Semantic Clustering of Scientific Articles with Use of DBpedia Knowledge Base*

Marcin Szczuka, Andrzej Janusz, and Kamil Herba

Faculty of Mathematics, Informatics, and Mechanics, The University of Warsaw
Banacha 2, 02-097 Warsaw, Poland
szczuka@mimuw.edu.pl, janusza@mimuw.edu.pl, k.herba@students.mimuw.edu.pl

Abstract. A case study of semantic clustering of scientific articles related to Rough Sets is presented. The proposed method groups the documents on the basis of their content and with assistance of DBpedia knowledge base. The text corpus is first treated with Natural Language Processing tools in order to produce vector representations of the content and then matched against a collection of concepts retrieved from DBpedia. As a result, a new representation is constructed that better reflects the semantics of the texts. With this new representation, the documents are hierarchically clustered in order to form partition of papers that share semantic relatedness. The steps in textual data preparation, utilization of DBpedia and clustering are explained and illustrated with experimental results. Assessment of clustering quality by human experts and by comparison to traditional approach is presented.

Keywords: Text mining, semantic clustering, DBpedia, document grouping, rough sets.

1 Introduction

This article presents a case study of semantic clustering of scientific articles related to the area of Rough Sets. We have undertaken this study in order to answer the need for developing semantic methods for document processing expressed in the major project (SYNAT) we are involved in.

The SYNAT project (abbreviation of Polish "**SY**stem **NA**uki i **T**echniki") is a large, national R&D program of Polish government aimed at establishment of a unified network platform for storing and serving digital information in widely understood areas of science and technology. Within the frame of this project we are concerned with devising and implementing methods that would allow for indexing, searching, and retrieving the documents from the possibly vast collection using their semantic content (the knowledge their contain).

* The authors are supported by the grant N N516 077837 from the Ministry of Science and Higher Education of the Republic of Poland and by the National Centre for Research and Development (NCBiR) under Grant No. SP/I/1/77065/10 by the strategic scientific research and experimental development program: "Interdisciplinary System for Interactive Scientific and Scientific-Technical Information".

R. Bembenik et al. (Eds.): Intelligent Tools for Building a Scient. Info. Plat., SCI 390, pp. 61–76.
springerlink.com

The idea that we pursue in this study is to perform semantic grouping (clustering) of documents based on their associations with concepts drawn from the DBpedia knowledge base. If done right, such clustering would make a good start for, e.g., a system with extended search features, capable of returning results that are topically close to the search terms, not just those that actually contain the terms from the query (semantic vs. syntactic). This approach is in line with the general trend of finding semantic similarities between documents with assistance of additional knowledge sources (ontologies, thesauri, taxonomies, Wikipedia) in order to obtain more meaningful and useful results, as described in [4,9,14,8].

In our case study we use a text corpus consisting of scientific papers related to Rough Sets. We hope that in this way we will gain some additional insight into our own field of research, verify (positively or negatively) some hypotheses and common beliefs, and possibly find some new. At the same time, since we know the document corpus well, we can use our own expertise to judge the quality of clustering solution.

The article is organized as follows. Section 2 describes the methodology and motivation behind our approach. Then we describe our data set (Section 3) and DBpedia knowledge base (Section 4), providing some details about their characteristics and the way they were collected and prepared for experiments. Section 5 contains description of the actual experiment and explanation of its results. We finish with conclusions and directions for further work in Section 6.

2 The Purpose and Methodology of the Study

The purpose of this experimental study is two-fold.

1. We want to test and verify methodology for document grouping (clustering) based on their semantic content and using a knowledge base. In particular, we want to identify the best configuration for various steps in the process, one that is both computationally feasible and produces meaningful clusters of documents. The goal is to establish a procedure that we will be able to apply semi-automatically to various future text corpora. Since the area of Rough Sets is close to us, we are able to better evaluate the results of experiments on the corpus of texts collected in this field of research. As a consequence, we can identify strengths and weaknesses of the method under scope.
2. We want to learn as much as the methodology permits about our corpus of documents (research papers) related to the area of Rough Sets. Since the individual documents used for this case study (Section 3) are familiar to us, we want to discover the semantic structure of the corpus as a whole and draw some conclusions regarding the features of publications in this scientific area. In particular, we are interested in identifying the most prevalent concepts that characterize this corpus.

Figure 1 shows the general layout of the method that we employ in our case study. The methodology of our was inspired by the Explicit Semantic Analysis (ESA)

