words before and after the word in question – for disambiguation.

One sense per discourse. We assume that only one sense for a given word is used throughout a whole document. With this assumption, we can make use of any occurrence of the word in the text and thus get a more stable disambiguation result.

Both assumptions have been examined and verified (Gale, Church, & Yarowsky, 1992; Yarowsky, 1993). As Yarowsky's approach is not completely unsupervised – a small set of pretagged senses is needed as seed – we only make use of the two assumptions, but use a much simpler approach: Word-sense disambiguation based on a Jaccard comparison (cf. Ramakrishnan, Prithviraj, and Bhattacharyya (2004)).

For this comparison, we define two sets of words: W as the context of an occurrence of the ambiguous word w, and C as the context of a candidate concept c, respectively. We then compute the Jaccard measure as follows:

$$\text{Jaccard}(W, C) = \frac{|W \cap C|}{|W \cup C|} \tag{5.1}$$

Based on the assumption "One sense per discourse," we assign each occurrence of w the concept c that was mostly assigned in the document, i.e. got in most cases the highest Jaccard value. If only abstracts are available for indexing, this procedure can be further simplified by just assuming the whole abstract as the context for each occurrence of w, which leads to the direct assignment of the concept c with the highest Jaccard value.

As context of an ambiguous word w, we either define a window of 100 words before and after the word or just use the whole document in case of short texts, like abstracts. The context of a concept c is defined as the union of all labels of the concept, its direct child concepts, its parent concepts and the direct children of the parent concepts, i.e. its siblings. Other definitions are of course possible, for example the weighting of words and labels depending on the distance to the word or concept, but for our purpose as part of a simple baseline indexer, our approach is sufficient.

5.2.4 Weighting

The last step in the indexing pipeline is the weighting of the assigned concepts. As the baseline indexer so far assigns every concept that can be identified by an occurring word, the weighting of these concepts is vitally important to determine which concepts are important and descriptive for the given text and which concepts are only marginally touched. It is also desirable to give concepts a higher weight when they are not used in the majority of documents, because these concepts usually only denote common terms and are not important for the indexing result.

The common approach for this kind of weighting is *tf-idf*, which is based on the term frequency $tf_{c,d}$ of a term (in our case concept c) in a given document d and on the document frequency df_c of a concept c, i.e. the number of documents, where the concept appears:

$$w(c,d) = tf_{c,d} \cdot \log \frac{D}{df_c} \tag{5.2}$$

D denotes the total number of documents in the indexed set. The last term is called inverse document frequency (*idf*), as the overall weight becomes smaller the higher df_c is.

5.2.5 Example

To show the weaknesses and strengths of LOHAI, we investigate an example of an indexing result. Again, we use the German STW Thesaurus for Economics (c.f. Section 1.5). A concept in the STW consists of preferred and alternative labels, both in English and in German. For example, there is the concept MIGRATION THEORY with alternative labels "Economics of migration" and "Theory of migration."

Figure 5.12 shows an example abstract that we indexed. LOHAI produces the output as shown in Figure 5.4. Additionally, we listed the intellectually assigned concepts by a librarian. It can easily be seen that the characteristics of the results are quite different. But if one takes the weighting into account, it can be seen that there are no wrong assignments with a weight above 0.3. Below that threshold, there are especially common terms that form a concept in the thesaurus and that are either not helpful or wrongly assigned, as EXCHANGE. GOVERNMENT for example, seems to be correct, but is rather a coincidence, as it is assigned due to the verb "govern" in the text – an indication for a mistake during the POS tagging. On the other hand, the very abstract concepts that are assigned by the librarian (besides THEORY) are not found by LOHAI, as the terms do not directly appear in the text in some form. It is no coincidence that these findings are reminiscent of the characteristics of automatic indexing that we identified in Chapter 3. Both the Collexis Engine and LOHAI use linguistic approaches and are structurally comparable.

All in all, the results are promising. Most assignments are correct, even if a human indexer would not assign all of them. The indexing quality correlates with the employed weighting, especially assignments with lower rank often contain more common concepts that sometimes are just wrong. A lot of these mistakes could be avoided if the thesaurus would be more precise about homonyms and would provide additional information to disambiguate them, when necessary. The indexer could be further improved, e.g. common concepts should not be assigned, if more specific concepts down the tree are found in the text (Like LAW and CONTRACT LAW above). On the other hand, we wanted to keep it simple. Such adaptions and improvements are easy to implement, if they are needed.

Title	Contractarianism: Wistful Thinking
Authors	Hardin, Russell
Abstract	The contract metaphor in political and moral theory is misguided. It is a poor metaphor both descriptively and normatively, but here I address its normative problems. Normatively, contractarianism is supposed to give justifications for political institutions and for moral rules, just as contracting in the law is supposed to give justification for claims of obligation based on consent or agreement. This metaphorical association fails for several reasons. First, actual contracts generally govern prisoner's dilemma, or exchange, relations; the so-called social contract governs these and more diverse interactions as well. Second, agreement, which is the moral basis of contractarianism, is not right-making per se. Third, a contract in law gives information on what are the interests of the parties; a hypothetical social contract requires such knowledge, it does not reveal it. Hence, much of contemporary contractarian theory is perversely rationalist at its base because it requires prior, rational derivation of interests or other values. Finally, contractarian moral theory has the further disadvantage that, unlike contract in the law, its agreements cannot be connected to relevant motivations to abide by them.
Journal	Constitutional Political Economy, 1 (2) 1990: 35-52

Figure 5.12: Document example.

Constitutional economics
Influence of government
Ethics
Theory

(a) Intellectual indexing

Contract Law (1.21)
Contract (0.76)
Social contract (0.64)
Law (0.51)
Politics (0.37)
Prisoner's dilemma (0.34)
Theory (0.32)
Rationalism (0.24)
Association (0.23)
Exchange (0.20)
Knowledge (0.19)
Government (0.16)
Information (0.12)

(b) LOHAI

Table 5.4: Example annotations: Intellectual indexing vs. LOHAI.

5.3 Related Work

Luhn (1957) envisioned an automatic indexing system already in 1957, where a thesaurus is built as part of the indexing process to encode documents in a uniform way. Luhn was very ambitious and sketches solutions for a lot of problems that arise from the ambiguous use of the natural language in written texts. Actually, Luhn wanted to replace traditional intellectual indexing by his system, something that has not been achieved even today – like with other fields of the strong artificial intelligence–, mainly due to the problems with the interpretation of natural language by computers. Nevertheless, the general approach that uses statistics as the frequency of words, formed the basis for later approaches and also influenced Salton and Lesk (1965), who presented the SMART retrieval system, which was a pioneer work and dominated the research on information retrieval for the next decades. The SMART system was a complete document retrieval system that used several hundred different methods to analyze documents and search requests. As such, it used various preprocessing steps on the text like stemming and other forms of language processing. Ultimately, it assigned thesaurus concepts to documents, either based on syntactical or statistical analysis. Silvester, Genuardi, and Klingbiel (1994) describe the MAI system, the machine-aided indexing system that is used at the National Aeronautic and Space Administration (NASA). MAI suggests thesaurus concepts based on an extensive knowledge base containing mappings from natural language to concepts and rules how to decide for a specific concept based on the context.

A completely different approach is the use of machine learning techniques to let the system learn the assignment rules from a training set. Such an approach is for instance used by KEA (Witten, Paynter, Frank, Gutwin, & Nevill-Manning, 1999) and KEA++ (Medelyan & Witten, 2006b), respectively, which introduces the assignment of KOS concepts, while KEA was built to extract arbitrary meaningful key phrases. Recently, Medelyan released Maui, a further improved indexing system that was developed as part of her thesis (Medelyan, 2009) and presented in (Medelyan, Perrone, & Witten, 2010).

In the medical domain, several projects exist that automatically assign concepts from the Medical Subject Headings (MeSH) to abstracts. The Medical Text Indexer is developed and presented by the National Library of Medicine as part of their Indexing Initiative.[12] The NLM investigates methods whereby automated indexing methods partially or completely substitute for current indexing practices. During the project, several prototypes and approaches were evaluated (cf. Kim, Aronson, and Wilbur (2001)). The actual Medical Text Indexer (MTI) is presented by Aronson et al. (2004) and uses machine learning techniques to learn from manually indexed documents. A detailed description is available (NLM, 2006). The application as supporting system for human indexers is well studied, results of a

[12] http://ii.nlm.nih.gov/

user survey are reported by Ruiz and Aronson (2007). Neveol et al. (2006) present an automatic indexer that is also intended to be used as preliminary indexer or final indexer after human revision. It is embedded in the CISMeF project,[13] a quality-controlled subject gateway, catalog and index of French-language Health Internet resources. With CADIS, the Computer Aided Document Indexing System, Kolar et al. present another approach that is strictly focused on the support of human indexers (Kolar, Vukmirovic, Basic, & Snajder, 2005).

The CERN provides a solution for digital libraries, the CDS Invenio Document Server[14] that contains a module for automatic indexing with a thesaurus that is presented by Montejo-Raez (2002). Ruch (2006) presents a different approach of assigning concepts to documents using a combination of pattern matcher and vector-space retrieval engine.

5.4 Conclusion

In this chapter, we shed light on implementational aspects that are nevertheless crucial for the theoretical findings in the other chapters. Semtinel is an elaborated framework and contains already a lot of functionality that was needed to implement the approaches presented in this thesis. The core functionality that is described in this chapter is stable and easy to use. The generalization of the Experiment API follows the modular approach of the statistical framework underlying the ICE-Map Visualization. A core requirement that results from its specific design is the abstraction from specific weight functions. We extended this modularity to the upper levels and ensured that every aspect is exchangeable that finally leads to a visualization. The Explanations are a consequent extension, as such a modular approach can lead to confusion how the different components affect the visualization. By introducing this API, we ensured that the whole process from the underlying data to the actual visualization is transparent to the user.

If Semtinel is further developed, then the next steps should focus on the usability of data integration. At the periphery, especially when it comes to import functionalities, the system is more prototypical, as we developed these plugins merely to suit our needs. The same holds true for the integration of indexing systems which could be much more generalized. Another aspect is the access to the documents. There is a document set viewer, but it is not very intuitive and provides only basic functionality. These aspects could all be improved to make sure that the first steps with Semtinel become easier. Currently, we usually prepare a database with the desired data sets and ship it together with the software to interested users. A wizard would be great that allows everyone to get started.

