Automatic Concept Retrieval with Rubrico

Emanuel Reiterer[1], Heinz Dreher[2], Christian Gütl[1,2]

[1]Graz Technical University of Technology, Graz
[2]Curtin University, Perth

1 Introduction

With the vast repositories of knowledge available via computers nowadays there is an expectation that the relevant portion of this knowledge is accessed and used when creating contributions to the world's store of knowledge. Search engines have been popular and useful for some decades but there remains the key problem of narrowing down the search to find precisely what there in a given document of document set. While the Information Retrieval community has developed solutions to increase relevance and precision, the work is based largely on key-word searching (Zhu and Dreher 2009). Clearly, manual search and find options are useful only with very limited data sets and we need computers to help us achieve that final set for manual inspection and processing. Recent developments, especially in ontologies (W3C 2004) have made possible the treatment of text-based datasets in 'intelligent' ways. Our work falls into this category.

Here we report of the outcome of our effort to search for, or find, as we prefer to think of it, concepts rather than key-words. The software embodying the conceptual searching ideas being reported in this article has been named Rubrico and is intended to use for lecturers who can use this program for marking purposes or for getting a better impression of the concepts that are lectured in a specific topic. But it also can be used for any other task where ontologies should be retrieved out of an electronic text or ontologies have to be improved.

This article starts with background information regarding concept analysis, ontologies and available software toolkits. Rubrico our prototype software package to find concepts reposing within a document set, and assist users in structuring them into a rubric or (possibly) hierarchical framework, is explained. There are many applications for concept search and find and one strong motivator in our research group named SATM4BE[1] was our work in automated essay grading, and plagiarism detection. For this task concepts are retrieved from a reference or model and compared with the retrieved concepts in target documents. This permits forming

[1] SATM4BE – Semantic Analysis & Text Mining for Business & Education
www.eaglesemantics.com

an idea of the taught subject via its concepts and the possibility to match those concepts against a specific essay for checking purposes.

We describe a manual process of retrieving concepts and then present our automatic approach. An explanation of the algorithms and toolkits that are used nowadays precedes the real implementation of the software Rubrico, which retrieves an ontology out of a specific corpus, stores and visualizes that ontology, and provides a graphical interface for the human interaction functionality. After that the deviation of the manually and automatically retrieved concepts will be mentioned. Finally we will give a conclusion of our work and state further developments in this field.

2 Background

2.1 Concept Definition

Concepts are represented by a set of words related by a meaning. Another definition is that a concept can be described through a meaning triangle. (Sowa 2000, pp. 55-81) This triangle depicts that a concept, thought or reference describes or symbolizes an object that can in turn denote a specific item or referent. For example the concept of an object describes a cat (the object) and the symbol could be the representation of the name (for example Minka).

In Frege's book "On sense and reference" (Frege 1892, pp. 25-50) he defined a concept or sign through sense and reference. Another definition is used in Formal Concept Analysis (FCA) invented by Rudolf Wille (Priss 2006, pp. 521-543): He uses the terms extension and intension that are representing a "Gallois Connection". This is a connection between two types of items, which are related to each other. An extension is a set of formal objects and an intension a set of formal attributes. A related combination of a set of formal objects and formal attributes is called a formal concept. So if a set of formal objects is reduced the set of formal attributes is increasing.

In Cimiano's article about "Ontology Learning and Population from Text" (Bitelaar and Cimiano et al. 2005, pp. 1-10) a concept is defined as a triple: the intension of the concept, the extension, and the lexical realization in a corpus.

2.2 Manual Concept Retrieval

In the current era of readily accessible and vast repositories of knowledge there is an urgent need for help to find and summarize possible relevant publications and the concepts contained therein. Searching for keywords produces an approximate set of relevant publications with those keywords highlighted, but a further step to automatically identify relevant concepts would be a large advantage. Concepts are

represented by a set of words related by meaning. For example, the concept of "the color white" can be represented by:

- white, whiter, whiteness, etc.
- the brightest color
- a color without hue at one extreme end of the scale of grays, opposite to black. A white surface reflects light of all hues completely and diffusely. Most so-called whites are very light grays: fresh snow, for example, reflects about 80 percent of the incident light, but to be strictly white, snow would have to reflect 100 percent of the incident light. It is the ultimate limit of a series of shades of any color

Clearly, finding those parts of a publication that refer to the concept "the color white" are not found by only searching for the word "white" and its derivatives. We need a means by which we can search for concepts.

