[MRSt06] *Marek Meyer, Christoph Rensing, Ralf Steinmetz;* **Towards Using Wikipedia as a Substitute Corpus for Topic Detection and Metadata Generation in E-Learning;** The 3rd annual e-learning conference on Intelligent Interactive Learning Object Repositories, Montreal, November 2006, S. .

Towards Using Wikipedia as a Substitute Corpus for Topic Detection and Metadata Generation in E-Learning

Marek Meyer
SAP AG
SAP Research CEC Darmstadt
Bleichstr. 8, 64283 Darmstadt, Germany
marek.meyer@sap.com

Christoph Rensing, Ralf Steinmetz
KOM Multimedia Communications Lab
Darmstadt University of Technology
Merckstr. 25, 64283 Darmstadt, Germany
{rensing, steinmetz}@kom.tu-darmstadt.de

Abstract

Metadata is crucial for reuse of Learning Resources. Only with good metadata, there is a chance that a Learning Resource can be successfully found in a repository. However, many Learning Resources are still delivered with no or little attached metadata. Automatic metadata generation is used to put things right - either as assistance for the author, or as part of a repository's retrieval functionality.

Among the various metadata fields, those that cover the topic of a Learning Resource are the most important ones - especially keywords and categorization information.

This paper presents a novel approach for domain-independent classification and keyword extraction by utilizing the immense knowledge that is gathered in the free Wikipedia encyclopedia. Wikipedia is proposed as a substitute corpus for classification methods in E-Learning. To support this proposal, the co-occurrence of matching topics and statistical similarity between Learning Resources and Wikipedia articles is analyzed.

An algorithm for keyword generation based on the Wikipedia encyclopedia has been implemented and is described in detail in this paper. The results of the algorithm are presented and discussed.

1. Introduction

E-Learning has become more and more successful over the last years. One of the keys to this success is re-use of E-Learning course and Learning Objects, as multiple usage leads to more efficiency. But re-use of Learning Objects requires good metadata that describes the contents of the Learning Objects. Finding a desired Learning Resource in a large repository is only feasible if adequate metadata for each Learning Resource is available. Unfortunately, creation of good metadata is often neglected; this is where metadata generation and metadata extraction enter the stage [1]. Metadata can be created by several analysis methods that consider the contents of a Learning Object. Especially the generation of topic-related metadata, such as keywords and categories, is important, because most users search for Learning Objects by topic.

Ideally, keywords are not only some words that are somehow related to the contents, but represent concepts, which are covered by a Learning Resource. This would require an ontology which contains all possible concepts. As Learning Resources in a repository are typically located is very different domains, a really comprehensive world ontology would be needed. An Ontology or taxonomy is also used for classification. In some repositories, classification is restricted to exactly one class per Learning Resource.

This paper analyzes existing approaches for domain-independent keyword extraction and classification in E-Learning and proposes a new approach. This approach is based on the free encyclopedia Wikipedia, which meanwhile contains more than one million articles on a multitude of topics. Wikipedia is probably the most complete, freely accessible and structured collection of world knowledge in the Internet. Three hypotheses are formulated in this paper as a foundation for the approach. Starting from the assumption that a Wikipedia article on a particular topic has a certain similarity to Learning Objects on the same topic, we introduce a practical implementation approach and present first results. This approach is based on standard machine learning methods, but with a novel data source.

This paper is structured as follows. First some related work regarding metadata extraction is discussed in section 2. In section 3 a new approach for topic detection based on Wikipedia is introduced. Section 4 gives an overview on the characteristics of the Wikipedia collection, discusses performance issues and presents a practical approach for a first implementation. Classification results of that implementation are presented and discussed in section 5.

2. Metadata Generation

Metadata generation is a field of research that has been heavily worked on in the recent years. There are many approaches for metadata generation for documents in general [10] and for Learning Resources in particular [1]. Metadata generation methods can be classified by the type of metadata to generate, by the sources that are used, by the required prerequisites and the applied methods.

Possible target metadata types are for example content-related metadata (such as title, keywords and categories) process-related metadata (author, creation date, version) or didactically metadata (learning objective, target group, difficulty, activity level). Sources for metadata generation strongly depend on the target metadata types. Content-related metadata requires to analyze the contents of a document, whereas process metadata, such as author and creation date can be obtained from the authoring environment [4]. In the following, existing methods for content-related metadata extraction will be discussed.

