Towards a Librarian of the Web

Mario M. Kubek
Chair of Communication Networks
University of Hagen
Universitätsstr. 27, Hagen, Germany
mario.kubek@fernuni-hagen.de

Herwig Unger
Chair of Communication Networks
University of Hagen
Universitätsstr. 27, Hagen, Germany
herwig.unger@fernuni-hagen.de

ABSTRACT

If the World Wide Web (WWW) is considered to be a huge library, it would need a librarian, too. Google and other web search engines are more or less just keyword databases and cannot fulfil this person's tasks in a sufficient manner. Therefore, an approach to improve cataloguing and classifying documents in the WWW is introduced and its efficiency demonstrated in first simulations.

CCS Concepts

•Information systems → Content analysis and feature selection; •Computing methodologies → Semantic networks;

Keywords

librarian, text processing, co-occurrence graph, centroid term, text clustering, text similarity, web search

1. INTRODUCTION

Libraries are often lonesome places these days, because most of the information, knowledge and literature is made available in the omnipresent Internet. It seems that the times are forgotten, when librarians collected giant amounts of books, archived them using their (own) special system to put all of these documents in the right place in many floors consisting of a maze of shelves, and –finally– made them usable by huge catalogue boxes containing thousands of small cards. In addition to these tasks, they had time to support library users by giving them advises to find the wanted information quickly and maybe tell them the latest news and trends, too.

Establishing a real library needs a big effort and is definitely a time-consuming learning process in which the interaction with its users plays an important role, i.e. it is a process with a determined history. This also results in the observation that two librarians ordering documents (mostly books) may end up with completely different arrangements depending on their own experienced process of knowledge acquisition. It usually requires a deep study of the texts (if not even special knowledge on the considered subjects) in order to find out some major meaning and terms to be later used in the assignment of categories and the determination of their relations, a process which also involves an estimation of the semantic similarity and distance to other terms and texts available. Thus, only after a larger amount of knowledge is gathered, a first classification of documents may be carried out with the necessary maturity and a first, later expansible catalogue and archiving system may be established. In such a manner, a catalogue is a small and compact abstraction of details in each book and in a condensed form even a representation of human intelligence that was used in connecting related books with each other and deciding on the card placements accordingly.

It is definitely a huge merit of the WWW to make the world's largest collection of documents of any kind of contents easily available at any time and any place without respect to the number of copies needed. It can therefore be considered to be the knowledge basis or library of mankind in the age of information technology. Google, as the world's largest search engine with its main role to connect information and the place/address where it can be found, might be the most effective, currently available information manager. But can Google claim to be the librarian of the WWW? Is Google (or any of the big search engines) really an efficient information manager and can it compete with a librarian in her/his dusty (and surely much smaller) library?

2. CRITIQUE OF EXISTING WEB SEARCH ENGINES

The authors think that Google and Co. are just the mechanistic, brute force answer to the question of how to manage the complexity of the WWW. As already discussed e.g. in [1], a copy of the web is created by crawling it and indexing web content in big reverse index files, containing for each occurring word a list of files in which it occurs. Complex -yet fixed- algorithms find all documents containing all words of a given query. Since (simply chosen) keywords/query terms appear in potentially millions of documents, a ranking like PageRank [2] and others must avoid that all of these documents are touched and presented in advance to the user (see Figure 1). In the ranking process, the relative importance of a document in the web graph and the graph's linking structure (today, economic interests and results of personalisation efforts and interest prediction may be of influence, too) is evaluated.

ICCIP 2016 Singapore

DOI:

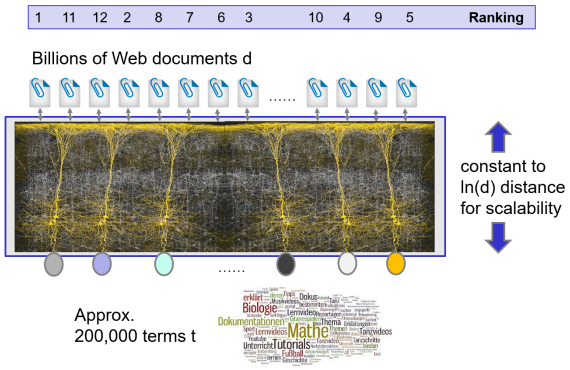

Figure 1: The dimensionality problem of the WWW

Judging from the well-known linear result lists generated this way, Google is nevertheless not able to abstract, classify or group contents following any semantic considerations and does not include or present any new linkage of documents, authors or contents as well. All those problems may be solved however by hard working and experienced librarians.

Certainly, Google greatly outperforms any librarian in processing simple queries and carrying out connectivity analyses and statistical calculations using its huge databases even when taking into account the rather limited amount of information in the books of a single library. However, the establishment of bibliographies and so-called signposts (pathfinder) is still a service of a librarian no search engine can offer so far, since every document is technically considered as an own instance and only relations between documents using links in them are considered. Finally, the authors argue that search engines cannot compete with the human classification process yet, because they do not support the already described maybe slow, but supervised learning and evaluation process and always apply the same standard techniques such as keyword processing to index all incoming documents, whereby the semantics or sequence of documents processed does not play a big role.

Information literacy [3], i.e. to have the ability to recognise when information is needed and to (proactively) locate, evaluate, and effectively access needed information, is not demonstrated by Google and Co. as it is not implemented. Also, it is hard for their users to gain it by using them, as the presented search results are merely pointers to documents with potentially relevant information. These documents must be manually inspected and evaluated with respect to their relevance to the information need at hand. Afterwards, this information need and the respective queries are (iteratively) adapted based on the newly learned contents. This is -especially for non-experts and novices in a field- an often time-consuming and error-prone process.