[13] http://www.cismef.org/
[14] http://cdsware.cern.ch/

LOHAI was developed out of necessity; the development of automatic indexing techniques is not the focus of this thesis. But to the best of our knowledge, there is no free indexer available that does not require any data preparation step or the creation of some training data. With LOHAI, we developed such an indexer by just using the standard approaches in natural language processing and information retrieval for the single steps in the indexing pipeline. Each step could be improved by employing new and more sophisticated approaches, but we intentionally restricted ourselves to the well-understood approaches that are state-of-the-art in information retrieval and natural language processing. We have shown in Chapter 3 that the indexer performs very well for our task of a proper KOS selection and we expect that the indexer would even be usable in more serious indexing projects. LOHAI is not a stupid indexer, it is a baseline indexer. All in all, the indexer consists of about 500 lines of code in Java, without the POS tagger and the Snowball stemmer.

Acknowledgements: Parts of this chapter have been published before. The automatic indexer LOHAI was presented at the Semantic Digital Archives Workshop 2011 (Eckert, 2011b).

Chapter 6

Conclusion

This thesis is entitled "Usage-driven KOS maintenance." There is a subtlety in this title that allows different interpretations. First, it means that the approaches presented in this thesis leverage information about the usage of concepts in indexing processes and about the usage of terminology in the documents to be indexed.

The title, however, also implies that KOS maintenance should be usage-driven, as otherwise these approaches would be of no help for the KOS maintainer. In information retrieval this is obviously true: fine-grained and elaborated parts of a KOS that are not used to annotate documents do not help the user to find the actually available documents. These parts are merely a waste of effort on the side of the KOS maintainer, as too fine-grained KOSs usually hardly affect the retrieval result negatively – an exception is the necessary disambiguation of additional homonyms during (automatic) indexing and during the retrieval. Worse are too coarse-grained parts of a KOS where many documents are available, as this directly affects the precision of the retrieval result. Usage-driven KOS maintenance in general, however, is not limited to information retrieval. The definition of *usage* only has to be adapted. For example, consider KOSs that are used as background-knowledge in applications like spell-checking, machine translation, or to control synonyms and variants in search applications. Frequently involved areas should be extended, i.e., statistics should be captured about the employed concepts.

At last, "usage-driven" presumes that KOSs are actually used. There is no doubt that systems for knowledge organization are needed. Applications are legion and range from information retrieval over computational linguistics to artificial intelligence. On the other hand, KOSs are legion, too. They have been developed based on a specific, local demand and some may lack maintenance due to capacity problems. An unmaintained KOS, however, is not usable for any application. The title of this thesis could therefore be reversed: Maintenance-driven KOS usage. The bottom line is that usage-driven KOS maintenance helps to balance supply and

demand; and it helps to use the precious – not to say costly – time of the KOS maintainers economically.

In this thesis, we examined the whole life cycle of a KOS, from the first creation, over the inevitable maintenance, to the possible reuse in a different scenario, as well as the actual use of the KOS for document indexing. For all these steps, we developed suitable approaches and evaluation techniques that make usage-information available to the KOS maintainer. In Chapter 1, we raised three research questions that we answer in the following based on our findings.

Research Question 1: How can the structure and usage of a concept hierarchy be visualized in a way that it provides meaningful information to the practitioner?

This is the motivating question for the development if the ICE-Map Visualization in Chapter 2. The central point of the visualization needs to be the concept hierarchy. It is the constant in the whole process of document organization and all efforts of human experts regarding the understanding and maintenance should be put into the concept hierarchy as a representation of the expert's background knowledge.

Concept hierarchies are trees and as such face the problem of exponential growth with increasing depth. This makes a complete visualization difficult. The treemap visualization not only deals with this problem, but at the same time allows the proper visualization of analysis results using coloring and the size of the fields. This makes it the perfect starting point for our purpose. The treemap works as a view from above on the concept hierarchy. It does not show everything; to reveal more details, you have to zoom into it. For the big picture, the underlying analysis and the resulting colorization needs to be designed in a way that results are propagated towards higher levels in the hierarchy, as it is the case for the statistical framework underlying the ICE-Map Visualization.

While the statistical framework of the ICE-Map Visualization is designed independently of a concrete use case, we provided weight-functions specifically tailored to the usage of concepts in indexing processes. We demonstrated how different applications are supported by different weight functions. For example, we used the weight function based on weighted concept assignments for the KOS evaluation based on topical overlap; for one to make sure that the topical focus is reflected correctly, but also to smooth out systematic errors of the automatic indexer that usually lead to assignments with low weights. In contrast, a discrete weight function based on the number of annotations is to be used to evaluate the characteristics of an indexing process, because in this case the systematic errors (or characteristics) have to be detected. We applied the ICE-Map Visualization for both use cases in Chapter 3. The datasets are specifically chosen to demonstrate the functionality and applicability of the ICE-Map Visualization, i.e., we employed KOSs and in-

dexing processes where the topical overlaps as well as the characteristics of the document sets and the indexing processes are known beforehand.

Besides general demonstrations on conferences, we presented the ICE-Map Visualization to practitioners, namely to the STW thesaurus maintainers of the German National Library of Economics (ZBW) and to the Department of Acquisition and Cataloging of the University Library Leipzig (UBL). One aspect regarding the efficiency of the ICE-Map Visualization is proven by the fact that in both cases, we could discuss specific details of the thesaurus, the indexing practice, and the focus of the collections with experienced practitioners immediately, without in-depth preparation or any kind of prior knowledge.

All participants were impressed by the power of the visualization and were interested in experimenting further as well as using it for their own research. The visualization admittedly did not reveal many facts that were not already known by the maintainers. This does not come as a surprise, as practitioners with such a huge experience know their KOS and their document collections like the back of their hand. While this confirms the applicability of our approach, this also indicates its main potential: It significantly reduces the barrier for new KOS maintainers and helps them to quickly get familiar with the characteristics of a KOS.

This becomes even more important considering that today the creation and maintenance of an own KOS is only one side of the coin. The creation of concordances to other KOSs is important, too, and part of many projects in libraries. In this case, making oneself familiar with an unknown KOS quickly becomes essential. Regarding the maintenance of a collection as in the case of the UBL, we see a similar scheme: Maintaining one's own collection is one part, yet new collections, as commercial databases of publications, have to be evaluated regarding their topical relevance; a task that can easily be supported by the ICE-Map Visualization.

Research Question 2: To which extent can and should alternative, usage-driven approaches be applied for the creation and maintenance of concept hierarchies?

Besides the visualization of concept usage, we developed and evaluated semi-automatic approaches based on usage information. We stated in the introduction that the fully-automatic creation of concept hierarchies is probably impossible due to the lack of a true world understanding in artificial intelligence. The proper support of a human expert, however, makes use of the same techniques that can and have been used to create KOSs automatically. While the maintainer still makes the actual decisions about specific maintenance steps, the limits of automatic techniques apply. In the light of new, inexperienced maintainers who rely on approaches like the ones presented in this thesis, the answer to the question to which extent the creation and maintenance *can* be automated cannot be given without considering the question, to which extent the creation and maintenance *should* be automated.

One aspect of the question can be answered as follows: if no KOS and only very limited resources for its creation and maintenance are available, the application of automatic or alternative approaches is unavoidable and recommended. In Chapter 4, we developed a virtually fully automatic system by means of crowdsourcing, i.e., from the user's perspective, a complete concept hierarchy is created from scratch. The resulting quality, however, mainly results from the input of human workers in the crowd. We have shown that by simple filtering and weighting of the workers based on the given answers to questions where the correct answer is known, a quality comparable to domain experts can be achieved (Section 4.1).

As the creation is always only the first step and the start of the KOS life cycle, more emphasis lies on the subsequent KOS maintenance, for which we identified the following tasks:

1. Adaptation of the concept hierarchy to changes in the vocabulary of the domain of interest by means of adding of new terms or concepts,
2. splitting, extension, or restriction of extensively used concepts,
3. deletion and/or merging of rarely used concepts,
4. review of the hierarchical structure to avoid extensive subclassing, and
5. identification of problematic concepts for the indexing software, i.e., concepts that are erroneous assigned or missing.

The extension of a concept hierarchy is a promising field for automated support. In Section 4.2, we have confirmed that the identification of new concepts – i.e., meaningful terms – is possible with a high accuracy. The problem lies in the proper allocation of the new terms within the existing concept hierarchy. By exploiting the search snippets of web search engines, we can provide candidate locations to the user, thus simplifying the KOS maintenance. This task, as well as our approach, is related to the creation of a KOS. Instead of workers in the crowd, webpages are used to gather evidence for a specific relation between terms and concepts. Likewise, the crucial part is the proper evaluation of the evidence to reach the desired quality. As the evidence is much more complex in the case of webpages, a machine learning approach had to be used to evaluate and combine the various pieces of evidence. This would be applicable for the evaluation of the crowdsourcing results as well. Machine learning, however, requires the existence of a training set, which makes it more suitable for the maintenance, where the existing KOS can be used to derive such a training set. A special advantage of our crowdsourcing approach is that no training data or gold standard is needed.

Regarding the second task, we transferred the splitting of a concept into the task of clustering the documents below an overpopulated concept to identify new subconcepts (Section 4.3). In this case, the naming of the new concepts is a main challenge. We developed a workflow that allows the maintainer to control the

splitting and naming process. Extensively used concepts can easily be spotted by means of the ICE-Map Visualization.

The ICE-Map Visualization also supports the remaining three tasks: rarely used concepts can be identified and the whole structure can easily be visualized and reviewed. The last task is special, as we added it to the maintenance tasks due to the new use of KOSs in combination with automatic indexing. As part of this answer, we postulate that the maintenance of KOSs should focus much more on their applicability for automatic indexing. We further elaborate on this task in the answer to the next question.

First, however, we consider to which extent the creation and maintenance *should* be automated. To answer this question, we extend the shell model of Krause (2006) regarding the quality of KOSs employed for document indexing. Krause basically states that high quality content analysis is only feasible for a core of highly relevant documents. Other documents with less relevance form shells around the core. With increasing distance from the core, the documents become less relevant and less effort can and should be invested in the content analysis and the indexing. In other words, a lower quality becomes acceptable, e.g., by translating indexing results from different vocabularies. This way, Krause provides guidance for the application of automatic and alternative indexing techniques – which he considers unavoidable – while he emphasizes the importance of intellectual indexing to retain the necessary quality for the core. Krause (2006, p. 100) concludes: "Whether considered right or wrong, the paradigm of forcing homogeneity by overarching standardization efforts is no longer sufficient."

