Fortschritt-Berichte VDI

VDI

Reihe 10

Informatik/
Kommunikation

Herwig Unger,
Mario M. Kubek

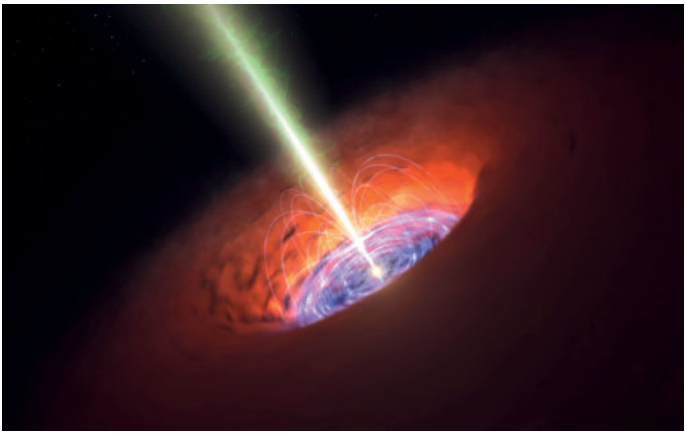
Nr. 863

Theory and Application of Text-representing Centroids

FernUniversität in Hagen
Schriften zur Informations- und Kommunikationstechnik

Theory and Application of Text-representing Centroids

Herwig Unger and Mario M. Kubek



Email: kn.wissenschaftler@fernuni-hagen.de
Website: https://www.fernuni-hagen.de/kn/

Fortschritt-Berichte VDI

Reihe 10

Informatik/
Kommunikation

Herwig Unger,
Mario M. Kubek

Nr. 863

Theory and Application of Text-representing Centroids

FernUniversität in Hagen
Schriften zur Informations- und Kommunikationstechnik

Herwig Unger, Mario M. Kubek
Theory and Application of Text-representing Centroids
Fortschr.-Ber. VDI Reihe 10 Nr. 863. Düsseldorf: VDI Verlag 2019.
152 Seiten, 49 Bilder, 10 Tabellen.
ISBN 978-3-18-386310-5, ISSN 0178-9627,
€ 57,00/VDI-Mitgliederpreis € 51,30.

Keywords: Text Processing – Text Centroid – Co-occurrence Graph – Spreading Activation – Text Categorisation – Librarian of the Web – P2P-system – Decentralised Search – WebEngine – Web Search Engine

Centroid terms are single, descriptive words that semantically and topically characterise text documents and thus can act as their very compact representation in automated text processing tasks that strongly rely on the semantic similarity of texts. Algorithms to classify and cluster them make use of this information. In this book, the novel, brain- and physicsinspired concept of centroid terms is introduced and deeply discussed. Furthermore, their unique properties and practical usage in major natural language processing and text mining tasks are covered. In this regard, a new graph-based method for their fast calculation is presented as well. In contrast to methods relying on the bag-of-words model, the derived centroid distance measure can uncover a topical relationship between texts even when their wording differs. As centroid terms can also represent short texts, the presented first fully integrated, P2P-based web search engine, called "WebEngine", therefore makes heavy use of centroid terms when interpreting queries and forwarding them to peers with matching documents, represented by their own centroid terms.

Bibliographische Information der Deutschen Bibliothek
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliographie; detaillierte bibliographische Daten sind im Internet unter www.dnb.de abrufbar.

Bibliographic information published by the Deutsche Bibliothek
(German National Library)
The Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliographie (German National Bibliography); detailed bibliographic data is available via Internet at www.dnb.de.

Schriften zur Informations- und Kommunikationstechnik
Herausgeber:
Wolfgang A. Halang, ehemaliger Lehrstuhl für Informationstechnik
Herwig Unger, Lehrstuhl für Kommunikationstechnik
FernUniversität in Hagen

Als Manuskript gedruckt. Printed in Germany.
ISSN 0178-9627
ISBN 978-3-18-386310-5

Contents

Is a 'Librarian of the Web' really needed?

Herwig Unger

Chair of Communication Networks, University of Hagen, Germany

During their entire existence, humans, and in particular scientists, were both: hunters for new knowledge and wisdom making their lives more efficient as well as collectors conserving their knowledge for themselves and their successors. Thus, the first libraries were established in ancient times[1], before the Common Era. The librarians' task was not only to act as intermediaries between knowledge and users, but also to acquire, manage and maintain the media knowledge was written on.

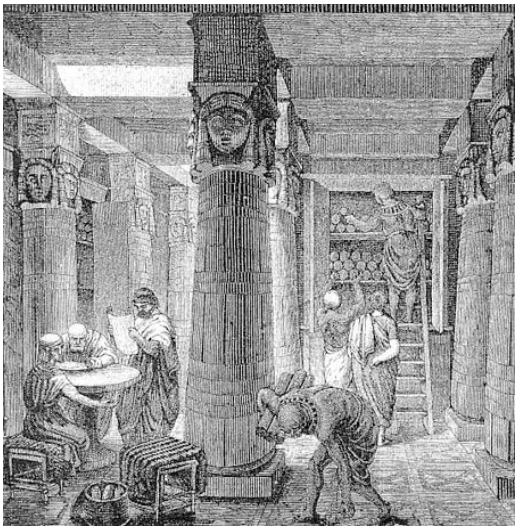


Fig. 1: The Great Library of Alexandria

[1]Fig. 1 has been reused and was originally published under: `https://commons.wikimedia.org/wiki/File:The_Great_Library_of_Alexandria,_O._Von_Corven,_19th_century.jpg`, original author: Igor Merit Santos, Creative Commons licence: CC-BY-SA-4.0

While the first scientists like ARCHIMEDES still might have been able to oversee the complete knowledge of their times, the amount of available materials has exploded after Gutenberg had invented letterpress. Categorisations, known in philosophy since at least PLATO as an approach to group objects based on similarities, allowed to cope with this problem in a systematic way. Books with similar content where placed adjacent in a shelf, while shelves with books of related content were located in close proximity and so on. The resulting, strictly hierarchical classification corresponds to the comfortable top-down or bottom-up thinking as prevailing in many sciences, but renders it almost impossible to find similarities, and to use or combine information from different areas of science in an interdisciplinary manner.

Finding good identifiers or names for all categories in a complex abstraction process is an art strictly depending on the background and intentions of the executing librarian: without any communication no two librarians would probably categorise an identical set of books by the same terms. Consequently, no classification, taxonomy, ontology or categorisation of human knowledge being nearly complete and accepted worldwide is available to this day.

In the 1990s, the introduction of the Internet and the World Wide Web gave their users the possibility to formulate their own content in the hypertext mark-up language and make it accessible using simple browsers. With no connection between a content and its place of appearance, early predecessors of search engines (like Archie [1]) just tried to compile locations with content offered. Shortly later, two concepts competed: the first one understood the WWW as a library and tried to develop the idea of a catalogue meeting the needs of the new media, while the second one was inspired by the idea of indexing the content of websites in big databases.

Owing to ever-increasing computer power, the brute-force approach of indexing prevailed and, therefore, currently all major search engines periodically read the majority of web pages and copy their content into big index files to provide correspondences between (indexed) terms and Uniform Resource Locators (URLs, i.e. addresses in the WWW) of web pages containing those terms. In other words, for any word of every language a database is built with the addresses of all other webpages containing the respective term. Thus, for a multi-keyword search, the intersection of all databases corresponding to each single query term must be determined with huge computational effort.

The authors of this booklet believe that returning to the first approach mentioned above and employing it in combination with effective document cate-

gorisation may be more efficient for a future Internet. To this end, a categorisation (not necessarily a strictly hierarchical taxonomy) had to be found, which is (differing from ontologies) not necessarily connected to human semantics and can be determined in a fully formal manner. Such a categorisation was found in form of Text-representing Centroids (TRCs or "Bedeutungsschwerpunkte" in German) as introduced in this booklet.

It was inspired by some considerations on natural language processing, physics as well as neuroscience. Comparing different learning approaches, the application of complex backpropagation or other difficult and computationally expensive methods was never found in nature so far.

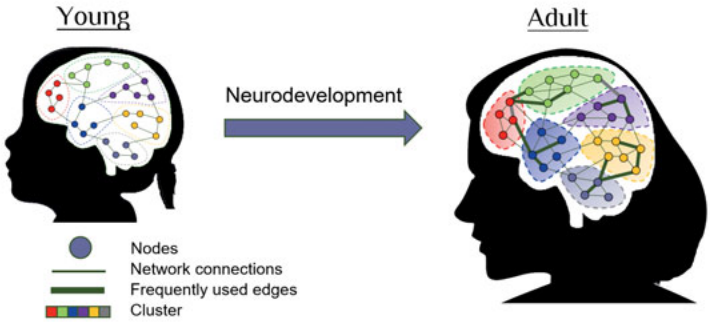


Fig. 2: The modular segregation process in the brain while learning

Fig. 2 is inspired by a neuroscience publication [2] on modular segregation and shows a somehow inspiring comparison. The empty brain first learns words, preferably those occurring often. At the same time, connections between words are added if they often occur together in certain contexts. As a result it can be observed that a few words are connected with many others and some central, strongly connected positions in the brain's word network are obtained.