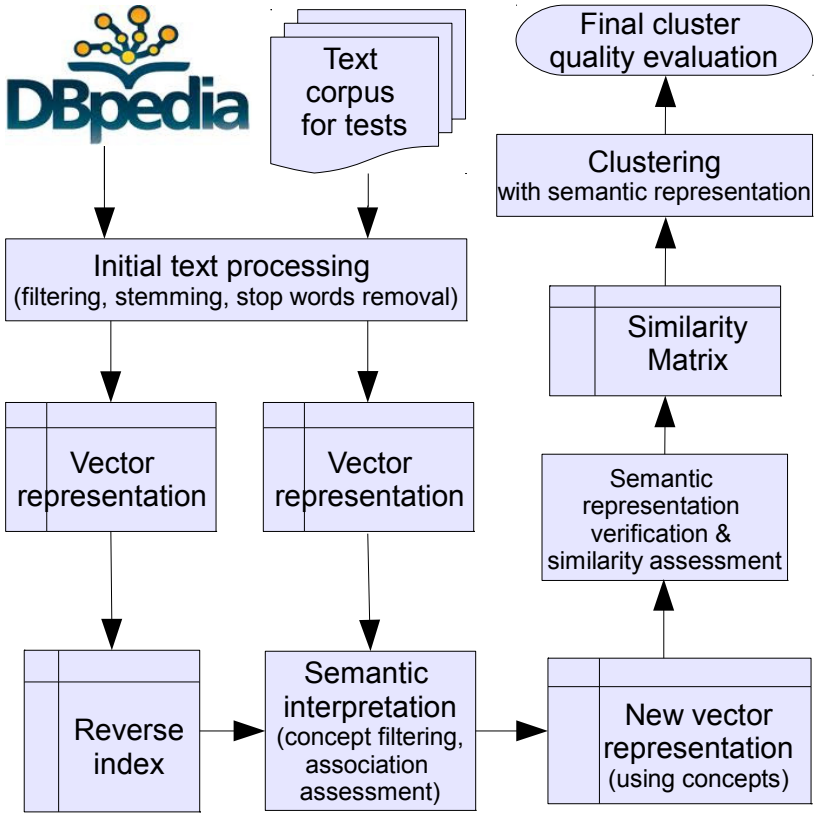

Fig. 1. The general scheme of the experiment

approach presented in IJCAI paper [4]. Since this article is quite involved, the method which it discusses requires more detailed explanation. The data sources, i.e., collection of documents and DBpedia knowledge base, together with the NLP[1] methods for their pre-processing ("Initial text processing" box in Fig. 1) are described in Sections 3 and 4, respectively. In our approach, after initial processing, both collections of texts (the corpus and the DBpedia abstracts) are converted to the *bag-of-words* (word-vector) representation. The bag-of-words representation of a text (document) is a vector based on vocabulary, i.e., the collection of unique words (stems) in the corpus.

Assume, that after initial processing of a text corpus $D = \{T_1, \ldots, T_M\}$ we have collected a vocabulary consisting of n unique terms (stems) $w_1, \ldots, w_n$. Then, any text (document) T_j in the corpus is represented by a vector of the form $\langle v_1, \ldots, v_n \rangle \in {\mathbb{R}_+}^n$, where each coordinate v_i is a value of importance measure for i-th term (word, stem) in vocabulary (w_i), relative to this document. The

[1] Natural Language Processing (NLP) tools as in [3].

most common measure used to calculate v_i is the *tf-idf* (term frequency-inverse document frequency) index (see [3]) defined, relative to document T_j and corpus D, as:

$$v_i = tf_{i,j} \times idf_i = \frac{n_{i,j}}{\sum_{k=1}^{N} n_{k,j}} \times \log\left(\frac{M}{|\{j : n_{i,j} \neq 0\}|}\right) \tag{1}$$

where $n_{i,j}$ is the number of occurrences of the considered word w_i in the document T_j.

To speed up semantic interpretation, we build an inverted index, which maps each word into a list of DBpedia concepts $c_1, \ldots, c_N$ in which it appears. With the inverted index we run a semantic interpreter (Semantic Interpretation in Fig.1). Given a text from the corpus, the semantic interpreter iterates over words it contains, retrieves corresponding entries from the inverted index and merges them into a weighted vector of concepts that represents the given text.

Let $T = \langle w_i \rangle_{i=1}^{n}$ be input text, and let $\langle v_1, \ldots, v_n \rangle$ be its tf-idf vector, where v_i is the weight of word w_i. Let k_{ij} be an inverted index entry for word w_i, where k_{ij} quantifies the strength of association of word w_i with knowledge base concept c_j, $j \in \{1, ..., N\}$. Then, the new vector representation of T is calculated as:

$$\langle \sum_{i:w_i \in T} v_i k_{ij} \rangle_{j=1}^{N}. \tag{2}$$

We will refer to this new vector representation using the notion of *bag-of-concepts*. Note, that in the sum in formula (2) we only consider the words that actually appear in the text T.

The new vector (bag-of-concepts) representation makes it possible to examine relations between concepts and documents, identify and filter key concepts for the given document corpus, and – most importantly – calculate semantic similarity between texts by comparing their bag-of-concepts representations. For technical reasons we choose to store all semantic similarity values for pairs of texts in a structure called *Similarity Matrix*. Entries in this matrix are used to numerically represent the proximity between documents (their bag-of-concept representations), which in our case is calculated using cosine distance.

The fact that we can calculate semantic similarity (distance) between documents gives us the means to perform clustering. Considering that we want to obtain a meaningful grouping of documents we decided to use an agglomerative hierarchical clustering. In order to decide for how many clusters we should divide our data we use a cluster quality measure, in particular the silhouette coefficient. For detailed description of agglomerative clustering, silhouette coefficient, and cluster distance refer to [12].