Figure 6.1 visualizes our extended shell model based on Krause (2006), with a new dimension of KOS quality. By adding this dimension to the shell model, we can distinguish the KOS quality and the quality of the content analysis (i.e., the indexing process), as both are independent factors. The techniques considered in this thesis are located as examples in this diagram. Near the core, i.e., the point of origin, is the traditional library catalog using intellectual indexing with intellectually maintained KOSs. Going to the right, processes with less quality are shown, like author-supplied concepts or KOS-based automatic indexing. At the same time, the quality of the KOS can decrease as well, e.g., by creating it (semi-)automatically or simply putting less effort into the maintenance. The extreme example is full text indexing, where no intellectual effort and no KOS are involved. More sophisticated techniques, e.g., semantic search applications, improve the quality by creating KOSs automatically; these are therefore closer to the center. Tagging is special: the rationale for its position is that usually basic structures are derived from the existing tags, like tag recommendations or spelling-corrections to ensure a minimal control.

These considerations leave the question open, which examples could be used for the remaining upper left area of the diagram. The extreme would be an automatically generated KOS that is used by domain experts for high quality intellectual

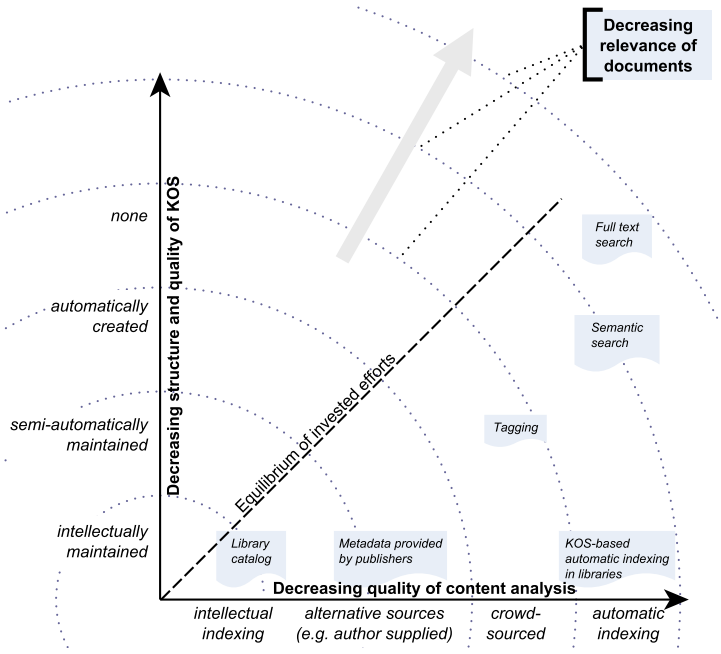


Figure 6.1: Extended shell model based on Krause (2006).

indexing. This does not happen for a comprehensible reason: intellectual indexing is a cumbersome task that does not scale. If this effort is taken, it has to be assured that the quality is not compromised by issues of the underlying KOS. In return, intellectual efforts are invested best into the creation and maintenance of the KOS, as this is the constant factor in the retrieval system. Therefore, we introduce the equilibrium line of invested efforts that answers this part of the question: The creation and maintenance of a concept hierarchy can and should be automated, as long as the reduced quality is in line with the reduced effort and quality of the content analysis and the reduced relevance of the documents to be indexed.

Research Question 3: What are the characteristics of different indexing processes regarding concept usage and how does that affect KOS maintenance?

In Chapter 3, we investigated three different indexing approaches by means of the ICE-Map Visualization: intellectual indexing, automatic indexing, and tagging.[1] We have demonstrated that the ICE-Map Visualization is a powerful means to visualize indexing results as a whole enabling the user to monitor and evaluate an indexing process efficiently. We could see that automatic indexing strongly depends on the labels assigned to the concepts in the KOS, as a concept can only be

[1] Our evaluation of tagging is limited, as we restricted the possible tags on terms available in the KOS and evaluated tags provided by only one user (cf. Chapter 3). The evaluation of a single tagger is interesting as it is the comparison of a layman to an information professional. The strength of tagging, however, lies in the cooperation of many taggers.

recognized if a label occurs in the text. There were many systematic errors, partly due to weaknesses of the indexer, partly due to misleading labels in the KOS that cannot be used without a proper word sense disambiguation.

Tagging is not as prone to wrong assignments as automatic indexing, but still it can be seen that usually simpler terms occurring actually in the text are preferred. This suggests that more labels should be added to make the whole KOS more accessible. In turn, this improves the information retrieval as well, as more search terms of users can be associated to KOS concepts during the retrieval.

This shows that usage characteristics of indexing processes have to evaluated and constantly monitored. Efficient monitoring has to concentrate on the concept hierarchy as the only constant in the process. The main effort regarding the improvement of the indexing processes should be invested in this background knowledge.

Today, this simple principle is not yet commonly used among practitioners. When an automatic indexing process produces errors, the first place to seek for improvements should be the concept hierarchy. Instead, usually a second source of background knowledge is created that contains additional rules and exceptions specifically formulated for the automatic indexer. To give an example, KOSs need information which concepts contain potentially ambiguous terms like homonyms, together with information, how to distinguish them, if possible. By this, the automatic indexer can use the information to perform a word-sense disambiguation or, if this is not possible, refrain from assigning this concept. Currently, most KOSs are created by domain experts to be used by domain experts. There are scope notes that can contain such information to warn the human indexer. What we need are machine-interpretable scope notes.

6.1 Summary of Contributions

To find answers to these research questions was the scientific goal of this thesis. In pursuing this goal, we particularly made the following theoretical and practical contributions.

The ICE-Map Visualization. In Chapter 2, we presented our main theoretical contribution: the ICE-Map Visualization. It is based on a sound statistical framework and the established treemap visualization. We demonstrated the applicability of this visualization in this thesis for the evaluation of indexing results, the evaluation of concept hierarchies, the selection of a suitable concept hierarchy, and the exploration of huge document collections.

While the ICE-Map Visualization was developed as an answer to our first research question, we generalized the statistical framework from concept usage by

basing it on general weight functions. Thus, the ICE-Map Visualization can be employed for various purposes, as long as it is possible to define a weight for a concept. For instance, by assigning a weight to each concept based on the money that was spent on literature indexed with this concept, a visualization can be created that shows how the money is distributed over different topics in the KOS. Then, publications of the institution can be indexed, like for the calculation of the topical overlap in Chapter 3. The result can be compared against the money distribution, resulting in a visualization that can be interpreted as an indicator if the money is spent wisely, for instance as part of a portfolio analysis.

While such an application would be highly controversial among scientists, librarians, and administrators, it certainly would lead to interesting insights. This example shows strikingly that informed decisions still are decisions that have to be made under consideration of all available information and aspects. The ICE-Map Visualization supports this. However, it rarely gives a simple answer but mostly leaves the interpretation to the user. We argue that this makes it valuable and superior to any kind of numbers game.

Application of the ICE-Map Visualization for the proper selection of a KOS. In Chapter 3, we combined the ICE-Map Visualization with a simple automatic indexer to calculate and visualize the topical overlap of a KOS and a set of documents. We demonstrated how unknown KOSs and document collections can be explored and evaluated, with the main goal to select a proper KOS for retrieval applications. The challenge was to develop an approach that does not require prior knowledge or elaborate data preparation. Regarding potential errors of the automatic indexers, we found that they mostly are smoothed out when the visualization is adjusted by means of the provided *tf-idf*-based weighting. This confirms the important role of such a weighting for information retrieval systems, especially when automatic indexing is involved, as systematic errors generally lead to low weights of the assignments, putting the wrongly assigned concepts close to typical stop words, i.e., words bearing no significance that are usually ignored for indexing purposes.

Comprehensive evaluation of indexing processes. We have demonstrated the use of the ICE-Map Visualization for the evaluation of indexing results, be they manually created by a librarian, automatically created, or created in a tagging-like environment by a layman. Beside the demonstration of the general usability of the ICE-Map Visualization for this task, we delivered interesting insights in the different characteristics of the indexing methods. All methods can and should be used to complement each other, especially for documents that are not indexed intellectually today, such as online publications. By means of the ICE-Map Visualization, the quality of such indexing processes can be monitored at large scale and thus the ICE-Map Visualization can significantly contribute to a successful employment.

Bootstrapping a KOS by means of crowdsourcing. In Chapter 4, we examined the characteristics of crowdsourcing in comparison to a volunteering user community. We were able to show that by means of redundancy, a quality can be achieved comparable to or even out-performing the volunteers. To achieve this, we designed an experiment for acquiring concept hierarchies from arbitrary web users using Amazon Mechanical Turk and compared the results provided by non-experts with the results reported in (Niepert et al., 2009). We proposed effective methods for filtering non-expert feedback based on quality-diagnosing questions and could show that the "wisdom of the crowd" can indeed be used to create a KOS from scratch.

Usage-driven KOS extension. The first and main task for KOS maintenance is the constant extension of the KOS by means of new terms. Therefore, we developed a method to identify relevant terms in a set of documents and to provide suggestions for locations in the KOS hierarchy. In this case, we specifically identified and adapted existing approaches to carry out the necessary steps. We applied these methods in a large-scale experiment and extended parts of the MeSH thesaurus with new terms extracted from documents. We presented detailed results on the use of web search engines as a means for generating feature sets for learning the correct relation of new and existing terms. Ultimately, we were able to narrow down the number of choices for the KOS maintainer drastically (on average 14 out of 1,797).

Usage-driven concept splitting. While our approach for KOS extension exploits terminology usage in documents, we investigated the possibility to exploit concept usage for indexing to split and name of concepts based on the assigned document. Therefore, we proposed a new implementation of the description-comes-first paradigm, preserving its advantages without putting the burden on the user to actually name the new concepts first. We integrated this approach with straightforward content-based k-Means clustering to ensure that the splitting of the concept best reflects the actual content of the documents and evaluated the method under laboratory conditions. We could show that the maintainer is able to create new subconcepts with matching documents assigned at an average precision of 96.5%.