Around those words, usually the formation of clusters, i.e. regions of the network with a significantly higher connectivity, can be observed. If those clusters

are interpreted as categories with their central terms as identifiers, categorisation can be explained as an automatic, simple (statistic) process. This neuroscience scenario has a direct correspondence in natural language processing, where co-occurrence graphs are a commonly used model and tool.

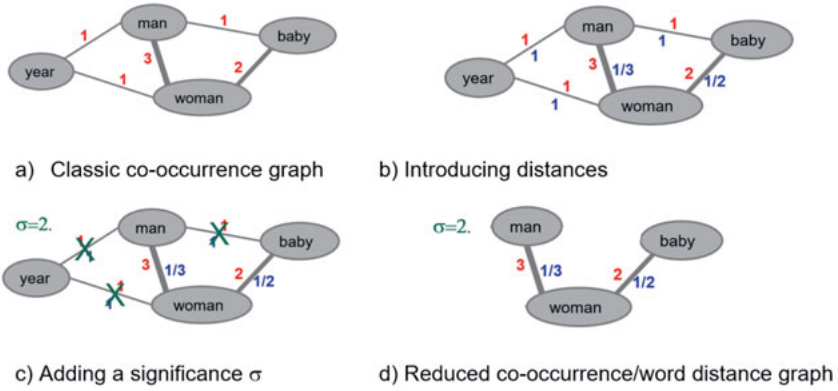


Fig. 3: Co-occurrence and word distance graph obtained from the example corpus

Given the following simple example corpus containing the four sentences

1. ***A man met a woman.***
2. ***A year later, the man and the woman got married.***
3. ***Then, the man and the woman got a baby.***
4. ***The woman takes care of the baby.***

the co-occurrence graph in Fig. 3a) may be obtained, if the text's words (nouns only) are considered to be the graph's nodes, while an edge is added whenever two words occur together in a sentence. Two words appear to be close (or have some similar meaning/context), if they occur often in a sentence. Thus, it makes sense to define a distance between words by the reciprocal of their frequency of co-occurrence. Using it, from the co-occurrence graph an isomorphic word-distance graph can be obtained as shown in Fig. 3b). If only significant co-occurrences are considered as to be seen in Fig. 3c), i.e. co-occurrences whose number exceeds a given threshold σ, the respective co-occurrence and

word-distance graphs can often be reduced significantly. Fig. 3d) shows such a reduced state.

Labelling the words of a query or document in a (bigger) word-distance graph and comparing the obtained configuration in the graph's discrete neighbourhood environment with the physical analogue of the centre of mass in continuous three-dimensional space, the definition of TRCs is quite obvious, as given as a starting point of further considerations in the subsequent Chapter 2 of this booklet. It is followed by some considerations on fast methods to calculate TRCs (Chapter 3) and a first discussion of their properties in Chapter 4. Any distance or similarity metric between (text-) objects may also be used in group-building or clustering algorithms as derived in Chapter 5 and as well as for document categorisation and search. A concept for the latter is presented in Chapter 6. Ring-like structures (similar to the CHORD system [3]) turn out to be adequate for storing the <category, address> relation as they do not restrict the users to solely hierarchic, tree-like structures. Matching them, a suitable self-organising distributed associative memory is introduced in Chapter 7 and employed in Chapter 8 as the central part of the first fully integrated, decentralised web search engine, our so-called WebEngine. A generalisation of the concept of TRCs is presented and thoroughly discussed in Chapter 9 which completes this booklet.

References

[1] Li, W.: The First Search Engine, Archie, In: *A Brief History of Search Engines*, 2002 `https://web.archive.org/web/20070621141150/http://isrl.uiuc.edu/~chip/projects/timeline/1990archie.htm`

[2] Baum, G. L. et al.: Modular Segregation of Structural Brain Networks Supports the Development of Executive Function in Youth, In: *Current Biology*, Vol. 27, Issue 11, pp. 1561—1572, Elsevier Ltd., 2017

[3] Stoica, I. et al.: Chord: A scalable peer-to-peer lookup service for internet applications, In: *ACM SIGCOMM Computer Communication Review*, 31(4):149, 2001

Centroid Terms as Text Representatives

Mario M. Kubek and Herwig Unger

Chair of Communication Networks, University of Hagen, Germany

Abstract: The calculation of semantic similarities between text documents plays an important role in automatic text processing. For example, algorithms to topically cluster and classify texts heavily rely on this information. Standard methods for doing so are usually based on the bag-of-words model and thus return only rough estimations regarding the relatedness of texts. Moreover, they are unable to find generalising terms or abstractions describing the textual contents. Therefore, a new graph-based method to determine centroid terms as text representatives will be introduced. It is shown, that – among further application scenarios – this method is able to compute the similarity of texts even if they have no terms in common. In first experiments, its results and advantages will be discussed in detail.

1 Motivation

After only a few lines of reading, a human reader is able to determine which category of texts and which abstract topic category a given document belongs to. This is a strong demonstration of how well and fast the human brain, especially the human cortex, can process and interpret data. It is able to not only understand the meaning of single words – as representations of real-world entities – but a certain composition of them [1], too.

In many text mining applications, the topical grouping of texts and terms/words contained in them are common tasks. In order to group semantically related terms, unsupervised topic modeling techniques such as LDA [2] have been successfully applied. This technique tries to infer word clusters from a set of documents based on the assumption that words from the same topic are likely to appear next to each other and therefore share a related meaning. Here, deep and computationally expensive (hyper)parameter estimations are carried out and for each word, the probability to belong to specific topic is computed in order to create those constructions. The graph-based Chinese Whispers algorithm [3] is another interesting clustering technique that can be used in the field

of natural language problems, especially to semantically group terms. It is usually applied on undirected semantic graphs that contain statistically significant term relations found in texts.

Also, it is usual to apply the k-means algorithm [4] to group terms. For this purpose, it is necessary to determine their semantic distance. Here, several methods can be applied. The frequency of the co-occurrence of two terms in close proximity (in a window of n words or on sentence level) is a first indication for their semantic distance. Terms that frequently co-occur together are usually semantically related. Several graph-based distance measures [7, 8] consult manually created semantic networks such as WordNet [5], 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 closest semantic relationship between two given input terms or concepts. It is also common to measure the similarity of term contexts [6] that contain terms that often co-occur with the ones in question. Technically, these contexts are realised as term vectors following the bag-of-words model.

The same approach is applied when the semantic similarity or distance of any two documents should be determined. Here, the term vectors to be compared contain the texts' characterising terms and their score (typically, a TF-IDF-based statistic [9] is used) as a measure for their importance. The similarity of two term vectors can be determined using the cosine similarity measure or by calculating the overlap of term vectors, e.g. using the Dice coefficient [10]. The commonly used Euclidean distance and the Manhattan distance are further examples to measure the closeness of term vectors at low computational costs.

However, in some cases, these measures do not work correctly (with respect to human judgement), mostly if different people write about the same topic but are using a completely different vocabulary for doing so. The reason for this circumstance can be seen in the isolated view of the words found in documents to be compared without including any relation to the vocabulary of other, context-related documents. Moreover, short texts as often found in posts in online social networks or short (web) search queries with a low number of descriptive terms can therefore often not be correctly classified or disambiguated. Another disadvantage is that these measures cannot find abstractions or generalising terms by just analysing the textual data provided. For this purpose, static

lexical databases such as WordNet [5] must be consulted as a reference. Despite their usefulness, these resources are – in contrast to the human brain – not able to learn about new concepts and their relationships.

In order to address these problems, this article presents a new graph-based approach to determine centroid terms of text documents. It is shown that those terms can actually represent text documents in automatic text processing, e.g. to determine their semantic distances. In the next section, the fundamentals of this method are presented. Afterwards, section 3 describes its mathematical and technical details. Section 4 proves the validity of this approach by explaining the results of first experiments. In section 5, the method's working principles and advantages are discussed. Section 6 presents numerous application scenarios for it in the fields of text mining and information retrieval while also elaborating on technological aspects of its practical implementation. Section 7 summarises the article and suggests further application fields of the introduced method.

2 Fundamentals

For the approach presented herein, co-occurrences and co-occurrence graphs are the basic means to obtain more detailed information about text documents than term frequency vectors etc. could ever offer. The reason for this decision is that co-occurrence graphs are able to accumulate a certain knowledge obtained from a few selected or all documents of a text corpus while (at least to some extent) maintaining the semantic connection of terms found in them.