In the work reported here, we have used machine learning algorithms as a first step in automatically retrieving concepts from a large set of documents. The resultant ontology may then be manually adjusted.

2.3 Automated Concept Retrieval

We chose to use ontologies as representation for the retrieved concepts because within ontologies classes are retrieved, which are similar to concepts. Furthermore it is possible to store relations. For this task programs were developed such as TextToOnto (University of Karlsruhe 2005), KASO (Wang and Völker et al. 2006, S. 1-8), Text2Onto (Cimiano and Voelker 2005), and Mo'K (Wang et al. 2006, pp. 1-8).

Those programs often use WordNet (Princeton University 2009), which is a large lexical database. For natural language processing GATE (General Architecture for Text Engineering) (University of Sheffield 2009), and Apache Lucene (Apache Software Foundation 2009) are used. Protégé (Standford University 2009) and NeOn(NeOn 2009) toolkit are well known development kits for creating and working with ontologies.

Next an automated approach for retrieving concepts from electronic documents will be described. Therefore we developed a prototype, which is called Rubrico and able to retrieve concepts from electronic data in the form of PDF, HTML or text files. This prototype uses several machine learning algorithms that are implemented in the Text2Onto (Cimiano and Voelker 2005) toolkit. Furthermore, the functionality to rename concepts, change relevance values of concepts, and the possibility of adding and removing concepts are implemented so the ontology can be adjusted or changed through human interaction according to human intelligence rather than artificial intelligence.

3 Rubrico

3.1 Design Considerations

For us it was necessary to further the development of our tool for automated essay grading. Therefore the idea arose to work with concepts. With concepts it is possible to extract the meaning of a text. So, if the lecturer is in the possession of a good conceptual design of a topic it should be possible to match those concepts against the retrieved ones from a student essay.

Automatic concept extraction, using machine learning or other statistical techniques, from a small word-count text like an essay is impossible without a concept rubric or model which can be referenced and checked by a human user. A possible solution to this problem is to submit enough information to an analysis program for the purpose of matching the text of the essays against an ontology. So, if it is possible to deliver an ontology out of a topic with a first cut of well structured concepts and with special relations, and secondly with enough instances (instances can be seen as keywords) of each concept it would be possible to process and grade essays automatically.

Because this program is only viable when the rest of the grading will be done automatically as well, the functionality should be extended to make it feasible for processing the whole essay automatically without human intervention - spell, style, and plagiarism checking. But a plagiarism checker can be realized with automatically retrieved concepts as well, which could be interesting for further research.

Another idea emerged in using 'non-concepts' as well as concepts. Non-concepts are necessary because if the subject is for example "airplanes", "trains" must not be part of the topic and should be mentioned as a non-concept in the marking process. Another advantage is to have an exact opposite to a correct concept and so it is easier to calculate the validity that the correct concepts are available in the text. Thus, the concepts form what could be called a 'white-list' and the non-concepts form a 'black-list'.

We proceeded to implement a prototype, which firstly extracts concepts out of electronic data related to a specific subject, stores those concepts in an ontology together with their relations, and retrieves a concept rubric for automatic essay grading, and other purposes. This rubric, which is represented with the help of an ontology, contains all necessary concepts with their individual components for marking purposes, and the marking levels the students can achieve when mentioning the specific concept in their essays.

