DOCUMENT INDEXING - PROVIDING A BASIS FOR SEMANTIC DOCUMENT ANNOTATION

H. Peter, H. Sack, C. Beckstein
Institut für Informatik
Friedrich-Schiller-Universität Jena
D-07743 Jena
Germany
{hpeter, sack, beckstein}@minet.uni-jena.de

Abstract: A document index represents a concise ordered compilation of the document's most important topics. It provides direct and fast access to the document parts related to the index information. Together with structural knowledge of the document itself in connection with general knowledge about indexing a 2-layered Index Graph is defined that is further mapped to an ontology representation. By defining suitable metrics it is shown how the Index Graph can be utilised to augment semantic applications. We have developed a system for supporting the author of a document in the process of index compilation. Other possible applications include document visualisation, and semantic document annotation.

1 Introduction

The index is an essential part of any document, no matter if we consider a book, an issue of a magazine, or any other information source. The purpose of the index is to facilitate fast and efficient random access to any important subject within the document. Especially for the world wide web (WWW) the indexing of web documents has become a fundamental task to achieve high quality search engine performance. Considering the implementation we distinguish traditional *linear indexes*, as e.g. indexes in printed books, and *hypertext indexes*. In linear indexes index entries usually are connected to corresponding document parts by page numbers. In hypertext indexes the index entries are connected to corresponding document parts by hyperlinks. In general the process of index creation is not trivial. While generating a search index for web search engines is performed automatically with the help of information retrieval techniques, traditional document indexing requires extensive intellectual efforts: Appropriate headings must be chosen, index entries must be defined sophistically, synonymy, ambiguities and other relationships between index entries must be detected and handled properly. In the end, the creation of a sound index also affects the corresponding document because it provokes text restructuring and disambiguation of the used vocabulary.

A document index is supposed to contain the most important concepts that are embodied in the document itself. Each index entry refers to a distinguished part of the document, which covers a topic closely related to the index entry. This semantic relationship can be utilised for additional semantic document annotation. Semantic document annotation typically is accomplished manually or with the help of information retrieval techniques. Manual semantic annotation is rather time-consuming and expensive, and therefore doesn't scale. Semantic annotation by information retrieval techniques on the other hand requires extensive processing and often delivers unreliable result. The high amount of intellectual effort that is invested in document indexing can be further utilised as being a resource for semantic document annotation.

Current indexing software for traditional document indexing (e.g. LaTeX's MakeIndex [Lam87] or MACREX[CC97]) supports the author only in mechanical indexing tasks, e.g. simple management or sorting of index entries. This type of software also does not assist the author in the much more complex and creative task of originating accurate and sound index entries. An entirely automated indexing process requires text understanding capabilities that are beyond the ability of prevailing computer systems.

We have developed an architecture — the SMARTINDEXER [PSB06b] — that supports the author in the creative tasks of the indexing process. For this purpose, we had to design a knowledge representation (in the following referred to as *Index Ontology*), which summarizes general knowledge about all index elements and their relationships to each other. Index quality strongly depends on the amount of its inherent semantics. An index can be regarded as a network — the *Index Graph* — where the index entries are represented by the nodes. Subentry relationship between two index entries as well as different cross-references among index entries constitute the arcs of the Index Graph. This network embodies the semantic interrelationships inherent in the index. The Index Graph serves as a suitable basis for index visualisation, thus enabling convenient user interfaces for efficient index navigation and index manipulation. Furthermore, distance measures defined on the Index Graph enable new ways to determine the quality of an index.

The paper is structured as follows: Section 2 defines the basic principles of indexing and gives an outline of the Index Ontology. Section 3 illustrates how the Index Graph is defined while in section 4 several examples are shown how to exploit the index metadata, which is represented by this graph. Section 5 concludes the paper with an outlook on ongoing and future work.

2 Document Indexes

There are several means that simplify the access to the content of a document. First, there is the table of contents (TOC), which is a hierarchically organized list of parts of a document. Besides chapter titles, it often lists section titles within the chapters as well, and occasionally even subsections. The TOC of a linear document indicates those page numbers where each part starts, while online ones offer links to go to each section. The entries of a TOC are arranged in linear reading order of the underlying document.

Then, there is the document index. According to the British Indexing Standard [Mul94] an index is a systematic arrangement of entries designed to enable users to locate information in a document or specific documents in a collection. This arrangement of the entries provides a basis for an efficient identification and access to information contained in a document independent of the order of its occurrence in a document.