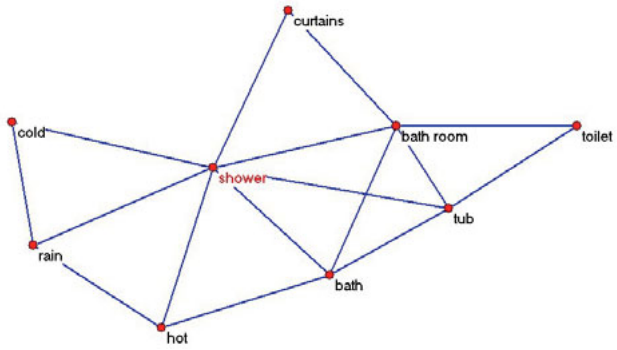


Fig. 1: A co-occurrence graph for the word 'shower'

Two words w_i and w_j are called *co-occurrents*, if they appear together in close proximity in a document D. The most prominent kinds of such co-occurrences are word pairs that appear as immediate neighbours or together in a sentence. A *co-occurrence graph* $G = (W, E)$ may be obtained, if all words of a document or set of documents W 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. A weight function $g((w_a, w_b))$ indicates, how significant the respective co-occurrence is in a document. If the significance value is greater than a pre-set threshold, the co-occurrence can be regarded as significant and a semantic relation between the words involved can often be derived from it. Commonly used significance measures are the Dice coefficient [10], the mutual information measure [11], the Poisson collocation measure [12] and the log-likelihood ratio [13].

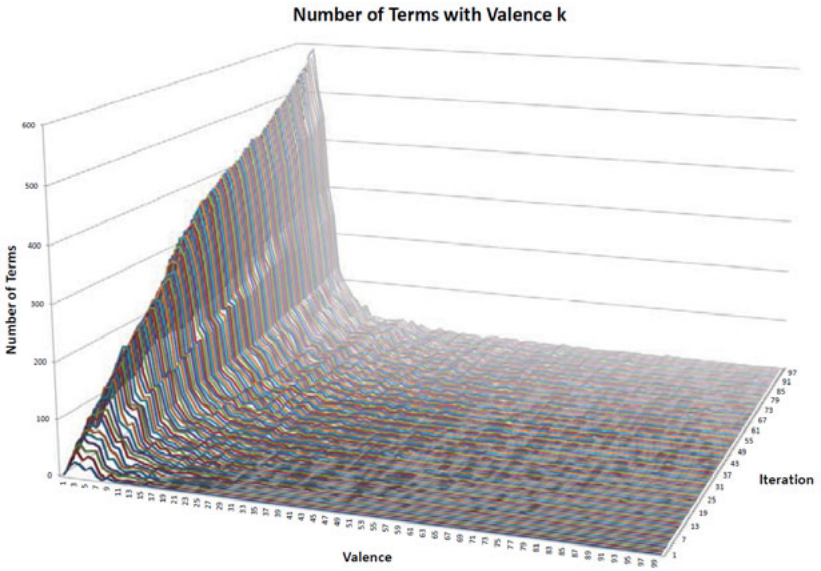


Fig. 2: Distribution of out-degrees in a co-occurrence graph over time

A co-occurrence graph – similarly to the knowledge in the human brain – may be built step by step over a long time taking one document after another into consideration. From the literature [6] and own experiments (see Fig. 2), it is

known that the out-degrees of nodes in co-occurrence graphs follow a power-law distribution and the whole graph exhibit small-world properties with a high clustering coefficient as well as a short average path length between any two nodes. This way, a co-occurence graph's structure also reflects the organisation of human lexical knowledge.

The use of the immediate neighbourhood of nodes in a co-occurrence graph has been widely considered in literature, e.g. to cluster terms [3] and to determine the global context (vector) of terms in order to evaluate their similarity [6] or to derive paradigmatic relations between them [14]. In the authors' view, indirect neighbourhoods of terms in co-occurrence graphs (nodes that can be reached only using two or more edges from a node of interest) and the respective paths with a length ≥ 2 should be considered as well as indirectly reachable nodes may still be of topical relevance, especially when the co-occurence graph is large. The benefit of using such nodes/terms in co-occurrence graphs has already been shown by the authors for the expansion of web search queries using a spreading activation technique applied on local and user-defined corpora [15]. The precision of web search results can be noticably improved when taking those terms into account, too.

The field of application of indirect term neighbourhoods in co-occurrence graphs shall be extended in the next section by introducing an approach to determine centroid terms of text documents that can act as their representatives in further text processing tasks. These centroid terms can be regarded as the texts' topical centers of interest (a notion normally used to describe the part of a picture that attracts the eye and mind) that the authors' thoughts revolve around.

3 Finding Centroid Terms

In physics, complex bodies consisting of several single mass points are usually represented and considered by their so-called center of mass, as seen in Fig. 3. The distribution of mass is balanced around this center and the average of the weighted coordinates of the distributed mass defines its coordinates and therefore its position.

For discrete systems, i.e. systems consisting of n single mass points $m_1, m_2, \ldots, m_i$ in a $3D$-space at positions $\vec{r}_1, \vec{r}_2, \ldots, \vec{r}_i$, the center of mass $\vec{r}_s$ can be found by

$$\vec{r}_s = \frac{1}{M} \sum_{i=1}^{n} m_i \vec{r}_i, \tag{1}$$

whereby

$$M = \sum_{i=1}^{n} m_i. \tag{2}$$

Usually, this model simplifies calculations with complex bodies in mechanics by representing the whole system by a single mass at the position of the center of mass. Exactly the same problem exists in automatic text processing: a whole text shall be represented or classified by one or a few single, descriptive terms which must be found.

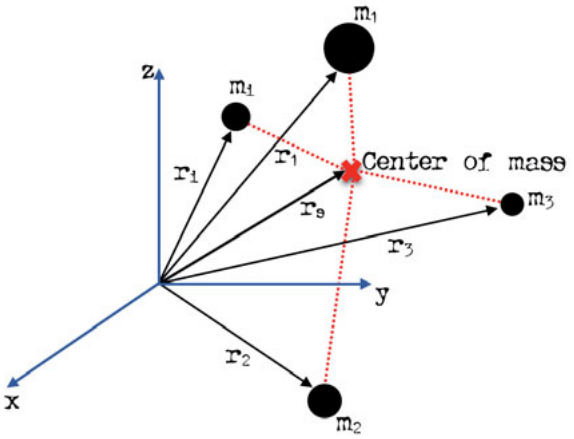


Fig. 3: The physical center of mass

To adapt the situation for this application field, first of all, a *distance d* shall be introduced in a co-occurrence graph G. From literature it is known that two words are semantically close, if $g((w_a, w_b))$ is high, i.e. they often appear together in a sentence or in another predefined window of n words. Consequently, a distance $d(w_a, w_b)$ of two words in G can be defined by

$$d(w_a, w_b) = \frac{1}{g((w_a, w_b))}, \tag{3}$$

if w_a and w_b are co-occurrents. In all other cases (assuming that the co-occurrence graph is connected[1]) there is a shortest path $p = (w_1, w_2), (w_2, w_3), \ldots, (w_k, w_k + 1)$ with $w_1 = w_a$, $w_{k+1} = w_b$ and $w_i, w_{i+1} \in E$ for all $i = 1(1)k$ such that

$$d(w_a, w_b) = \left(\sum_{i=1}^{k} d((w_i, w_{i+1})) \right) = MIN, \tag{4}$$

whereby in case of a partially connected co-occurrence graph $d(w_a, w_b) = \infty$ must be set. Note, that differing from the physical model, there is a distance between any two words but no direction vector, since there is no embedding of the co-occurrence graph in the 2- or 3-dimensional space. Consequently, the impact of a word depends only on its scalar distance.

In continuation of the previous idea, the distance between a given term t and a document D containing N words $w_1, w_2, \ldots, w_N \in D$ that are reachable from t in G can be defined by

$$d(D, t) = \frac{\sum_{i=1}^{N} d(w_i, t)}{N}, \tag{5}$$

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! Also, it might happen in some cases that the minimal distance is not uniquely defined, consequently a text may have more than one centroid term (as long as no other methods decide which one is to use). In order to define the centroid-based distance ζ between any two documents D_1 and D_2, let t_1 be the center term or *centroid term of* D_1 with $d(D_1, t_1) = MIN$. If at the same time t_2 is the centroid term of D_2,

$$\zeta(D_1, D_2) = d(t_1, t_2) \tag{6}$$

can be understood as the semantic distance ζ of the two documents D_1 and D_2. In order to obtain a similarity value instead,

$$\zeta_{sim}(D_1, D_2) = \frac{1}{1 + \zeta(D_1, D_2)} \tag{7}$$

can be applied.

It is another important property of the described distance calculation that documents regardless of their length as well as single words can be assigned a centroid term by one and the same method in a unique manner. The presented

[1] This can be achieved by adding a sufficiently high number of documents to it during its building process.

approach relies on the preferably large co-occurrence graph G as its reference. It may be constructed from any text corpus in any language available or directly from the sets of documents whose semantic distance shall be determined. The usage of external resources such as lexical databases or reference corpora is common in text mining: as an example, the so-called difference analysis [6, 16] which measures the deviation of word frequencies in single texts from their frequencies in general usage (a large topically well-balanced reference corpus is needed for this purpose) is an example for it. The larger the deviation is, the more likely it is that a term or keyword of a single text has been found. Furthermore, the presented distance measure is not only based on a physical analogon and bears (at least to a certain extent) resemblance to the well-known difference analysis as discussed, the measure's approach is brain-inspired, too. Further considerations in this respect will be discussed in section 5.