Content-related metadata is the most important type of metadata for retrieval of documents and especially Learning Resources. Common content-related metadata fields are title, keywords, classification and an abstract or brief description. Using these fields is usually more efficient than using full text search and produces more relevant search results. The discussion of methods will focus on keywords and classification. Keywords are terms that give a hint on the topics that are covered by a document; these keywords can be any words without restrictions. Classification, in contrast, is restricted to a fixed taxonomy or ontology, from which concepts can be taken to describe the contents of a document. Hence, the methods for generation of keywords and classification information also differ: for classification a mapping to known terms is required, whereas arbitrary words may be produced as keywords.

Classification problems are addressed by classification and clustering methods. Classification here means again that a document is assigned to one or multiple predefined classes. Clustering algorithms build new classes based on the similarity of documents.

Classification methods are a traditional focus of machine learning. If a large enough set of classified examples - also called corpus - is available, it can be used for training a system to automatically assign new documents to the existing classes. Examples for such systems are artificial neural networks, support vector machines (SVM) and the nearest neighbor algorithm. These methods are also called supervised learning, because desired outputs are known in the training phase. Another approach for classification of documents are rule-based systems, such as the ontology-based metadata generation described in [13].

The unsupervised pendent to classification is clustering. Clustering algorithms calculate a distance between documents and build groups of documents, which are near to each other or have common attributes. A common clustering method is the K-means algorithm.

A method for clustering news articles is presented in [9]. A set of 400 clusters is calculated based on the co-occurrence of entities, such as persons, organizations and places. Each of these clusters represents a hot topic that has been extensively discussed in the media.

Latent Semantic Indexing (LSI) is probabilistic method which is similar to clustering. The term-document matrix of a set of documents is transformed into a low-rank approximation by merging terms to concepts. This transformation can be used to calculate the covered concepts of a document. However, the concepts produced by LSI do not necessarily have a real meaning. Therefore, LSI is not suited for generating human-interpretable classes or keywords[9]. Similar to LSI is the Random Indexing method, which lowers rank by using random dimensions[11]. Random Indexing provides comparable results to LSI, but avoids the complex calculation of term co-occurrences.

Some approaches for classification of documents have been presented above - methods for keyword extraction will follow. Keyword extraction methods can be classified by their coverage: Domain-dependent methods are limited to a particular domain but usually provide better results. Domain-independent keyword extraction methods can be applied universally, but are less precise. Domain-dependent methods are based on a domain model, which contains relevant terms for the particular domain. Documents are searched for these terms for determining keywords. [6] demonstrates how to build a domain model out of existing documents. Matsuo and Ishizuka have introduced a domain-independent method for extracting keywords from a single document without having a large corpus of documents [8]. This approach is based on the specific distributional characteristic of terms.

Another method for extracting keywords from web pages is to exploit the structure of a document [5]. Opposing to the approaches above, Kruschwitz uses only those terms as keywords, that appear in at least two different contexts within a document; the considered contexts are meta information, document headings, document title and emphasized parts of a document.

To summarize this section, useful metadata generation methods exist for classification if a large corpus of documents is available for training. In the area of keyword extraction, some domain-independent approaches exist, but most of them also depend on a training corpus. Only the method of Matsuo and Ishizuka works domain-independently on a single document. Some statistical methods can also be applied for keyword extraction in a domain-dependent case.

3. Using Wikipedia as a Substitute Corpus

As the last section has shown, classification of Learning Resources based on topics requires two prerequisites: predefined topic classes and a training corpus which contains several examples per class. There are good exemplary corpora for news articles for web pages, e.g. the Reuters corpora for news or the TREC corpora for web pages [7, 2]. For E-Learning repositories, however, there is no suitable corpus yet. One major problem of current E-Learning repositories is that they contain too few Learning Resources. Combined with the fact, that Learning Resources are not restricted to a certain topic domain, but many cover any topic, there is only little hope that a suitable Learning Resource corpus for automated topic classification will be available in the near future.