In order to access specific information contained within a document, one may consult the TOC first for getting an overview, where the desired information might be dissembled. The information provided by the TOC often is not as detailed as the information provided by the index. By looking up a specific term in the index, the user is guided directly to the location of the information within the document.

2.1 The Syntactical Structure of an Index

An index is a vertically ordered arrangement of entries. Each index entry consists of a heading (or main heading) and at least one of the following components: a subentry, a reference locator, or a cross-reference. Fig. 1 shows the five basic components of an index. In the following the function and the syntactical structure of these components will be discussed.

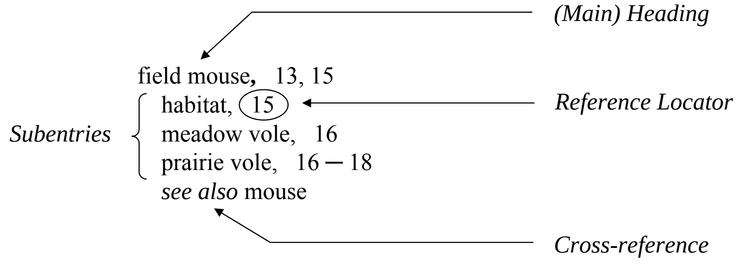

Figure 1: The Index Entry

Main headings of an index establish the primary access for users if they look for information about a topic in a document. They are normally nouns or noun phrases such as persons, places or objects.

If a main heading has more than five reference locators, it should be further subdivided into appropriate subentries. Thus, the user of the index doesn't need to spend too much time to locate the information it seeks. A subentry is similarly structured as an index entry and is subordinated to a main heading. It is composed of a (sub)heading, one or more reference locators, and — only rarely — cross-references.

A subentry can have further index entries — so called sub-subentries. The above mentioned statements about subentries hold analogously for sub-subentries. In general it is not recommended to go beyond the level of sub-subentries.

A reference locator is a unique identifier for a document unit. A document unit can be e. g. a sentence, a paragraph, or a page. Reference locators follow a heading and indicate that document unit, where information related to the heading can be found. In linear documents, they are usually page numbers, section numbers, line numbers, or ranges thereof. The most common reference locator in web documents is the URL. Regardless of the type of reference locators long series of undifferentiated locators after a heading should be avoided.

Cross-references are links between one heading and another. They constitute internal navigation guides within the index because they connect related information. An index of a linear document usually provides two kinds of cross-references: *see* references and *see also* references. The first kind is used for variant spellings, synonyms, aliases, abbreviations, and so on. *See also* references are used to guide the user to another closely related heading that supplies additional information.

2.2 The Semantics of an Index

In order to describe the semantics of an index we have to distinguish between ’concepts’ and ’terms’.

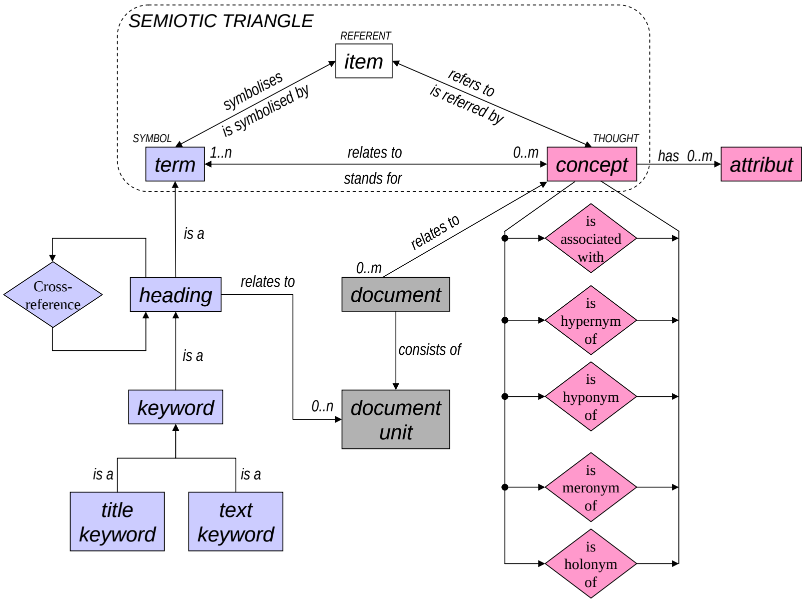

Figure 2: The Index Ontology

2.2.1 Concepts and Terms

According to Fugmann [Fug99] a concept is the sum of all essential propositions that can be made about an item. Each proposition constitutes an attribute of the concept. Thereby, we consider an item as everything whereof a proposition can be made.