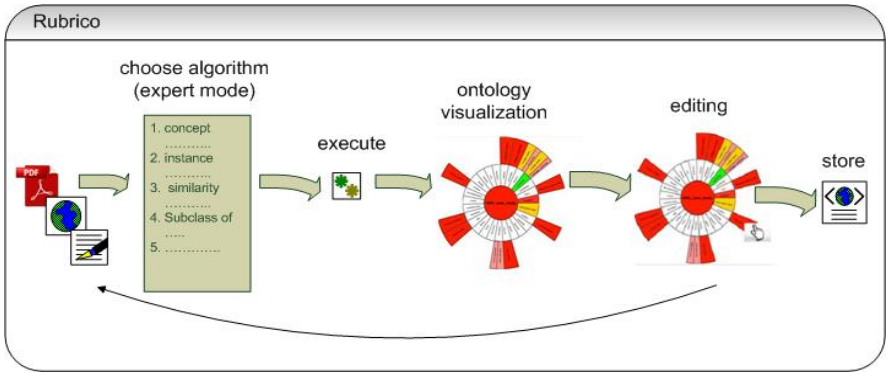

Figure 1: Rubrico process

3.2 The manual algorithms mapped to the automatic ones

One of the main intentions in retrieving concepts out of a document is to get knowledge of the document or a summary of it. A good generic summary (Villalon and Calvo 2009, pp. 221-225) should contain the main topics and keep redundancy at a minimum. Thus the topics or concepts should not exceed a specific number because if too many concepts are retrieved the text may as well be read directly by the human user.

We researched how nature handles the topics and tried adopting a similar approach to the automated task of finding and retrieving ontologies automatically. As mentioned above, concepts are represented by a set of words related by meaning but can also be described through a meaning triangle (Bitelaar and Cimiano et al. 2005, pp. 1-10). Thus, the derived steps are to: read (or parse) the text; retrieve keywords; group these keywords; and combine those groups to form a concept. For retrieving concepts automatically the same steps have to be applied. This will be mentioned in the next section.

3.3 The application of the manual steps into a software

Accessing text automatically is relatively easy if it is already in a computer readable form. The more complicated part is to retrieve keywords because of the necessity to 'understand' the meaning of the text. Approaches to handle this problem are divided into several steps. At first, text operations like elimination of stop-words, extraction of word stems, and enumeration of the occurrences of those retrieved word-stems have to be executed. Afterwards the result has to be brought into a meaningful representation. Several algorithms can be applied. These algorithms are described in the next subsection. The last step is to visualize those retrieved concepts in a human readable form and to create a computer processable representation.

The human readable representation is necessary to give the user the chance to adjust the retrieved concepts to better match the desired concept rubric or model, and the computer processable representation is useful to search in those concept hierarchies.

Manual concept extraction can be roughly divided into the following steps:

1. read the text
2. retrieve keywords
3. group keywords

In the paper called "Ontology learning from text" (Bitelaar and Cimiano et al. 2005, pp. 1-10) an ontology learning layer cake is proposed. This cake is divided into six layers. For concept retrieval the first four are interesting but the further two layers can be interesting as well because they are handling relations and axioms, which can be used to classify the retrieved concepts.

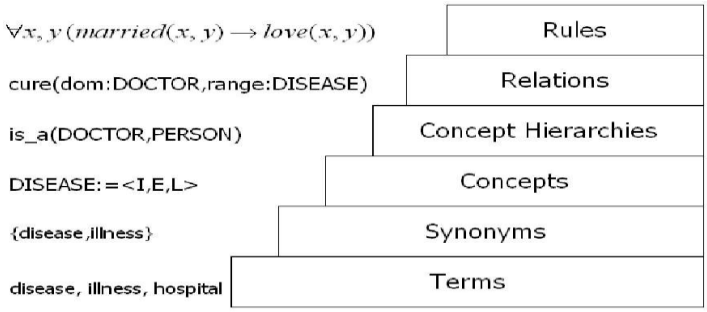

Figure 2: Ontology Layer Cake (Bitelaar and Cimiano et al. 2005, pp. 1-10)

The first four layers are Terms, Synonyms, Concept, and Concept Hierarchies. Next the ontology layer cake will be applied using the manual algorithms.

Manual step of reading the text and retrieving the Keywords

Steps 1 and 2 can be mapped to the layers Terms, Synonyms, and Concepts.