Therefore, a new approach is proposed by this paper. Instead of a real corpus of Learning Resources, a substitute corpus shall be used, whose entities bear enough resemblance to real Learning Resources. The free encyclopedia Wikipedia [14] is suggested as such a substitute corpus. Wikipedia is a free, web-based encyclopedia, which is written and updated by a large community of volunteers. This large community ensures that all topics that seem relevant to anyone already are or probably will be described by a Wikipedia article. Wikipedia also is available in several languages, is continually updated and still grows over time. By April 2006, the English Wikipedia database contained more than one million articles. But Wikipedia is not just a collection of articles: it also provides a classification system: Each article may be assigned to one or more hierarchically organized categories.

The important research question is: Is Wikipedia suitable as a substitute corpus for Learning Resources? This paper addresses this question and works towards an answer. The underlying general hypothesis is:

Hypothesis 1 (General Wikipedia Hypothesis)
Learning Resources and articles of the Wikipedia encyclopedia both are knowledge transfer texts. As such, they bear a resemblance. If a Learning Resource and a Wikipedia cover the same topic, a similarity between them can be measured.

If this similarity between Learning Resources exists, it should be exploitable by Information Retrieval methods. Therefore, statistical similarity measurements (e.g. a distance function in a document vector space) are used as a basis to formulate a more specific hypothesis.

Hypothesis 2 (Specific Wikipedia Hypothesis)
Whenever a Learning Resource is statistically similar to a particular Wikipedia article, there is also a similarity in the covered topics. If the statistical similarity exceeds a certain threshold, the Learning Resource covers the same or a closely related topic as the article.

This second hypothesis raises some additional questions. First of all, which statistical methods are suitable to deduce topic similarity from statistical similarity? Second, which minimal threshold value assures a sufficient accurate classification? And finally, the choice of topics and a definition of topic matching has to be defined.

Furthermore, there are large Learning Resources that cover multiple topics. For classification of these Learning Resources, an additional definition of subtopics is helpful. For this purpose, each contiguous extract of a Learning Resource is regarded as a Learning Resource fragment.

Hypothesis 3 (Fragment Hypothesis)
Hypothesis 2 also applies accordingly to Learning Resource fragments. Whenever a topic has been determined as topic of a Learning Resource fragment, it is also considered to be a subtopic of the embracing Learning Resource.

If these hypotheses are true, they can serve as a foundation of using the Wikipedia as a substitute corpus for Learning Resource classification. Two basic classification approaches based on Wikipedia articles are thinkable: Coarse classification using Wikipedia categories as classes or fine-grained classification by regarding each article - and thereby each individual topic - as one class. The second approach, regarding each article as a class, takes into account that each article addresses exactly one well-defined and disambiguated topic; the article title is suited for naming the class.

Keyword generation is a further application of Wikipedia-based topic determination. Keywords for a Learning Resource should be a very brief description of the contents. If matching topics are determined for all relevant fragments of a Learning Resource, the resulting topics and subtopics are very well suited as keywords. In contrast to most other methods, this approach does not depend on useful headlines or particular structures.

4. Proof-of-Concept Implementation

For testing the hypotheses, a first implementation of the approach has been realized. Main goal of the implementation is to prove the first hypothesis and identify critical factors for proving the second and third hypotheses. Hypothesis 1 expresses, that having a Learning Resource and a Wikipedia article about the same topic implicates a similarity of the two documents. That means that there is a co-occurrence of topic similarity and document similarity. The hypothesis though does not specify the way similarity is measured. In Information Retrieval, a common method for determining the similarity of texts is to compare them in

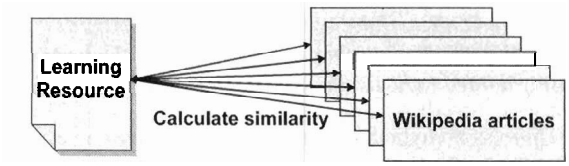

Figure 1. Basic approach for finding similar Wikipedia articles.

Table 1. Statistics on Wikipedia.

	English	German
Number of articles	1.3 M	435 K
Articles > 200 char.	1.2 M	422 K
Average article size [Bytes]	3133	3498
Size of database dump (zipped)	1.2 GB	524 MB
Size of database dump (plain)	5.2 GB	2.0 GB

a Vector Space Model (VSM) [12]. Hence, this method is also applied for the first proof-of-concept implementation. For proving the first hypothesis, it is not necessary to identify the method that calculates similarity best; a method that provides a good similarity function is sufficient.