In the following section, the quality and properties of the centroid terms and the new centroid-based distance measure shall be investigated and discussed.

4 First Experiments

For all of the exemplary experiments (many more have been conducted) discussed herein, 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 [10]. The particularly used sets of documents will be described in the respective subsections[2].

4.1 Centroids of Wikipedia Articles

As the centroid terms are the basic components for the centroid-based distance measure, it is useful to get a first impression of their quality in terms of whether they are actual useful representatives of documents. Table 1 therefore presents the centroid terms of 30 English Wikipedia articles. The corpus used to create the reference co-occurrence graph G consisted of 100 randomly selected articles (including the mentioned 30 ones) from an offline English Wikipedia corpus from `http://www.kiwix.org`. It can be seen that almost all centroids properly represent their respective articles.

[2]Interested readers can download these sets (1.3 MB) from: `http://www.docanalyser.de/cd-corpora.zip`

Table 1: Centroids of 30 Wikipedia articles

Title of Wikipedia Article	Centroid Term
Art competitions at the Olympic Games	sculpture
Tay-Sachs disease	mutation
Pythagoras	Pythagoras
Canberra	Canberra
Eye (cyclone)	storm
Blade Runner	Ridley Scott
CPU cache	cache miss
Rembrandt	Louvre
Common Unix Printing System	filter
Psychology	psychology
Religion	religion
Universe	shape
Mass media	database
Rio de Janeiro	sport
Stroke	blood
Mark Twain	tale
Ludwig van Beethoven	violin
Oxyrhynchus	papyrus
Fermi paradox	civilization
Milk	dairy
Corinthian War	Sparta
Health	fitness
Tourette syndrome	tic
Agriculture	crop
Finland	tourism
Malaria	disease
Fiberglass	fiber
Continent	continent
United States Congress	Senate
Turquoise	turquoise

4.2 Comparing Similarity Measures

In order to evaluate the effectiveness of the new centroid-based distance measure, its results will be presented and compared to those of the cosine similarity

measure while the same 100 online news articles from the German newspaper "Süddeutsche Zeitung" from the months September, October and November of 2015 have been selected (25 articles from each of the four topical categories 'car', 'travel', 'finance' and 'sports' have been randomly chosen) for this purpose. As the cosine similarity measure operates on term vectors, the articles' most important terms along with their scores have been determined using the extended PageRank [17] algorithm which has been applied on their own separate (local) co-occurrence graphs (here, another term weighting scheme such as a TF-IDF variant [9] could have been used as well). The cosine similarity measure has then been applied on all pairs of the term vectors. For each article A, a list of the names of the remaining 99 articles has been generated and arranged in descending order according to their cosine similarity to A. An article's A most similar article can therefore be found at the top of this list.

In order to apply the new centroid distance measure to determine the articles' semantic distance, for each article, its centroid term has been determined with the help of the co-occurrence graph G using formula (5). The pairwise distance between all centroid terms of all articles in G has then been calculated. Additionally, to make the results of the cosine similarity measure and the centroid distance measure comparable, the centroid distance values have been converted into similarity values using formula (7).

The exemplary diagram in Fig. 4 shows for the reference article ("Abgas-Skandal – Schummel-Motor steckt auch in Audi A4 und A6") its similarity to the 50 most similar articles. The cosine similarity measure was used as the reference measure. Therefore, the most similar article received rank 1 using this measure (blue bars). Although the similarity values of the two measures seem uncorrelated, it is recognisable that especially the articles with a low rank (high similarity) according to the cosine similarity measure are generally regarded as similar by the centroid distance measure, too. In case of Fig. 4, the reference article dealt with the car emissions scandal (a heavily dicussed topic in late 2015). The articles at the ranks 3 ("Abgas-Affäre – Volkswagen holt fünf Millionen VWs in die Werkstätten"), 7 ("Diesel von Volkswagen – Was VW-Kunden jetzt wissen müssen") and 12 ("Abgas-Skandal – Was auf VW- und Audi-Kunden zukommt") according to the cosine similarity measure have been considered most similar by the centroid distance measure, all of which were indeed related to the reference article. The strongly related articles at the ranks 1, 4, 6 and 9 have been regarded as similar by the centroid distance measure, too. In many experiments, however, the centroid distance measure considered articles as similar although the cosine similarity measure did not. Here, another implicit yet

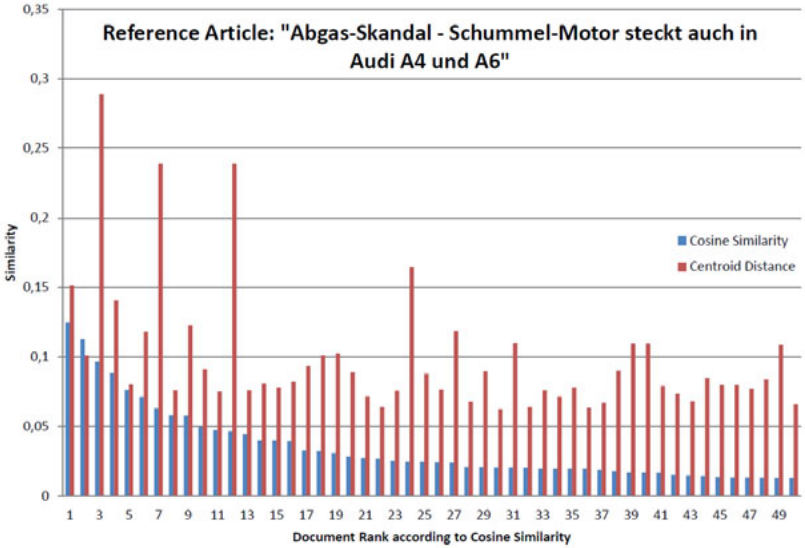


Fig. 4: Cosine similarity vs. centroid distance (topic: car emissions scandal)

important advantage of the new centroid distance measure becomes obvious: two documents can be regarded as similar although their wording differs (the overlap of their term vectors would be small or even empty and the cosine similarity value would be very low or 0). The article at rank 49 ("Jaguar XF im Fahrbericht – Krallen statt Samtpfoten") is an example for such a case. The centroid distance measure uncovered a topical relationship to the reference article, as both texts are car-related and deal with engine types.

Fig. 5 depicts another case of this kind: the article with rank 29 received the highest similarity score from the centroid distance measure. A close examination of this case revealed that the centroids of the reference article ("Deutschland – Ausgebucht") and the article in question ("Briefporto – Post lässt schon mal 70-Cent-Marken drucken") are located close to each other in the reference co-occurrence graph. The reference article's main topic was on financial investments in the German hotel business and the article at rank 29 dealt with postage prices of Deutsche Post AG. Both articles also provided short reports on business-related statistics and strategies.

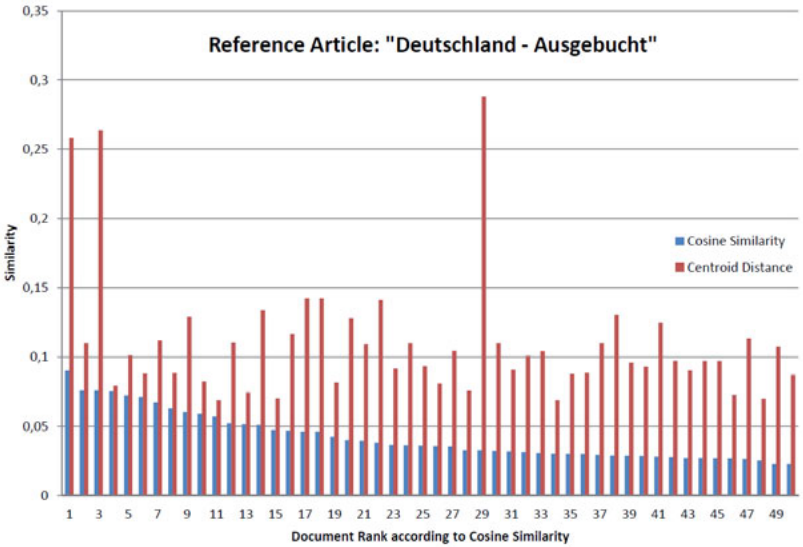


Fig. 5: Cosine similarity vs. centroid distance measure (topic: business-related statistics and strategies)

4.3 Searching for Text Documents

The previous experiments suggest that the centroid distance measure might be applicable to search for text documents, too. In this sense, one might consider a query as a short text document whose centroid term is determined as described before and the k documents whose centroid terms are closest to the query's centroid term are returned as matches. These k nearest neighbours are implicitly ranked by the centroid distance measure, too. The best matching document's centroid term has the lowest distance to the query's centroid term.

The following two tables show for two exemplary queries "VW Audi Abgas" (centroid term: "Seat") and "Fußball Geld Fifa" (centroid term: Affäre) their respective top 10 articles from the German newspaper "Süddeutsche Zeitung" along with their own centroid terms whereby the distances from the queries' centroid terms to all 100 mentioned articles' centroid terms in the co-occurrence graph G have been calculated.