Concepts can be in relationship with each other, as e.g. hypernymy or hyponymy. A concept A is a hypernym of another concept B, if A is missing at least one attribute from B. In this way a hierarchy of abstraction can be established. Hyponymy is the opposite relation to hypernymy. If we add an attribute to a concept A a new and more specific concept B will be created. B is then called a hyponym of A.

Another important relationship between concepts is the part-of-relation. There, a distinction is drawn between meronymy and holonymy. A concept A is a meronym of another concept B, if A is a part or a member of B. The concept B is then a holonym of the concept A.

A term is a symbolic representation of a concept. The terms 'concept', 'item' and 'term' correspond to the three elements 'thought', 'referent', and 'symbol' of the well known semiotic triangle [OR23] (see also Fig. 2).

It is possible that several different concepts are related to the same term. Such an ambiguous term is called homonym. In contrast to that, if there are different terms standing for the same concept we refer to them as synonyms. Synonymy and homonymy are essential relationships that have to be considered when creating an index. According to Mulvany [Mul94] synonyms can be used to control the scattering of information in an index, to anticipate the language of the index user, and to reconcile the language of the document with the users' language.

2.2.2 Headings and their Relationships

Headings are terms, which represent concepts in the document. They should be predictable for a user of the index, i. e. a user should be able to guess the index term for a given concept that he is interested in. The nouns or noun phrases that constitute the headings need not necessarily be contained in the document unit that the index entry refers to. If the terms occur in the corresponding document unit, we call them *keywords*. Keywords can be distinguished according to their occurrences in the document: If they are extracted from the document's title, they are called *title keyword*; if they are taken from a paragraph they are *text keywords*.

An author who puts a subheading below a main heading signals that there is an important semantic relationship between the corresponding concepts, as e. g. hypernymy or meronymy.

2.2.3 Cross-references

Cross-references establish semantical relationships between the concepts denoted by headings. Usually, there are two kinds of cross-references in linear documents: *see* references and *see also* references.

See References *See* references express the fact that two terms represent the same concept. If there is a *see* reference between a heading h_1 and h_2, then the corresponding concepts of h_1 and h_2 are meant to be the same. Without *see* references indexes of linear documents would become unnecessarily long. They allow the author to define a preferred term that can be reused in the index part of the document for all its equivalent terms.

See also References *See also* references relate to terms, which denote concepts that are semantically related, but not equal concepts. It is good practice to always have an inverse *see also* reference for each such cross-reference in the index.

2.3 The Index Ontology

The Index Ontology provides general knowledge about the components of an index and their relationships with each other. It is given as graphical representation of the index components and their associated semantic relationships (see Fig. 2). The Index Ontology consists of three main parts: The semiotic triangle, which characterizes the relationship between concepts, items, and terms defines the first part. The second part is given by the conceptual relationships. The third part specifies the index structure itself.

We have designed an OWL DL knowledge base that describes the Index Ontology. OWL is an XML based knowledge representation language and has been defined by the World Wide Web Consortium (W3C) as the recommended standard for (Web) ontologies [MvH04] . With OWL the semantics of knowledge can be described in a machine-accessible way. The W3C standard allows the interchangeability, portability and reusability of ontologies. Therefore, it is the first choice for our index tool set SMARTINDEXER. OWL Lite is a less complex version of OWL. We had to resort to the more complex OWL DL because our ontology requires arbitrary cardinalities, which is prohibited with OWL Lite. The current OWL version of our Index Ontology can be found at [PSB06a].

- field mouse, 13, 15
  - habitat, 15
  - meadow vole, 16
  - prairie vole, 16
  - *see also* rodent

- rodent, 1
  - beaver, 10, 11
  - dentision
    - incisor, 4
    - rotation of teeth, 5
  - hamster, 2 – 4
  - *see also* field mouse

Figure 3: Excerpt of a Linear Index

3 The Index as a Network

An index can be regarded as a directed 2-layered graph $G = (N, E)$ — in the following called *Index Graph* — that splits into two subgraphs: a *structure graph* G_{IH} and a *document graph* G_{PO}. The node set N of G is the union $N_{IH} \cup N_{PO}$ of the set of all index headings N_{IH} and the set of all document units N_{PO}. The edge set E of G is the union $E_{IH} \cup E_{RL} \cup E_{PO}$, where $E_{IH} \subseteq N_{IH} \times N_{IH}$ is the set of all arcs representing relationships between index entries, $E_{RL} \subseteq N_{IH} \times N_{PO}$ is the set of reference locators, and $E_{PO} \subseteq N_{PO} \times N_{PO}$ represents the part-of structure of the document.