For the first test, a Learning Resource on the topic "Network Calculus" is used. There is also a Wikipedia article available on that topic. To determine if there is a significant similarity between both documents, the Learning Resource is compared to all existing Wikipedia articles (cf. Fig. 1). Hypothesis 1 can be assumed true if the similarity of the Learning Resource to the "Network Calculus" article is significantly higher than the average similarity values. For proving Hypothesis 2, the "Network Calculus" article would be required to be the best-matching article.

As the Wikipedia-based topic detection approach is targeted not only for repositories, but also for client-side classification and metadata generation, the implementation should be designed with regard to the run-time performance workstations.

4.1. Analysis of Wikipedia

Before starting the design process, some characteristics of the Wikipedia encyclopedia have been analyzed. Some up-to-date statistics on the size and usage of Wikipedia in different languages can be found online [15]. This paper focuses mainly on the English version, and as comparison also on the German one. As of June 2006, the English Wikipedia contains about 1.300.000 articles; for the German Wikipedia version, 435.000 are listed. Most of these articles contain at least 200 visible characters.

For further considerations and an implementation the provided complete databases dumps have been downloaded.

Table 2. Number of terms per range of document frequencies.

Document frequencies	Terms (EN)	Terms (DE)
1	1.898.542	1.598.058
2	518.047	360.442
3	223.730	163.643
4-5	204.943	160.157
6-10	186.834	156.739
11-20	116.379	102.823
21-30	45.387	40.750
31-50	40.761	373.04
51-100	34.540	32.356
101-1.000	43.495	39.514
1001-10.000	9.668	7085
10.001-100.000	2.421	1041
100.001-200.000	140	47
200.001-300000	20	16
300.001-400.000	5	80
400.001-∞	7	2
$\sum$	3.324.919	2.699.985

For the English version, the database dump from the 20th of April 2006 is used. The German database dump dates from the 4th of June. Once the database dumps are unzipped into plain XML files, they consume several Gigabytes of disk space (see Table 1). The dumps contain pages which do not represent articles, but special pages, images pages or redirects - these pages are not included as articles throughout this paper.

Important for processing text documents in the Vector Space Model are terms. We consider only one-word terms; common stemming algorithms are used to map words to their basic word stem. We have counted the document frequency for each occurring term. The document frequency indicates in how many different articles a term occurs. Table 2 presents the number of terms that fall into different ranges of document frequencies for the English (EN) and German (DE) version of Wikipedia. In sum, there are over 3 million different terms in the English Wikipedia. But more than half of the terms occur in only one article. On closer examination, most of these terms seem to be words from different languages, fantasy words or unfamiliar names. All terms that occur in at most two documents form two-thirds of the whole vocabulary.

4.2. Performance Considerations

The implementation is considered to run on typical workstations. We therefore assume a computer with a 3

GHz desktop CPU, 2 GB main memory and Java as programming language as the target platform. Due to the operating system, overhead and other influences, only 1 to 1.5 GB of RAM are effectively available for an application. Furthermore, performing the algorithm on a workstation usually implies, that a user is sitting in front of the computer and waiting for a result; this leads to a desire for fast execution. Real-time behavior - delivering results within some seconds - should be aimed at as optimum. Based on these conditions, performance considerations are discussed in this section.

The three most important reasons for performance bottlenecks are

- Disk memory
- Size of in-memory representation
- Structure in-memory representation

High consumption of disk space also means many disk operations, which are very slow. The size of RAM footprint mainly matters if the amount of required memory exceeds the RAM size - in this case expensive swapping is needed. And finally, the structure of the run-time representation of data has an impact on the complexity of the comparison algorithm.

Assume that a non-optimized Vector Space Model implementation is used. The original English Wikipedia database dump is 5 GB. A rule of thumb for the dimension of a document classification indices is to multiply the size by two; this leads to an index size of 10 GB - definitely too large for the target main memory. A memory size optimization is required.