It can be seen that most of the documents can actually satisfy the information need expressed by the queries. This kind of search will, however, not return

Table 2: Top 10 documents for the query "VW Audi Abgas" (Seat)

Filename of News Article	Centroid Term
auto_abgas-skandal-vw-richtet...	Audi
geld_aktien-oeko-fonds-schmeissen-volkswagen-raus	Ethik
auto_bmw-siebener-im-fahrbericht-luxus-laeuft	S-Klasse
auto_abgas-affaere-volkswagen-ruft...	Schadstoffausstoß
auto_abgas-skandal-schummel-motor...	Schadstoffausstoß
geld_briefporto-post-laesst-schon...	Marktanteil
auto_abgas-skandal-was-auf-vw-und-audi...	EA189
auto_abgas-skandal-acht-millionen-vw-autos...	Software
auto_diesel-von-volkswagen-was-vw-kunden...	Motor
auto_abgas-affaere-schmutzige-tricks	Motor

Table 3: Top 10 documents for the query "Fußball Geld Fifa" (Affäre)

Filename of News Article	Centroid Term
sport_affaere-um-wm-mehr-als-nur-ein-fehler	Fifa
sport_angreifer-von-real-madrid-karim-benzema...	Videoaufnahme
sport_affaere-um-wm-vergabe-zwanziger-schiesst...	Zwanziger
sport_affaere-um-fussball-wm-wie-beckenbauers...	Organisationskomitee
sport_affaere-um-wm-zwanziger-es-gab-eine...	Organisationskomitee
sport_affaere-um-wm-vergabe-zwanziger-legt...	Gerichtsverfahren
sport_affaeren-um-wm-vergaben-die-fifa...	Zahlung
sport_affaere-um-wm-netzer-wirft-zwanziger...	Fifa-Funktionär
geld_ehrenamt-fluechtlingshilfe-die-sich...	Sonderausgabe
sport_affaere-um-wm-wie-zwanziger-niersbach...	Präsident

exact matches as known from the popular keyword-based web search. Instead, documents will be returned that are in general topically related to the query. As the query and the documents to be searched for are both represented by just one centroid term, an exact match is not possible when applying this approach.

However, this method can still be of use when a preferably large set of topically matching documents is needed. This kind of recall-oriented search is of interest e.g. for people that want to get an overview of a topic or during patent searches when exact query matches might lower the chance of finding possibly relevant documents that nevertheless do not contain all query terms but related terms instead. A typical precision-oriented search would then be harmful. In these

cases, a search system would first determine documents that contain the input query terms using its inverted index and then rank these documents by computing e.g. their term vectors' cosine similarity with the query. That means that a highly relevant document will contain (almost) all query terms.

In order to optimise both recall using the centroid distance measure and also the precision for the k top documents (precision@k) using a variant of the aforementioned procedure, it might be sensible to calculate a combined rank that factors in the rankings of both approaches. Also, it is imaginable to use the centroid distance measure (as a substitute for the Boolean model) to pre-select those documents that are in a second step ranked according to the cosine similarity measure. Still, other well-known techniques such as expanding queries using highly related and synonymous terms [15] are suitable options to increase recall as well. More experiments in this regard taking all these approaches into account will be conducted.

Also, in the experiments presented herein, mostly topically homogeneous texts (except for the book analyses) have been used in order to demonstrate the validity of the centroid distance measure and the role of centroid terms as text representatives. In future experiments, it will be interesting to evaluate the effectiveness of this approach when it is applied on more topically heterogeneous documents.

4.4 Analysing Full Books

Additionally, full books (not in combination with other texts) have been analysed to determine their centroid terms. In these cases, the books' own co-occurrence graphs G have been used to determine their important terms and to find their respective centroid terms (one for each book). In case of the English King James version of the Holy Bible, the centroid term determined that has the shortest average distance in the book's (almost fully connected) graph G to all other 7211 reachable terms is 'Horeb'. This experiment has been repeated while only using the k (k=25, 50, 75...) most frequent terms for this purpose. Here, besides 'Horeb' and others, the terms 'God' and 'gladness' have been determined as the centroid terms. It is to be pointed out that all of these terms have a low distance to each other in the co-occurrence graph G, meaning they are all good representations of the text no matter what actual centroid term is used for further considerations and applications. This also shows, that it is sufficient to take into account only a few prominent terms of a text in order to determine its cen-

troid term in the co-occurrence graph G while at the same time the algorithm's execution time is drastically lowered.

5 Discussion

The presented approach of using a reference co-occurrence graph to determine the semantic distance of texts is brain-inspired, too. Humans naturally, unconsciously and constantly learn about the entities/objects and their relationships surrounding them and build representaions of these perceptions in form of concept maps as well as their terminologies in their minds. New experiences are automatically and in a fraction of a second matched with those previosuly learned. The same principle is applied when using the centroid distance measure. An incoming text A – regardless of whether it was previously used to construct the co-occurrence graph G or not – whose centroid term shall be found, must at least partially be matched against G. In this sense, G takes on the role of the brain and acts as a global and semantic knowledge base. The only prerequisite is that the graph G must contain enough terms that the incoming text's terms can be matched with. However, it is not necessary to find all of A's terms in G for its at least rough topical classification. The human brain does the same. A non-expert reading an online article about biotechnoloy may not fully understand its terminology, but can at least roughly grasp its content. However, in doing so, this person will gradually learn about the new concepts, a process that is not yet carried out in the herein presented approach. In later publications, the inclusion of this process will be examined.

In order to find proper centroid terms for documents whose topical orientation is unknown, it is important to construct the co-occurrence graph G from a preferably large amount of texts covering a wide range of topics. That is why, in the previous section, the 100 documents to build the respective corpora have been randomly chosen to create G as a topically well-balanced reference. However, the authors assume that topically oriented corpora can be used as a reference when dealing with documents whose terminology and topical orientation is known in advance, too. This way, the quality of the determined centroid terms should increase as they are expected to be better representations for the individual texts' special topical characteristics. Therefore, a more fine-grained automatic classification of a text should be possible. Further experiments are planned to investigate this assumption.

The bag-of-words model that e.g. the cosine similarity measure solely relies on is used by the centroid-based measure as well, but only to the extend that

the entries in the term vectors of documents are used as anchor points in the reference co-occurrence graph G (to 'position' the documents in G) in order to determine their centroid terms. Also, it needs to be pointed out once again that a document's centroid term does not have to occur even once in it. In other words, a centroid term can represent a document, even when it is not mentioned in it.

However, as seen in the experiments, while the cosine similarity measure and the centroid distance measure both often regard especially those documents as similar that actually contain the same terms (their term vectors have a significantly large overlap), one still might argue that both measures can complement each other. The reason for this can be seen in their totally different working principles. While the cosine similarity measure will return a high similarity value for those documents that contain the same terms, the centroid distance measure can uncover a topical relationship between documents even if their wording differs. This is why it might be sensible to combine both approaches in a new measure that factors in the results of both methods. Additional experiments in this regard will be conducted.

Additionally, the herein presented experiments have shown another advantage of the centroid distance measure: its language-independence. It relies on the term relations and term distances in the reference co-occurrence graph G that has been naturally created using text documents of any language.

6 Application Scenarios

The presented centroid distance measure can naturally be applied by text mining algorithms that topically cluster or classify documents. These algorithms make heavy use of similarity and distance measures in order to group semantically similar documents or terms. Here, the new measure can be perfectly applied as an alternative to the well-known measures mentioned above. It will be especially useful, when it comes to grouping topically documents that – despite their topical relatedness – have only a limited amount of terms in common.

However, as shown in the experiments, search applications can make use of this measure, too. Also in this case, documents can be found that do not even share a single query term, yet are highly relevant to the query. Even so, as users are often interested in documents that actually contain the entered query terms but make mistakes in finding the right terms for their information needs, it might be sensible to expand the original query terms with the determined centroid term along with some of its neighbouring terms in the co-occurrence graph G.

Matching documents containing these terms could be ranked in reverse order of their similarity to the expanded query. By using this approach, the search results' recall and precision are both expected to increase as common terms in a topical field (the included centroid term and/or its immediate neighbours) as well as the original query terms are used to find matching documents. This approach will be examined and discussed in further publications.

Interactive search applications such as "DocAnalyser" [18] that aim at helping users to find topically similar and related documents in the World Wide Web could benefit from employing the centroid distance measure, too. Starting with a document of the user's interest, the application could determine the document's centroid term as described before and send this term (to increase the search results' recall) as well as some characteristic terms of the document as an automatically formulated query to a web search engine which will (hopefully) return relevant documents.

From the technological point of view, it becomes obvious that it is necessary to be able to manage large graph structures efficiently and effectively. Graph databases such as Neo4j [19] are specifically designed for this purpose. They are also well-suited to support graph-based text mining algorithms [20]. This kind of databases is not only useful to solely store and query the herein discussed co-occurrence graphs, with the help of the property graph model of these databases, nodes (terms) in co-occurrence graphs can be enriched with additional attributes such as the names of the documents they occur in as well as the number of their occurrences in them, too. Also, the co-occurrence significances can be persistently saved as edge attributes. Graph databases are therefore an urgently necessary tool as a basis for future and scalable text mining solutions.