Google for instance fails, if it is asked e.g. 'What is the science predicting the future by stars?'. In the suggestions, astrology appears after entering 'predicting the future', however none of the top-ranked search results contain it in their title. The reason is that Google's indexing process only considers words contained in the documents, but not topically related or close ones. The same problem arises during the search using keywords but is usually not recognised as such due to the plenty of returned results to each query.

The authors argue that a decentralised WWW librarian system would be a proper solution to these problems as it would suitably combine search functionalities with semantic text analysis methods and would replace the more or less centralised *crawling-copying-indexing-searching* procedure with a decentralised *learn-classify-divide-link&guide* method, that

1. contains a document grouping process based on a successive category determination and refinement (including mechanisms to match and join several categorisations/clusters),
2. is based on a fully decentralised, document management process that (largely) avoids the copying of documents,
3. allows for search inquiries based on keywords to be classified and forwarded by the same decision process that carries out the grouping of the respective target documents to be found and
4. supports a user-access-based ranking to avoid a network flooding with messages.

A basic, new concept that meets these requirements shall be derived and explained in the subsequent sections.

3. CONCEPT

In order to realise the suggested WWW librarian with its main task to manage and topically group text document collections, it is important to programmatically identify characteristic terms (e.g. keywords, concepts), words and word forms from those documents and to determine the degree of their semantic relatedness as well. The problem arising in this context is that word forms are basic elements in texts and, therefore, carry no explicit attributes to characterise their semantic orientation and meaning which could be used to compare their similarity.

However, as natural language text is unstructured data (from the machine's point of view), it needs to be cleaned and transformed into information or another representation that can be uniformly handled by algorithms. That is why linguistic preprocessing is usually applied in text analysis pipelines [4] on the documents to be analysed whereby

- running text from text files in different file formats is extracted,
- the used language is identified,
- sentence borders are detected,
- stop words (short function words from closed word classes that carry no meaning) are removed,
- parts-of-speech of the word forms found are determined and
- only nouns (usually in their base form), proper nouns, names and phrases (partly adjectives and verbs in the field of sentiment analysis, too) are extracted for further considerations.

Those important words actually determine the meaning of texts as they represent real-world entities, their properties as well as actions concerning them. In this understanding, a

word is an equivalence class of related word forms (inflected form of a given root word) usually found in texts. To simplify the following elaborations, the words 'term' and 'word' are used synonymously and mostly concern nouns, proper nouns and names.

3.1 State of the Art: Word Importance and Relatedness

The selection of characteristic and discriminating terms in texts through weights, often referred to as keyword extraction or terminology extraction, plays an important role in text mining and information retrieval. The goal is to identify terms that are good separators that make it possible to topically distinguish documents in a corpus. In information retrieval and in many text mining applications, text documents are often represented by term vectors containing their keywords and scores while following the bag-of-words approach (the relationship between the terms is not considered). Text classification techniques as an example rely on properly selected features (in this case the terms) and their weights in order train the classifier in such a way that it can make correct classification decisions on unseen contents which means to assign them to pre-defined categories.

As a first useful measure, variants of the popular TF-IDF statistic [5] can be used to assign terms a weight in a document depending on how often they occur in it and in the whole document corpus. A term will be assigned a high weight, when it often occurs in one document, but less often in other documents in the corpus. However, this measure cannot be used when there is no corpus available and just a single document needs to be analysed. Furthermore, it does not take into account the semantic relations between the terms in the text.

Another approach from statistical text analysis to find discriminating terms is called difference analysis [6]. Terms in a text are determined and assigned a weight according to the deviation of word frequencies in single (possibly technical) texts from their frequencies in general usage (a large topically well-balanced reference text corpus such as a newspaper corpus which reflects general language use is needed for this purpose). The larger the deviation is, the more likely it is that a (technical) term or keyword of a single text has been found. If such a reference corpus is not available, this method cannot be used, too.

Under the assumption that local weights for terms even in single texts need to be determined, it is sensible to consider the semantic relations between terms contained in order to determine their importance. Approaches following this idea would not require external resources such as preferably large text corpora as a reference. For instance, two state-of-the-art solutions for graph-based keyword and search word extraction that implement this idea are based on extensions of the well-known algorithms PageRank [7] and HITS [8]. As a prerequisite, it is necessary to explain, how those semantic term relations can be extracted from texts which can then be used to construct term graphs (or word nets) that are analysed by these solutions.

Semantic connections of terms/words come in three flavours: synonymy, similarity and relatedness. In case of synonymy, two words are semantically connected because they share a meaning. Their semantic distance -to quantify this relation- is 0. However, words can be semantically connected although they do not share a meaning. In this case, the semantic distance is greater than 0 and can reflect either similarity and/or relatedness of the words involved. As an example, 'cat' and 'animal' are similar and related. However, 'teacher' and 'school' are related, but not similar.

In the simplest case, a resource like Roget's Thesaurus [9] is available and thus it is possible to directly check for the words' synonymy. However, the task is getting more difficult when only text corpora or standard dictionaries are at hand. In these cases, synonymy of words cannot be directly derived or even taken for granted. Here, measures to quantify the semantic distance between words can be applied. A very low distance is often a sign for word synonymy, especially if they often appear together with the same words (have the same neighbours). Recently, a very effective method [10] for synonym detection using topic-sensitive random walks on semantic graphs induced by Wikipedia and Wiktionary has been introduced. This shows, that such tasks can be carried out with high accuracy even when static thesauri or dictionaries are unavailable.

Also, in recent years, several graph-based distance measures [11, 12] have been developed that are knowledge-based and make use of external resources such as the manually created semantic network WordNet [13], a large lexical database containing semantic relationships for the English language that covers relations like polysemy, synonymy, antonymy, hypernymy and hyponymy (i.e. more general and more specific concepts), as well as part-of-relationships. These measures apply shortest path algorithms or take into account the depth of the least common subsumer concept (LCS) to determine the semantic distance between two given input terms or concepts. With the help of these resources, it is instantly possible to determine their specific semantic relationship as well.