Second, consider the calculation effort to compute a simple distance calculation - the inner product of two vectors - in the VSM. For the ease of estimation, the number of articles is rounded down to 1.000.000 articles and the number of dimensions down to 3.000.000. A total number of 1.000 Learning Resource fragments is assumed. As a result, 3×10^{15} floating point multiplications have to be calculated. If one multiplication is executed per CPU cycle, the algorithm runs for about 277 hours. The execution time is also inappropriate and has to be decreased.

Of course, some information retrieval libraries provide generic performance optimizations. But the known characteristics of the particular application can be utilized for a tailored optimization. This means especially to find an accurate trade-off between memory consumption, execution time and classification quality. Objectives for the implementation are

- Make run-time representation small enough to fit completely into main-memory

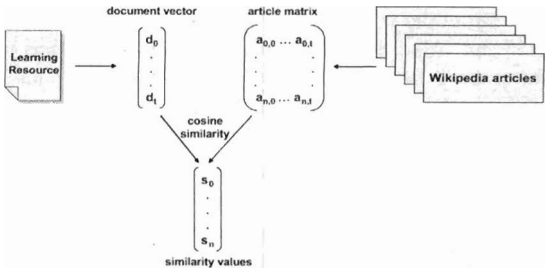

Figure 2. Calculation of similarity values.

- Optimize in-memory structure and algorithms for fast execution
- Reduce consumption of disk space to minimize disk operations

4.3. Implementation

The implementation is based on a Vector Space Model. All words that are used in any Wikipedia article are used as dimensions after a stemming algorithm and a stop list have been applied. Java 1.5 has been chosen as programming language.

In a preparation phase, all Wikipedia articles are transformed into document vectors. First the whole Wikipedia database is scanned for all used words. The stemmed forms of these words are stored as a global word list that serves as a description of the vector space dimensions. In a second pass over the database dump, a vector is created for each article. The vector is weighted by TF.IDF and normalized to a length of 1. The resulting article vectors are stored to disk. This preparation step has to be performed only once per version of the database dump.

At run-time, all article vectors are read into main memory again. Then a Learning Resource is divided into several fragments. For each of these fragments, the contained text is extracted and transformed into a document vector; this transformation is performed analogue to the previous transformation of articles. Each fragment vector is compared to all article vectors. Currently, the inner product is applied as similarity function (see Fig. 2). The highest matches for each fragment are interpreted as classification result. Depending on the mode, either a fixed number of matches or all matches with a similarity value above a certain threshold are used.

In the following, some of the implementation details are explained.

Top priority is to reduce memory consumption. First of all, a sparse vector representation is used, which means that only non-zero elements of vectors are stored. Considering, that most articles contain only some hundred different

words out of the vocabulary of 3 million words, the effect is significant. Also, the precision of vector elements has been minimized in favor of low vector sizes: using the data type float instead of double saves 25% of memory consumption after all.

As a first approach, the number of vector space dimensions has been reduced. The vector dimensions represent the words that occur in a document; hence a decision to reduce the number of dimensions has to take into account the relevance of different words. A common method is to remove words with a very high document frequency, because they are considered to have only little relevance. Furthermore, subsection 4.1 has shown, that a very large amount of words occur in only one or two articles. On the one hand, rare words generally are considered to have a very high significance. On the other hand, in the case of Wikipedia articles, most of these terms seem to be names, words from a foreign language or misspellings. Also, a term, which occurs in only one or two articles, might be a rather unknown word. Removing terms with a document frequency of only 1 and two could significantly decrease the number of dimensions. The implementation allows to define minimum and maximum document frequencies to minimize the vector space dimensions. However, because a sparse vector format is used, removing very frequent terms saves much more memory than removing very rare terms.

The run-time representation of article vectors is realized using the Compressed Sparse Row (CSR) format, which contains only none-zero values plus two index vectors: a column index, which determines the position within a vector, and a row index which indicates where each vector starts [3]. Vectors of Learning Resource fragments are stored as hash tables. Each hash table entry represents a non-zero element of the vector; the element position within the vector is used as key for allowing fast random access to non-zero elements.