7 Conclusion

A new physics-inspired method has been introduced to determine centroid terms of particular text documents which are strongly related to them and yet do not need to occur in them. As text representatives, these terms are useful to determine the semantic distance and similarity of text documents. Especially, texts with similar topics yet different descriptive terms, may be classified more precisely than by commonly used measures. As the text length's influence does not play a role in doing so, even short texts or (search) queries may be matched with other texts using the same approach. It may therefore be applied in future (decentralised) search engines and text clustering solutions.

References

[1] Hawkins, J., Blakeslee, S.: *On Intelligence*, Times Books, New York, NY, USA, 2004

[2] Blei, D. M., Ng, A. Y., Jordan, M. I.: Latent Dirichlet Allocation, In: *The Journal of Machine Learning Research*, Vol. 3, pp. 993-–1022, 2003

[3] Biemann, C.: Chinese Whispers: An Efficient Graph Clustering Algorithm and its Application to Natural Language Processing Problems, In: *Proceedings of the HLT-NAACL-06 Workshop on Textgraphs-06*, pp. 73–80, ACL, New York City, 2006

[4] MacQueen, J. B.: Some Methods for Classification and Analysis of Multivariate Observations, In: *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, Vol. 1, pp. 281–297, University of California Press, 1967

[5] Miller, G. A.: WordNet: A Lexical Database for English, In: *Communications of the ACM*, Vol. 38, Issue 11, pp. 39–41, Nov. 1995

[6] Heyer, G., Quasthoff, U., Wittig, T.: *Text Mining - Wissensrohstoff Text*, W3L Verlag Bochum, 2006

[7] Budanitsky, A., Hirst, G.: Evaluating WordNet-based measures of semantic distance, In: *Computational Linguistics*, Vol. 32, Issue 1, pp. 13-–47, 2006

[8] Resnik, P.: Using information content to evaluate semantic similarity, In: *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pp. 448-–453, Montreal, Canada, 1995

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

[10] Dice, L. R.: Measures of the amount of ecologic association between species, In: *Ecology*, Vol. 26, No. 3, pp. 297–302, 1945

[11] Church, K. W., Hanks, P.: Word association norms, mutual information, and lexicography, In: *Computational Linguistics*, Vol. 16, Issue 1, pp. 22–29, Mar. 1990

[12] Quasthoff, U., Wolff, C.: The poisson collocations measure and its application, In: *Workshop on Computational Approaches to Collocations*, Wien, Austria, 2002

[13] Dunning, T.: Accurate methods for the statistics of surprise and coincidence, In: *Computational Linguistics*, Vol. 19, Issue 1, pp. 61–74, MIT Press, Cambridge, 1993

[14] Biemann, C., Bordag, S., Quasthoff, U.: Automatic acquisition of paradigmatic relations using iterated co-occurrences, In: *Proceedings of LREC2004*, Lisboa, Portugal, 2004

[15] Kubek, M., Witschel, H. F.: Searching the Web by Using the Knowledge in Local Text Documents, In: *Proceedings of Mallorca Workshop 2010 Autonomous Systems*, Shaker Verlag Aachen, 2010

[16] Witschel, H. F.: *Terminologie-Extraktion: Möglichkeiten der Kombination statistischer und musterbasierter Verfahren*, Ergon-Verlag, Würzburg, 2004

[17] Kubek, M., Unger, H.: Search Word Extraction Using Extended PageRank Calculations, In: *Autonomous Systems: Developments and Trends*, Studies in Computational Intelligence, Vol. 391, pp. 325–337, Springer Berlin Heidelberg, 2011

[18] Kubek, M.: DocAnalyser - Searching with Web Documents. In: *Autonomous Systems 2014, Fortschritt-Berichte VDI*, Vol. 10, Nr. 835, pp. 221–234, VDI-Verlag Düsseldorf, 2014

[19] Website of Neo4j, `https://neo4j.com/`, 2016, Last retrieved on 07/22/2016

[20] Efer, T.: Text Mining with Graph Databases: Traversal of Persisted Token-level Representations for Flexible On-demand Processing, In: *Autonomous Systems 2015, Fortschritt-Berichte VDI*, Vol. 10, Nr. 842, pp. 157–167, VDI-Verlag Düsseldorf, 2015

Spreading Activation: A Fast Calculation Method for Text Centroids

Mario M. Kubek[1], Thomas Böhme[2] and Herwig Unger[1]

[1]Chair of Communication Networks, University of Hagen, Germany

[2]Institute for Mathematics, Ilmenau University of Technology, Germany

Abstract: Centroids are comfortable instruments to represent queries and whole texts by single descriptive terms. They can be used to determine the similarity of textual contents and to (hierarchically) cluster sets of documents. However, their computation strictly following the concept's definition may use a plenty of time and hinder any practical application. A more demonstrative view on the meaning and topological interpretation of the definition leads to the derivation of a graph-based algorithm using the well-known spreading activation technique, which is described in this contribution. The experimental results obtained using co-occurrence graphs of varying sizes underline the high performance of this method which is – last but not least – brought about by its clear local working principle.

1 Motivation

Text centroids – inspired from the centre of mass in physics – and their application have been introduced in [1] and further articles like [2] have been used to deeply discuss their properties and to derive some interesting applications.

Differing from other methods, the determination of text representing centroids depends on some general knowledge and experience of a user, agent or program represented in the condensed structure of a co-occurrence graph and a defined metrics on it.

Any two words w_i and w_j are called co-occurrents, if they appear together in one sentence (or any other well-defined environment, context or window). This co-occurrence relation may be used to define a respective graph $G = (W, E)$. Therefore, the set words of a document corresponds to the set of nodes $w_a \in W$ and two nodes are connected by an edge $(w_a, w_b) \in E$, if w_a and w_b are co-occurrents. A weight function $g((w_a, w_b))$ can be introduced to represent the

frequency of a co-occurrence in a document, while usually only co-occurrences of a high significance $\sigma > 1$, $\sigma \leq g((w_a, w_b))$ are taken into account.

Distances can be defined on G, if two words are considered to be closely related, if they appear often together, i.e. $g((w_a, w_b))$ is big enough. If $g(w_a, w_b) > 0$, the distance $d(w_a, w_b)$ of the co-occurring words w_a and w_b is defined by

$$d(w_a, w_b) = \frac{1}{g((w_a, w_b))}.$$

For word pairs that do not co-occur, the shortest path $p = \{(w_a, w_2), (w_2, w_3), \ldots, (w_k, w_b)\}$ with $(w_i, w_{i+1}) \in E$ is considered and the distance is defined by

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

Consequently, the distance of two words w_a and w_b being isolated nodes or situated in two, not connected sub-graphs is set to

$$d(w_a, w_b) = \infty.$$

By replacing frequencies of co-occurrences with distances, the co-occurrence graph is transformed into an isomorphic (word) distance graph.

In order to determine a centroid term $\chi(D)$ of a document D, the set of N words $W(D) = w_1, w_2, \ldots, w_N \in D$ (filled by starting with the most frequent words) is considered with the nodes in a fully connected (sub-)component of the co-occurrence graph, such that pairwise distances $d(w_a, w_b) < \infty$ are guaranteed.

The centroid term $\chi(D)$ of a document is the term with the minimal average distance to all words of the document represented in $W(D)$[1], i.e. $d(D, \chi(D)) = MINIMAL$ for

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

The above definition is quite straight forward and descriptive. However, its direct application within a calculation algorithm would require to check all nodes of the fully connected (sub-)component of the co-occurrence graph, whether

[1] Note, that not necessarily $\chi \in W(D)$.

they fulfil the above defined minimum property. This results in an average complexity of $\mathbf{O}(|W|^3)$. Since co-occurrence graphs may contain up to 500.000 nodes (including nouns, names, compounds etc.), significant calculation times in the range of several minutes may appear even on powerful machines.

Therefore, a slight adaptation may be indicated which additionally avoids computationally expensive division operations. In the following, $\chi(D)$ denotes the centroid term of a document D, the term which minimises the longest distance to any of the words in the document, i.e. $d'(D, \chi(D)) = MINIMAL$ for all $t \in W(D)$ and

$$d'(D,t) = MAX(d(w_i,t)|i = 1 \ldots N).$$

In numerous experiments, it was found that the deviations caused by this adaptation are not too big. With these changes, a much faster, locally working method to determine the centroid of any document can be presented.

2 Conceptual Approach

2.1 Idea

To understand the idea of the herein presented method which utilises the spreading activation technique [3] to address this problem, the physical correspondence of the text centroids, i.e. the centre of mass must be considered again, as presented in [1]. In a physical body, the centre of mass is usually expected to be inside a convex hull line of the (convex or concave) body, in case of a homogeneous one and is situated more or less in its middle (Fig. 1[2]).

If a set of uniform, discrete mass points in a 2-dimensional, Euclidean plane with an underlying rectangular grid is considered, one would try to fix the centre or mass in the intersection of concentric cycles of the same radius around those points (Fig. 2).

