Control and Cybernetics

vol. **39** (2010) No. 2

Semantic information within the BEATCA framework*

by

Krzysztof Ciesielski[1], Dariusz Czerski[1], Michał Dramiński[1], Mieczysław A. Kłopotek[1,3], Sławomir T. Wierzchoń[1,2]

[1] Institute of Computer Science, Polish Academy of Sciences
Ordona 21, 01-237 Warszawa, Poland

[2] Institute of Informatics, University of Gdańsk
Wita Stwosza 57, 80-952 Gdańsk, Poland

[3] Institute of Informatics, University of Podlasie in Siedlce, Poland

Abstract: In this paper we investigate the impact of semantic information on the quality of hierarchical, fuzzy-based clustering of a collection of textual documents. We show that via a relevant tagging of a part of the documents one can improve the quality of overall clustering, both of tagged and un-tagged documents.

Keywords: semantic information, clustering with partial multilabel supervision

1. Introduction and related works

A fundamental difference between a computer system and a human being, when processing a text document, is that the computer perceives it as a formal set of meaningless symbols, while the human being attaches meaning to it immediately.

Technically, the meaning of a piece of text is described by its "semantics", albeit the specific usage of this term does not necessarily cover its intended understanding, and is rather a more or less accurate approximation.

The current state of information technology does not permit computers to grasp a meaning out of the human readable information present on the Web, just because this meaning is not applicable to computer "awareness", as it usually neither acts nor perceives anything in the "real world". However, this deficiency can be slightly reduced. Namely, a web-application should be designed in such a way that instead of acting upon a human-readable information itself, it preprocesses available information in an intelligent way. The result of such

*Submitted: May 2009; Accepted: September 2009.

preprocessing should support the human recipient in his/her decision making – particularly through context-aware presentation of information[1].

In order to support human decision making, based on natural language texts, approximations of its semantics are necessary, which by now are restricted more or less to interrelationships between concepts (related to words, phrases etc.). Typically, *is-a-part-of* hierarchies (see Winston et al., 1987) are used, though more sophisticated relations may be captured. These relationships may be crisp or fuzzy in nature, depending on the logic applied.

In this sense, inclusion of information on semantics into search engine mechanisms is a subject of increased interest of the scientific community. By 2006, for example, the US National Science Foundation awarded almost 500 grants for research in this and related domains.

Generally, semantic search is understood as an extension of traditional document information retrieval with the semantic web technology, exploiting ontology-based metadata. In particular "the Semantic Web is an extension of the current web in which information is given well-defined meaning (semantics), better enabling computers and people to work in cooperation."[2].
This subarea of Semantic Web is referred to in this paper.

Within this context, we can speak of "semantic content" as a kind of "unsaturated information", that is, the queries that one can pose to a set of data. We will speak about "semantic information" as the "saturated information", that is, the query plus the true answer to it. See Floridi (2005) for a more elaborate presentation of understanding of semantic information.

The semantic information about a document stems from three distinct sources: (a) the text of the document itself, (b) the link information, and (c) the user inserted semantic tags. Depending on the way this information is exploited, we distinguish three main paradigms:

Context Based Semantic Search Engines, intended to enhance performance of traditional search engines (measured e.g. in terms of their precision and recall, see Baeza-Yates and Ribeiro-Neto, 1999). For this purpose, the context information (in terms of domain ontology and metadata) is used. After having retrieved the documents using word matching, RDF graphs are used to enrich their content, and so to obtain better quality of results. An overview of different approaches developed within this paradigm can be found in Esmaili and Abolhassani (2006). Among the systems developed within this area we can mention: OWLIR (Ding et al., 2004), QuizRDF (Davies, Weeks and Krohn, 2002), InWiss (Priebe, Schlaeger and Pernul, 2004), Corese (Corby, Dieng-Kuntz and Faron-Zucker, 2004), SHOE (Heflin and Hendler, 2000), DOSE (Bonino, Corno and Farinetti, 2003), SERSE (Tamma et al., 2004), OntoWeb (Spyns et al., 2002), Score (Sheth et al., 2002, and Zhu et al., 2002).

[1] Consult `http://reasoningweb.org/2006/Objectives.html`.

[2] See `http://www.w3.org/RDF/Metalog/docs/sw-easy`

Supplementary Search Engines, supporting the process of collecting information on specific topic, and expanding traditional search with directions to include external sources of additional information (e.g. while looking for a singer, biography, posters, albums etc.). External meta data is predominantly used (e.g. CDNow, Amazon, IMDB as mentioned in Guha, McCool and Miller, 2003). Sample systems are W3C Semantic Search (Guha, McCool and Miller, 2003) and ABC (Guha, McCool and Miller, 2003).

Semantic Query Expansion Engines, which use an ontology like WordNet, for query expansion to modify user queries targeted at, e.g., Google search engine. An example of such an approach is Semantic Search Facilitator/assistant (Terziyan et al., 2004).

However, the experiments performed with these engines are usually limited to carefully selected well-defined sets of documents. Such restrictions allow for elaborated representations of the domain knowledge in terms of ontological languages like OWL[3].

Our goal is to provide a framework enabling exploitation of domain knowledge for large-scale searches, with documents collections assigned explicitly (with diversified degree of membership) to different semantic categories. Therefore, we try to incorporate the semantic information right into the whole search process, starting from grasping the documents by the spider (crawler) and ending at the query answering module. To get manageable results, we restricted the permitted representation of semantic information to tags (attributes being simple terms) attached to documents either manually or via an extraction process, to additionally provided hierarchies of these terms, and natural language texts eventually attached to terms in these hierarchies (ontologies).

We allow coexistence of semantically tagged and un-tagged documents[4] as well as multiple ontologies for representing content of the documents. The representation of search results extends beyond typical lists of results, namely to the maps of document collections (Becks, 2001). Both the content and the semantic tags may be used for document collection indexing in terms of a document map.

Our framework is based on the so-called contextual model of a document collection. The contextual model nay be viewed as a method of clustering documents. Clustering may be viewed as a data compression method. Well known limitation on encoding efficiency for loseless compression states that at least $\sum_{i=1}^{m} -n_i \log \frac{n_i}{N}$ bits are necessary to encode a sequence of N symbols from an alphabet of cardinality m, where the "letter" i occurs n_i times in the sequence. In our case, the "alphabet" are the terms, the sequence is the contents of documents. This efficiency is approximately reached by the so-called arithmetic encoding. But this lower bound holds for an unpredictable sequence of symbols.

[3] Consult e.g. http://www.w3.org/TR/2004/REC-owl-features-20040210/.

[4] We did not consider tagging of parts of documents yet.

If we know the order, the bounds can be changed downwards. So the contextual model means recursive approach to clustering.

While the impact of our proposal is multifold, we concentrate in this paper on the clustering aspect of map-based search engine operation. High quality of clustering is crucial for the successful response to user queries. We will in particular demonstrate that inclusion of semantic information for a part of documents increases the quality of final grouping of the whole collection. Furthermore, the user may influence the clustering by providing a (partial) outline of his vision of clustering. We will also show that the time complexity for clustering documents with semantic information is feasible even for larger collections.