In [14], the authors of the present paper have pointed out that the statistical significance of the co-occurrence of two terms/words in any order in close proximity in a text or text corpus is another reliable indication for an existing semantic relatedness. The technique of statistical co-occurrence analysis can be used to extract those word pairs. Moreover, a *co-occurrence graph* $G = (W, E)$ may be obtained, if all words W of a document or a text corpus are used to build its set of nodes which are then connected by an edge $(w_a, w_b) \in E$ if $w_a \in W$ and $w_b \in W$ are co-occurrents (the words that co-occur). A weight function $g((w_a, w_b))$ indicates, how significant the respective co-occurrence is in the given content. It was shown in [14] as well that the distance d of any two nodes (terms) w_a and w_b in a fully connected graph G can be obtained by computing the shortest path between them:

$$d(w_a, w_b) = \sum_{i=1}^{k} d((w_i, w_{i+1})) = MIN, \qquad (1)$$

whereby in case of a partially connected co-occurrence graph $d(w_a, w_b) = \infty$ must be set.

The distance between a given term $t \in G$ and a document D containing N words $w_1, w_2, .., w_N \in D$ that are reachable from t in G can then be defined by

$$d(D, t) = \frac{\sum_{i=1}^{N} d(w_i, t)}{N}, \qquad (2)$$

i.e. the average sum of the lengths of the shortest paths between t and all words $w_i \in D$ that can be reached from it. Note that -differing from many methods found in literature-

it is not assumed that $t \in D$ holds! The term $t \in G$ is called the centre term or *centroid term of* D when $d(D,t) = MIN$ applies. Thus, the semantic distance ζ between any two documents D_1 and D_2 with their respective centroid terms t_1 and t_2 can be derived by

$$\zeta(D_1, D_2) = d(t_1, t_2). \quad (3)$$

The centroid terms obtained this way generally represent their documents very well. Also, this distance method is able to detect a similarity between topically related documents that, however, do not share terms or only have a limited number of terms in common. The cosine similarity measure (when relying on the bag-of-words model) would not be able to accomplish this.

3.2 Text Clustering using Document Distances

Based on these definitions and findings, it can be assumed that this new distance measure is well-suited to be applied in text clustering solutions. In order to proof the correctness of this assumption, a number of experiments have been carried out and will be described in the next section in detail.

At this point, it is intended to describe the building process of a WWW library which is carried out in strictly the same way a usual book collector or librarian would perform this task. Starting with a few text documents, the collection will grow until it cannot be managed at one location anymore. Consequently, the set of documents must be divided into two subsets (dichotomy) and stored separately. Documents contained in one location shall have similar or related contents that significantly differ from texts at the other place. Following this idea, the centroid terms in these collections such as names, categories, titles, major subjects can be identified which are the building parts of the collection and are used as the content of a small, descriptive guiding catalogue (in analogy to the small cards). Later, the described steps can be applied in the same manner to each sub-collection.

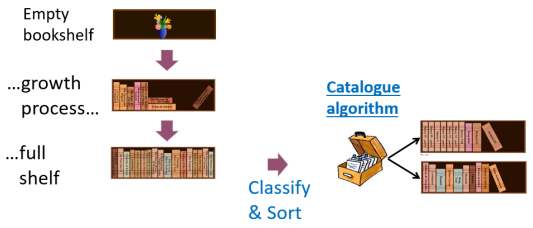

Figure 2: Growth and division of a book collection

This working principle shall now be implemented in unsupervised algorithms for the librarian management and the grouping of text documents. Here, the distance of centroid terms in a co-occurrence graph is used as a metric to determine the semantic closeness of documents. Moreover, the results are generated in an almost fully decentralised manner. In the following elaborations, the term librarian refers to the software that actually performs the document clustering algorithm and management tasks described later. A new librarian starts with an initially generated root node R_1. The refinement level of the librarian k is initially set to $k(R_1) = 1$ and the classification term t of the root node is set to $t(R_1) = NIL$. Links to documents are managed in a local database of each node, denoted by $L(R_k)$. Every node R_k can generate two child nodes, whereby the node identifications R_{2k} and R_{2k+1} are kept on R_k. The local co-occurrence graph $G(R_k)$ represents the state of the knowledge of word relations and their distances and corresponds to the respective knowledge of the librarian at this level. R_1 starts with the execution of the following algorithm I. It is started on any generated child node R_k, too.

ALGORITHM I (Librarian Management)

1. Receive the initial node data.
2. **REPEAT** //Growth loop
   (a) receive and store document links l in $L(R_k)$ on R_k and add their co-occurrences to the local co-occurrence graph $G(R_k)$
   (b) receive search queries and answer them using the evaluation of documents addressed by $L(R_k)$.

   UNTIL the memory is full.
3. //Classify and sort
   Use the following algorithm II or IIa to find a division (dichotomy) of all documents $D(L)$ addressed by links $l_i \in L(R_k)$ into two sets $D_x(l_i)$, $x \in \{1,2\}$. For later use, define $f_d(T, R_k, D_1, D_2)$ as a function returning the index of either D_1 or D_2 to which any query or text document T is most similar. This index can be determined e.g. using the centroid-based distance, a naive Bayes classifier or any other suitable similarity function.
4. Generate two child nodes of R_k, R_{2k} and R_{2k+1} and send them their respective value for k and t which corresponds to the centroid terms of either one of the dichotomy sets D_1 or D_2. Also send them a copy of $G(R_k)$ for later extension.
5. Move (not copy) the link to every document in D_1 to the node R_{2k} and in D_2 to R_{2k+1}, respectively.
6. **WHILE** //Catalogue and order loop
   (a) Receive and calculate for all obtained text links l – i.e. either for incoming, new documents or (sequences of keywords of) search queries $T(l)$ – $x = f_d(T(l), R_k, D_1, D_2)$.
   (b) If $x = 1$ move/forward respective document link or search query T to $R_{2k+(x-1)}$.