The quality of resulting clusters is evaluated manually with help of experts in the area of Rough Sets as well as compared with results of a different clustering method. In order to have a reference point we perform a "classical" agglomerative clustering on the bag-of-words representation of documents, without any use of knowledge base. The resulting partition of documents is then compared with

our approach using various measures for cluster consistency as well as manual evaluation of cluster meaningfulness.

3 Data Acquisition and Preparation

For our case study we have used 349 documents in PDF format. These documents are selected from the collection of papers published by the members and associates of the Group of Logic at the University of Warsaw. The subset used for our experiment is significantly smaller than the entire collection, which consists of over 600 publications. While choosing documents for this subset we have used the following criteria:

- We restricted publications to those published in last 15 years (between 1996 and 2011) and written in English.
- We have only chosen "regular" articles, i.e., standard journal, book, and conference papers. They roughly correspond to BibTEX categories: `article`, `inproceedings`, and `incollection`.
- Papers that are very short (extended abstracts) or unusually long (mini-monographs) have been left out.
- Some articles have been removed from the study due to technical difficulties they posed. This was mostly due to problems with incorrect PDF format and usually concerned older (pre-2003) publications.

There were several reasons for using the above criteria in the process of constructing initial data sample for our study. The most important are as follows:

- We wanted the corpus of documents to be relatively *regular*. Since our ultimate goal is the grouping (clustering), we tried to eliminate outliers early on. The idea is to have well-comparable documents and then do the clustering on the basis of their semantic content rather than attributes of their syntactic composition, such as size, level of complication or number of words.
- We have chosen the collection of documents that were created over the years in our group in order to have good understanding of the corpus from the very beginning. Since we know the field and in many cases have direct contact with authors, we can evaluate the outcome with greater ease and confidence. This is a big advantage, especially for an initial, explorative study such as the one that we conduct. It gives us the ability to clearly identify strong and weak points in our methodology.
- We have decided to use this particular number of documents (349) because we wanted to construct a corpus which would be as representative for the area of Rough Sets as it is possible in the given circumstances. The 349 documents in our collection correspond to roughly 10% of all documents of this kind listed in the Rough Set Database system (RSDS [5]). At the moment of writing, the RSDS contains 3641 bibliographical notes that belong to categories that we are interested in.

– Last, but not the least, we selected this particular document corpus because we have both access to their PDF versions and the limited copyrights that allow us to re-use (but not re-distribute) them.

The original PDF documents were first converted to a pure text format with the use of Python script based on PDFMiner library [11]. All documents were divided into blocks of plain text. Based on certain text statistics the script extracted only the text contained in paragraphs, sections and their titles. It has to be underlined that author names article and page headers, footers, tables, equations and other parts of text which were irrelevant and could bias further analysis were discarded. The purpose of this step was to remove various artifacts and clarify text files before attempting to calculate word frequencies and clustering.

This step, although it may appear simple, proved to be troublesome at times. Typical problems at this stage are associated with conversion of hyphenated (broken between lines) words and ligatures (e.g., **fi** in "classification") back to their original (textual) form. These problems were partly resolved with use of an English dictionary which made it possible to guess the right encoding of some characters by determining whether words created after substitution of missing characters were proper English terms. Articles contained also a great amount of mathematical symbols which were encoded in PDF files in various, sometimes very unexpected way. These unusual characters were filtered out as well. Additionally, the bibliography section (references) was removed from each of selected text files. It was done in order to assure that we perform analysis on actual semantic content of the document and to reduce the influence of certain words contained in references, like: publisher, journal name, etc.

The corpus of 349 plain text files was then processed in order to calculate word-vector (bag of words) representations in the next step. First, stop words were removed and then we have performed stemming on the set of words contained in these documents. For stemming of both documents and DBpedia abstracts (as described in Section 4) we use a version of popular Porter's algorithm (cf. [7]). Initially, the corpus contained 35507 unique words (excluding stop words). After stemming we have obtained 26800 unique words (stems) to work with. On average a single document in the collection contains 3524 stems, with minimum of 362 and maximum of 13640.

4 The DBpedia Knowledge Base

According to its creators, the DBpedia (cf. [13,1]) is a community effort to extract structured information from Wikipedia (cf. [15]) and to make this information available on the Web. DBpedia allows to ask queries against Wikipedia data and structure, and to link other data sets to Wikipedia data. In layman terms, DBpedia is a snapshot of the original Wikipedia with mostly preserved structure, but reduced content.

For the purpose of our study we needed to use DBpedia as an enriched dictionary. The version of DBpedia that we use (version 3.5.1 for English Wikipedia) contains 3,257,133 notions (so called *things*). Each DBpedia *thing* represents

a single Wikipedia concept (a single Wikipedia page including disambiguation pages and lists). Due to the distributed and asynchronous nature of the process in which the Wikipedia is created by members of its community, there are some consistency and regularity issues with it. Much of these issues are inherited by DBpedia, which results in some problems related to conflicting or expired names for concepts and categories.