The paper is organized as follows: Section 2 provides brief overview of the role and the use of semantic information in information retrieval. One possible way of representing such an information is described. Details concerning exploitation of semantic information in so-called semantic clustering are given in Section 3, and the whole procedure of constructing contextual models is provided in Section 4. Performed experiments are described in Section 5. Lastly, Section 6 concludes the paper.

2. Semantic information in information retrieval

2.1. Sources of semantic information

The search process can be enhanced by smart exploitation of knowledge coming from various sources. In case of searching for documents we distinguish at least three such sources.

First, semantic information about the end-user may be available, including demographic information collected automatically (e.g. when the user IP and/or hardware and software features can be mapped to a geographic location, or to a branch of business, or to income category etc.), or via questionnaires, filled by the user. It may also stem from the analysis of the user's past behavior (earlier queries, other activities like purchase, reaction to advertisement etc.).

Second, diverse semantic information is available about documents in collections. It includes categorizations of Web pages or Web sites, product catalogues, library categories etc. Semantic information may be also derived automatically from textual or structural document content itself, via so called Information Extraction (Grishman, 1997). It may have a simple form, e.g. identification that the document contains a phone number, an authoring information, proper names (of people, firms, geographic locations etc.), extraction of keywords or assignment to thesaurus. Even hyponym/hypernym hierarchies may be created automatically under some circumstances for a document collection (Tjong, Sang and Hofmann, 2007).

Lastly, semantic information may be extracted from the links between Web pages. The address of the Web page (the directory hierarchy) may contain indications of broader/narrower concepts (Han and Kamber, 2001). The personalized random walks starting at given page may, for instance, identify the cluster it belongs to, so that one gets perhaps other pages tagged semantically to transfer to the given page (though there exist obvious difficulties when establishing relation between links and content, see Björneborn, 2004).

2.2. Impact of semantics on document clustering and classification

One would wish to have self-contained documents, since their content can then be easily understood. However, in practice, this requirement is rarely, or even never, satisfied and to understand properly a given document we have to refer to some wider context. This context may be explicitly or implicitly expressed in terms of semantic information, which is provided in a way substantially different from the text of the document as such. In other words, semantic information, although associated with the document, has to be stored and handled in a different way than the entire document.

Frequently, the documents collected by a Web crawler are represented as the points in the vector space spanned by the terms (words, word stems or phrases) occurring in the document collection. Physically, each such a point is implemented as a vector, or list, whose entries represent quantitative (statistical) properties of the terms from this document. Then, the so-called *tfidf*, i.e. term frequency – inverse document frequency (Baeza-Yates and Ribeiro-Neto, 1999; Jones, 1972) approach is used. Namely, the weight $w_{t,d}^D$ of the term t from the document d belonging to a collection D of all documents is computed as

$$tfidf\,(t,d) = w_{t,d}^D = f_{t,d} \cdot \log\left(\frac{|D|}{f_t^D}\right) \tag{1}$$

where $f_{t,d}$ is the number of occurrences of term t in document d, f_t^D specifies how many documents from D contain the term t and $|D|$ is the total number of documents in D.[5]

This way each document $d \in D$ is represented by (rather sparse) vector

$$\mathbf{d} = (w_{t_1,d}^D, w_{t_2,d}^D, \ldots, w_{t_{|T|},d}^D) \tag{2}$$

where T is the set of terms, or dictionary, taken into account, and $|T|$ stands for the cardinality of the dictionary.

To attach semantic information to the documents we enrich their representation. The simplest approach is to introduce additional attributes of documents (of the form: <attribute name, attribute value>, for example <"document about","graph theory">, <"document contains a phone number","yes">, <"phone number in the document","++4822634567">) that will represent manually assigned categories, to which the document belongs (taken from a

[5] This representation is subject to diverse normalization operations, e.g. one may require that $f_{t,d}$ sums up to 1 within the document, that the $tfidf$ weights form a unit vector for a document etc.

subject index, topics index, the list of authors, etc.), or other information. In other words, each document d is represented now as a vector in extended space (T, C), where C stands for the additional attributes. These attributes themselves may be grouped into broader categories (e.g. one can define a group of attributes concerning authors, a group of attributes concerning topics in mathematics etc.). Further, the relationships among the attributes can be modeled via crisp or fuzzy relations. The attributes may be accompanied by additional textual information, explaining in more detail the meaning of the attribute.

When operating with such extended representation, we must design proper similarity measure between documents reflecting both the statistical information about the document (expressed in terms of its *tfidf*'s) and semantic information[6] associated with analyzed documents.

The statistical and semantical points of view can be integrated under the so-called contextual view of the document collection (Ciesielski and Kłopotek, 2006, 2007). It says that the term importance in the document space should not be evaluated by taking into account all collected documents, but rather by considering a subset $G \subset D$ of documents representing a particular topic. By a "topic" we mean here uniform clusters obtained during an initial clustering of documents, using the standard *tfidf* representation (1) of documents. Having identified the clusters G, called hereafter "contexts", we look at the context when further processing the documents. Thus, the inverse document frequency is computed within a context G, implied by specific topic, i.e. the quotient $|D|/f_t^D$ in equation (1) has to be replaced by $|G|/f_t^G$. This leads to substantial diversification of importance of terms in different contexts.

More precisely, the contextual model consists of the following steps: (1) cluster the original document set, using e.g. fuzzy-c-means, supported by CF-trees, into clusters of size manageable by structure-generating clustering algorithms (that is, by WebSOM, Kohonen et al., 2003; GNG, Fritzke, 1997; aiNet, de Castro and Timmis, 2002), that are later used as contexts, (2) create document maps for each context, (3) create whole-collection document map using representatives of each context.

In this way, a small hierarchy of clusters is created. The contexts are the groups of documents at the highest level. Within each context we create a document map for the context documents, which means that the documents within a cluster are put together into groups being cells of the document map. In some versions of our method we group also documents within a cell within a context via e.g. GNG or aInet.

Note that this approach to the issues of clustering differs from the hierarchical approach in Jing, Ng and Huang (2007), Frigui and Nasraoui (2004), because we allow for different clustering strategies for the whole collection and within the identified contexts, while Jing, Ng and Huang (2007) insist on having

[6] Though the *tfidf* may be viewed as a simple form of semantic information, we will refer to "semantic information" as to the more elaborate information about the document.

a unified process of cluster identification in the whole space and in the subspaces, and allow for k-means only. The approach of Cheung and Zeng (2007) is restricted to Gaussian mixtures. Beside this, in our approach, the resulting clusters are structured, either in terms of a WebSOM map, or GNG or aiNet network. Therefore, subsequently we will reserve the term "hierarchical" to the approaches using the same clustering methods at all hierarchy levels, while our concept with differentiation of them will be termed "contextual".