The structure graph $G_{IH} = (N_{IH}, E_{IH})$ represents the index components and their interrelationships. Each node in N_{IH} correlates with a heading of the index. There are three different kinds of arcs in E_{IH}. Let $a, b \in N_{IH}$, where a represents the node that corresponds to an index heading A and b corresponds to an index heading B. The first kind of arcs (SUB-arcs) is given by the subentry relationship between two headings. If B is a subheading of A, then the graph contains a directed SUB-arc from a to b. The two other kinds of arcs represent cross-references between two headings. A *see* reference from A to B constitutes a SEE-arc from a to b. Analogously, a *see also* reference from A to B constitutes a SEA-arc from a to b. Furthermore, the graph contains an artificial root node (*index root*) that is linked via a SUB-arc with all nodes that represent a main heading. If we consider an index without cross-references, the structure graph is a tree with maximum depth 3. Cross-references might cause cycles in the tree.

The document graph $G_{PO} = (N_{PO}, E_{PO})$ represents the hierarchical structure of a document. It consists of several document units such as chapters, sections, subsections, and paragraphs. It is a refinement of the TOC representing all the part-of links between the root of the document and its parts, the parts and their chapters, all the way down to the links between paragraphs and document pages. Fig. 4 shows a graph resulting from an example index (see Fig. 3).

4 Generating and Exploiting Index Metadata

Being a condensed extract of the most important information within a document, the index itself in addition to the knowledge about the index provided by the Index Ontology can be further utilised. We consider three possible applications: smart index generation, semantic annotation of documents, document visualisation and navigation.

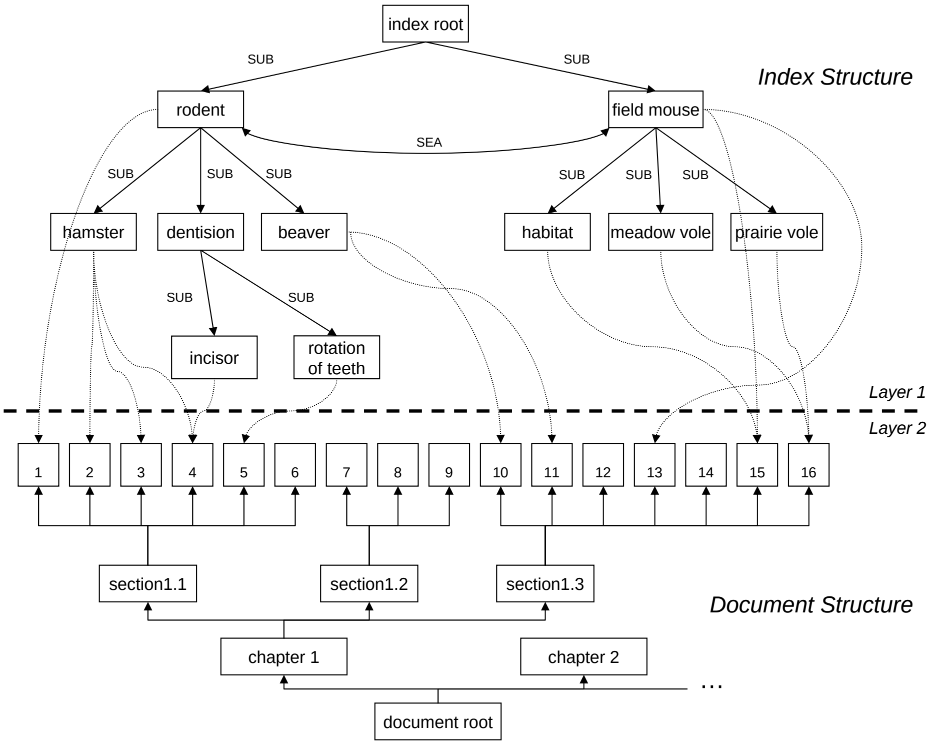

Figure 4: The Graph Resulting from the Example in Fig. 3

4.1 Smart Index Generation

We have developed an architecture for supporting the author in document indexing — the SMARTINDEXER — that can be applied as an auxiliary means for a given word processing application. By utilizing semantic relationships of terms provided by the lexical database WordNet [Fel98], SMARTINDEXER assists the user to complete the document index. For each new potential index entry SMARTINDEXER reviews synonyms, hyponyms, hypernyms, holonyms, meronyms, and sister terms, and in accordance with the Index Ontology gives suggestions about where and how to integrate the new index entry into the already existing document index. In this way SMARTINDEXER ensures index consistency and index integrity (for an outline of the SMARTINDEXER work flow see Fig. 5). For a more detailed discussion of SMARTINDEXER and its underlying algorithms see [PSB06b].