   END.

The clustering method to determine the dichotomy of documents will significantly influence the effectiveness of this approach. Both of the following algorithms to do so borrow some ideas from the standard but discrete $k-means$ clustering algorithm [15] with the parameter k (number of clusters to be generated) set to $k = 2$.

ALGORITHM II (Document Dichotomy)

1. Choose two documents $D_1, D_2 \in D(l_i)$, i.e. addressed by an $l_i \in L(R_k)$, such that for their centroid terms

t_1 and t_2 in R_k respectively, $d(t_1, t_2) = MAX$. (antipodean documents). If there are several pairs having (almost) the same high distance, choose a pair, for which both centroids have an almost similar, high valence.
Set $D(L) := D(L) \setminus \{D_1, D_2\}$.

2. Randomly choose another document $D_x \in D(L)$ and determine its centroid t_x.
3. If $d(t_x, t_1) \leq d(t_x, t_2)$ in R_k set $c = 1$ and otherwise $c = 2$.
4. Build $D_c := D_c \cup D_x$. In addition, set $D(L) := D(L) \setminus D_x$.
5. While $D(L) \neq \emptyset$, GoTo 2.
6. Determine the new centroid terms $t_c(D_c)$ using R_k for both document sets obtained for $c = 1, 2$, i.e. D_1 and D_2.

In this case, $f_d(T, R_k, D_1, D_2) = 1$, if for a given text or query T with the centroid term t $d(t, t(D_1)) \leq d(t, t(D_2))$ in R_k and otherwise $f_d(T, R_k, D_1, D_2) = 2$. In contrast to the classic k-means algorithm, the repeated calculation of the updated centroids of both obtained clusters is avoided and carried out only once such that the algorithm runs faster. In order to overcome the (possible) loss of exactness in this sequential process, a modification is made as follows:

ALGORITHM IIa (Document Dichotomy)

1. Choose two documents $D_1, D_2 \in D(l_i)$, i.e. addressed by an $l_i \in L(R_k)$, such that for their centroid terms t_1 and t_2 in R_k respectively, $d(t_1, t_2) = MAX$. If there are several pairs having (almost) the same high distance, choose a pair, for which both centroids have an almost similar, high valence.
Set $D(L) := D(L) \setminus \{D_1, D_2\}$.
2. Choose another, remaining document $D_x \in D(L)$ such that its centroid t_x is as close as possible to t_1 or t_2.
3. For both document sets D_1 and D_2, i.e. for $i = 1, 2$ calculate the average distance $d(t_x, D_i)$ of the centroid of the newly chosen document D_x to all centroids of texts $D_{i,1}..D_{i,|D_i|}$, which are already assigned to D_1 or D_2 by

$$d(t_x, D_i) = \frac{1}{|D_i|} \sum_{j=1}^{|D_i|} d(t_x, t(D_{i,j})).$$

If $d(t_x, D_1) \leq d(t_x, D_2)$ set $c = 1$ and otherwise $c = 2$.
4. Build $D_c := D_c \cup D_x$. In addition, set $D(L) := D(L) \setminus D_x$.
5. While $D(L) \neq \emptyset$, GoTo 2.
6. Determine the new centroid terms $t_c(D_c)$ using R_k for both document sets obtained for $c = 1, 2$, i.e. D_1 and D_2.

In this case, the determination of $f_d(T, R_k, D_1, D_2)$ remains the same as presented in algorithm II.

Since a formal analysis of the described (heuristic) mechanisms seems to be impossible, in the following section, some important properties and clustering results shall be investigated by simulative experiments.

4. CLUSTER EVALUATION

For all of the exemplary experiments discussed in this section, linguistic preprocessing has been applied on the documents to be analysed whereby stop words have been removed and only nouns (in their base form), proper nouns and names have been extracted. In order to build the undirected co-occurrence graph G (as the reference for the centroid distance measure), co-occurrences on sentence level have been extracted. Their significance values have been determined using the Dice coefficient. The composition of the used sets of documents will be described in the respective subsections[1].

The aim of the following preliminary experiments is to show that among the first k documents returned (they have the lowest centroid distance to a reference document) according to the centroid distance measure, a significant amount of documents from the same topical category is found. This experiment has been carried out 100 resp. 200 times for all the documents in the following two datasets (each document in these sets has been used as the reference document). The datasets used consist of online news articles from the German newspaper "Süddeutsche Zeitung" from the months September, October and November of 2015. Dataset 1.1 contains 100 articles covering the topics 'car' (25), 'money' (25), 'politics' (25) and 'sports' (25); dataset 1.2 contains 200 articles on the same topics with each topic having 50 documents. The articles' categories (tags) have been manually set by their respective authors. On the basis of these assignments (the documents/articles to be processed act as their own gold-standard for evaluation), it is possible to easily find out, how many of the k nearest neighbours (kNN) of a reference document according to the centroid distance measure share its topical assignment. The goal is that this number is as close to $k = 5$ resp. $k = 10$ as possible. For this purpose, the fraction of documents with the same topical tags will be computed.