So, by enabling extended representation, we can distinguish between *hierarchical* and *contextual* model. In the former, the set of terms, with *tfidf* weights defined in equation (1), is identical for each subgroup of documents, while in the latter each subgroup is represented by different subset of terms weighted in accordance with equation (6). Finally, when we do not split the entire set of documents and we construct a single, "flat", representation for whole collection – we will speak of a *global* model.

2.3. Representation of semantic information within the BEATCA system

The approach described above was successfully implemented in the search engine BEATCA (Ciesielski and Kłopotek, 2007; Kłopotek et al., 2007) equipped with a map visualization of the content of collected documents. Such a map consists of a set of cells (usually hexagonal or rectangular) tiled in a plane or another surface in Euclidean or other space. Each cell is assigned a set of documents (in fact, the set of cells represents a clustering of documents) in such a way that the documents close on the map (same cell, neighboring cells etc.) are close by their content.

A well known representative of such a paradigm is WebSOM[7], although other systems are available – consult Becks (2001) or Kłopotek et al. (2007) for more detailed discussion of this subject. It is important to notice that the process of map creation with the help of WebSOM, as applied in WebSOM, is time consuming. Hence, other self-organizing models, like GNG (growing neural gas[8]), aiNet (an artificial immune network technique, Ciesielski, Wierzchoń and Kłopotek, 2006) as well as their hierarchical and contextual versions, were implemented and tested within the BEATCA (Kłopotek et al., 2007) framework.

All these techniques operate on the vector representation of the documents, mentioned in previous subsection and proper definition of the distance, or similarity measure between the documents determines their efficiency. We discuss this problem shortly in what follows.

3. Semantic-based clustering approach

The approach to clustering a collection of documents, as performed by the BEATCA system, relies upon effective exploitation of specific information about

[7] See e.g. http://websom.hut.fi/websom/

[8] See e.g. http://www.ki.inf.tu-dresden.de/~fritzke/research/incremental.html

the document content and context. It is a multistage process exploiting the extended representation over the space (T, C), as described in Subsection 2.2.

First, the document collection is split into smaller, more homogenous subsets, called "contexts". Conventional clustering methods are applied at this stage, i.e. k-means or its fuzzy variant Fuzzy-ISODATA (Bezdek and Pal, 1991) for small contexts, or fuzzified versions of the BIRCH algorithm (Zhang, Ramakrishnan and Livny, 1996) in case of large collections. Next, treating each context separately, its clustering is performed, and results of this clustering are represented in the form of an appropriate sub-map.

The term "representative" does not mean here the traditional cluster centroid or medoid, because we have the option to use histograms of term weights within each context (see Ciesielski and Kłopotek, 2007), as they reflect better the content of the respective document group.

3.1. k-means algorithm with modified similarity measure

k-means clustering is a popular method of splitting data into disjoint subsets. To fit it into our framework two essential modifications are needed. First, we replace the traditional concept of centroid as an averaged point in vector space by term weight histograms. A much richer characterization of the importance of a term for a context becomes possible, so that a group representative is not just a point in vector space, but each dimension is equipped with a distribution. Given this, we need next to create a method of calculating the distance between documents and the new type of centroid must be proposed. In this section we describe the second topic only. The first issue was handled extensively in Ciesielski and Kłopotek (2007).

With the extended representation $\mathbf{d} = (\mathbf{w}, \mathbf{c})$, where $\mathbf{w}$ is the vector with *tfidf* entries and $\mathbf{c}$ represents semantic information, we define supervised similarity sim_s of the two documents d_i, d_j simply as cosine of the angle between the two above-defined vectors:

$$sim_s(d_i, d_j) = \cos\{(\mathbf{w_i}, \mathbf{c_i}), (\mathbf{w_j}, \mathbf{c_j})\}. \tag{3}$$

According to Heaps' law (Baeza-Yates and Ribeiro-Neto, 1999), the number of different content terms in the collection usually surpasses the number of assigned categories. Thus, we control the impact of the two subspaces – unsupervised (textual content) T and supervised (multilabel categorization) C on the joint similarity of two documents by defining sim_s as the weighted sum of two components:

$$sim_s(d_i, d_j) = (1 - W) \cdot \cos(\mathbf{w}_i, \mathbf{w}_j) + W \cdot \cos(\mathbf{c}_i, \mathbf{c}_j) \tag{4}$$

where $W \in [0, 1]$ is a user-defined parameter. For $W = 0$ we obtain standard unsupervised cosine similarity measure, while for $W = 1$ we ignore textual similarity, focusing on the semantic attributes only. From our experiments it follows that in general, the weight W should be proportional to the quotient

$|T|/|C|$ (cardinality of the set of terms divided by the cardinality of the set of attributes). Particularly, in the experiments described in Section 5 we set $W = 0.9$, focusing primarily on the attributes, not on the terms occurring in the documents.

3.2. Semantic-based competitive clustering algorithms

Competitive clustering algorithms, like WebSOM or GNG, are attractive because of at least two reasons. First, they adaptively fit to the internal structure of the data. Second, they offer natural possibility of visualizing this structure by projecting high-dimensional input vectors to a two-dimensional grid structure, called a map. This map preserves most of the topological information of the input data.

Each cell of the map is described by the so-called reference vector, or "centroid", being a concise characterisation of the microgroup, defined by such a cell. These centroids attract other input vectors with the force proportional to their mutual similarity. In effect, weight vectors are ordered according to their similarity to the cells of the map. Further, the distribution of weight vectors reflects the density of the input space. Reference vectors of cells neighboring on the map are also closer (in original data space) to one another than those of distant cells.

The centroids are in fact "averaged" documents, i.e. they represent rather abstract entities. In our approach we use "typical" (for a given cluster) instead of "averaged" documents. The typicality is defined by means of the histogram of terms occurrence in the cluster. More precisely, the quantity $H_{t,G}(i)$ defines the fraction of documents, for which the value of the weighting function $w^G_{t,d}$ belongs to the i-th subinterval $[a_i, a_{i+1}] \in [0, 1]$. Of course, to use such histograms, we have to normalize the document vectors $\mathbf{d}$ first, and next, we split the domain $[0, 1]$ into appropriate number of subintervals I. Details describing full procedure can be found in Ciesielski and Kłopotek (2007).

To justify the use of the histograms, note that the weight of a term in a document depends usually on three factors: the number of its occurrences in the document, the number of documents containing this term, and the length of the document. The term which occurred several times in a short document and does occur in only a few other documents, is awarded by high weight, as it is characteristic for such a group. Terms that occur everywhere, or those occurring one time in a very long document, will have low weight.

The histogram then reflects the probability distribution that a particular term occurs with a given weight in the documents forming particular context. The terms, that have only low weights in the documents, are not important within the context. Those with strong share of high weight occurrences can be considered important in discriminating the documents within the context. Now, as "typical" we understand the document containing only those terms that are labeled as important for a given context.

Analogously to content terms, one can build histograms of the distribution of additional semantic attributes within the context and the "typical" document can be defined now as the one sharing important (typical) semantic attributes with this context.