In DBpedia, pages from the original Wikipedia are represented only by their abstracts. For most of the DBpedia concepts there is also additional information derived from Wikipedia, such as classification to Wikipedia categories. There are 3,144,262 abstracts available in DBpedia 3.5.1, but they are very diverse in their length and quality. The length of abstracts vary from empty (0 words) to quite long ones (the longest has 16850 words), with an average of 101 words per abstract. Most of those texts are well formatted and structured but there are exceptions, e.g., some contain only LaTeX-styled source code of tables or figures which were, probably unintentionally, placed in the abstract section of the corresponding Wikipedia article. There are also cases when a whole text of Wikipedia page is placed in the abstract which results in considerably longer DBpedia representations.

Taken altogether, DBpedia 3.5.1 entries constitute a text corpus consisting of 316,631,010 words (after filtration). The number of unique words, before stemming and filtering, is 2,818,483. There are 560,049 *categorical notions* (Wikipedia categories) of which 449,140 are *direct*, i.e., contain some concepts and the rest are *indirect*, i.e., they contain only other categories.

5 Experimental Evaluation of the Approach

Our experiment was conducted in three main steps which we implemented in R System ([10]). First, DBpedia and the selected text collection were preprocessed. Each DBpedia entry and a document was was cleaned, in particular: its encoding was changed to UTF-8, words that contained special characters (!@#$%_&*+–=) or numbers were removed, the most common shortcuts were expanded, and the most common words from a special *stop word list*[2] were removed. The Porter's algorithm [7], implemented in the *Rstem* library, was used for finding stems of words. The stems that occurred less than three times in DBpedia were also eliminated from the texts. Finally, the concepts that were represented by less than 10 unique stems were removed from the knowledge base. As a result, the size of the knowledge base was reduced to around 2.5 million concepts described by approximately 850 thousands of unique stems.

In the second step, the bag-of-concepts representations of texts from the rough set corpus were created using the method described in Section 2. A modified tf-idf index was used to assess the relevance of words (stems) to documents and to concepts. For each text, the frequencies of words, i.e., the tf component in tf-idf formula (1), were smoothed by taking their square root. This modification was

[2] A standard stop word list from *openNLP* library was extended by the 100 most common words from DBpedia abstracts.

Table 1. List of ten most relevant DBpedia concepts for three exemplary documents, with degree of association included

(LTF-C): Architecture, Training Algorithm and Applications of New Neural Classifier		
[1]	9.19	"Neural_Lab"
[2]	9.17	"Echo_state_network"
[3]	8.75	"Auto-encoder"
[4]	8.30	"Interneuron"
[5]	8.09	"Oja's_rule"
[6]	8.08	"Multilayer_perceptron"
[7]	8.06	"Biological_neural_network"
[8]	8.06	"Artificial_neural_network"
[9]	8.00	"Artificial_neuron"
[10]	7.84	"Neuroevolution"
Judgment of satisfiability under incomplete information		
[1]	8.21	"Definable_set"
[2]	8.08	"Schaefer's_dichotomy_theorem"
[3]	7.96	"Formal_semantics_of_programming_languages"
[4]	7.85	"Empty_domain"
[5]	7.78	"Tautology_(logic)"
[6]	7.68	"Equisatisfiability"
[7]	7.54	"Method_of_analytic_tableaux"
[8]	7.38	"Conditional_quantifier"
[9]	7.36	"Model_checking"
[10]	7.32	"Satisfiability_and_validity"
Combination of Metric-Based and Rule-Based Classification		
[1]	8.92	"K-nearest_neighbor_algorithm"
[2]	6.19	"Backmarking"
[3]	6.08	"Wolfe_conditions"
[4]	5.90	"Evolutionary_data_mining"
[5]	5.66	"Event_condition_action"
[6]	5.64	"Transduction_(machine_learning)"
[7]	5.63	"Soft_independent_modelling_of_class_analogies"
[8]	5.63	"Ground_truth"
[9]	5.56	"Proximity_problems"
[10]	5.50	"Dominating_decision_rule"

dictated by a fact, that many of the documents which we use are of technical nature and as such contain many repetitions of specific terms (or single words). The strength of bounds between the concepts and the rough set articles was computed using the equation (2). Following the intuition, that it is meaningless to associate any document with a large number of specific concepts, we have restricted the number of concepts associated with each document. We have decided to use no more than 35 most related concepts for characterization of any given text. This number (35) was selected because it corresponds to around 1% of the average number of stems appearing in the single document in the corpus, which in turn gives more compact and comprehensive representation. Table 1