Table 1: Average number of documents that share the reference documents' category for their $k = 5$ resp. $k = 10$ most similar documents

Aver. number of doc. / median	$k = 5$	$k = 10$
Dataset 1.1	3,9 / 5	7,6 / 9
Dataset 1.2	3,9 / 5	7,5 / 9

As an interpretation of table 1, for dataset 1.1 and $k = 5$, the centroid distance measure returned in average 3,9 documents with the reference document's topical assignment first. For the $k = 10$ returned documents first, in average 7,6 documents shared the reference document's tag. The median in both cases is even higher.

These good values indicate that with the help of the centroid distance measure it is indeed possible to identify semantically close documents. Furthermore, the measure is able to group documents with the same topical tags which is a necessity when building a librarian-like system whose performance should be comparable to human judgement, even when no such assessment is available. This is a requirement that cannot be met by measures relying on the bag-of-words model. The centroid distance measure's ap-

[1]Interested researchers may download these sets (1,3 MB) from: http://www.docanalyser.de/cd-clustering-corpora.zip

plication in kNN-based classification systems seems therefore beneficial as well. The findings further suggest that the centroid distance measure is useful in document clustering techniques, too.

That is why, the clustering algorithms II and IIa apply this measure and are presented and evaluated using a set of experiments in the following subsections. Their effectiveness will be compared to the (usually supervised) naive Bayes algorithm [16] which is generally considered a suitable baseline method for classifying documents into one of two given categories such as 'ham' or 'spam' when dealing with e-mails. As the introduced algorithms' aim is to generate a dichotomy of given text documents, their comparison with the well-known naive Bayes approach is therefore reasonable. While generally accepted evaluation metrics such as entropy and purity [17] will be used to estimate the general quality of the clustering solutions, for one dataset and for all three clustering/classification approaches, the parameters of the resulting clusters will be discussed in detail. In doing so, the effects of the algorithms' properties will be explained and their suitability for the task at hand evaluated.

For these experiments, three datasets consisting of online news articles from the German newspaper "Süddeutsche Zeitung" from the months September, October and November of 2015 have been compiled:

1. Dataset 2.1 consists of 100 articles covering the topics 'car' (34), 'money' (33) and 'sports' (33).

2. Dataset 2.2 covers 100 articles assigned to the categories 'digital' (33), 'culture' (32) and 'economy' (35).

3. Dataset 2.3 contains 100 articles on the topics 'car' (25), 'money' (25), 'politics' (25) and 'sports' (25).

Although three of the four topical categories of dataset 2.1 are found in dataset 2.3 as well, a different document composition has been chosen. As in the preliminary set of experiments, the articles' categories have been manually set by their respective authors and can be found in the articles' filenames as tags. On the basis of these assignments, it is possible to apply the well-known evaluation metrics entropy and purity. However, these assignments will of course not be taken into account by the clustering/classification algorithms when the actual clustering is carried out.

4.1 Exp. 1: Clustering using Antipodean Documents

In the first experiment discussed herein, algorithm II is iteratively applied in two rounds on the clustering dataset 2.1: first on the dataset's initial cluster (root) and then again on its two subclusters (child clusters) created. This way, a cluster hierarchy (binary tree of clusters) is obtained. In Figure 3, this hierarchy is shown. The clusters contain values for the following parameters (if they have been calculated, otherwise N/A):

1. the number of documents/articles in the cluster

2. the number of terms in the cluster

3. the cluster radius (while relying on the respective co-occurrence graph G of the father cluster)

4. the fraction of topics in the set of documents

5. the centroid term of the first document (antipodean document) in the cluster

Moreover, the distance between the centroid terms of the antipodean documents in two child clusters and the intersection of two child clusters (number of terms they have in common) are given in the cluster hierarchy.

It is recognisable that already after the first iteration, the two clusters exhibit dominant topics. E.g. 28 articles with car-related contents are grouped in the first cluster (from left to right). The second cluster contains -in contrast- altogether 51 articles on the topics 'money' and 'sports'. This grouping is not suprising as many of the sports-related articles dealt with the 2015 FIFA corruption case. In the second iteration, this cluster is split again (while using its own documents to construct the co-occurrence graph G) creating one cluster with the dominant topic 'sports' (17 articles) and another one with the dominant topic 'money' (22 documents). Also, the recognisable topical imbalance of the first two clusters is a sign that the clustering solution actually works. The documents with the topics 'money' and 'sports' are semantically closer to each other than to the car-related documents. If this unbalance would not occur, it would mean that the clustering would not work properly.

As it can be seen in the clusters, the centroid terms of the first documents (the antipodean documents) in them are very distant from each other in the respectively used co-occurrence graph G (e.g. the distance of the centroid term 'Rollenprüfstand' to the term 'Radio' is 17,84). This means, that their topics differ as well. It is also logical that the documents in D_x to be assigned to one of the two clusters in each iteration share a topical relatedness with one of the two antipodean documents. In all cases, the cluster radii are much smaller than the distance between the centroid terms of the antipodean documents in G.

The calculated cumulated entropy for the generated four clusters in the second iteration is 0,61 and the cumulated purity of these clusters is 0,70. This reasonable result shows that the basic algorithm II (of course depending on the number and the size of the documents to be grouped) is able to return useful clusters after only a few clustering iterations.

4.2 Exp. 2: Clustering using Centroid Terms

In the second experiment, algorithm IIa is also applied in two rounds on the clustering dataset 2.1: first on the dataset's initial cluster (root) and then again on its two subclusters created. In this case, a cluster hierarchy is obtained as well. Figure 4 presents this hierarchy after two clustering iterations. Its structure follows the one given in the first experiment.

In contrast to the experiment 1, the first cluster contains one more document from the category 'car' and no document from the category 'sports'. The second cluster contains with 62 documents from the categories 'money' and 'sports' 11 documents more than the second cluster from experiment 1. Here, the topical imbalance of the first two clusters is recognisable, too. Also, the clusters generated in the second iteration exhibit a clear topical orientation. The cluster radii are much smaller than the distance between the centroid terms of the antipodean documents in G, too.