4. Contextual models

The text documents are not uniformly distributed in the space of terms. Frequency distribution of a particular term depends strongly on document location (and is expected to be similar for the neighboring documents) in the vector space. As already stated, in our approach, the set of documents is initially divided automatically into a number of homogenous and disjoint subgroups, each of which is described by unique (but usually overlapping) subset of terms. Then the selection of significant and insignificant terms for efficient calculation of similarity measures during map formation step appears to be more robust (Ciesielski and Kłopotek, 2007).

It has to be stressed that when producing contextual models we operate simultaneously in two spaces: the space of documents and (extended) space of terms. The whole algorithm iteratively adapts: (a) document representation, (b) description of contexts by means of the histograms, and (c) the degrees of membership of documents in the contexts as well as weights of the terms $w_{t,d}^G$. As a result of such a procedure we obtain homogenous groups of documents together with the description fitted individually to each group (Ciesielski and Kłopotek, 2007).

At the beginning, the whole document set D is split arbitrarily into a few, say 2–5, groups. Next, each group is recursively divided until the documents inside a group meet required homogeneity or quantity criteria. After such a process we obtain hierarchy, represented by a tree of clusters and each cluster corresponds to a context. In the last stage, the groups with cardinality smaller than a predefined value, are merged with the closest group. Similarity measure is defined as a single-linkage cosine angle between the vectors representing both cluster centroids.

The second, crucial, phase of contextual document processing is the division of terms space (dictionary) into – possibly overlapping – subspaces. In this case it is important to calculate fuzzy membership values representing importance of a particular term in different contexts (and implicitly, ambiguity of its meaning). The fuzzy within-group membership of the term, m_t^G, is estimated according to the equation

$$m_t^G = \frac{\sum_{d \in G} f_{t,d} \cdot m_d^G}{f_t^D \cdot \sum_{d \in G} m_d^G} \tag{5}$$

where m_d^G denotes the degree of membership of document d in the group G.

Next, vector-space representation of a document is modified so as to take into account the document context. This representation increases the weights of terms which are significant for a given contextual group and decreases the weights of insignificant terms. In the extreme case, insignificant terms are ignored, what leads to the (topic-sensitive) reduction of representation space dimensionality. The significance of the term in a given context is computed as

$$w_{t,d}^{G} = m_t^G \cdot f_{t,d} \cdot \log\left(\frac{|G|}{f_t^G}\right). \tag{6}$$

The main idea behind the proposed approach is to replace a single flat model by a set of independently created contextual models and to merge them together into a hierarchical model. Training data for each model is a single contextual group. Each document is represented as a standard referential vector in the terms space. However, *tfidf* weights (1) in vector components are replaced by $w_{t,d}^G$.

5. Experimental results

The experiments reported in this section have several goals. First of all, we show that exploiting semantic information (in our case – partially labelled, multicategorical data) increases the quality of clustering, both in terms of textual content clustering as well as grouping of other attributes (even those not present in vectors representing documents, but only implicitly correlated with semantic information we used, e.g. book authors).

Second, we argue that semi-supervised clustering (e.g. exploiting manually labeled subset of the document collection) enables more control on the clustering process, and produces results that are agreeable with user expectations and also with user's subjective view on what is ”natural clustering” (results profiling).

Finally, we verify that a combination of the contextual approach with semantic information, represented as additional dimensions (attributes) in vector space, is admissible from the computational point of view, i.e. its time complexity is quite acceptable even in the case of larger collection of documents, and leads to high-quality results (Kłopotek et al., 2007; Ciesielski and Kłopotek, 2006, 2007).

The experiments were conducted on the real dataset, consisting of data on 5615 books from the library of the Institute of Computer Science[9].

5.1. Library dataset

Library dataset contains brief textual information on each book: title, authors, publisher, publication year, ISBN, etc. Publication abstracts are not available. 5449 out of 5615 books have been manually classified to one or more ACM Classification categories[10]. 368 book subcategories (so-called *third-level categories*) are organized into taxonomy, rooted into 11 top-level categories. Basic statistics

[9] Available online at `http://www.ipipan.waw.pl/cgi-bin/klopotek/oj`

[10] see `http://www.acm.org/about/class`

are presented in Table 1. Please note that categories are overlapping, and so the total number of entries in the third column is greater than the total number of books and sums up to 16,209.

Table 1. Top-level category statistics

Top-level category	**Category name**	**Total no. books**	**Total no. subcategories**
[a]	General Literature	59	5
[b]	Hardware	380	56
[c]	Computer Systems Organization	1541	29
[d]	Software	2738	48
[e]	Data	509	8
[f]	Theory of Computation	2765	27
[g]	Mathematics of Computing	908	24
[h]	Information Systems	1868	43
[i]	Computing Methodologies	4113	75
[j]	Computer Applications	763	10
[k]	Computing Milieux	565	43
	Total number of entries	**16209**	**368**

Each book may belong to more than one category. The distribution of the number of subcategories assigned to books is presented in Table 2. Average number of subcategories assigned to a single book is 16209/5615 $\approx$ 2.88.

Table 2. The total number of books having a given number of the third-level subcategories assigned

# subcategories	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
# books	166	1126	653	502	437	456	380	317	216
# subcategories	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**
# books	109	36	11	20	5	10	4	2	1

For instance, the proceedings volume of the International Workshop on Artificial Neural Networks (IWANN'93) has been classified to 17 different subcategories: twice to subcategories of category [b], 3 times to subcategories of [c], 3 to [f], 1 to [h], 6 to [i], 2 to [j].

As described earlier, beside standard representation of the book content via $tfidf$ weights one can build its representation by referring to the categories to which it belongs. The weight of an attribute (or category) a for book b is defined as the product $w_{a,b} = f_{a,b} \cdot \log(N/f_a)$, where $f_{a,b}$ is the weight of a for book b (for subcategories equal to 1, for top-level categories equal to the total number of assigned subcategories), and f_a is the total number of books assigned to a particular category.

Such a representation shares most statistical properties with standard $tfidf$ representation of textual document content. In particular, frequency of categories follows power-law distribution (the most frequent category was [a.2] "Artificial Intelligence", assigned to 619 books; at the other end there were 62 categories with zero frequency - usually "Miscellaneous" subcategory of various top-level categories).

Beside librarian classification, there was another type of attributes extracted from book descriptions, i.e. authors of each book. Intuitively, author-related attributes should be positively correlated with their research areas and thus also with book categories. It should be stressed that only attributes related to ACM Classification categories (either top-level categories or subcategories) were used by similarity measure (4). Still, as experiments showed, through positive correlation, exploitation of supervised similarity measure affected also the distribution of author-related attributes within clusters.