Semtinel and LOHAI. Significant efforts were made to provide the necessary tools to support the approaches presented in this thesis. Details of these implementations are provided in Chapter 5. The main practical contribution is the definition of the framework Semtinel that is used to implement most of the approaches developed in this thesis. It focuses on the creation and maintenance of KOSs, as well as the analysis and evaluation of the KOS usage in indexing processes. The second implementation described is LOHAI, a complete, simple, and knowledge-poor indexer that uses standard approaches in natural language processing and information retrieval and does not need any kind of training data. The indexer is specifically

tailored to the requirements of the KOS selection in Chapter 3. LOHAI, Semtinel, and all included approaches are published under an open source license.[2]

6.2 Future Work

This thesis provides a number of starting points for future work, both on the level of the big picture and on the level of details for the single approaches presented. To start with the former, the whole methodology of usage analysis could be extended to the retrieval part, i.e., information need of the users expressed by search queries could be taken into account. There are two aspects of this idea: first, the search queries can be exploited as another source for usage information to further enhance the KOS with the goal to reflect the terminology that is used for searching within the KOS. Second, this information can be used to visualize the topical overlap of the documents available in the system and the information need of the users.

When we presented the ICE-Map Visualization, a frequently asked questions was, if it could be used to visualize retrieval results. While this is not in the scope of this thesis, it is possible to visualize the topical overlap of the documents in the retrieval result, either using existing annotations or by indexing them with LOHAI – provided that at least abstracts are available. The question is if such a visualization supports the user in navigating the result set, possibly to narrow down the results, e.g., by selecting relevant areas in the KOS.

Regarding the visualization of indexing results, a promising extension would be the visualization of differences between more than two sources, e.g., the differences of annotations at several points in time. Currently, this is supported for two points in time, as two annotation sets can be compared. Semtinel already supports the provision of a series of annotation sets. So far, however, we did not use it. One idea is to directly visualize interesting concepts, e.g., by simply summarizing all information contents. The hypothesis is that concepts whose value constantly increases or decreases would become visible, while concepts with oscillating values would have results close to zero. A totally different approach would be an animated visualization of the continuous changes. There are several potential use cases: Based on publication dates and with a sufficiently big document base, global changes in research activities can be identified, and therefore areas in the KOS that need more attention. Based on the date of purchase or search queries, the result rather reflects the local research activities and could be used for portfolio maintenance.

A further application of the ICE-Map Visualization is the visualization of alignments between two KOSs, i.e., relations between two concepts of two different KOSs. Currently, it is possible to visualize the distribution of alignments for each

[2] http://www.semtinel.org/

KOS separately. With an approach to determine the overlap of two KOSs at a coarse-grained level (possibly an extension of our approach for KOS selection), it could be determined, which parts of the KOS probably need further alignments. There is certainly a demand for such approaches for *Usage-driven KOS Alignment*.

For the application of the approaches presented in this thesis, we envision an integration alongside the extended shell model as introduced above. This means that systems have to be developed that cannot only deal with different qualities of KOSs, but even with different qualities within one KOS. This requires the management and utilization of provenance information on a detailed level. Concepts that have been added to a KOS automatically should only be used for automatic indexing. Reviewed parts, e.g., by means of crowdsourcing, can be used for applications like tagging or as background knowledge for NLP applications. The idea is that all these extensions and results of (semi-)automatic maintenance serve as supporting means for the actual KOS maintainer who approves them and this way maintains the KOS with the desired high quality. Only this certified core of the KOS should be used for intellectual indexing by domain experts. The realization of such an integrated system comprises many interesting starting points for future research.

For the approaches presented in Chapter 4, we also envision various potential improvements. Regarding the crowdsourcing of the KOS creation, possible next steps include further refinement of the learning process using Amazon Mechanical Turk and a transfer of this approach from the domain of philosophy to other domains. Another promising avenue of future work is the employment of more sophisticated algorithms such as machine learning to classify MTurk workers according to their feedback quality. It would be interesting to extend the approach to continuous KOS maintenance, e.g., by including the workers for decisions about the placement of new concepts or by asking them about proper names for new concepts. Maybe it is possible to design *games with a purpose* (Ahn & Dabbish, 2008) for one or more of the maintenance tasks or to find other exploitable sources of the wisdom of the crowds.

For our approach of KOS extension using web search engines, we identified the time it takes to query the web search engine as a bottleneck. We reduced the number of pairs by using a co-occurrence similarity measure. It would be worthwhile to investigate additional methods to reduce the number of concept positions that have to be visited in the KOS. For instance, having strong evidence that a candidate concept is not a hyponym of a concept c we can immediately infer that it can also not be a hyponym of any of c's descendants. This would allow us to prune entire sub-trees in the KOS, drastically reducing the number of pairs that have to be send to the web search engine. Another idea is to not only apply shallow parsing strategies to extract lexical pattern but also more sophisticated approaches such as part-of-speech tagging and deep syntax parsing.

The approach for splitting and naming of concepts has to be applied to real KOSs. Based on our experiments, we expect that it works well for classifications,

where a clear distinction of documents is desired. This is a result of the employed clustering algorithm that separates the documents into disjunct clusters. For KOSs that do not focus on classificatory aspects, fuzzy clustering algorithms could be promising, as they would be able to extract concepts from the documents without requiring the documents to strictly belong to one of these concepts and to none of the others.

Most of the approaches are implemented within Semtinel – with the exception of the KOS creation based on crowdsourcing and the concept splitting. To make all approaches directly usable for KOS maintainers, all of the approaches have to be integrated. This is not difficult, as all implementations are available, but only the starting point. For this thesis, we have put all effort into the usability of Semtinel for the development and employment of evaluation techniques like the ICE-Map Visualization. A lot of work needs to be done to make the integration of own KOSs and the access to the necessary data more intuitive. For a practical application, an integration with existing KOS maintenance systems and library systems is desirable. By releasing Semtinel under an open source license, we express the hope that its development continues towards this goal.

6.3 Final remarks

Based on our findings, we close this thesis with five short postulations:

1. **Focus on the KOS:** Intellectual effort in an information retrieval system is invested best in the maintenance of the KOS, as this is the constant factor that reflects the domain knowledge.
2. **KOSs for automatic indexing:** KOSs should be enhanced to be applicable for automatic indexing. Machine-interpretable scope notes are needed.
3. **Flexible usage and maintenance:** KOSs have to be extended and adapted automatically or by alternative means, despite the lack of quality, to quickly reflect changes in the domain. These versions can be used for automatic indexing approaches and function as input for the actual KOS maintainer.
4. **Interpretable visualizations instead of numbers games:** Only meaningful, interpretable visualizations help the user to understand the characteristics of the KOS and the indexing process. Bare numbers pretend objectivity, but do not show the actual problems and therefore rarely lead to informed decisions.
5. **Usage-driven KOS Maintenance:** Maintenance decisions should be driven by the analysis and visualization of concept usage to ensure that the effort is targeted at the actual application of the KOS.

This thesis has been created at the intersection of computer science and information science. Working in the border zone of a field, where it intersects with

another one, is exciting and often very fruitful. It allows the researcher to approach the other field with a disarming naivety – in a positive sense of an unburdened, innocent approach – and at the same time to bring new and different ways of thinking into this field.

In the introduction of Chapter 2, we told the story how we tried to evaluate a thesaurus-based automatic indexing system and how it turned out that the established evaluation techniques were not satisfying. Our main questions were not answered: "Where are the problems?", "How can we improve such a system?" At this time, we made our first steps in the library world. We first needed to see the thesaurus and we wanted to see the indexing result. Hence, we created the first prototype of the ICE-Map Visualization and it opened the door that allowed us to see; it gave us insights that otherwise only would have been possible with years of experience. Maybe this is the main strength of the ICE-Map Visualization: it allows the user to assume the position of the outsider, to step back from the details. It facilitates the understanding of the big picture, which is sometimes easier with a hint of naivety.

References

All online resources were checked for availability on April 20th, 2012. Links to articles, where available, are given for convenience only but might no longer be available or refer to slightly different versions (pre-prints) and thus should be used with care.

Ahn, L. von, & Dabbish, L. (2004). Labeling images with a computer game. In *CHI'04: Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 319–326). New York, NY, USA: ACM.

Ahn, L. von, & Dabbish, L. (2008). Designing games with a purpose. *Communications of the ACM*, *51*(8), 58–67. Available from http://doi.acm.org/10.1145/1378704.1378719

Ahn, L. von, Kedia, M., & Blum, M. (2006). Verbosity: A Game for Collecting Common-Sense Knowledge. In *ACM Conference on Human Factors in Computing Systems, CHI Notes.* New York, NY, USA: ACM.

Ahn, L. von, Maurer, B., McMillen, C., Abraham, D., & Blum, M. (2008). reCAPTCHA: Human-Based Character Recognition via Web Security Measures. *Science*, *321*, 1465–1468.

Aitchison, J. (1970). The thesaurofacet: a multipurpose retrieval language tool. *Journal of Documentation*, *26*(3), 187–203.

Aitchison, J., & Dextre Clarke, S. (2004). The Thesaurus: A Historical Viewpoint, with a Look to the Future. *Cataloging & Classification Quarterly*, *37*, 5–21. (Printed as Book: The Thesaurus - Review, Renaissance and Revision)

Aitchison, J., Gilchrist, A., & Bawden, D. (2004). *Thesaurus construction and use: a practical manual*. Europa Publications, Abingdon.

Alonso, O., Rose, D. E., & Stewart, B. (2008). Crowdsourcing for relevance evaluation. *SIGIR Forum*, *42*(2), 9-15.

Altenhöner, R., Hannemann, J., & Kett, J. (2010). Linked Data aus und für Bibliotheken: Rückgratstärkung im Semantic Web. In M. Ockenfeld (Ed.), *Semantic Web & Linked Data - Elemente zukünftiger Informationsinfrastrukturen, 1. DGI-Konferenz, 62. Jahrestagung der DGI, Frankfurt am Main, 7. bis 9. Oktober 2010.* Deutsche Gesellschaft für Informationswissenschaft und Informationspraxis, Frankfurt.