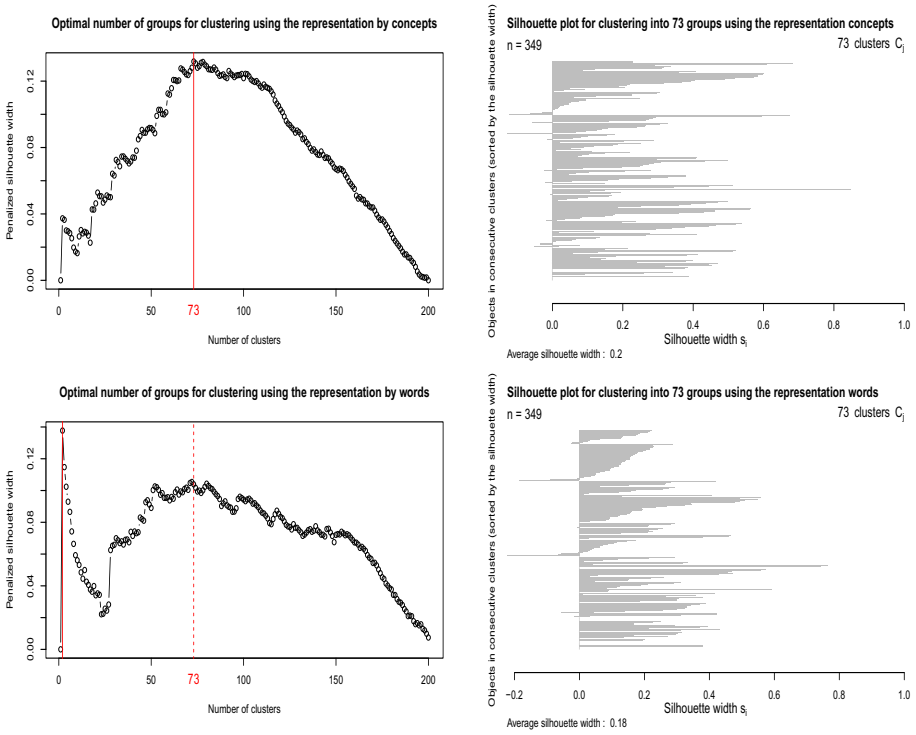

Fig. 2. The plot of silhouette coefficient values across clusters used to establish the optimal number of groups (on the left) and the silhouette coefficient values of individual documents for the selected clustering (on the right). The results obtained from the representation by concepts (on top) is compared to representation by words (at the bottom).

presents associations of top 10 concepts to three exemplary articles from the corpus.

The last step of our experiment involved computation of distances between documents from the rough set corpus which we then use for clustering. Due to an extremely sparse representation of our texts (only 35 non-zero values out of ≈ 2.5 million) the *cosine distance* was employed, which is a commonly used measure in high-dimensional information retrieval tasks ([3],[12]). For the clustering we utilized an agglomerative hierarchical approach with the "average" as a linking function (see [12]). The optimal number of groups (clusters) was decided using the *silhouette width coefficient* which was additionally penalized for selecting larger number of clusters. Figure 2 illustrates the values of average silhouette coefficient w.r.t. the growing number of clusters, along with the silhouette coefficient values of individual documents for the selected clustering. From this picture one can see that the highest cluster separability is achieved when we use 73 of them.

Figure 3 presents the clustering tree corresponding to the partition into 73 groups. Apart from using the silhouette coefficient, quality of the 73 clusters was also assessed with the aid of human experts. Mutual relatedness of documents from several groups has been evaluated. In order to gain another point of reference we have also performed clustering using the original bag-of-words[3] representation of texts in the corpus (Figure 4).

The results of this comparison are encouraging. The consecutive partitions obtained using the bag-of-concepts representation yielded much more stable silhouette coefficients than those for the original word-vector (bag-of-words) one. The optimal number of groups for the latter is two, which corresponds to meaningless grouping of documents. This number is also not very stable as it may vary wildly between 2 and 157 if we alter the penalty for producing excessive clusters. Moreover, if with the bag-of-words representation we make the clustering algorithm produce 73 groups (optimal number for the bag-of-concepts), then brief analysis of this partition reveals a significant imbalance in the size of clusters (Figure 4). The largest cluster obtained in this way contained 60 papers and there were 29 singletons (clusters that contained only a single document). To make things worse, many of the larger groups constructed in this manner contain semantically unrelated documents and are very difficult to label. In contrast, size of the largest group resulting from utilization of the bag-of-concepts representation was 27 and there were only 19 singletons.

The observation, that employment of domain knowledge improves the quality of clustering was confirmed by domain experts. For instance, Table 2 shows members of three exemplary clusters taken from distinct branches in the clustering tree (Figure 3). Labels that briefly summarize contents of those groups were given by experts. Among 13 papers that belong to the cluster 21, 12 were recognized by experts as related to the notion of neural computing and artificial neural networks. The same subset of papers, partitioned based on the bag-of-words representation, was broken between three different clusters of which only one was semantically homogeneous and meaningful.

It it also worth mentioning that, even though information about authors and bibliography was removed from the corpus during the preprocessing phase, 12 out of 14 articles grouped in the cluster 39 were written by a single author (Anna Gomolińska). In those papers, the author consider a problem of partial satisfiability and validity of formulas (such as decision rules) under incomplete or uncertain information.

It seems that with the bag-of-concept representation, the clustering algorithm was able to conceptually discern them from other research topics of this particular author. The articles of the same author that belong to other research direction, the theory of approximation spaces, are located in another cluster. These articles (six of them) are placed in the cluster 36 (Figure 3). In comparison, when the representation by bag-of-words is used, almost all publications of Anna Gomolińska from our corpus (21 out of 22) are placed in a single group. That last fact, in our opinion, is probably due to usage of a characteristic and

[3] For consistency, we used the smoothed tf-idf vector representation.