5.2. Experimental setting and quality measures

When presenting experimental results, we focused only on standard supervised and unsupervised measures used to evaluate WebSOM clustering results, although other more precise measures are possible (see Ciesielski, Wierzchoń and Kłopotek, 2006). The two supervised measures are average cluster purity (ACP) and average cluster entropy (ACE) introduced by Frigui and Nasraoui (2004). The two unsupervised measures are average distance of neighbor cells, also known as average map quantization (AMQ) and average cluster diameter (ACD), which is analogous to the average document quantization (ADQ) measure, but is based on complete-linkage distance between books in a cluster rather than the distance from centroid (see Kohonen et al., 2003). The lower values of AMQ and ADQ measures correspond, respectively, to more "smooth" inter-cluster transitions and more "compact" clusters.

In general, we search for the partitions characterized by high purity and low entropy. High values of ACP correspond to the situation when the clustering agrees with given criteria, while low values of ACE signalize homogenous distribution of the categories within each group. On the other hand, low values of the unsupervised measures indicate well formed and compact clusters.

Each of the four measures was calculated for four different vector representations of a document (i.e. single book): (a) standard, content-based representation, (b) pure semantical representation pertaining to book subcategories, (c) aggregated weights of top-level categories, and (d) representation restricted to weights of author-related attributes.

Beside different representations, subsets of books used to compute these measures were varied, i.e. we used

- full collection,
- books having at least one third-level subcategory assigned
- books without any third-level subcategory assigned
- 11 subsets consisting of books having particular top-level category ([a]-[k]) assigned (see Table 1)

The figures presented in the following sections contain four box-and-whisker plots. Each box-and-whisker plot evaluates 10 WebSOM models, trained on 10-fold crossvalidated sample of books. Four different WebSOM models are presented in each plot:

- **cos/H**: standard WebSOM model, with *tfidf* weights and cosine similarity measure,
- **cos/C**: contextual WebSOM model, with contextual $w_{t,d}^G$ weights and cosine similarity measure,
- **sup/H**: WebSOM model with *tfidf* weights and supervised similarity measure (4),
- **sup/C**: WebSOM model with contextual weights and supervised similarity measure (4).

5.3. Supervised evaluation of library books

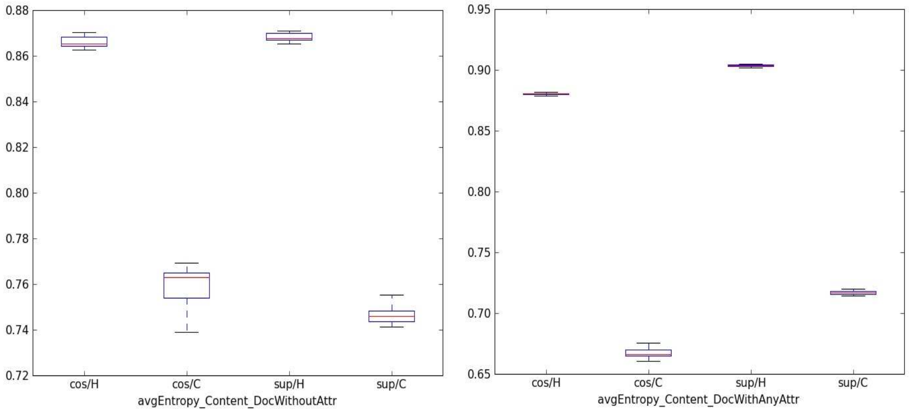

Figure 1. Average entropy of content terms in books: (a) books without and (b) books with assigned third-level subcategories

In both cases, i.e. of books without and with assigned third-level subcategories, contextual (C) models are better than hierarchical ones (H). We have discussed this issue in our earlier papers, see, e.g., Ciesielski and Kłopotek (2006). The difference is statistically significant and quite large, especially in case of books with assigned third-level subcategories (Fig. 1(b)), where best models feature mean entropy values below 0.7, while hierarchical models feature 0.9.

In case of books without assigned subcategories (Fig. 1(a)), the **sup/C** model, i.e. supervised contextual model is the best among all models, while in case of documents with subcategories assigned, the best model is unsupervised one **cos/C**. Even though the difference of entropy between **cos/C** and **sup/C** is quite small, this seems to be counterintuitive. We suppose that this phenomenon is implied by the fact that significantly better clustering of book categories in case of **sup/C** model (see Fig. 2) improves also clustering of the remaining part of the dataset, i.e. books without subcategories. At the same time, the supervised similarity measure, which operates on the content terms, subcategories and author-related attributes, sacrifices content clustering for better clustering of subcategories.

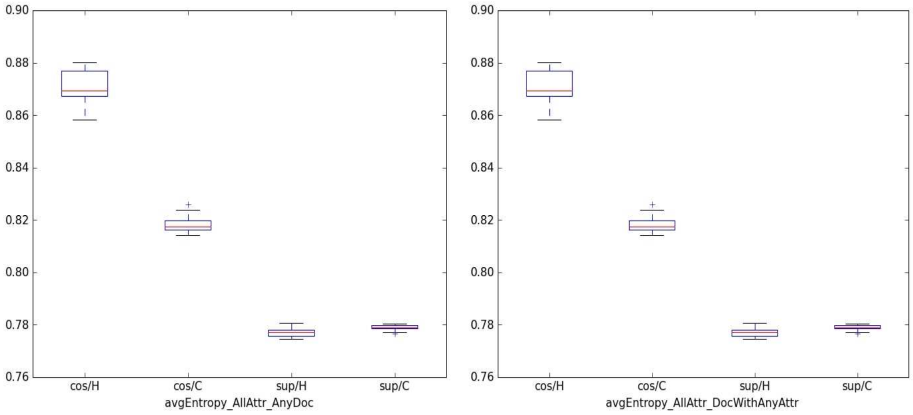

Figure 2. Average entropy of all additional attributes describing: (a) books without and (b) books with assigned third-level subcategories

Results in Fig. 2 are not surprising. Here we observe clustering of all additional attributes, i.e. librarian categories and book authors. The two supervised models – hierarchical and contextual – have lower entropy and thus show better clustering of additional attributes. There is almost no difference between hierarchical and contextual case here (although the variance of contextual model is lower). The worst (in terms of mean entropy as well as its variance in cross-validated samples) model is provided by the standard **cos/H** approach, i.e. hierarchical WebSOM model with cosine similarity measure.

A similar, although not presented graphically, behavior is observed when the entropy of only book category attributes (without author-related attributes) is computed. There is statistically significant (and large) difference in favor of the supervised measure. But in this case the difference between the two supervised cases – hierarchical and contextual one – is also statistically significant.