Aronson, A. R., Mork, J. G., Gay, C. W., Humphrey, S. M., & Rogers, W. J. (2004). The NLM Indexing Initiative's Medical Text Indexer. In *Medinfo 2004* (pp. 268–272). Amsterdam, NL: IOS Press. Available from http://citeseer.ist.psu.edu/aronson04nlm.html

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Boston, MA, USA: Addison Wesley. Available from http://people.ischool.berkeley.edu/~hearst/irbook/

Baker, S. (2010). *Helping computers understand language.* Published online. Available from http://googleblog.blogspot.com/2010/01/helping-computers-understand-language.html

Barlow, T., & Neville, P. (2001). A Comparison of 2-D Visualizations of Hierarchies. In *Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS'01).*

Bederson, B. B., Shneiderman, B., & Wattenberg, M. (2002). Ordered and quantum treemaps: Making effective use of 2D space to display hierarchies. *ACM Transactions on Graphics*, *21*(4), 833–854. Available from http://hcil.cs.umd.edu/trs/2001-18/2001-18.pdf

Bollegala, D., Matsuo, Y., & Ishizuka, M. (2007). Measuring semantic similarity between words using web search engines. In *Proceedings of the WWW 2007, May 8-12, 2007, Banff, Alberta, Canada.* (pp. 757–766).

Borgman, C. L. (1996). Why are online catalogs still hard to use? *Journal of the American Society for Information Science*, *47*(7), 493–503. Available from http://dx.doi.org/10.1002/(SICI)1097-4571(199607)47:7⟨493::AID-ASI3⟩3.0.CO;2-P

Brabham, D. C. (2008). Crowdsourcing as a model for problem solving: An introduction and cases. *The International Journal of Research into New Media Technologies*, *14 (1)*, 75–90. Available from http://con.sagepub.com/cgi/content/abstract/14/1/75

Brank, J., Grobelnik, M., & Mladenic, D. (2008). Predicting Category Additions in a Topic Hierarchy. In *The Semantic Web - 3rd Asian Semantic Web Conference, ASWC 2008, Bangkok, Thailand, December 8-11, 2008. Proceedings.* (pp. 315–329).

Broughton, V. (2006). The need for a faceted classification as the basis of all methods of information retrieval. *Aslib Proceedings: New Information Perspectives*, *58*, 49–72.

Bruls, M., Huizing, K., & Wijk, J. J. van. (2000). Squarified treemaps. In *Joint Eurographics and IEEE TCVG Symposium on Visualization, IEEE Computer Society* (pp. 33–42). Available from http://www.win.tue.nl/~vanwijk/stm.pdf

Buchanan, B. (1979). *Theory of library classification*. München, London, New York, Paris: C. Bingley / K.G. Saur.

Calmet, J., & Daemi, A. (2004a). *Assessing Conflicts in Ontologies* (Tech. Rep.). IAKS Calmet, University Karlsruhe (TH), Germany. Available from http://avalon.ira.uka.de/iaks-calmet/papers/WSEAS_2004.pdf

Calmet, J., & Daemi, A. (2004b). *From entropy to ontology* (Tech. Rep.). Institute for Algorithms and Cognitive Systems (IAKS), University of Karlsruhe (TH), Germany. Available from http://iaks-www.ira.uka.de/calmet/papers/AT2AI4.pdf

Campbell, D. G., & Fast, K. V. (2004). Panizzi, Lubetzky, and Google: How the modern web environment is reinventing the theory of cataloguing. *Canadian Journal of Information & Library Sciences*, *28*(3), 25–38. Available from http://www.redi-bw.de/db/ebsco.php/search.ebscohost.com/login.aspx?direct=true&db=lxh&AN=17663772&site=ehost-live

Cimiano, P., Pivk, A., Schmidt-Thieme, L., & Staab, S. (2004). Learning taxonomic relations from heterogeneous sources. In *Proceedings of the ECAI 2004 Ontology Learning and Population Workshop.* Available from citeseer.ist.psu.edu/cimiano03learning.html

Curran, J. R. (2002). Ensemble methods for automatic thesaurus extraction. In *EMNLP '02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing* (pp. 222–229). Morristown, NJ, USA: Association for Computational Linguistics.

DaCosta, D., & Venturini, G. (2006). An interactive visualization environment for data exploration using points of interest. In *Advanced Data Mining and Applications, Second International Conference, ADMA 2006, Xi'an, China, August 14-16, 2006, Proceedings* (pp. 416–423).

Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., & Harshman, R. A. (1990). Indexing by latent semantic analysis. *JASIS*, *41*(6), 391–407.

DIN. (1987). *DIN 32705: Klassifikationssysteme; Erstellung und Weiterentwicklung von Klassifikationssystemen (engl: Classification systems; establishment and development of classification systems).*

Downs, J. S., Holbrook, M. B., Sheng, S., & Cranor, L. F. (2010). Are your participants gaming the system? Screening mechanical turk workers. In *CHI '10 Proceedings of the 28th international conference on Human factors in computing systems* (pp. 2399–2402). New York, NY, USA: ACM.

Eckert, K. (2011a). *The ICE-Map Visualization* (Tech. Rep. No. TR-2011-003). University of Mannheim, Department of Computer Science. Available from http://ub-madoc.bib.uni-mannheim.de/29611

Eckert, K. (2011b). LOHAI: Providing a baseline for KOS based automatic indexing. In *Proceedings of the first International Workshop on Semantic Digital Archives (SDA) at the International Conference on Theory and Practice of Digital Libraries (TPDL) 2011, Sep 29 2011, Berlin.* Available from http://ceur-ws.org/Vol-801/

Eckert, K., Hänger, C., & Niemann, C. (2009). Tagging and Automation - Challenges and Chances for Academic Libraries. *Library Hi Tech*, *27*(4). Available from http://dx.doi.org/10.1108/07378830911007664

Eckert, K., Meusel, R., & Stuckenschmidt, H. (2011). User-centered Maintenance of Concept Hierarchies. In W. Wong, W. Liu, & M. Bennamoun (Eds.), *Ontology learning and knowledge discovery using the web: Challenges and recent advances.* IGI Global.

Eckert, K., Niepert, M., Niemann, C., Buckner, C., Allen, C., & Stuckenschmidt, H. (2010). Crowdsourcing the Assembly of Concept Hierarchies. In *Proceedings of the Joint Conference on Digital Libraries JCDL-2010, Brisbane, Australia.*

Eckert, K., Ritze, D., & Pfeffer, M. (2012). Does it fit? KOS evaluation using the ICE-Map visualization. In *Proceedings of the 9th Extended Semantic Web Conference (ESWC12), May 27-31, 2012, Heraklion, Greece.*

Eckert, K., Stuckenschmidt, H., & Pfeffer, M. (2007). Interactive Thesaurus Assessment for Automatic Document Annotation. In *Proceedings of The Fourth International Conference on Knowledge Capture (K-CAP 2007), Whistler, Canada.*

Eckert, K., Stuckenschmidt, H., & Pfeffer, M. (2008). Semtinel: Interactive Supervision of Automatic Indexing. In *JCDL '08: Proceedings of the 2008 conference on Digital libraries.* New York, NY, USA: ACM.

Euzenat, J. (2007). Semantic Precision and Recall for Ontology Alignment Evaluation. In *IJCAI* (pp. 348–353). Available from http://ijcai.org/papers07/Papers/IJCAI07-054.pdf

Eversberg, B. (2002). Grundsätze und Ziele des Katalogisierens. In *Die Bibliothek zwischen Autor und Leser : 92. Deutscher Bibliothekartag in Augsburg 2002* (pp. 113–126). Klostermann, Frankfurt. Available from http://www.allegro-c.de/formate/tlcse.htm (The online version is a translated, newer version of the printed article and might be slightly different.)

Fellbaum, C., Tengi, R., Jose, L., Schulam, P., Julien, I., & Miller, G. A. (2010). WordNet 3.0 Reference Manual [Computer software manual]. Princeton, NJ, USA. Available from http://wordnet.princeton.edu/wordnet/documentation/

Fidel, R. (1991a). Searchers' Selection of Search Keys: II. Controlled vocabulary or free-text searching. *Journal of the American Society for Information Science*, *42*(7), 501–514.

Fidel, R. (1991b). Searchers' Selection of Search Keys: III. Searching styles. *Journal of the American Society for Information Science*, *42*(7), 515–527.

Fidel, R. (1991c). Searchers' Selection of Search Keys: I. The Selection Routine. *Journal of the American Society for Information Science*, *42*(7), 490–500.

Fluit, C., Sabou, M., & Harmelen, F. van. (2005). Ontology-based Information Visualisation: Towards Semantic Web Applications. In V. Geroimenko (Ed.), *Visualising the Semantic Web (2nd edition).* Heidelberg: Springer.

Gale, W. A., Church, K. W., & Yarowsky, D. (1992). One sense per discourse. In *Proceedings of the workshop on speech and natural language* (pp. 233–237). Stroudsburg, PA, USA: Association for Computational Linguistics. Available from http://dx.doi.org/10.3115/1075527.1075579

Gast, H. (2009). Towards a Modular Extensible Isabelle Interface. In S. Berghofer & M. Wenzel (Eds.), *Theorem Proving in Higher Order Logics - Emerging Trends Proceedings.* Technische Universität München. (Technical Report TUM-INFO-08-I0916-0/1.-FI)

Gilchrist, A. (2003). Thesauri, taxonomies and ontologies – an etymological note. *Journal of Documentation*, *59*(1), 7–18.

Gillam, L., Tariq, M., & Ahmad, K. (2005). Terminology and the Construction of Ontology. *Terminology*, *11*, 55–81.

Gordon-Murnane, L. (2006). Social bookmarking, folksonomies, and Web 2.0 tools. *Searcher: The Magazine for Database Professionals*, *14*(6), 26–38. Available from http://www.scopus.com/record/display.url?view=extended&origin=resultslist&eid=2-s2.0-33745173160

Granitzer, M., Kienreich, W., Sabol, V., Andrews, K., & Klieber, W. (2004). Evaluating a System for Interactive Exploration of Large, Hierarchically Structured Document Repositories. In *Infovis* (pp. 127–134). Available from http://dblp.uni-trier.de/db/conf/infovis/infovis2004.html\#GranitzerKSAK04

Greenberg, J. (2004). User Comprehension and Searching with Information Retrieval Thesauri. *Cataloging & Classification Quarterly*, *37*(3/4), 103–120.

Grefenstette, G. (1994). *Explorations in Automatic Thesaurus Discovery*. Heidelberg: Springer.

Gross, T., & Taylor, A. G. (2005). What have we got to lose? The effect of controlled vocabulary on keyword searching results. *College and Research Libraries*, *66*(3), 212–230.