Based on the CSR representation for article vectors and a hash table representation of Learning Resource fragment vectors, an optimized algorithm for calculating the similarity has been implemented. Goals for the implementation have been a low complexity and the optimal usage of CPU cache. Most of the calculation effort for determining the inner product (cosine similarity) of two vectors might be wasted on multiplying zeros by generic implementations. As a consequence, the product algorithm does not iterate over all elements of a vector, but only over the non-zero elements of the CSR representation. The column index is used to obtain the corresponding element from the fragment's hash table. This algorithm also makes good use of the CPU cache, because the fragment vectors may remain in the cache for the complete calculation, whereas each article vector is transferred into the cache only once for a short time.

Table 3. Size of Wikipedia articles after transformation into Vector Space Model.

Language	Used terms (doc. freq.)	Size of VSM data
English	3 - 2M	1.12 GB
German	all	630 MB
German	3 - 200K	558 MB

The file format for article vectors is a sparse vector format again. Because of the large amount of data, a binary format instead of the more common plain text format has been chosen. Each vector contains the key (identifier) of the Wikipedia article, the number of non-zero elements and uses 8 Bytes per non-zero element (4 Bytes for index and 4 Bytes for the value). Hence, an article, which contains about 100 different words, needs 808 Bytes disk storage.

5. Test Results and Interpretation

For the English and German Wikipedia databases have been transformed into the Vector Space Model for first tests. The used word lists have been varied to find out the effect of reduced vector space dimensions. For example, from the German Wikipedia all words that occur in less than three documents or more than 200.000 documents have been removed. The result was that the word list itself significantly shrinks, but the vector information decreases only slightly from 630 MB to 558 MB. Some sizes of VSM representations are given in Table 3. However, determining the impact on classification performance would require a large-scale experiment. For the English version, only a limited word list has been used because of limited main memory resources. The transformation process is very time consuming and took about two days for the German Wikipedia and four days for the English Wikipedia on a standard workstation.

In contrast, the run-time performance of the classification algorithm is faster. Performing a classification of an English sample Learning Resource took about 30 minutes. However, the bottleneck is the transfer of data from disk into main memory. 28 minutes were consumed by loading the article vectors, but only 24 seconds were needed for determining the similarity between a given Learning Resource vector and all articles. Creating a vector representation of a text document has taken less than one second. The total time of the method is quite high. But once the vector data is available in main memory, the classification works at an acceptable speed.

For the sample Learning Resource on "Network Calculus", one vector for the whole course has been created. This vector has been used as input for the classification method.

Table 4. Result of topic classification for "Network Calculus" course including category pages.

Article	Similarity
Category:Algebraic curves	0.445
Network calculus	0.397
Category:Economics curves	0.320
Category:Elliptic curves	0.285
Singularity (mathematics)	0.279
Singular points	0.278
Curved Bar	0.272
Category:Spirals	0.271
Category:Packets	0.258

For testing the first Hypothesis, the similarity value for the article "Network Calculus was of interest. The measurement has produces a similarity of 39.66% for that article, whereas the average similarity value was only 1.17%. The most similar articles are listed in Table 4. This result supports Hypothesis 1.

The classification results show some interesting characteristics. First of all, category pages - which had not been removed from the articles database before - are obviously overrated. This can be explained by the different linguistic structure of category pages: A category page consists mainly only of titles of several related articles; therefore it contains many high-rated keywords, in contrast to those words in natural language, which have less significance. Nevertheless, the classified categories are not completely off-topic, because the network calculus course uses mathematical curves for modeling network traffic.

If category pages are removed from the result list (see Table 5), "Network Calculus" is the article with the highest similarity to the given Learning Resources. Furthermore, the next regular article in the list has a similarity value of only 28%, which is more than 10% lower than the correct match. If the top match is used, the Wikipedia-based classification method provides a good result for this sample course. Using a threshold of one third (33%) shows the same result. For making a general statement on how to best select topics for metadata, another test with a larger set of Learning Resources has to be carried out.

6. Conclusions and Outlook

This paper has introduced a new approach for metadata generation based on using the Wikipedia encyclopedia as a substitute corpus. This approach uses standard machine learning technology, but with a different data source. The test results that have been presented are very promising.

Table 5. Result of topic classification for "Network Calculus" course without category pages.