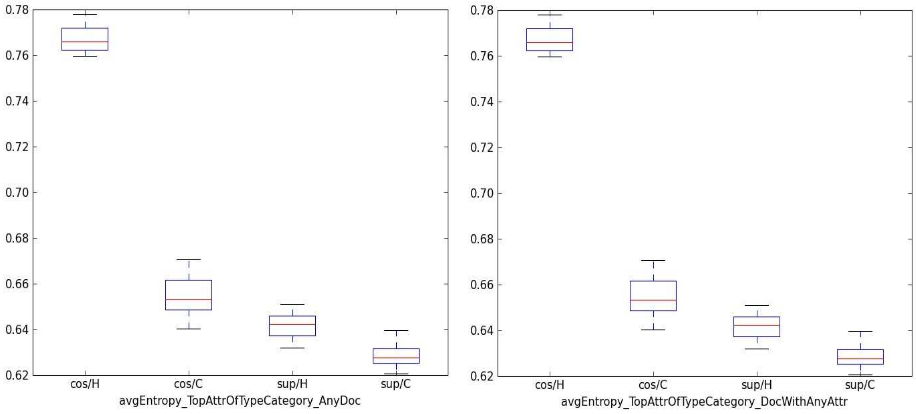

Figure 3. Average entropy of aggregated (top-level) categories: (a) books without and (b) books with assigned third-level subcategories

When aggregating attributes related to book third-level subcategories and computing the entropy of the distribution of top-level categories within clusters (Fig. 3), we observe significant difference between hierarchical (H) and contextual (C) models. The absolute values of the entropy of aggregated top-level categories are lower than the entropy of third-level subcategories (mainly because the number of different attributes is much lower in the former representation). Hence, the documents from the top-level categories form well separated and compact clusters, and these clusters occupy distinct areas of the map.

Finally, considering the entropy of the distribution of author-related attributes only (Fig. 4), we state again that both supervised models (**sup/H** and **sup/C**) show better clustering properties than models trained with cosine measure (**cos/H** and **cos/C**). The differences between hierarchical (H) and contextual models (C) trained with the same similarity measure are insignificant. One should notice quite high value of entropy in all four cases. We consider it to be partially due to the fact that a handful of authors appear in more than one book. The other reason might be the fact that we parsed author-related information from textual content by a set of approximate parsing rules, and we found several mistakes afterwards.

The results for cluster purity are analogous to those for cluster entropy and due to the lack of space, we do not present all plots here.

It should also be noted that in case of aggregated (top-level category) attributes, purity results above 0.5 (with very low variance) indicate very good clustering of categories (Fig. 5). This applies particularly to the case when it is not possible to attain maximal, equal 1, value of purity, mainly because most of the books have more than one subcategory assigned (see Table 2).

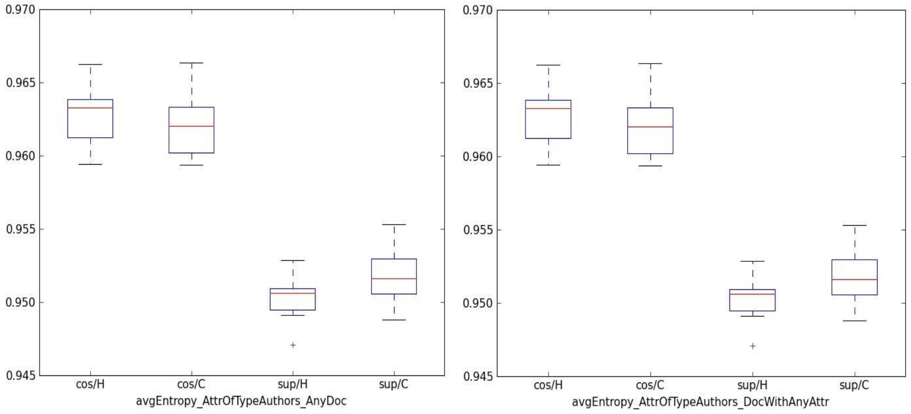

Figure 4. Average entropy of book author-related attributes: (a) books without and (b) books with assigned third-level subcategories

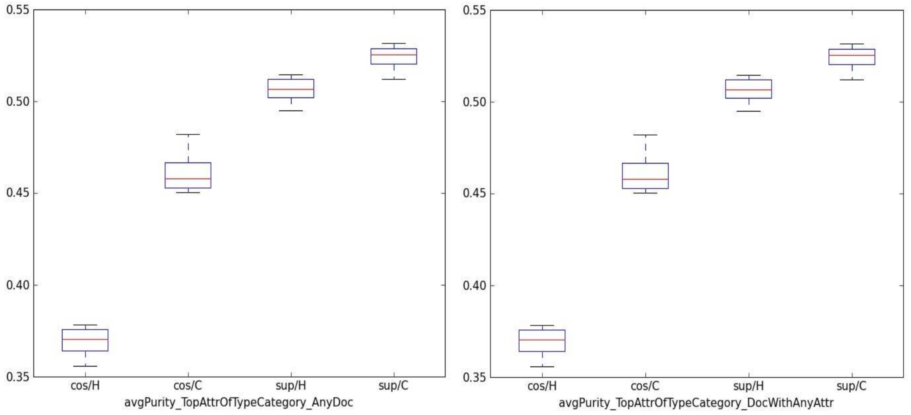

Figure 5. Average purity of aggregated (top-level) categories: (a) books without and (b) books with assigned third-level subcategories

5.4. Unsupervised evaluation of library books

We just briefly conclude on experiments with analysis of unsupervised measures for WebSOM clustering models. Recall, that this kind of evaluation does not allow comparing the distribution of categories with respect to different similarity measures applied. However, some interesting properties of the resulting models can also be noticed here.

The lower value of map quantization means that adjacent cells on a map are similar to each other, thus the distribution of particular entities (e.g. content terms or additional semantic attributes) is smooth, when translocating from one

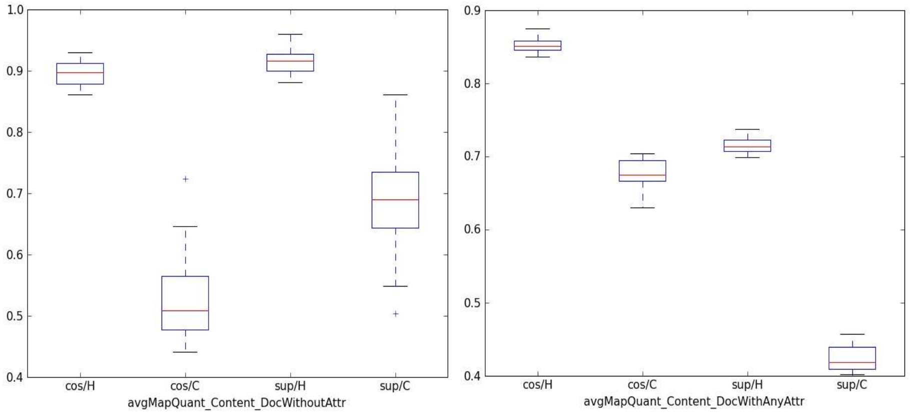

Figure 6. Average map quantization of content terms: (a) books without and (b) books with assigned third-level subcategories

cell to another. The best map quantization with respect to document content terms is achieved by contextual models (Fig. 6(a)), and particularly by the cosine contextual model **cos/C**. This supports observations we made earlier (see Ciesielski and Kłopotek, 2006) concerning crucial role of identification of important terms in contextual document vector subspaces and their impact on text clustering. Particularly, in case of books with additional semantic attributes (i.e. book categories or authors), the best quantization is achieved for **sup/C** model; this value is almost half of **cos/C** – the second-best model (Fig. 6(b)).