Things may look more complex in a usual co-occurrence graph, since it can normally not be embedded in a 2- or 3-dimensional space, which humans can easily imagine. However, similar ideas of a neighbourhood allocation have already been used to provide a graph clustering method [4]. The so-called *Chinese Whispers* algorithm [5] is another interesting and related solution for efficient graph clustering that relies on a label propagation technique. The algorithm in the

[2]Modified from `https://commons.wikimedia.org/wiki/File:Bird_toy_showing_center_of_gravity.jpg`, original author: APN MJM, Creative Commons licence: CC BY-SA 3.0

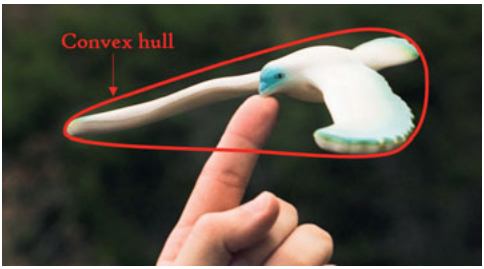


Fig. 1: A convex hull curve of a bird-toy and its centre of mass

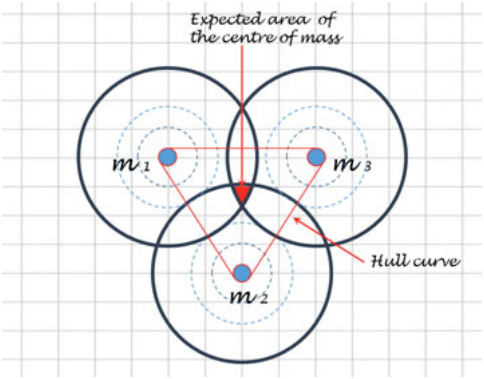


Fig. 2: Locating the centre of mass in the 2-dimensional plane

next subsection adapts the idea described above and can be applied on large co-occurrence graphs quite fast.

2.2 Algorithm

Usually, the co-occurrence graph can be kept on every machine. Therefore, the calculation of text centroids can be carried out in a local, serial manner. If only shortest paths are considered between any two nodes in the co-occurrence graph, a metric system is built.

In the following considerations, a query set Q of s words $Q = \{w_1, w_2, \ldots, w_s\}$ shall be considered. Q is called a **query set**, if it contains (usually after a respective preprocessing) only words $w_1, w_2, \ldots, w_s$, which are nodes within a

single, connected component of the co-occurrence graph $G = (W, E)$ denoted by $G' = (W', E')$.

A fast calculation method for the centroid term $\chi(Q)$ of a query set Q shall be presented at this point.

For computation purposes, a vector $\bar{v}(w') = [v_1, v_2, \ldots, v_s]$ is assigned to each $w' \in W'$ with the components being initialised to 0. With this preparations, the following, **spreading activation algorithm** is executed.

1. Determine (or estimate) the maximum of the shortest distances d_{max} between any pair (w_i, w_j) with $w_i, w_j \in Q$, i.e. let

$$d_{max} = sup(d(w_i, w_j)|(w_i, w_j) \in Q \times Q)$$

2. Choose a radius $r = d_{max}/2 + \Delta$, where Δ is a small constant of about $0.1 \cdot d_{max}$ ensuring that an overlapping area will exist.

3. Apply (or continue) a breadth-first-search algorithm from every $w_i \in Q$ and activate (i.e. label) each reached, recent node w' for every w_i by

$$\bar{v}(w')[v_i] = d(w_i, w') \leftrightarrow d(w_i, w') \leq r.$$

   Stop the activation, if no more neighbourhood nodes with $d(w_i, w') \leq r$ can be found.

4. Consider all nodes $w' \in W'$ with

$$\forall i, \quad i = 1 \ldots s : \bar{v}(w')[v_i] \neq 0$$

   and choose among them the node with the **minimal**

$$MAX(\bar{v}(w')[v_i])$$

   to be the centroid $\chi(\{w_1, w_2, \ldots, w_s)\}$.

5. If no centroids are found, set $r := r + \Delta$ and GoTo 3, otherwise *STOP*.

The greatest benefit of the described method is that it generally avoids the 'visit' of all nodes of the co-occurrence graph as it solely affects the local areas around the query terms $w_1, w_2, \ldots, w_s \in Q$.

The supremum $sup(d(w_i, w_j)|\forall(w_i, w_j) \in (Q \times Q))$ of a search query Q is called the **diversity** of a (search) query. The smaller the diversity is, the more a query

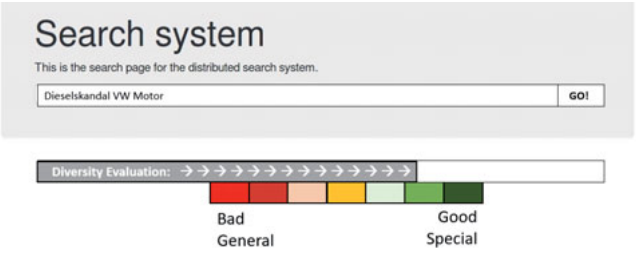


Fig. 3: Using the query diversity as a user's guide

targets a designated, narrow topic area, while high values of the diversity mark a more general, common request. This may be used to provide additional guidance and support for users during interactive search sessions on the usually keyword-oriented search engines (see Fig. 3).

3 Experimental Evaluation

In this section, the performance of the presented algorithm will be evaluated in a number of experiments. All measurements have been performed on a Lenovo Thinkpad business-class laptop equipped with an Intel Core i5-6200U CPU and 8 GB of RAM to show that the algorithm can even be successfully applied on non-server hardware. The four datasets[3] used to construct the co-occurrence graphs consist of either 100, 200, 500 or 1000 topically classified (topical tags assigned by their authors) online news articles from the German newspaper "Süddeutsche Zeitung". In order to build the (undirected) co-occurrence graphs, linguistic preprocessing has been applied on these documents whereby sentences have been extracted, stop words have been removed and only nouns (in their base form), proper nouns and names have been considered. Based on these preparatory works, co-occurrences on sentence level have been extracted. Their significance values have been determined using the Dice coefficient [6]. These values and the extracted terms are persistently saved in an embedded Neo4j (`https://neo4j.com`) graph database using its property-value store provided for all nodes (represent the terms) and relationships (represent the co-occurrences and their significances).

[3]Interested readers may download these datasets (4.1 MB) from: `http://www.docanalyser.de/sa-corpora.zip`

3.1 Exp. 1: Average Processing Time

In the first sets of experiments presented here, the goal is to show that for automatically generated queries of five different sizes (queries consisting of two to six terms) in six different ranges of diversity the average processing time to find their centroid terms is low when the spreading activation method is applied. The queries and the used co-occurrence graph have been generated using the dataset "Corpus-100" from which 4331 terms have been extracted. In order to determine the average processing time for each query size and for the six ranges of diversity, 20 different queries have been generated for these two parameters. Therefore, altogether 600 queries have been created.

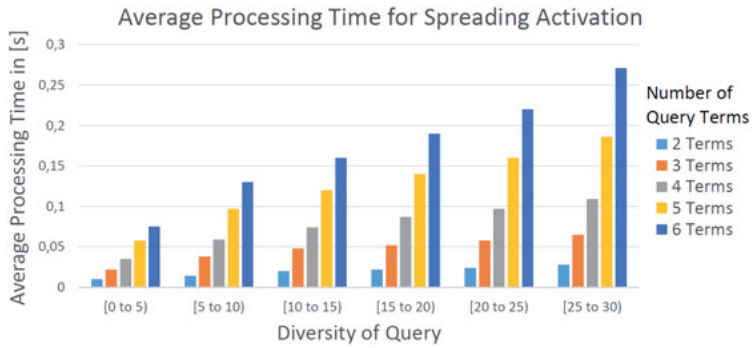


Fig. 4: Average processing time for spreading activation

Fig. 4 shows that even for an increased number of query terms and diversity values the average processing time stays low and increases only slightly. As the average processing time stays under half a second for all cases, the algorithm is clearly suited for application in interactive search systems.

Table 1: Processing times of the original algorithm

Number of query terms	2	3	4	5	6
Processing time in [s]	185	252	317	405	626

In order to demonstrate the great improvement in processing time of this algorithm, the original algorithm (as the direct application of the centroid definition given above) has been run on five queries consisting of two to six terms in the diversity range of [10-15) as a comparison while using the dataset "Corpus-100" to construct the co-occurrence graph as well.

Table 1 presents the absolute processing times in [s] needed by the original algorithm to determine the centroid terms for those queries. Due to these high and unacceptable values, the original algorithm – in contrast to the herein presented one – cannot be applied in interactive search systems that must stay responsive at all times.

3.2 Exp. 2: Node Activation

The second set of experiments examines the average number of nodes activated when the presented algorithm is run starting from a particular centroid term of a query while restricting the maximum distance (this value is not included) from this starting node. As an example, for a maximum distance of 5, all activated nodes by the algorithm must have a smaller distance than 5 from the centroid. In order to be able to determine the average number of activated nodes, the algorithm has been started for 10 centroid terms under this restriction. The maximum distance has been varied (increased) from 1 to 25. For this experiment, the dataset "Corpus-100" has been used again.