A term, which is extracted in the Term layer, is a single or multi-word compound containing a meaning and can be already seen as a concept. (Bitelaar and Cimiano et al. 2005, pp. 1-10) The extraction of most relevant ones is crucial therefore. The Synonym layer retrieves synonyms that are used to get an objective meaning of concepts. They can be used to match concepts from different texts because not everyone is using the same words for the same meaning. The Concept layer instead depicts concepts that can be defined as a triplet of the intension, their

extension, and the lexical realization in the corpus. Intension is a natural language description of the intuitive meaning of a concept. It can be seen similar to glosses in WordNet (Princeton University 2009) or a collection of attributes, which are handled in FCA (Formal Concept Analysis) (Priss 2006, pp. 521-543). (Bitelaar and Cimiano et al. 2005, pp. 1-10)

The Manual Step of Grouping keywords

For implementing the aggregation to a main concept and for retrieving subclasses of concepts, additional algorithms to retrieve concept hierarchies, relations and axioms are useful. These algorithms are processed in the layers 4 (Concept Hierarchies), 5 (Relations), and 6 (Rules).

With the Concept Hierarchies layer it is possible to eliminate similar concepts and to classify them. The Relations layer helps find concepts standing in some non-taxonomic ontological relations, appropriate labels and relation identifiers, to determine the right level of abstraction, and to learn a hierarchical order between relations. (Bitelaar and Cimiano et al. 2005, pp. 1-10) The axioms or Rules layer is for handling disjointness or equivalence axioms for concepts and transitivity, and symmetry axioms for relations.

Choice for the automatic concept retrieval steps of the prototype

The Text2Onto toolkit is used for the Rubrico prototype and therefore the concept retrieval task is divided into the following steps, which are applied when a new ontology is calculated. The steps are: concept retrieval; instance extraction; similarity calculation; subclass of extraction; instance of extraction; and disjointness.

In the first version of our prototype and for the standard perspective, which will be described later, concept retrieval, instance extraction, and subclass-of extraction algorithms are used.

3.4 The Ontology and its Representation

After the execution of the algorithms an ontology containing the concepts and their relation is retrieved. But to understand this ontology and to be able to work with it a good representation and visualization is beneficial.

Ontology Representation

There are several representations available nowadays. The different representations are divided in frame-based ontologies like F-Logic (Kifer and Lausen et al. 1990, pp. 741-843), description-based ontologies such as OWL (Web Ontology Language) (W3C 2004), and first order logic based representations. Because the repre-

sentation should be stored in a computer readable way, and OWL can be handled by a lot of programs, we chose that syntax.

Ontology Visualization

From all the different existing ontology visualization methods such as indented lists, node link and tree, zoomable or space filling, focus and context or distortion, or 3D information and landscapes (Katifori and Halatsis et al. 2007) we chose an *indented list* and a *radial space filling tree* because they are clearly laid out and human readable. They also allow multiple manipulation opportunity for better usage when the ontology is edited by human support.

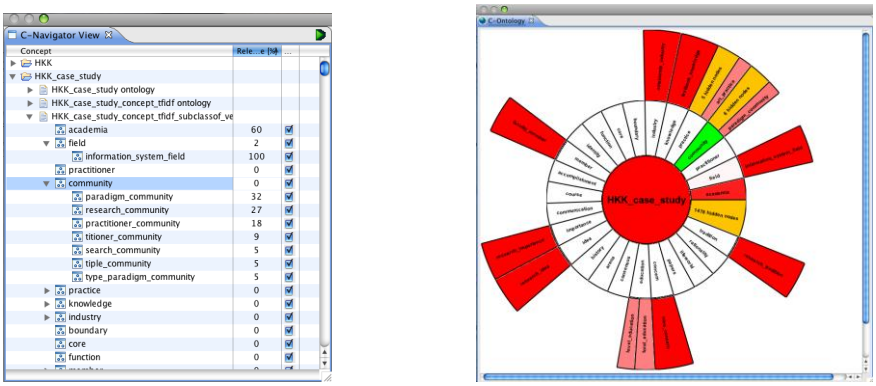

Figure 3: LHS: indented list; RHS: radial space filling tree

3.5 A short introduction to the graphical user interface of the prototype

The prototype Rubrico itself was developed as a plug-in for the NeOn-Toolkit (NeOn 2009), which is an Eclipse RCP (Rich Client Platform) application (Eclipse Foundation 2009).

Next the views, the perspectives, and how the information is stored will be explained. Views are windows inside an Eclipse application. A perspective is described by a set of views opened at specific positions in the program.