Surprisingly, quite opposite situation is observed in case of additional attribute quantization (Fig. 7). Here the lowest value of this measure is 0.8 for **cos/H** model. We suppose it is caused by the very high sparsity of attribute information, which leads to almost-orthogonal vectors representing attribute distribution in adjacent cells. This claim is supported by the observation of aggregated 11 top-level categories. In this case, average map quantization is 0.33 for **cos/H** model, and around 0.55 in the other three cases. In case of book author-related attributes, quantization value is again near 1, meaning orthogonal vectors (the best of four models is **cos/C**, with map quantization 0.96).

Summarizing, we can say that the analysis of average document quantization, measuring compactness of clusters represented by a set of books assigned to a single cell, confirms that additional semantic information affects the quality of text clustering. The difference of document quantization values is the biggest in case of aggregated (top-level) book categories, where both supervised models have quantization around 2.5, versus 3.0 for the models with cosine similarity (Fig. 8). Similar advantage of supervised models is observed also in case of unaggregated, third-level categories; in this case the difference is not so radical but still statistically significant (about 2.2 for supervised models, and 2.5 for the models with cosine similarity).

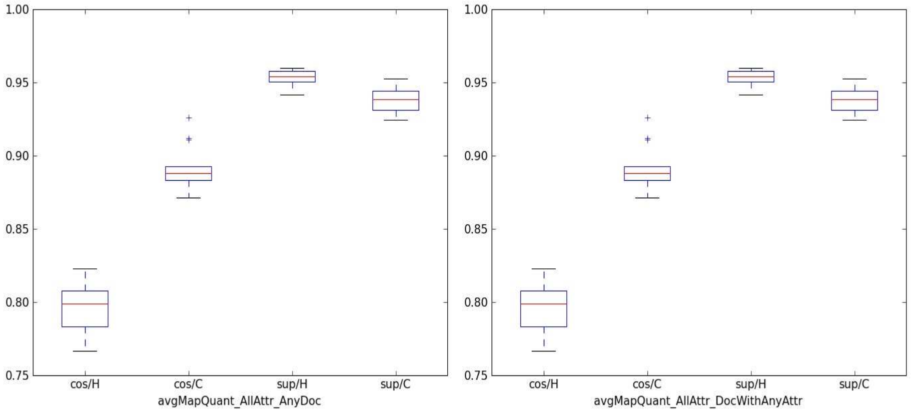

Figure 7. Average map quantization of additional attributes: (a) books without and (b) books with assigned third-level subcategories

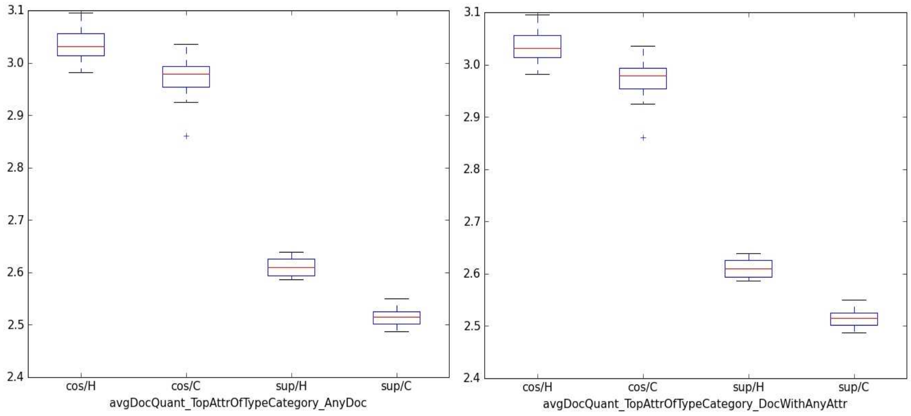

Figure 8. Average document quantization of aggregated (top-level) categories: (a) books without and (b) books with assigned third-level subcategories

6. Conclusions and future research

A general conclusion is that by introducing semantic information we can improve the whole process of gathering and representing information about a domain.

First of all, we gain an extension of the search engine possibilities by clustering with human-readable and human-controllable criteria, such as good distribution of book categories or book authors within clusters.

Implicit integration of well-defined clustering criteria via exploitation of manually prepared supervised information substantially improves the quality of clusters. This is particularly useful in cases where textual-only information is of low-quality and/or unavailable (as in the case of book descriptions mentioned earlier). Furthermore, we have observed a kind of positive correlation effect between various kinds of attributes, in our case-book categories and authors. While clustering with exploitation of book categories only, we observed improvement of quality of clusters measured via purity of author-related attributes.

In our future research, we plan to investigate whether the user profile, describing his/her information needs, can also be taken into account as additional dimension(s) in the document space, so that a conceptual framework for personalization of documents clustering and document classification is swiftly achievable. Such a form of integration may be of importance, e.g., for recommender systems, especially for content-based recommenders, that suffer badly from the "cold start" problem of missing initial usage information of the item by the user – see Mobasher (2005).

Lastly, let us notice that the process of learning contextual models is almost the same as that of learning standard, non-contextual, models. But since each constituent model (and the corresponding contextual map) can be processed independently, the whole process of acquiring contextual maps can be distributed and calculated in parallel. Also a partial incremental update of such models appears to be much easier to perform, both in terms of model quality, stability and time complexity. The possibility of incremental learning stems from the fact that the very nature of the learning process is iterative. So, if new documents come, we can consider the learning process as having been stopped at some stage and it is resumed now with all the documents. We claim that it is not necessary to start the learning process from scratch neither in the case when the new documents "fit" the distribution of the previous ones nor when their term distribution is significantly different – deeper discussion of this topic can be found in Kłopotek et al. (2007).

Acknowledgement

This research has been partially supported from the science budget for the years 2007-2009 as research Project N516 015 32/1906 "A growing map of object space as a new paradigm for recommender systems".

References

Baeza-Yates, R. and Ribeiro-Neto, B. (1999) *Modern Information Retrieval.* ACM Press.

Becks, A. (2001) *Visual Knowledge Management with Adaptable Document Maps.* GMD Research Series, **15**.

Bezdek, J.C. and Pal, S.K. (1991) *Fuzzy Models for Pattern Recognition: Methods that Search for Structures in Data.* IEEE Press, New York.

Björneborn, L. (2004) Small-world link structures across an academic web space: A library and information science approach. PhD dissertation. www.db.dk/LB.

Bonino, D., Corno, F. and Farinetti, L. (2003) DOSE: a Distributed Open Semantic Elaboration Platform. In: *ICTAI 2003, The 15th IEEE International Conference on Tools with Artificial Intelligence*, Sacramento, California. IEEE Computer Society.