The calculated cumulated entropy for the generated four clusters in the second iteration is 0,55 and the cumulated purity of these clusters is 0,77. This result is even better than the one from algorithm II and shows that it is sensible

to not solely base the classification decision on the distance of a document's centroid term to one of the antipodean documents' centroid terms, but to take into account its average distance to all of the already added centroid terms found in one of the two clusters, too.

4.3 Exp. 3: Clustering using Naive Bayes

In experiment 3, the well-known naive Bayes algorithm [16] is applied to iteratively and hierarchically group the documents of dataset 2.1. This supervised algorithm is usually applied to classify documents into two categories such as 'ham' or 'spam' when incoming e-mails need to be filtered. It is often regarded as a baseline method when comparing classification techniques. Therefore, it makes sense to also use this algorithm to classify the documents of the mentioned datasets into one of two groups in each classification step. However, in order to correctly classify unseen documents, a classifier using the naive Bayes algorithm needs to be trained with particular sets of documents from the categories of interest. Small sets are usually sufficient. For this purpose, the automatically determined antipodean documents are used as this training set. Based on the features (terms) in these documents, the naive Bayes algorithm can determine the probabilities of whether a document from the set D_x rather belongs to either one cluster or the other. A newly classified document (its features) is then automatically taken into account to train the classifier for the next documents from D_x to be classified/clustered.

Here, however, a problem arises (especially when only a few documents from D_x have been classified so far): it might be that the majority of those documents will be assigned to only one category, which is undesirable. The classification probabilities might be shifted in favour of exactly this category and new documents might be wrongly classified, too. Therefore, a preselection of the next document to be classified is applied before the actual naive Bayes classification is executed. In this step, the desired category is determined in an alternating way and the best suited document from the (remaining) non-empty set D_x is selected based on the shortest distance to the centroid term of the antipodean document of this specific category. The aim of this approach is that the two clusters grow at almost the same rate. This approach resembles the human (i.e. manual and supervised) (pre-)classification of documents, before the classifier is trained based on their features. As an example, an e-mail can usually be instantly and without much effort categorised by a human reader. The same principle is applied here, only that in the case at hand this preselection is carried out fully automatically. In this setting, the naive Bayes algorithm is applied in an unsupervised way. Although the mentioned constant growth of the two clusters is practically unreachable due to the datasets' characteristics, this preselection is however sensible as described before.

Also in this experiment, a cluster hierarchy is obtained. Figure 5 presents this hierarchy after two clustering iterations in the same fashion as in the previous experiments. In contrast to the first and second experiment, the first generated cluster contains almost all documents from the root cluster. Its second child cluster contains with 77 documents covering all topics still a large fraction of all given documents. The remaining clusters are practically unusable due to their topical mixture and the low number of documents (one cluster is actually made up of the initial antipodean document) they contain. Also, no dominant topics can be identified when analysing the clusters in the hierarchy.

This bad result is also reflected in the values of the cumulated entropy and purity. The calculated entropy for the generated four clusters in the second iteration is 0,98 and the purity of these clusters is 0,39. The reasons for this result can be found in both the dataset used and the working principle of the naive Bayes classifier. First, the dataset's documents have many terms in common. The (sub)topic 'money' is found in the documents of the categories 'sports' and 'car', too. For instance, many car-related documents dealt with the car emissions scandal and financial penalties for the car companies involved and a lot of sports-related documents covered the 2015 FIFA corruption case. Second, based on just two training documents (the antipodean documents), the naive Bayes algorithm was not able to separate the given dataset properly. As this algorithm does not make use of the term relations in the respective co-occurrence graphs G, only the features (terms) in the documents, which are supposed to be independent, determine the classification probabilities.

In order to improve the algorithm's clustering performance, an idea was to increase the training set (although the naive Bayes algorithm can be trained on small sets, too). For this purpose and in a small modification of the presented setting, not only the antipodean documents have been initially put into the respective two child clusters, but their two closest documents (in terms of their centroid terms' distance in the co-occurrence graph G), too. This way, any child cluster initially contained three documents. However, even with this modification, the quality of the clusters did not increase.

4.4 General Evaluation and Discussion

The algorithms II, IIa and the naive Bayes algorithm have been applied on the datasets 2.2 and 2.3, too. Table 2 presents the values for the cumulated entropy (a value near 0 is wished for) and cumulated purity (a value near 1 is desired) for all algorithm/dataset combinations. In all these cases, it is to be expected that a topical imbalance occurs in the clusters as seen in the experiments 1 and 2. As datasets 2.1 and 2.2 cover three topical categories, it can therefore be assumed that after already two clustering iterations, a clear topical separation should be visible (in case the algorithms work properly). For dataset 2.3, however, a clear topical separation should be visible after at most three iterations as it contains documents of four topical categories. Therefore, for datasets 2.1 and 2.2, the cumulated entropy and purity values have been computed after two clustering iterations, whereas for dataset 2.3, these values have been determined after three iterations.