Gruber, O., Hargrave, B. J., McAffer, J., Rapicault, P., & Watson, T. (2005). The Eclipse 3.0 platform: Adopting OSGi technology. *IBM Systems Journal*, *44*(2), 289–299.

Guy, M., & Tonkin, E. (2006). Tidying up Tags. *D-Lib Magazine*. Available from http://www.dlib.org/dlib/january06/guy/01guy.html (Online)

Hammond, R. (2001). Negotiating the medical maze. *The Library Association Record*, *103*(4), 218–220.

Hearst, M. A. (1992). Automatic Acquisition of Hyponyms from Large Text Corpora. In *Proceedings of the Fourteenth International Conference on Computational Linguistics, Nantes, France.*

Heymann, P., & Garcia-Molina, H. (2006). *Collaborative creation of communal hierarchical taxonomies in social tagging systems* (Tech. Rep. No. 2006-10). Computer Science Department. Available from http://dbpubs.stanford.edu:8090/pub/2006-10

Hildreth, C. R. (1995). *Online Catalog Design Models: Are We Moving in the Right Direction?* (Tech. Rep.). Palmer School of Library and Information Science, Long Island University. Available from http://myweb.cwpost.liu.edu/childret/clr-opac.html (A Report Submitted to the Council on Library Resources)

Hodge, G. (2000). *Systems of Knowledge Organization for Digital libraries. Beyond traditional authority files* (Tech. Rep. No. CLIR Pub91). The Council on Library and Information Resources, Washington, DC. Available from http://www.clir.org/pubs/abstract/reports/pub91

Horton, J. J. (2011). The condition of the turking class: Are online employers fair and honest? *Economics Letters*, *111*(1), 10–12. Available from http://www.sciencedirect.com/science/article/pii/S0165176510004398

Howe, J. (2006). The Rise of Crowdsourcing. *Wired*, *14*(6). Available from http://www.wired.com/wired/archive/14.06/crowds.html

Hsueh, P.-Y., Melville, P., & Sindhwani, V. (2009). Data quality from crowdsourcing: a study of annotation selection criteria. In *HLT '09: Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing* (pp. 27–35). Morristown, NJ, USA: Association for Computational Linguistics.

ISO. (2011). *ISO/DIS 25964-1: Information and documentation – Thesauri and interoperability with other vocabularies – Part 1: Thesauri for information retrieval.*

Johnson, B. S. (1993). *Treemaps: visualizing hierarchical and categorical data*. Doctoral dissertation, College Park, MD, USA. (UMI Order No. GAX94-25057)

Johnson, S. (2006). *The Ghost Map - The Story of London's Deadliest Epidemic - and How It Changed the Way We Think about Disease, Cities, Science, and the Modern World*. Riverhead.

Kageura, K., Tsuji, K., & Aizawa, A. N. (2000). Automatic thesaurus generation through multiple filtering. In *COLING 2000: Proceedings of the 18th International Conference on Computational Linguistics* (pp. 397–403). Morristown, NJ, USA: Association for Computational Linguistics.

Kaji, N., & Kitsuregawa, M. (2008). Using hidden markov random fields to combine distributional and pattern-based word clustering. In *COLING '08: Proceedings of the 22nd International Conference on Computational Linguistics* (pp. 401–408). Morristown, NJ, USA: Association for Computational Linguistics.

Kekäläinen, J., & Järvelin, K. (2002). Using graded relevance assessments in IR evaluation. *Journal of the American Society for Information Science and Technology (JASIST)*, *53*(13), 1120–1129.

Kermanidis, K. L., Thanopoulos, A., Maragoudakis, M., & Fakotakis, N. (2008). Eksairesis: A domain-adaptable system for ontology building from unstructured text. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08).*

Kerr, D. (2012). *Google plans major revamp for search engine.* Published online. Available from http://news.cnet.com/8301-1023_3-57397782-93/google-plans-major-revamp-for-search-engine/

Khoo, C. S. G., & Na, J.-C. (2006). Semantic relations in information science. *Annual Review of Information Science and Technology*, *40*, 157–228.

Kim, W., Aronson, A. R., & Wilbur, W. J. (2001). Automatic MeSH term assignment and quality assessment. *Journal of the American Medical Informatics Association (JAMIA)*, 319–323. Available from http://view.ncbi.nlm.nih.gov/pubmed/11825203

Kittur, A., Chi, E. H., & Suh, B. (2008). Crowdsourcing user studies with Mechanical Turk. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems* (pp. 453–456). New York, NY, USA: ACM.

Koch, T. (2004). The Map as Intent: Variations on the Theme of John Snow. *Cartographica*, *39*(4), 1–13.

Kolar, M., Vukmirovic, I., Basic, B. D., & Snajder, J. (2005). Computer Aided Document Indexing System. In *Proceedings of the 27th International Conference on Information Technology Interfaces.*

Korfhage, R. R. (1991). To see, or not to see – is That the query? In *SIGIR '91: Proceedings of the 14th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 134–141). New York, NY, USA: ACM.

Krause, J. (2006). Shell Model, Semantic Web and Web Information Retrieval. In I. Harms, H.-D. Luckhardt, & H. W. Giessen (Eds.), *Information und Sprache: Beiträge zu Informationswissenschaft, Computerlinguistik, Bibliothekswesen und verwandten Fächern; Festschrift für Harald H. Zimmermann* (pp. 95–106). Munich: Saur.

Kruskal, J. B., & Landwehr, J. M. (1983). Icicle plots: Better displays for hierarchical clustering. *The American Statistician*, *37*(2), 162–168. Available from http://www.jstor.org/stable/2685881

Kuhlen, R., Seeger, T., & Strauch, D. (Eds.). (2004). *Grundlagen der praktischen Dokumentation und Information, Band 1*. Munich: Saur.

Kules, B., Capra, R., Banta, M., & Sierra, T. (2009). What do exploratory searchers look at in a faceted search interface? In *JCDL '09: Proceedings of the 9th ACM/IEEE-CS joint conference on Digital Libraries* (pp. 313–322). New York, NY, USA: ACM.

Kwasnik, B. H. (1999). The role of classification in knowledge representation and discovery. *Library Trends*, *48*(1), 22–47.

Lancaster, F. W. (2004). *Vocabulary control for information retrieval* (2nd ed.). Washington, D.C., USA: Information Resources Press.

Lehmann, S., Schwanecke, U., & Dörner, R. (2010). Interactive visualization for opportunistic exploration of large document collections. *Information Systems*, *35*(2), 260–269. Available from http://dx.doi.org/10.1016/j.is.2009.10.004

Luhn, H. P. (1957). A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of Research and Development*, *1*(4), 309–317.

Mai, J.-E. (2006). Contextual Analysis for the Design of Controlled Vocabularies. *Bulletin of the American Society for Information Science & Technology*, *33*, 17–19. Available from http://www.asis.org/Bulletin/Oct-06/mai.html

Manning, C. D., & Schuetze, H. (1999). *Foundations of Statistical Natural Language Processing*. Cambridge, Massachusetts, USA: MIT Press.

Matsuo, Y., Sakaki, T., Uchiyama, K., & Ishizuka, M. (2006). Graph-based word clustering using a web search engine. In *EMNLP '06: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing* (pp. 542–550). Morristown, NJ, USA: Association for Computational Linguistics.

Maynard, D. (2005). 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. Available from http://gate.ac.uk/sale/ahm05/ahm.pdf

McCray, A. T. (2006). Conceptualizing the world: Lessons from history. *Journal of Biomedical Informatics*, *39*(3), 267–273. Available from http://www.sciencedirect.com/science/article/B6WHD-4H6PP1M-1/2/65a082e05afa3b727bb87075b0b40e94

McCulloch, E. (2004). Multiple terminologies: an obstacle to information retrieval. *Library Review*, *53*(6), 297–300.

McCulloch, E. (2005). Thesauri: practical guidance for construction. *Library Review*, *54*, 403–409.

McGuffin, M., & Robert, J.-M. (2010). Quantifying the Space-Efficiency of 2D Graphical Representations of Trees. *Information Visualization*, *9*(2), 115–140.

McLeod, K. S. (2000). Our sense of Snow: the myth of John Snow in medical geography. *Social Science & Medicine*, *50*(7-8), 923–935. Available from http://www.sciencedirect.com/science/article/B6VBF-3YDG0NP-3/2/4aa97b73f3392c7b3110f84ba28decb7

Medelyan, O. (2009). *Human-competitive automatic topic indexing*. Doctoral dissertation, University of Waikato.

Medelyan, O., Perrone, V., & Witten, I. H. (2010). Subject Metadata Support Powered by Maui. In *Proceedings of the Joint Conference on Digital Libraries JCDL-2010, Brisbane, Australia.*

Medelyan, O., & Witten, I. H. (2006a). Measuring inter-indexer consistency using a thesaurus. In *JCDL '06: Proceedings of the 6th ACM/IEEE-CS joint conference on Digital Libraries* (pp. 274–275). New York, NY, USA: ACM.

Medelyan, O., & Witten, I. H. (2006b). Thesaurus based automatic keyphrase indexing. In *JCDL '06: Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries.*

Meusel, R., Niepert, M., Eckert, K., & Stuckenschmidt, H. (2010). Thesaurus Extension using Web Search Engines. In G. Chowdhury, C. Khoo, & J. Hunter (Eds.), *The Role of Digital Libraries in a Time of Global Change – 12th International Conference on Asia-Pacific Digital Libraries, ICADL 2010, Gold Coast, Australia, June 21-25, 2010, Proceedings.* Heidelberg: Springer.

Montejo-Raez, A. (2002). Toward conceptual indexing using automatic assignment of descriptors. In S. Mizzaro & C. Tasso (Eds.), *Proceedings of the AH 2002 Workshop on Personalization Techniques in Electronig Publishing, Malaga, Spain.* Available from http://users.dimi.uniud.it/~stefano.mizzaro/AH2002/proceedings/

Mulligen, E. M. van, Eijk, C. van der, Kors, J. A., Schijvenaars, B. J., & Mons, B. (2002). Research for Research: Tools for Knowledge Discovery and Visualization. In *Proceedings of the 2002 AMIA Symposium.*

Müller, F. von, & Goethe, J. W. von. (1870). *Goethes Unterhaltungen mit dem Kanzler Friedrich v. Müller* (C. A. H. Burckhardt, Ed.). Stuttgart: Cotta.