4.2 Semantic Annotation of Documents

Semantic Annotation of a document can be achieved by using the index metadata represented by the Index Ontology in addition with an already existing document index. For this purpose we consider the following metrics defined on the document index

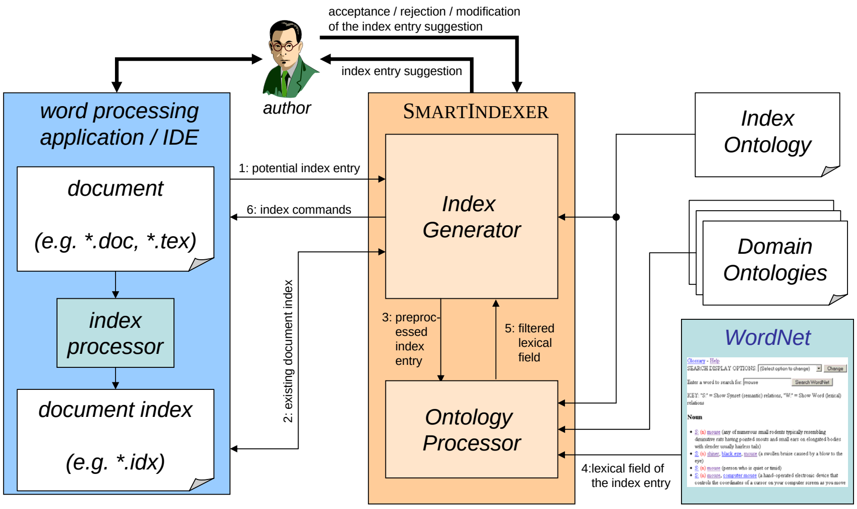

Figure 5: Indexing Process with SMARTINDEXER

being represented as a directed Index Graph $G = (N, E)$. Let $x, y \in N$:

$$d(x,y) = \begin{cases} \infty & \text{if there is no directed path from x to y} \\ \mathrm{lsp(x,y)} & \text{otherwise} \end{cases}$$

where $\mathrm{lsp(x,y)}$ is the length of the shortest path from x to y. $d(x, y)$ is closely related to the semantic distance of two index headings x and y. If $d(x, y)$ is small, it indicates a strong relationship between x and y. It can be used to indicate the relationship of the contents of document parts and thus, offers a way of guiding the user's reading direction.

Next, we can define a semantic weight $w(u)$ of a document unit $u \in N_{PO}$. For sequential indexes a document unit usually refers to a page. For this purpose, we use the Index Graph in the same way as an inverted index by simply considering the number of ingoing arcs of a specific document unit u:

$$w(u) = \mathrm{indegree_E(u)}$$

where $\mathrm{indegree_E(u)}$ denotes the indegree of node $u \in N_{PO}$ in $G = (N, E)$. The weight $w(u)$ indicates how many index headings refer to a given document unit u. The higher $w(u)$, the more topics are related to u. This might serve the user as an advice on which document units are important and which are not.

The relation given by the index elements and their corresponding reference locators specifies, which document unit can be annotated with the index heading that refer to that unit. In the index creation process, the author usually collects only those concepts for the index that are essential considering the semantic content of the document. Therefore, the index heading related to a specific document unit can also be regarded as keywords denoting the semantic content of that unit. The reader as well as a software agent can use those keywords to decide whether this document unit is important or not.

In addition, each index heading can also be connected with lexical information. By linking each index heading with its corresponding entry of the WordNet dictionary, the