In Rubrico, an expert and a standard perspective are implemented, which means that a different set of application windows or views will be shown. In the expert perspective, for example, the workflow view will be shown, which is useful if the user wants to decide which algorithms should be executed. For a standard user instead the best workflow should be chosen automatically because in the first place he/she is normally only interested in the result.

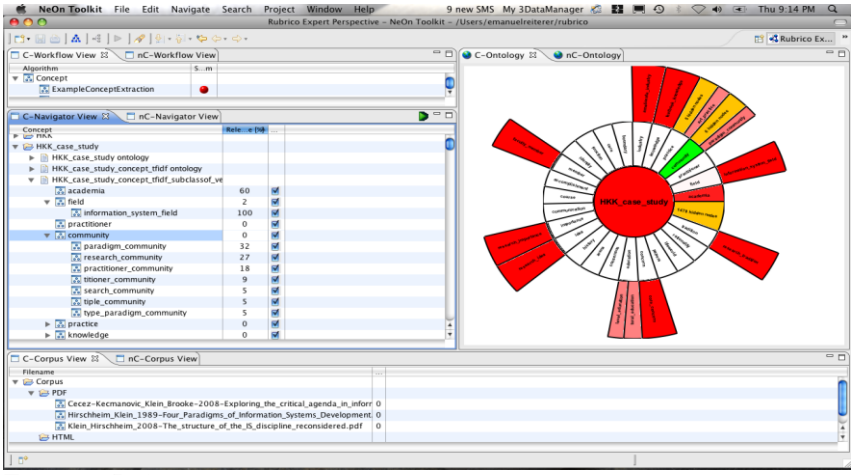

Figure 4: Rubrico

The views, which are handled by those perspectives, are the corpus view, the indented list view, the radial space filling tree view, and the workflow view.

The corpus view is used to handle the corpus data. It is possible to add files to, and delete from the corpus.

The indented list and the radial space filling view instead are for representing the ontology. Additionally, the former can be used for project navigation as well. It shows an indented list of all projects and automatically retrieved ontologies as well as cloned ontologies, which are used for adapting with human support. The radial space filling tree view or visualization was made for reflecting the concepts. We chose to work with the ZEST toolkit (Eclipse Foundation ZEST 2009) that is already implemented for the Eclipse framework. Therefore we needed to implement the specific nodes of such a tree and many more – and this involved considerable work in extending the framework.

The last view implemented is the workflow view, which can be used to select algorithms for use with a specific project.

Once the ontologies are retrieved and the user editable cloned ontologies are created they are stored in an OWL/XML format. Any other configurations such as the workflow and the corpus information are saved in XML files. The originals of the added documents (the source data) are stored in the project directory as well.

4 Case study

Having spent very considerable resources creating Rubrico, and fixing and extending component toolkits we wanted to apply it to some real data as a case study. We chose a selection of articles authored or co-authored by the late Heinz K. Klein (Hirschheim and Lyytinen 2009, pp. 59-62), influential interpretivistic Information

Systems researcher. Those papers were analyzed automatically with the Rubrico prototype using a selection of machine learning algorithms to generate ontologies that were subsequently adjusted and fine-tuned by humans to improve the retrieved concepts representation and structure. The case study itself is the subject of a special study, hopefully to be published as a contribution to the life's work of HKK in a future publication.

5 Conclusion and further work

Here we take the opportunity to reflect on aspects of automatic retrieval of concepts, and some considerations for further development of that topic.

5.1 Artificial Concepts

An interesting aspect we confronted is the problem of "artificial concepts". Because the retrieved terms and concepts are handled as word stems it may be that the retrieved concepts are just parts of a word and because of that fact sometimes sub-concepts are retrieved, which are matched to this artificial concept even if the terms have no standing to reason similarity between each other. Sometimes however, such concepts have something in common in a second attempt at thinking about them and such considerations may open a new perspective to the research topic the user is working on.

5.2 Concept and non-concept

Worthy of mention is the 'concept' and 'non-concept' strategy. This is especially useful in the educational domain because if it is required to automatically mark an essay it could be useful not to search only for necessary concepts of a specific subject, but also for concepts that are the complete opposite and therefore should not appear or be used.