Navigli, R. (2009). Word sense disambiguation: A survey. *ACM Computing Surveys*, *41*(2), 1–69.

Nelson, P. E. (1992). Site Report for the Text REtrieval Conference. In *NIST Special Publication 500-207: The First Text REtrieval Conference (TREC-1)* (pp. 287–296).

Neubert, J. (2009). Bringing the "Thesaurus for Economic" on to the Web of Linked Data. In *Proceedings of the Workshop on Linked Data on the Web (LDOW) 2009, April 20, 2009, Madrid, Spain.*

Neveol, A., Rogozan, A., & Darmoni, S. (2006). Automatic indexing of online health resources for a french quality controlled gateway. *Information Processing and Management*, *42*, 695–709.

Nguyen, D. P. T., Matsuo, Y., & Ishizuka, M. (2007). Exploiting syntactic and semantic information for relation extraction from wikipedia. In *IJCAI Workshop on Text-Mining & Link-Analysis (TextLink 2007).*

Nielsen, M. L. (2004). Thesaurus construction: key issues and selected readings. *Cataloging & Classification Quarterly*, *37*, 57–74.

Niepert, M., Buckner, C., & Allen, C. (2007). A dynamic ontology for a dynamic reference work. In *JCDL '07: Proceedings of the 7th ACM/IEEE-CS joint conference on Digital Libraries* (pp. 288–297). New York, NY, USA: ACM.

Niepert, M., Buckner, C., & Allen, C. (2008). Answer set programming on expert feedback to populate and extend dynamic ontologies. In *Proceedings of the Twenty-First International Florida Artificial Intelligence Research Society Conference, May 15-17, 2008, Coconut Grove, Florida, USA* (pp. 500–505).

Niepert, M., Buckner, C., & Allen, C. (2009). Working the crowd: Design principles and early lessons from the social-semantic web. In *Proceedings of the Workshop on Web 3.0: Merging Semantic Web and Social Web at ACM Hypertext, Turin, Italy.*

NLM. (2006). *Medical Text Indexer: MTI Processing Flow Explained* (Whitepaper). National Library of Medicine. Available from http://ii.nlm.nih.gov/resource/Medical_Text_Indexer_Processing_Flow.pdf

Ogden, C. K., & Richards, I. A. (1923). *The meaning of meaning: A study of the influence of language upon thought and of the science of symbolism*. London: Routledge & Kegan Paul.

Olson, H. A., & Wolfram, D. (2006). Indexing consistency and its implications for information architecture: A pilot study. In *Proceedings of the ASIS&T Information Architecture Summit (IA Summit 2006)* (pp. 23–27).

Osinski, S., Stefanowski, J., & Weiss, D. (2004). Lingo: search results clustering algorithm based on singular value decomposition. In *Intelligent Information Processing and Web Mining, Proceedings of the International IIS: IIPWM'04 Conference held in Zakopane, Poland, May 17-20, 2004* (pp. 359–368). Heidelberg: Springer.

Panzer, M., & Zeng, M. L. (2009). Modeling classification systems in SKOS: some challenges and best-practice recommendations. In *Proceedings of the 2009 International Conference on Dublin Core and Metadata Applications* (pp. 3–14). Dublin Core Metadata Initiative. Available from http://dcpapers.dublincore.org/ojs/pubs/article/view/974

Pfeffer, M., Eckert, K., & Stuckenschmidt, H. (2008). Visual Analysis of Classification Systems and Library Collections. In *ECDL '08: Proceedings of the 12th European conference on Research and Advanced Technology for Digital Libraries, Aarhus, Denmark* (pp. 436–439). Heidelberg: Springer. Available from http://dx.doi.org/10.1007/978-3-540-87599-4_57

Plas, L. van der, & Tiedemann, J. (2006). Finding synonyms using automatic word alignment and measures of distributional similarity. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions* (pp. 866–873). Morristown, NJ, USA: Association for Computational Linguistics.

Porter, M. (2001). *Snowball: A language for stemming algorithms.* Published online. Available from http://www.snowball.tartarus.org/texts/introduction.html

Quintarelli, E. (2005). *Folksonomies: power to the people.* Paper presented at the ISKO Italy – UniMIB meeting, Milan, June 24, 2005. Available from http://www.iskoi.org/doc/folksonomies.htm

Ramakrishnan, G., Prithviraj, B., & Bhattacharyya, P. (2004). A gloss-centered algorithm for disambiguation. In *SENSEVAL-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text, Barcelona, Spain, July 2004*. New York, NY, USA: ACM.

Ranganathan, S. R. (1937). *Prolegomena to library classification*. Madras: Madras Library Association.

Ranganathan, S. R. (1945). *Elements of library classification*. Poona: N. K. Publishing House.

Rayson, P., & Garside, R. (2000). Comparing corpora using frequency profiling. In *Proceedings of the workshop on comparing corpora – volume 9* (pp. 1–6). Stroudsburg, PA, USA: Association for Computational Linguistics. Available from http://dx.doi.org/10.3115/1117729.1117730

Razikin, K., Goh, D. H.-L., Chua, A. Y.-K., & Lee, C. S. (2008). Can social tags help you find what you want? In B. Christensen-Dalsgaard, D. Castelli, B. A. Jurik, & J. Lippincott (Eds.), *ECDL '08: Proceedings of the 12th European conference on Research and Advanced Technology for Digital Libraries, Aarhus, Denmark* (Vol. 5173, pp. 50–61). Heidelberg: Springer. Available from http://dblp.uni-trier.de/db/conf/ercimdl/ecdl2008.html#RazikinGCL08

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*. Available from http://arxiv.org/pdf/cmp-lg/9511007

Rolling, L. (1981). Indexing consistency, quality and efficiency. *Information Processing & Management*, *17*(2), 69–76. Available from http://www.sciencedirect.com/science/article/B6VC8-468H7YY-1D/2/b22cec63272fee629d120dec6ff2c3aa

Rosa, C. D. (2006). *College Students' Perceptions of Libraries And Information Resources*. Dublin, Ohio, USA: OCLC.

Ross, J., Irani, L., Silberman, M. S., Zaldivar, A., & Tomlinson, B. (2010). Who are the crowdworkers? Shifting demographics in mechanical turk. In *Proceedings of the 28th International Conference on Human Factors in Computing Systems (CHI 2010)* (pp. 2863–2872). New York, NY, USA: ACM. Available from http://doi.acm.org/10.1145/1753846.1753873

Ross, J., Zaldivar, A., Irani, L., & Tomlinson, B. (2009). *Who are the Turkers? Worker Demographics in Amazon Mechanical Turk.* (Tech. Rep.). Department of Informatics, University of California, Irvine, USA, Technical Report SocialCode-2009-01.

Ruch, P. (2006). Automatic assignment of biomedical categories: toward a generic approach. *Bioinformatics*, *22*(6), 658–664. Available from http://bioinformatics.oxfordjournals.org/cgi/content/abstract/22/6/658

Ruiz, M. E., & Aronson, A. (2007). *User-centered evaluation of the medical text indexing (MTI) system* (Tech. Rep.). National Library of Medicine. Available from http://ii.nlm.nih.gov/resources/MTIEvaluation-Final.pdf

Salton, G., & Lesk, M. E. (1965). The SMART automatic document retrieval systems – an illustration. *Communications of the ACM*, *8*, 391–398. Available from http://doi.acm.org/10.1145/364955.364990

Salton, G., & Yang, C. (1973). On the specification of term values in automatic indexing. *Journal of Documentation*, *29*(4), 351–372.

Santorini, B. (1990). *Part-of-speech tagging guidelines for the Penn Treebank Project (3rd revision)* (Tech. Rep.). University of Pennsylvania. Available from http://repository.upenn.edu/cis_reports/570/

Schulz, H.-J., Hadlak, S., & Schumann, H. (2011). The design space of implicit hierarchy visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, *17*(4), 393–411.

Schwens, U., & Wiechmann, B. (2009). Netzpublikationen in der Deutschen Nationalbibliothek. *Dialog mit Bibliotheken*, *1*(1), 10–13.

Schöning-Walter, C. (2011). Automatische Erschließung – Herausforderung und Chance (Bericht über den PETRUS-Workshop). *Dialog mit Bibliotheken*, *23*(2), 49-51.

Seco, N., Veale, T., & Hayes, J. (2004). An Intrinsic Information Content Metric for Semantic Similarity in WordNet. In *Proceedings of the 16th European Conference on Artificial Intelligence* (pp. 1089–1090). Valencia, Spain. Available from http://eden.dei.uc.pt/~nseco/ecai2004b.pdf

Shearer, J. R. (2004). A Practical Exercise in Building a Thesaurus. *Cataloging & Classification Quarterly*, *37*, 35–56.

Sheng, V. S., Provost, F., & Ipeirotis, P. G. (2008). Get another label? Improving data quality and data mining using multiple, noisy labelers. In *KDD '08: Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 614–622). New York, NY, USA: ACM.

Shirky, C. (2005). *Ontology is Overrated: Categories, Links, and Tags.* Published online. Available from http://www.shirky.com/writings/ontology_overrated.html

Shneiderman, B. (1992). Tree visualization with tree-maps: 2-d space-filling approach. *ACM Transactions on Graphics*, *11*(1), 92–99.

Shneiderman, B. (1996). The eyes have it: a task by data type taxonomy for information visualizations. In *Proceedings of the 1996 IEEE Symposium on Visual Languages (VL '96)* (pp. 336–343). Washington, DC, USA: IEEE Computer Society. Available from http://www.cs.uta.fi/~jt68641/infoviz/The_Eyes_Have_It.pdf

Shneiderman, B., & Wattenberg, M. (2001). *Ordered Treemap Layouts.* Published online. Available from ftp://ftp.cs.umd.edu/pub/hcil/Reports-Abstracts-Bibliography/2001-06html/2001-06.pdf

Silvester, J. P., Genuardi, M. T., & Klingbiel, P. H. (1994). Machine-aided indexing at NASA. *Information Processing & Management*, *30*(5), 631–645. Available from http://www.sciencedirect.com/science/article/B6VC8-469WV4C-21/2/8d4f38d91e4851c0e5fa6a4db6ce5b4e

Smith, G., Czerwinski, M., Meyers, B., Robbins, D., Robertson, G., & Tan, D. S. (2006). FacetMap: A Scalable Search and Browse Visualization. *IEEE Transactions on Visualization and Computer Graphics*, *12*(5), 797–804.