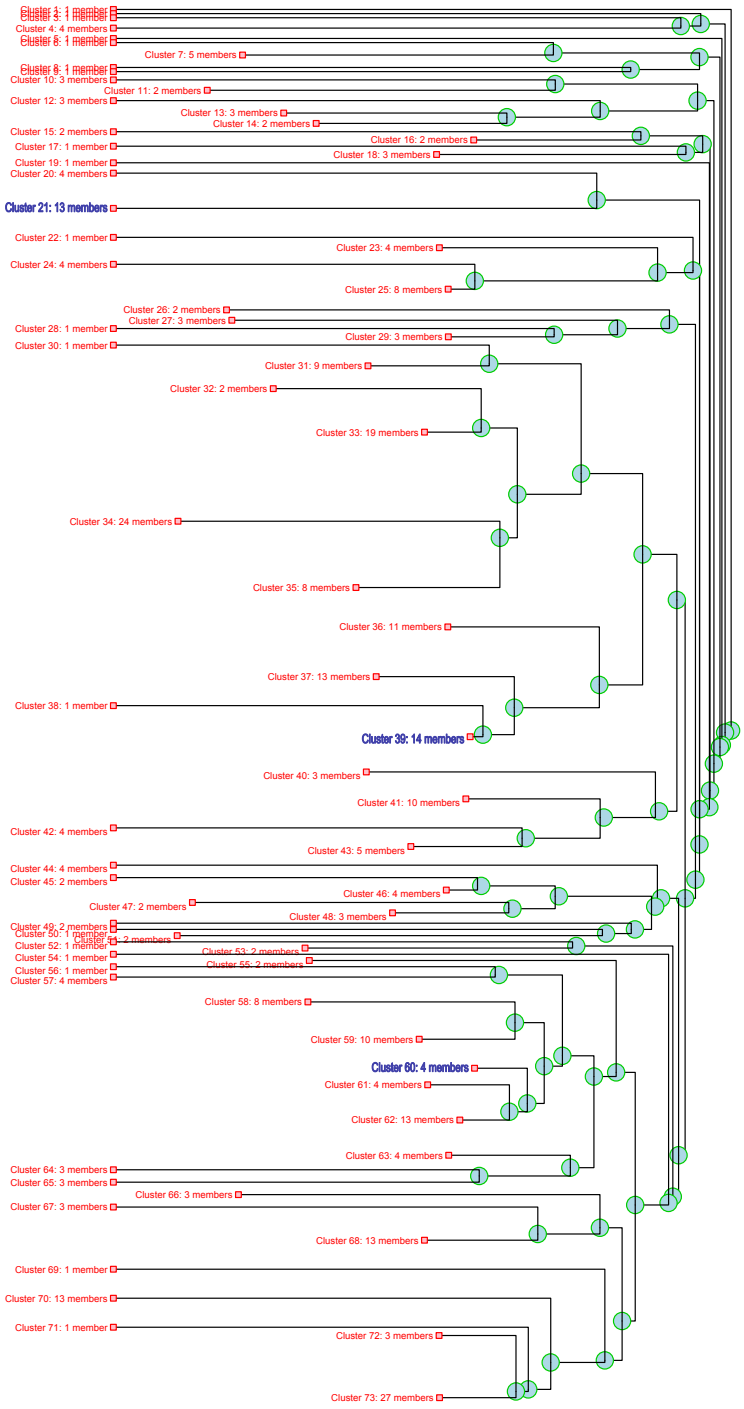

Fig. 3. Truncated tree of clusters (dendrogram) with 73 leafs, based on the bag-of-concepts representation. Clusters 21,39, and 60 that are detailed in Table 2 are marked with different font.