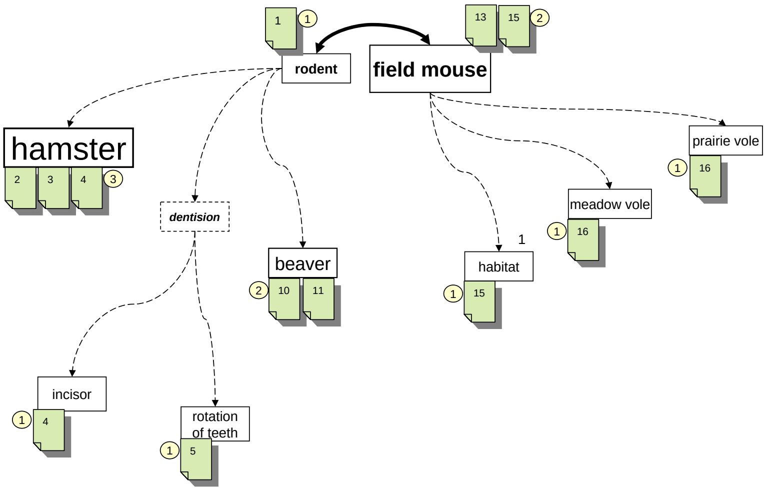

Figure 6: Weighted Index Graph for Document Visualisation and Navigation

user can access the semantic environment of the index heading. By connecting index headings with corresponding domain ontologies, this semantic environment can be further expanded. Thus, processing required for text comprehension can be simplified and the quality of its result can be enhanced.

4.3 Document Visualisation and Navigation

A view on the structure graph itself can serve as an alternative visualisation of the sequentially printed index. It guides the reader on its way gathering quickly the most important information of a document. This view can be generated w.r.t. a personalized user profile, thus enabling each user a different way of navigation. For this purpose, relevance weights can be used to define a heuristics indicating the importance of certain index entries or paths through the Index Graph:

- To weight a node in the Index Graph, we can use the number of its corresponding subentries. The more subentries, the more information loaded the index entry and the more details are given on a specific index entry.
- The number of reference locators associated with an index heading gives advice about its importance. The more reference locators are associated with an index heading, the more document units are referring to the concept represented by it.

For a visual presentation of the Index Graph, index headings with a higher relevance weight can be displayed larger than index headings with a lower relevance weight. This allows to guide the users attention while traversing the document. See Fig. 6 for an example of the weighted Index Graph.

5 Conclusions and Outlook

Document index compilation is a sophisticated task. It requires conceptual knowledge about the topic discussed in the document as well as structural knowledge of the document itself in connection with general knowledge about indexing. This knowledge is represented in the 2-layered Index Graph that is used by semantic applications like the SMARTINDEXER to augment their knowledge as required for index compilation, document visualisation, or semantic document annotation.

Future work will investigate how semantic applications based on the Index Graph can be improved by deploying suitable metrics. This will be an important step towards the overall goal of unifying seemingly different structuring devices such as table of contents, index, bibliographic references, and topic maps into a general approach of semantic document annotation.

References

[CC97] Hilary Calvert and Drusilla Calvert. MACREX Manual for Version 6.5., 1997.

[Fel98] C. Fellbaum. *WordNet – An Electronic Lexical Database*. MIT Press, 1998.

[Fug99] Robert Fugmann. *Inhaltserschliessung durch Indexieren: Prinzipien und Praxis*, volume 3 of *Reihe Informationswissenschaft der DGD*. Deutsche Gesellschaft fr Dokumentation, Frankfurt am Main, 1999.

[Lam87] Leslie Lamport. MakeIndex: An Index Processor for LaTeX, 1987.

[Mul94] Nancy C. Mulvany. *Indexing Books*. The University of Chicago Press, Chicago, 1994.

[MvH04] Deborah L. McGuinness and Frank van Harmelen. OWL Web Ontology Language: Overview, W3C Recommendation. `http://www.w3.org/TR/2004/REC-owl-features-20040210/`, 10 February 2004.

[OR23] C. K. Ogden and I. A. Richards. *The Meaning of Meaning*. Harcourt, Brace, and Co., Inc., 1923.

[PSB06a] Heiko Peter, Harald Sack, and Clemens Beckstein. Current OWL version of the Index Ontology. `http://www.informatik.uni-jena.de/~sack/IndexOntology.owl`, 2006.

[PSB06b] Heiko Peter, Harald Sack, and Clemens Beckstein. SmartIndexer - Amalgamating Ontologies and Lexical Resources for Document Indexing. In *OntoLex 2006 - Interfacing Ontologies and Lexical Resources for Semantic Web Technologies*, Genoa, Italy, 24-26 May 2006.

[vHBF$^+$01] Frank van Harmelen, Jeen Broekstra, Christiaan Fluit, Herko ter Horst, Arjohn Kampman, Jos van der Meer, and Marta Sabou. Ontology-Based Information Visualisation. In *International Conference on Information Visualisation, IV 2001*, pages 555–562, London, England, 25-27 July 2001.