Article	Similarity
Network calculus	0.397
Singularity (mathematics)	0.279
Singular points	0.278
Curved Bar	0.272
Average similarity	0.012

Three hypothesis have been postulated as a foundation of the Wikipedia-based approach. The first two hypotheses have been supported by test results: The comparison of a Learning Resource with Wikipedia articles can be used for determining the topic of the Learning Resource. This information can be used either for classification or for generating keywords for a metadata record. For proving the hypotheses, further expanded tests will be required, including a consideration of statistical significance.

However, the approach might produce results only for a particular granularity of Learning Resources. This has to be checked in further experiments. For large Learning Resource, which covers a variety of topics, the approach might possibly produce less accurate results. Therefore, the third hypothesis becomes relevant. Subtopics for several fragments of a Learning Resource can be determined; then an overall topic is built out of fragment subtopics. The links between Wikipedia articles might be used to find the overall topic. Also, subtopics of fragments may be used as keywords for the Learning Resource metadata.

Some additional ideas have to be tested if they can improve the classification method. For example, different similarity functions, such as binary comparison or a word recall rate, have to be evaluated. Furthermore, if a topic domain of Learning Resources to classify is already known in advance, the corpus for comparison could be limited to articles of that particular domain. The domain-limited corpus for classification of Learning Resources on medical science could for instance be much smaller than the general purpose corpus.

7. Acknowledgments

This work is supported by the German Federal Ministry of Economics and Technology in the context of the project Content Sharing.

References

[1] S. Bergstraesser. Automatisierung der Erstellung von Metadaten. Diploma thesis, Darmstadt University of Technology,

Mar 2005.

[2] Commonwealth Scientific and Industrial Research Organisation. TREC-2004 Web Research Collections. http://es.csiro.au/TRECWeb/, [Online; last visited 28th August 2006].

[3] N. Goharian, A. Jain, and Q. Sun. Comparative analysis of sparse matrix algorithms for information retrieval. *Journal of Systemics, Cybernetics and Informatics*, 2003.

[4] S. Hoermann, T. Hildebrandt, C. Rensing, and R. Steinmetz. ResourceCenter - A Digital Learning Object Repository with an Integrated Authoring Tool Set. In P. Kommers and G. Richards, editors, *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications ED-MEDIA 2005*, pages 3453–3460, Montreal, Canada, June 2005. AACE.

[5] U. Kruschwitz. Exploiting structure for intelligent web search. In *Proceedings of the 34th Hawaii International Conference on System Sciences (HICSS)*, 2001.

[6] U. Kruschwitz. A rapidly acquired domain model derived from markup structure. In *Proceedings of the ESSLLIŠ01 Workshop on Semantic Knowledge Acquisition and Categorisation*, Oct. 17 2001.

[7] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2004.

[8] Y. Matsuo and M. Ishizuka. Keyword extraction from a single document using word co-occurrence statistical information. *International Journal on Artificial Intelligence Tools*, 13(1):157–169, 2004.

[9] D. Newman, C. Chemudugunta, P. Smyth, and M. Steyvers. Analyzing entities and topics in news articles using statistical topic models. In S. Mehrotra, D. D. Zeng, H. Chen, B. M. Thuraisingham, and F.-Y. Wang, editors, *Proceedings of the IEEE International Conference on Intelligence and Security Informatics, ISI 2006*, volume 3975 of *Lecture Notes in Computer Science*, pages 93–104. Springer, 2006.

[10] P. Noufal. Metadata: Automatic generation and extraction. In *7th MANLIBNET Annual National Convention on Digital Libraries in Knowledge Management: Opportunities for Management Libraries*. Indian Institute of Management Kozhikode, May 2005.

[11] M. Sahlgren. An introduction to random indexing. In *Proceedings of the Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, TKE 2005*, 2005.

[12] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[13] H. Stuckenschmidt and F. van Harmelen. Ontology-based metadata generation from semi-structured information. In *K-CAP '01: Proceedings of the 1st International Conference on Knowledge Capture*, pages 163–170, New York, NY, USA, 2001. ACM Press.

[14] Wikipedia. The free encyclopedia. http://en.wikipedia.org, 2006. [Online; last visited 28th August 2006].

[15] Wikipedia. Wikipedia statistics. http://stats.wikimedia.org/EN/TablesRecentTrends.htm, 2006. [Online; last visited 4th September 2006].