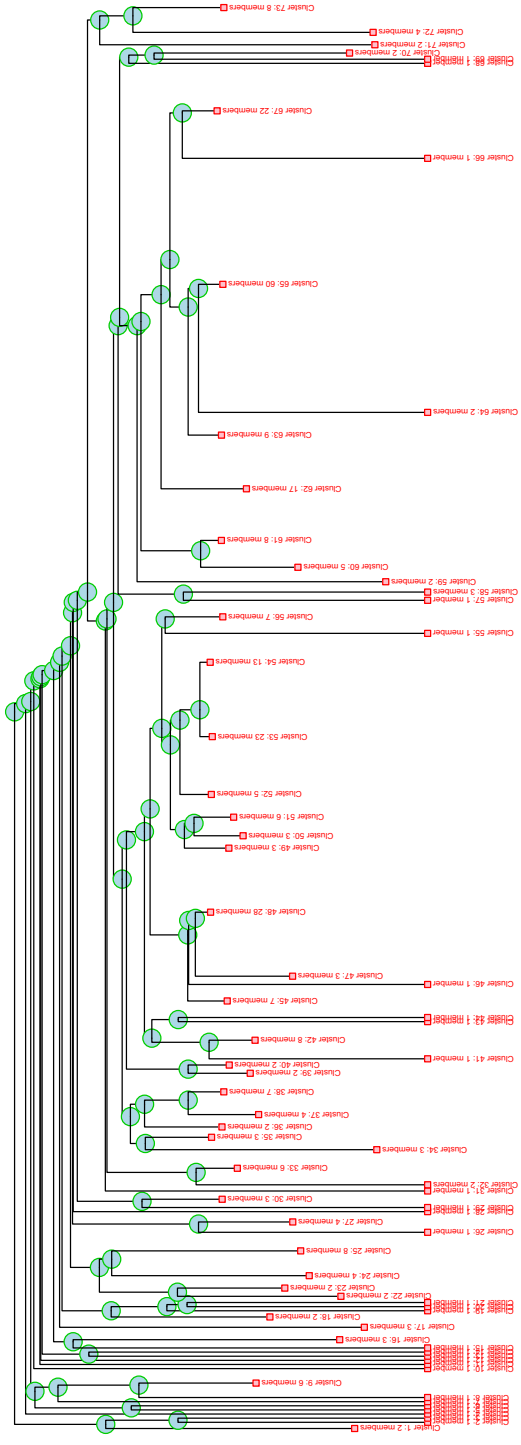

Fig. 4. A clustering tree based on the bag-of-words representation, truncated to 73 leafs

Table 2. Members of exemplary partitions, resulting from clustering with the bag-of-concepts representation. The IDs of branches from the clustering tree are given along with the labels assigned by domain experts and titles of corresponding documents.

Cluster 21: Neurocomputing and Artificial Neural Networks	
[1]	(LTF-C): Architecture, Training Algorithm and Applications of New Neural Classifier
[2]	Rough Neurons: Petri Net Models and Applications
[3]	Rough-Neural Computing: An Introduction
[4]	Toward Rough Neural Computing Based on Rough Membership Functions: Theory and Application
[5]	Rough Neurocomputing: A Survey of Basic Models of Neurocomputation
[6]	Design of rough neurons: Rough set foundation and Petri net model
[7]	Constructing Extensions of Bayesian Classifiers with use of Normalizing Neural Networks
[8]	Refining decision classes with neural networks
[9]	Harnessing Classifier Networks - Toward Hierarchical Concept Construction
[10]	Feedforward concept networks
[11]	Neural network design: Rough set approach to real-valued data
[12]	Hyperplane-based neural networks for real-valued decision tables
[13]	Rough Sets and Artificial Neural Networks

Cluster 39: Logical Satisfiability and Validity of Formulas	
[1]	Judgment of satisfiability under incomplete information
[2]	A graded applicability of rules
[3]	Toward rough applicability of rules
[4]	Satisfiability and meaning in approximation spaces
[5]	Satisfiability Judgment Under Incomplete Information
[6]	Reasoning Based on Information Changes in Information Maps
[7]	Rough validity, confidence, and coverage of rules in approximation spaces
[8]	Satisfiability and meaning of formulas and sets of formulas in approximation spaces
[9]	On rough judgment making by socio-cognitive agents
[10]	Rauszer's R-logic for multiagent systems
[11]	Rough rule-following by social agents
[12]	Satisfiability of formulas from the standpoint of object classification
[13]	Construction of rough information granules
[14]	Patterns in Information Maps

Cluster 60: Instance-based Learning	
[1]	Combination of Metric-Based and Rule-Based Classification
[2]	Rough Set Approach to CBR
[3]	Local Attribute Value Grouping for Lazy Rule Induction
[4]	Granulation in Analogy-based Classification

Table 3. Tags (concept labels) for three examples of clusters

Cluster 21: Neurocomputing and Artificial Neural Networks	
[1]	"ADALINE"
[2]	"Artificial_neural_network"
[3]	"Artificial_neuron"
[4]	"Auto-encoder"
[5]	"Delta_rule"
[6]	"Multilayer_perceptron"
[7]	"Universal_approximation_theorem"
[8]	"Echo_state_network"
[9]	"Neural_Lab"
Cluster 39: Logical Satisfiability and Validity of Formulas	
[1]	"Empty_domain"
[2]	"Formal_theorem"
[3]	"Limit-preserving_function_(order_theory)"
[4]	"Satisfiability_and_validity"
[5]	"Schaefer's_dichotomy_theorem"
[6]	"Tautology_(logic)"
[7]	"Well-definition"
Cluster 60: Instance-based Learning	
[1]	"Attribute_(computing)"
[2]	"Attribute_(network_management)"
[3]	"Integrity_constraints"
[4]	"K-nearest_neighbor_algorithm"
[5]	"Online_machine_learning"
[6]	"Relation_(database)"
[7]	"Structured_SVM"

highly specialized vocabulary that inadvertently biases the bag-of-words representation.

We have also investigated whether the bag-of-concepts representation may be used for the purpose of automated tagging (labeling) of clusters. For this purpose we associated each group (cluster) of articles with DBpedia concepts that appear in representations of at least 80% of its members. Table 3 presents these associations for the three exemplary clusters. From this example one can see that the selected concepts (cluster tags) are well in line with cluster labels assigned by the experts. Unfortunately, they seem to be too specific to express the semantic relatedness of the documents in the cluster by themselves. To overcome this issue, in the future we plan to employ knowledge about DBpedia categories and the structure of concepts to construct more general tags.