Table 2: Entropy and purity of the obtained clusters

Entropy (E)/ Purity (P):	Alg. II	Alg. IIa	Naive Bayes
Dataset 2.1 (100 doc. / 3 topics)	E=0,61 P=0,70	**E=0,55** **P=0,77**	E=0,98 P=0,39
Dataset 2.2 (100 doc. / 3 topics)	E=0,65 P=0,69	**E=0,46** **P=0,82**	E=0,97 P=0,37
Dataset 2.3 (100 doc. / 4 topics)	E=0,41 P=0,76	**E=0,35** **P=0,82**	E=0,49 P=0,62

As previously described, algorithm IIa performs best on dataset 2.1. For the datasets 2 and 3, this picture does

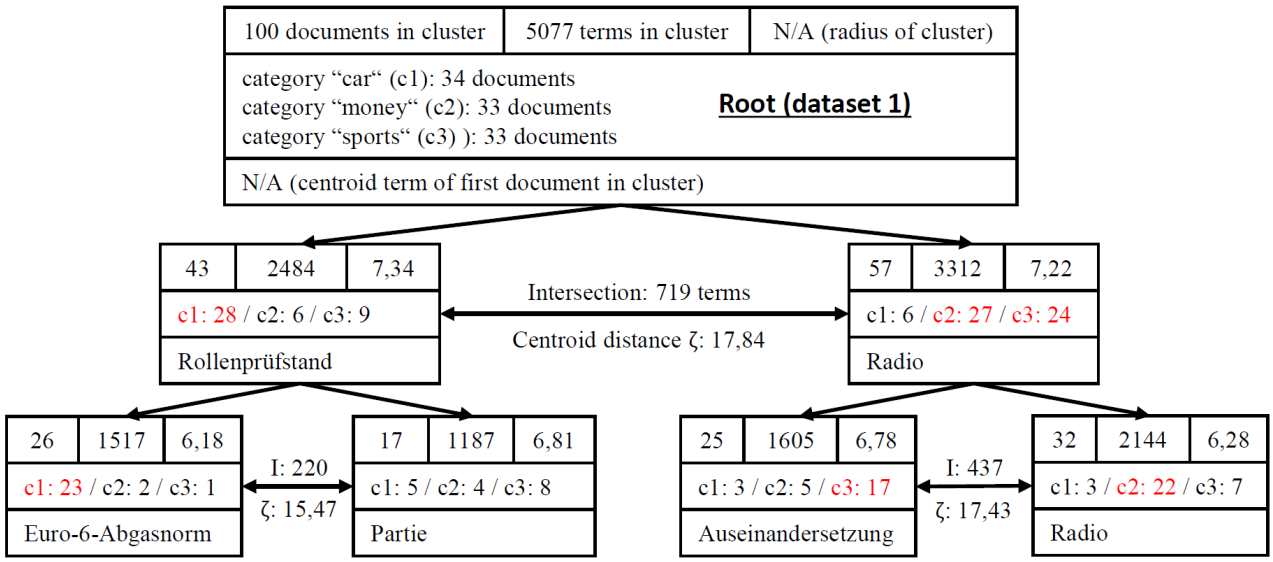

Figure 3: Basic approach: clustering using antipodes

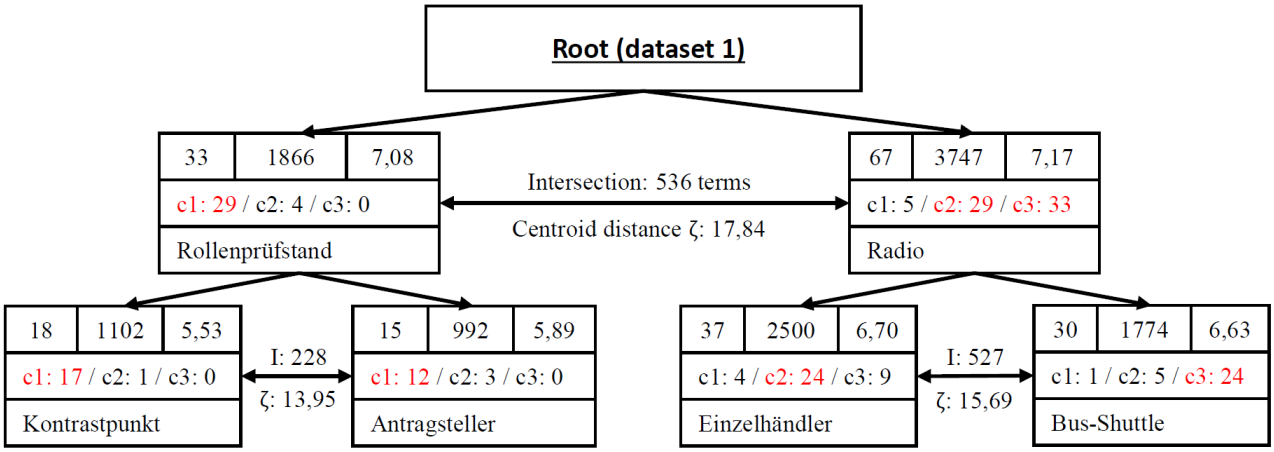

Figure 4: Centroid-based clustering

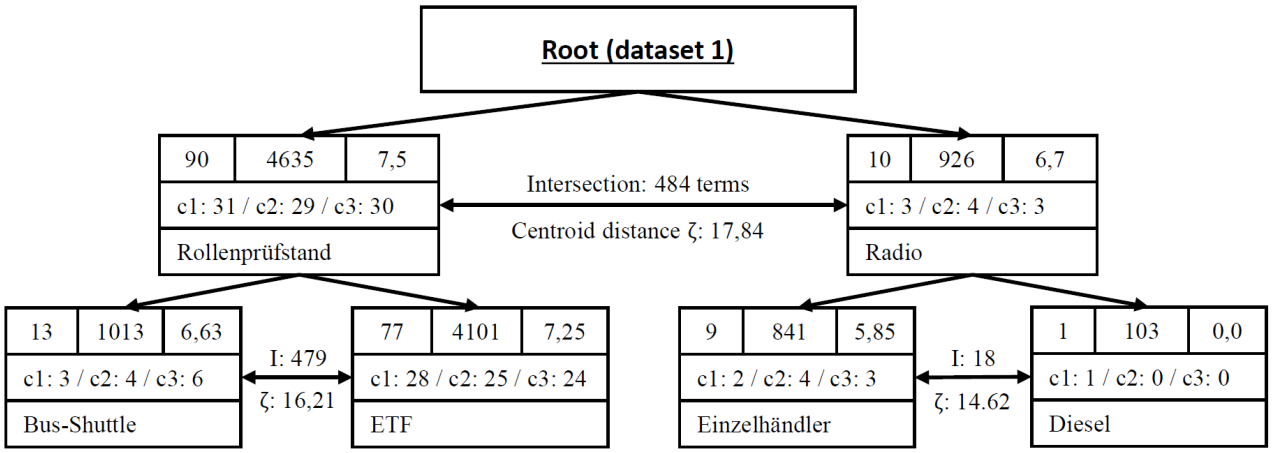

Figure 5: Clustering using a Naive Bayes classifier

not change as well. Also, in all cases, algorithm II achieved better entropy and purity values than the naive Bayes algorithm. This shows that the introduced algorithms II and IIa can clearly outperform the naive Bayes algorithm, even after a small number of cluster iterations.

In this regard, it is sensible to ask, when to start and stop the clustering process? In case of the librarian, the process will be started only once when the node's 'bookshelf' is full and is not carried out again afterwards by this node (but its child nodes). This parameter therefore depends on the hardware resources available at each node running the librarian. In future articles, this hardware-dependent parameter estimation will be thematised.

The main goal of the herein presented experiments and results was to demonstrate that centroid terms of text documents and their distances are well-suited for clustering purposes. It could be seen in the simulations that the decision base (the co-occurrence graph G) to put a document in either one of the two possible clusters is reduced in size in each clustering iteration. This means that with a growing number of iterations the probability to make the right decision (correctly order the stored documents) is reduced, too. Even so, it was shown that hierarchic clustering based on centroid distances works satisfying.

In case of the presented librarian (the real implementation following algorithm I), however, the decision base would not shrink as the clustering process would be carried out once after a node's local 'bookshelf'/memory is full. The local co-occurrence graph G is then handed down to its two child nodes and is at this place continuously specialised by incoming (more topically focused) documents. At the same time, the area of the original graph G for which a child node is primarily responsible (represented by the terms of its local document repository) is reduced in size. This decision area is therefore more specialised than the whole graph of the father node. Thus, child nodes can make sharp classification decisions on incoming documents of their topical specialisation, yet are able to classify documents of a different topical orientation correctly. After the generation of child nodes, the librarian only acts as a semantic router for incoming document links and search queries as described in step 6 of algorithm I.

5. CONCLUSION