Smith, M., & Fiore, A. (2001). Visualization components for persistent conversations. In *Proceedings of the SIG-CHI on Human factors in computing systems* (pp. 136–143). Available from http://research.microsoft.com/research/coet/Communities/chi2001/paper.pdf

Snow, J. (1855). Report on the Cholera Outbreak in the Parish of St. James, Westminster, during the Autumn of 1854. In The Cholera Inquiry Committee (Ed.), (chap. Dr. Snow's Report). Churchill, London. Available from http://johnsnow.matrix.msu.edu/work.php?id=15-78-55

Snow, R., O'Connor, B., Jurafsky, D., & Ng, A. Y. (2008). Cheap and fast – but is it good? evaluating non-expert annotations for natural language tasks. In *EMNLP '08: Proceedings of the Conference on Empirical Methods in Natural Language Processing* (pp. 254–263). Morristown, NJ, USA: Association for Computational Linguistics.

Soergel, D. (1999). The rise of ontologies or the reinvention of classification. *Journal of the American Society for Information Science*, *50*(12), 1119–1120. Available from http://www.dsoergel.com/cv/B70.pdf

Sorokin, A., & Forsyth, D. (2008). Utility data annotation with Amazon Mechanical Turk. *First IEEE Workshop on Internet Vision at CVPR 08*, 1–8.

Spärck Jones, K. (1972). Some thesauric history. *Aslib Proceedings*, *24*(7), 400–411.

Spärck Jones, K. (1991). *Two tutorial papers: Information Retrieval & Thesaurus* (UCAM-CL-TR-234 No. 234). University of Cambridge. (ISSN 1476-2986)

Sridhar, M. S. (2004). Subject searching in the OPAC of a special library: problems and issues. *OCLC Systems & Services*, *20*, 183–191.

Stasko, J., & Zhang, E. (2000). Focus+context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations. In *Proceedings of the IEEE Symposium on Information Vizualization 2000* (pp. 57–65). Washington, DC, USA: IEEE Computer Society. Available from http://dl.acm.org/citation.cfm?id=857190.857683

Stefanowski, J., & Weiss, D. (2007). Comprehensible and accurate cluster labels in text clustering. In *Large Scale Semantic Access to Content (Text, Image, Video, and Sound)* (pp. 198–209). Paris, France, France: Le centre de hautes etudes internationales d'informatique documentaire. Available from http://dl.acm.org/citation.cfm?id=1931390.1931410

Stork, D. (2010). *Automatic concept splitting and naming for thesaurus maintenance*. Unpublished master's thesis, University of Mannheim.

Stork, D., Eckert, K., & Stuckenschmidt, H. (2011). Cluster it! Semiautomatic Splitting and Naming of Classification Concepts. In *Proceedings of the Joint Annual Conference of the German Association for Pattern Recognition (DAGM) and the German Classification Society (GfKl).*

Stuckenschmidt, H., Waard, A. de, Bhogal, R., Fluit, C., Kampman, A., van Buel, J., et al. (2004). Exploring Large Document Repositories with RDF Technology - The DOPE Project. *IEEE Intelligent Systems, Special Issue on the Semantic Web Challenge*.

Suchanek, F. M., Vojnovic, M., & Gunawardena, D. (2008). Social tags: meaning and suggestions. In *CIKM '08: Proceeding of the 17th ACM Conference on Information and Knowledge Mining* (pp. 223–232). New York, NY, USA: ACM.

Svenonius, E. (2000). *The intellectual foundation of information organization*. Cambridge, Massachusetts, USA: MIT Press.

Telles, G. P., Minghim, R., & Paulovich, F. V. (2007). Normalized compression distance for visual analysis of document collections. *Computers & Graphics*, *31*(3), 327–337. Available from http://dx.doi.org/10.1016/j.cag.2007.01.024

Toutanova, K., Klein, D., Manning, C., & Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the Human Language Technology Conference (HLT-NAACL 2003)* (pp. 252–259).

Tudhope, D., & Koch, T. (2004). New applications of knowledge organization systems: introduction to a special issue. *Journal of Digital Information*, *4*(4). Available from http://journals.tdl.org/jodi/article/view/109/108

Turney, P. D. (2001). Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL. In *EMCL '01: Proceedings of the 12th European Conference on Machine Learning* (pp. 491–502). Heidelberg: Springer.

Turo, D., & Johnson, B. (1992). Improving the visualization of hierarchies with treemaps: Design issues and experimentation. In *Proceedings of the IEEE Conference on Visualization, October 1992*.

Tödter, K. (2007a). Eclipse Rich Client Platform und NetBeans Platform im Vergleich – Das Rad muss immer noch nicht neu erfunden werden! *Eclipse Magazin*, *13*, 24–32.

Tödter, K. (2007b). Eclipse Rich Client Platform und NetBeans Platform im Vergleich – Das Rad muss nicht neu erfunden werden! *Eclipse Magazin*, *12*, 19–27.

Vervenne, D. (1999). Advanced document management through thesaurus-based indexing: the IKEM platform. *CWI Quarterly*, *12*(2), 159–172.

Vickery, B. C. (1997). Ontologies. *Journal of Information Science*, *23*, 277–286.

Wal, T. V. (2005). *Explaining and Showing Broad and Narrow Folksonomies.* Blog post. Available from http://www.personalinfocloud.com/2005/02/explaining_and_.html

Weiner, J. M. (2005). Differences in indexing term vocabularies and agreement with subject specialists. *Electronic Journal of Academic and Special Librarianship*, *6*(1-2). Available from http://southernlibrarianship.icaap.org/content/v06n01/weiner_j01.htm

Wersig, G. (1978). *Thesaurus-Leitfaden – Eine Einführung in das Thesaurus-Prinzip in Theorie und Praxis*. Munich: Saur.

White, R. W., & Marchionini, G. (2007). Examining the effectiveness of real-time query expansion. *Information Processing & Management*, *43*(3), 685–704. Available from http://www.sciencedirect.com/science/article/B6VC8-4KRY930-2/2/780cc85808e10bf781e3853c0b0d2ead (Special Issue on Heterogeneous and Distributed IR)

Wijk, J. J. van, & Wetering, H. van de. (1999). Cushion Treemaps: Visualization of Hierarchical Information. In *Proceedings of the IEEE Symposium on Information Visualization (INFOVIS'99), San Francisco, October 25-26, 1999.*

Witschel, H. F. (2005). Using decision trees and text mining techniques for extending taxonomies. In *Proceedings of the workshop on learning and extending lexical ontologies by using machine learning methods (ontoml 05), bonn, germany.*

Witten, I. H., Paynter, G. W., Frank, E., Gutwin, C., & Nevill-Manning, C. G. (1999). KEA: Practical automatic keyphrase extraction. In *DL '99: Proceedings of the fourth ACM conference on Digital Libraries* (pp. 254–255).

Wolff, C., Heckner, M., & Mühlbacher, S. (2008). Tagging tagging. Analysing user keywords in scientific bibliography management systems. *Journal of Digital Information*, *9*(2). Available from http://journals.tdl.org/jodi/article/view/246

Wolfram, D., & Olson, H. A. (2007). A Method for Comparing Large Scale Inter-Indexer Consistency Using IR Modeling. In *Canadian Association for Information Science Conference Proceedings 2007.* Available from http://www.cais-acsi.ca/proceedings/2007/wolfram_2007.pdf

Wolters, C. (1997). *GOS Thesaurus-Handbuch* (Tech. Rep.). Berlin: Konrad-Zuse-Zentrum für Informationstechnik Berlin. Available from http://www.zib.de/Publications/Reports/TR-97-19.pdf

Wu, H., & Zhou, M. (2003). Optimizing synonym extraction using monolingual and bilingual resources. In *Proceedings of the Second International Workshop on Paraphrasing* (pp. 72–79). Morristown, NJ, USA: Association for Computational Linguistics.

Yarowsky, D. (1992). Word-sense disambiguation using statistical models of Roget's categories trained on large corpora. In *Proceedings of the 14th conference on Computational linguistics – Volume 2* (pp. 454–460). Stroudsburg, PA, USA: Association for Computational Linguistics. Available from http://dx.doi.org/10.3115/992133.992140

Yarowsky, D. (1993). One sense per collocation. In *Proceedings of the workshop on human language technology* (pp. 266–271). Stroudsburg, PA, USA: Association for Computational Linguistics. Available from http://dx.doi.org/10.3115/1075671.1075731

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics* (pp. 189–196). Stroudsburg, PA, USA: Association for Computational Linguistics. Available from http://dx.doi.org/10.3115/981658.981684

Yu, H., & Young, M. (2004). The Impact of Web Search Engines on Subject Searching in OPAC. *Information Technology & Libraries*, *23*(4), 168–180. Available from http://www.redi-bw.de/db/ebsco.php/search.ebscohost.com/login.aspx?direct=true&db=aph&AN=16072026&site=ehost-live

Zamir, O., & Etzioni, O. (1998). Web Document Clustering: A Feasibility Demonstration. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 46–54). New York, NY, USA: ACM.

Zapilko, B., & Sure, Y. (2009). *Converting the TheSoz to SKOS* (Tech. Rep.). GESIS – Leibniz-Institut für Sozialwissenschaften.

Zeng, M. L. (2008). Knowledge Organization Systems (KOS). *Knowledge Organization*, *35*(2-3), 160–182.

Zeng, M. L., & Chan, L. M. (2004). Trends and issues in establishing interoperability among knowledge organization systems. *Journal of the American Society for Information Science and Technology (JASIST)*, *55*(5), 377–395.

Zhang, D., & Dong, Y. (2004). Semantic, hierarchical, online clustering of web search results. In J. X. Yu, X. Lin, H. Lu, & Y. Zhang (Eds.), *Proceedings of the Sixth Asia Pacific Web Conference* (Vol. 3007, pp. 69–78). Heidelberg: Springer.

Zhao, S., McGuffin, M. J., & Chignell, M. H. (2005). Elastic hierarchies: Combining treemaps and node-link-diagrams. In *INFOVIS '05: Proceedings of the 2005 IEEE Symposium on Information Visualization* (pp. 57–64).

Zittrain, J. (2009). Work the New Digital Sweatshops. *Newsweek*, *December 8*. Available from http://www.newsweek.com/id/225629