Cheung, Y.M. and Zeng, H. (2007) A Maximum Weighted Likelihood Approach to Simultaneous Model Selection and Feature Weighting in Gaussian Mixture. In: *Artificial Neural Networks - ICANN 2007*, **LNCS 4668**, Springer, 78–87.

Ciesielski, K. and Kłopotek, M.A. (2006) Text data clustering by contextual graphs. In: L. Todorovski, N. Lavrac and K.P. Jantke, eds., *Discovery Science (DS-2006).* **LNAI 4265**, Springer-Verlag, 65–76.

Ciesielski, K. and Kłopotek, M.A. (2007) Towards Adaptive Web Mining: Histograms and Contexts in Text Data Clustering. In: M.R. Berthold and J. Shawe-Taylor, eds., *Intelligent Data Analysis (IDA-2007).* **LNCS 4723**, Springer-Verlag, 284–295.

Ciesielski, K., Wierzchoń, S.T. and Kłopotek, M.A. (2006) An Immune Network for Contextual Text Data Clustering. In: H. Bersini and J. Carneiro, eds., *5th International Conference on Artificial Immune Systems (ICARIS-2006).* **LNCS 4163**, Springer-Verlag, 432–445.

Corby, O., Dieng-Kuntz, R. and Faron-Zucker, C. (2004) Querying the Semantic Web With Corese Search Engine. In: R. Lopez de Mantaras and L. Saitta, eds., *Proc. 16th ECAI/PAIS*, Valencia, Spain. IOS Press, 705–709.

Davies, J., Weeks, R. and Krohn, U. (2002) QuizRDF: Search Technology for the Semantic Web. In: *WWW2002 Workshop on RDF and Semantic Web Applications.* Hawaii, USA.

de Castro, L.N. and Timmis, J. (2002) *Artificial Immune Systems: A New Computational Intelligence Approach.* Springer.

Ding, L., Finin, T., Joshi, A., Pan, R., Cost, R.S., Peng, Y., Reddivari, P., Doshi, V.C. and Sachs, J. (2004) Swoogle: A Search and Metadata Engine for the Semantic Web. In: *Proceedings of the Thirteenth ACM Conference on Information and Knowledge Management.* ACM Press.

Esmaili, K.S. and Abolhassani, H. (2006) A Categorization Scheme for Semantic Web Search Engines. In: *Proceedings of the 4th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA-06)*, Sharjah, UAE, 171–178.

Floridi, L. (2005) Semantic Conceptions of Information. In: *Stanford Encyclopedia of Philosophy.* http://plato.stanford.edu/entries/information-semantic/.

Frigui, H. and Nasraoui, O. (2004) Simultaneous Clustering and Dynamic Keyword Weighting for Text Documents. In: M. Berry, ed. *Survey of Text Mining.* Springer, 45–70.

Fritzke, B. (1997) A Self-Organizing Network that Can Follow Non-stationary Distributions. In: *ICANN '97: Proceedings of the 7th International Conference on Artificial Neural Networks*, Springer-Verlag, 613–618.

Grishman, R. (1997) Information extraction: Techniques and challenges. In: M.T. Pazienza, ed., *Information Extraction A Multidisciplinary Approach to an Emerging Information Technology.* **LNCS 1299**, Springer, 10–27.

Guha, R., McCool, R. and Miller, E. (2003) Semantic Search. In: *Proc. of the 12th International Conference on World Wide Web.* ACM, New York, NY, 700-–709.

Han, J. and Kamber, M. (2001) *Data Mining: Concepts and Techniques.* Morgan Kaufmann.

Heflin, J. and Hendler, J. (2000) Searching the Web with SHOE. In: *Proc. 17th National Conference on Artificial Intelligence (AAAI-2000).* AAAI/MIT Press, Menlo Park, 443-449.

Jing, L., Ng, M.K. and Huang, J.Zh. (2007) An Entropy Weighting k-Means Algorithm for Subspace Clustering of High-Dimensional Sparse Data. *IEEE Trans. on Knowl. and Data Eng.*, **19**(8), 1026–1041, doi:http://dx.doi.org/10.1109/TKDE.2007.1048.

Kłopotek, M.A., Wierzchoń, S.T., Ciesielski, K., Dramiński, M. and Czerski, D. (2007) *Conceptual Maps of Document Collections in Internet and Intranet. Coping with the Technological Challenge.* Institute of Computer Science of Polish Academy of Sciences, Warsaw.

Kohonen, T., Kaski, S., Somervuo, P., Lagus, K., Oja, M. and Paatero, V. (2003) Self-organization of very large document collections. Technical Report, University of Technology, Helsinki, Finland.

Mobasher, B. (2005) *Practical Handbook of Internet Computing.* Chapter: Web Usage Mining and Personalization, CRC Press, 342–380.

Priebe, T., Schlaeger, C. and Pernul, G. (2004) A Search Engine for RDF Metadata. In: *Proc. of the DEXA 2004 Workshop on Web Semantics (WebS 2004)*, Zaragoza, Spain.

Sheth, A., Bertram, C., Avant, D., Hammond, B., Kochut, K. and Warke, Y. (2002) Managing Semantic Content for the Web. *IEEE Internet Computing*, **6**(4), 80–87.

Sparck, J. (1972) A statistical interpretation of term specifity and its application in retrieval. *Journal of Documentation*, **28**, 111–121.

Spyns, P., Oberle, D., Volz, R., Zheng, J., Jarrar, M., Sure, Y., Studer, R. and Meersman, R. (2002) OntoWeb - A Semantic Web Community Portal. In: D. Karagiansis and U. Reiner, eds., *Proceedings of the 4th International Conference on Practical Aspects of Knowledge Management.* **LNAI 2569**, Springer, 189—200.

TAMMA, V., BLACOE, I., SMITH, B. and WOOLDRIDGE, M. (2004) SERSE: searching for semantic web content. In: *Proceedings of the 16th European Conference on Artificial Intelligence, ECAI 2004*, Valencia, Spain. IOS Press.

TERZIYAN, W., KAYKOVA, O., KLOCHKO, O., TARANOV, A., KHRIYENKO, O., KONONENKO, O. and ZHARKO, A. (2004) Semantic Search Facilitator. Technical Report, Industrial Ontologies Group.

TJONG, E., SANG, K. and HOFMANN, K. (2007) Automatic Extraction of Dutch Hypernym-Hyponym Pairs. taalunieversum.org/taal/technologie/stevin/documenten/clin2007_cornetto.pdf.

WINSTON, M.E., CHAFFIN, R. and HERRMANN, D. (1987) A taxonomy of part-whole relations. *Cognitive Science*, **11**(4), 417–444.

ZHANG, T., RAMAKRISHNAN, R. and LIVNY, M. (1996) BIRCH: An efficient data clustering method for very large databases. In: *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data.* ACM Press, 103–114.

ZHU, H., ZHONG, J., LI, J. and YU, Y. (2002) An Approach for Semantic Search by Matching RDF Graphs. In: *Proceedings of the 15th International Florida Artificial Intelligence Research Society Conference.* AAAI Press, 450–454.