5.3 Retrieving of deeper concept hierarchies

We noticed that it is difficult to retrieve deeper concept hierarchies. This will also be a point for further studies because it could be useful for plagiarism checks and in the educational sector as well where it is feasible to mark more accurately through the use of a finer subdivision of the subject or the concepts.

5.4 How to deal with multiple relations

The nature of an ontology is that every concept can have multiple parents but this can be a problem in displaying an indented list for example and it is a problem for

the readability as well. Therefore we took the parent concept with the highest relevance value as the main parent concept and eliminated the others. But as it distorts the retrieved ontology, other visualizations with good readability, could be used in addition to the mentioned ones.

5.5 Further Work on this topic and research field

At the present time we are finishing a study to determine the best workflow for scientific essays. It is also a goal to increase the usability of this tool to give the user an easy way to adapt the ontologies. Further the plug-in will be extended to be able to compare retrieved concepts to essays.

The execution time should be decreased and the number of concepts should be reduced to a minimum because only the most important and general ones are interesting for the user at a first glance. Another topic is to work with meta-ontologies for retrieving better concepts. Additionally there would be some algorithms that are worth adding either to the prototype or to the Text2Onto toolkit.

References

Apache Software Foundation (2009) Apache Lucene. http://lucene.apache.org/. Last visit 2009-09-17.

Bitelaar P, Cimiano P, Magnini B (2005) Ontology Learning from Text: An Overview. IOS Press, The Netherlands.

Cimiano P, Voelker J (2005) Text2Onto - A Framework for Ontology Learning and Data-driven Change Discovery. Institute AIFB, University of Karlsruhe.

Eclipse Foundation (2009) Eclipse. http://www.eclipse.org/. Last visit 2009-09-13.

Eclipse Foundation ZEST (2009) ZEST. http://www.eclipse.org/gef/zest/. Last visit 2009-09-17.

Frege G (1892) On Sense and Reference. Zeitschrift für Philosophie und philosophische Kritik: 25-50.

Hirschheim R, Lyytinen K (2009) To the memory of our friend and colleague Heinz K. Klein. Information and Organization Elsevier Vol 19: 59-62.

Katifori A, Halatsis C, Lepouras G, Vassilakis C, Giannopoulou E (2007) Ontology Visualization Methods—A Survey. ACM Computing Surveys, Vol. 39, No. 4, Article 10.

Kifer M, Lausen G, Wu J (1990) Logical Foundations of Object-Oriented and Frame-Based Languages. Journal of the ACM 42: 741-843.

NeOn (2009) NeOn toolkit. http://neon-toolkit.org/wiki/Main_Page. Last visit 2009-09-13.

Princeton University (2009) WordNet. http://wordnet.princeton.edu/. Last visit 2009-09-17.

Priss U (2006) Formal Concept Analysis in Information Science. Cronin, Blaise (ed.), Annual Review of Information Science and Technology 40: 521 - 543.

Sowa JF (2000) Ontology, Metadata, and Semiotics. Springer-Verlag: 55-81.

Standford University (2009) Protege. http://protege.stanford.edu/. Last visit 2009-09-17.

University of Karlsruhe (2005) KAON Tool Suite. http://kaon.semanticweb.org/. Last visit 2009-09-17.

University of Sheffield (2009) GATE. http://gate.ac.uk/. Last visit 2009-09-17.

Villalon J, Calvo RA (2009) Concept Extraction from student essays, towards Concept Map Mining. Proceedings of the 2009 Ninth IEEE International Conference on Advanced Learning Technologies - Volume 00: 221-225.

W3C (2004) OWL Web Ontology Language Overview. http://www.w3.org/TR/owl-features/. Last visit 2009-09-13.

Wang Y, Völker J, Haase P (2006) Towards Semi-automatic Ontology Building Supported by Large-scale Knowledge Acquisition. AAAI: 1-8.

Wolff KE (1993) A first course in formal concept analysis. Faulbaum, F. (ed.) SoftStat'93 Advances in Statistical Software 4: 429-438.

Zhu D, Dreher H (2009) Discovering Semantic Aspects of Socially Constructed Knowledge Hierarchy to Boost the Relevance of Web Searching. JUCS Volume 15 Issue 8.