6 Conclusions and Further Work

The conclusions drawn from this case study, just like the motivations presented in Section 2, are of two kinds.

Firstly, we can draw conclusions regarding the structure and characteristics of the corpus of 349 rough set related documents that were used as the basis for the study. The experimental results confirm, that our text corpus is fairly uniform and focused. It is quite clear that our articles share a lot of common concepts at the same time being separable from other research areas. Within the area of rough sets, the papers can be arranged into groups (clusters) in a really meaningful manner.

The second conclusion is that the proposed approach to clustering, based on ESA approach, has a significant potential and shall be seriously considered as an element in the future studies. During the experiments it was possible to establish some ground knowledge about features of the method used. That gives us some confidence about the viability of this approach and its potential to become an element of the prototype software solution that we are eyeing in frame of our main (SYNAT) project.

As usual with this kind of experimental study, there is a plethora of things we can do next. In shorter perspective, next steps should include testing more clustering methods, playing with parameters, coefficients and so on, to obtain more optimal and versatile solution. Also, it would be very interesting to investigate whether including more knowledge from DBpedia, for instance structural information about categories, helps to improve results or not. Another natural next step is to extend the corpus of (currently 349) documents and check if our findings remain valid for larger data set. This step, however, requires the access to larger sources of PDF documents. It may be possible to have more documents from other research areas in the future and then perform the comparative study. We hope that with progress of the overall SYNAT project we will obtain more material (documents) from other co-operating partners.

Another possible direction of a continuation of this study may regard different methods for assessment of similarity between pairs of scientific documents. Currently, only the cosine distance is being used. That fact restrains our ability to detect semantically similar texts since it enforces potentially undesirable properties of a metric on the similarity measure. We believe, that in order to capture more semantic resemblance of articles, the similarity measure should be more dependent on a domain from which the documents come. One way to achieve that is through utilization of some similarity learning methods, such as the Rule-based Similarity model described in [6].

In a long run, the follow-up of this study should produce a software module that could serve as a part of a search/recommender system supporting development of information platform in the SYNAT project. For that to happen the tools that we have used to hand-craft the experiments in this study will have to be implemented as (semi-)automatic software modules that will be integrated with the prototype of main system. This will require not only, re-writing of some code but, quite possibly, re-designing some algorithms (e.g., calculation of reverse index) in order to make them computationally efficient even for processing very large number of documents at once. We are currently considering using

the SuperMatrix library and tools (cf. [2]) as a candidate for foundation of such implementation.

References

1. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: DBpedia – a crystallization point for the web of data. Journal of Web Semantics: Science, Services and Agents on the World Wide Web 7, 154–165 (2009)
2. Broda, B., Jaworski, D., Piasecki, M.: Parallel, massive processing in SuperMatrix - a general tool for distributional semantic analysis of corpus. In: Proceedings of International Multiconference on Computer Science and Information Technology - IMCSIT 2010, pp. 373–379 (2010)
3. Feldman, R., Sanger, J. (eds.): The Text Mining Handbook. Cambridge University Press (2007)
4. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence, pp. 6–12 (2007)
5. Grochowalski, P., Suraj, Z.: RSDS - the Rough Set Database System - a bibliographic database on wide aspects of rough sets (2009), `http://rsds.univ.rzeszow.pl/`
6. Janusz, A.: Utilization of dynamic reducts to improve performance of the rule-based similarity model for highly-dimensional data. In: Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and International Conference on Intelligent Agent Technology - Workshops, pp. 432–435. IEEE (2010)
7. Jones, K.S., Willet, P.: Readings in Information Retrieval. Morgan Kaufmann, San Francisco (1997)
8. Maguitman, A.G., Menczer, F., Roinestad, H., Vespignani, A.: Algorithmic detection of semantic similarity. In: Ellis, A., Hagino, T. (eds.) WWW, pp. 107–116. ACM (2005)
9. Oleshchuk, V.A., Pedersen, A.: Ontology based semantic similarity comparison of documents. In: DEXA Workshops, pp. 735–738. IEEE Computer Society (2003)
10. R Development Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2009), `http://www.R-project.org`
11. Shinyama, Y.: PDFMiner: Python PDF parser and analyzer (2010), `http://www.unixuser.org/~euske/python/pdfminer/`
12. Tan, P.N., Steinbach, M., Kumar, V.: Introduction to Data Mining. Addison Wesley, Boston (2006), `http://www-users.cs.umn.edu/~kumar/dmbook/index.php`
13. The DBPedia Community: The DBPedia knowledge base (2011), `http://DBpedia.org/`
14. Varelas, G., Voutsakis, E., Raftopoulou, P., Petrakis, E.G.M., Milios, E.E.: Semantic similarity methods in wordnet and their application to information retrieval on the web. In: Bonifati, A., Lee, D. (eds.) WIDM, pp. 10–16. ACM (2005)
15. Wikipedia Community: Wikipedia - the free Encyclopedia (2011), `http://en.wikipedia.org/`