The classic concept of the (human) librarian has been analysed and generalised in algorithmic form for its use in the World Wide Web. Its decentralised, P2P- and structure-based approach to manage web documents is able to classify, link and return 100% recent results and avoids crawling as well as copying of the web into reverse index files. The introduced concept comprises a new hierarchic text clustering method that computes semantic distances of documents using their centroid terms. The method generates clusters of comparably high quality and enables the classification of both text documents and keyword-based queries in the same manner. For the librarian's proper technical realisation, an innovative approach using modified webservers has already been introduced in [1]. Future works will deal with further improvements of the clustering approach as well as with suitable structure-building methods to overcome problems caused by the low connectivity and the major role of root nodes of the generated tree-like node and document hierarchy.

6. REFERENCES

[1] R. Eberhardt, M. Kubek, and H. Unger. Why google isn¨t the future. Really not. In *Autonomous Systems 2015*, pages 268–281. VDI Verlag, 2015.

[2] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, 1999.

[3] A. L. A. P. Committee. Presidential committee on information literacy: Final report. Final report, American Library Association, 1989.

[4] H. Wachsmuth. *Text Analysis Pipelines - Towards Ad-hoc Large-Scale Text Mining*, volume 9383 of *Lecture Notes in Computer Science*. Springer, 2015.

[5] R. A. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

[6] G. Heyer, U. Quasthoff, and T. Wittig. *Text Mining: Wissensrohstoff Text – Konzepte, Algorithmen, Ergebnisse*. IT lernen. W3L-Verlag, Herdecke, 2008.

[7] M. Kubek and H. Unger. Search word extraction using extended pagerank calculations. In *Autonomous Systems: Developments and Trends*, pages 325–337. Springer Berlin Heidelberg, 2012.

[8] M. M. Kubek, H. Unger, and J. Dusik. *Correlating Words - Approaches and Applications*, pages 27–38. Springer International Publishing, Cham, 2015.

[9] P. Roget and S. Lloyd. *Roget's thesaurus of English words and phrases*. Longman, 1982.

[10] T. Weale, C. Brew, and E. Fosler-Lussier. Using the wiktionary graph structure for synonym detection. In *Proceedings of the 2009 Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources*, pages 28–31. Association for Computational Linguistics, 2009.

[11] A. Budanitsky and G. Hirst. Evaluating wordnet-based measures of lexical semantic relatedness. *Comput. Linguist.*, 32(1):13–47, 2006.

[12] P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'95, pages 448–453, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc., 1995.

[13] G. A. Miller. WordNet: A Lexical Database for English. *Commun. ACM*, 38(11):39–41, 1995.

[14] M. M. Kubek and H. Unger. Centroid terms as text representatives. In *Proceedings of the 2016 ACM Symposium on Document Engineering*, DocEng '16, pages 99–102, New York, NY, USA, ACM., 2016.

[15] J. Macqueen. Some methods for classification and analysis of multivariate observations. In *In 5-th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

[16] C. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. An Introduction to Information Retrieval. Cambridge University Press, 2008.

[17] Y. Zhao and G. Karypis. Criterion functions for document clustering: Experiments and analysis. Technical report, University of Minnesota, 2001.