As Fig. 5 shows, the average number of activated nodes visibly and constantly rises starting from the maximum distance of 7 (40 activated nodes). At the maximum distance of 25, in average, 3780 nodes have been activated. The result also shows that for queries with a low diversity (and a therefore likely high topical homogeneity) the number of activated nodes will stay low as well. In this example, for a low maximum distance of 10, the average number of activated nodes is only 88 (2 percent of all nodes in the used co-occurrence graph). Therefore and as wished-for, the activation stays local, especially for low diversity queries.

3.3 Exp. 3: Growing Co-occurrence Graph

As document collections usually grow, the last sets of experiments investigate the influence of a growing co-occurrence graph on the processing time of the introduced algorithm. For this purpose, the datasets "Corpus-100", "Corpus-200", "Corpus-500" and "Corpus-1000" respectively consisting of 100, 200, 500

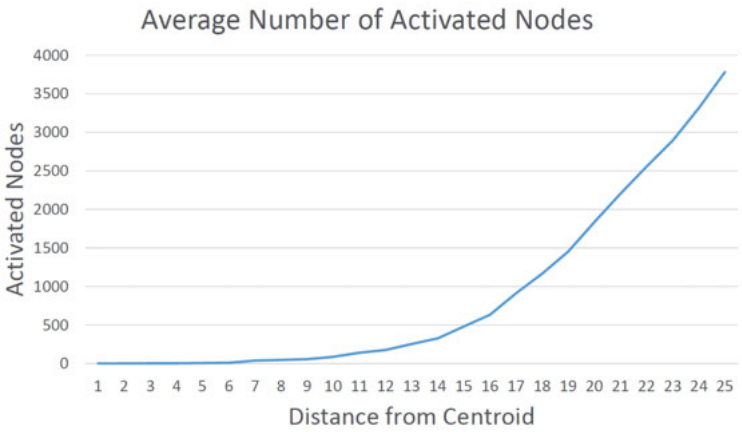


Fig. 5: Average number of activated nodes

and 1000 news articles have been used to construct co-occurrence graphs of increasing sizes with 4331, 8481, 18022 and 30048 terms/nodes. Also, as the document collection should be growing, it is noteworthy to point out that corpora of smaller sizes are included in the corpora of larger sizes. For instance, the articles in dataset "Corpus-200" are included in both "Corpus-500" and "Corpus-1000", too.

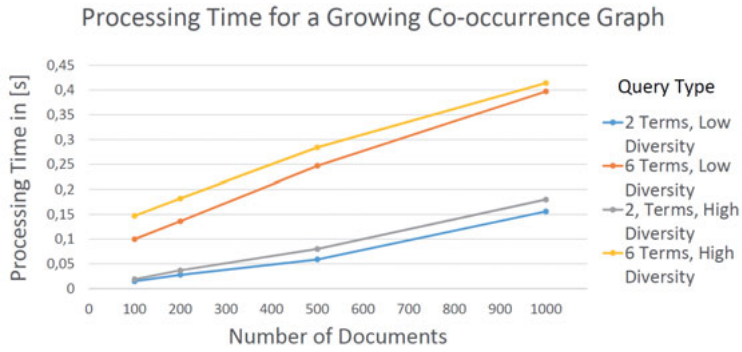


Fig. 6: Influence of a growing co-occurrence graph

In order to conduct the experiments, four queries have been chosen: one query with two terms and a low diversity in the range of [5-10), one query with two terms and a high diversity in the range of [20-25), one query with six terms and a low diversity in the range of [5-10) and one query with six terms and a high diversity in the range of [20-25). For each of these queries and each corpus, the absolute processing time in [s] (in contrast to the previous experiments that applied averaging) to determine the respective centroid term has been measured.

The curves in Fig. 6 show an almost linear rise in processing time for all four queries and the four growing co-occurrence graphs. Besides the size of the co-occurrence graph used, the query size is of major influence on the processing time. While the query's diversity plays a rather secondary role at this, it can clearly be seen that – even initially – the processing times for the queries with a high diversity are higher than for their equal-sized counterparts with a low diversity. However, even in these experiments and especially for the query with six terms and high diversity and the largest co-occurrence graph of 30048 nodes, the processing time stayed low with 0.41 seconds. While these experiments showed that the processing time will understandably increase when the underlying co-occurrence graph is growing, its rise is still acceptable, especially when it comes to handle queries in a (graph-based) search system. The reason for this is again the algorithm's strict local working principle. Node activation will occur around the requested query terms only while leaving most of the nodes in the graph inactivated.

4 Conclusion

A new graph-based algorithm to determine centroid terms of queries and text documents in a fast manner has been presented. In three sets of experiments conducted on modern laptop hardware, its performance has been positively evaluated for application in search systems. Due to its local working principle, it can be efficiently applied even when no server hardware is used. As the integrated spreading activation technique can be independently executed for every single initially activated node/term (e.g. from a query), the algorithm's core steps can be performed in parallel, e.g. in separate threads. This makes an effective utilisation of potentially available multiple CPU cores possible. Future optimisations of this algorithm will therefore focus on its parallelisation.

References

[1] 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.

[2] M. Kubek and H. Unger. Towards a librarian of the web. In *Proceedings of the 2nd International Conference on Communication and Information Processing*, ICCIP '16, pages 70–78, New York, NY, USA, ACM, 2016.

[3] S. E. Preece. *A Spreading Activation Network Model for Information Retrieval*. PhD thesis, Champaign, IL, USA, 1981.

[4] H. Unger and M. Wulff. Cluster-building in p2p-community networks. In *International Conference on Parallel and Distributed Computing Systems, PDCS 2002, November 4-6, 2002, Cambridge, USA*, pages 680–685, 2002.

[5] C. Biemann. Chinese whispers: An efficient graph clustering algorithm and its application to natural language processing problems. In *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing*, TextGraphs-1, pages 73–80, Stroudsburg, PA, USA, Association for Computational Linguistics, 2006.

[6] L. R. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, July 1945.

Empiric Experiments with Text-representing Centroids

Mario M. Kubek[1], Thomas Böhme[2] and Herwig Unger[1]

[1]Chair of Communication Networks, University of Hagen, Germany

[2]Institute for Mathematics, Ilmenau University of Technology, Germany

Abstract: Centroid terms are comfortable instruments to represent texts, compare them semantically and to even (hierarchically) cluster sets of documents using them. Their determination depends on their topical and conceptual context, i.e. the dynamically changing knowledge of a user represented by the co-occurrence graph. Herein, important properties of centroids as well as their applicability for tasks in natural language processing and text mining shall be discussed and their use justified by a set of experiments. Based on the obtained results, a new approach to detect fine-grained similarities between text documents is derived.

1 Introduction

Text centroids – inspired from the centre of mass in physics – have been introduced in [1] to represent sentences, paragraphs or whole texts by a single representing term. In addition, it could be shown that a distance measure among centroids may be defined which can be used to determine sematic text similarities and distances as well as to derive a hierarchical clustering algorithm [2] based on them. The introduction of centroids changes the methods of comparing texts in a significant manner. Two major approaches with practical relevance might be distinguished:

- The pairwise processing of two documents typically using the city block- or cosine distance. These methods are based on the vector space model [3] following the bag-of-words principle and work with any kind of term vectors of the two documents and consider – by nature – only words contained in one or the other document. These methods are quite simple but do not work well if the texts are written by authors using different sets of words to describe similar topics.

- The consideration of texts in the context of a corpus as used for instance in the technique Latent Semantic Indexing (LSI) [4] requires the calculation of the term-document matrix of a whole corpus and the computational expensive determination of a lower-dimensional approximation of its original semantic space. Document vectors in this lower-dimensional space can then be compared in the same manner.

The new, centroid-based method presented in [1] represents texts by a single centroid term (which is not necessarily contained in the document), which must be usually calculated, only once. Then, every comparison operation is just a single distance measurement on the respective co-occurrence graph, which can be considered a condensed, compact and easily extendible representation of the knowledge of an entity (e.g. a user) at a given moment and can be used to determine the centroid terms. Since the entity's knowledge may change/update/extend and texts may be subject to different modifications (merge, split, edit), properties of centroids must be investigated in a more detailed manner than it has been done so far, what is the goal of the presented work. After a short introduction on centroids, the influence of an entity's knowledge (i.e. the co-occurrence graph) for their calculation will be discussed followed by a detailed consideration of centroid properties obtained from a set of empiric experiments.

2 Fundamentals

To understand our subsequent discussions, some basic notations need be introduced. Any two words wi and wj are called co-occurrents, if they appear together in one sentence (or any other well-defined environment or context). This co-occurrence relation may be used to define a graph $G = (W, E)$. Therefore, the set of words of a document corresponds to the set of nodes $w_a \in W$ and two nodes are connected by an edge $(w_a, w_b) \in E$, iff w_a and w_b are co-occurrents. A weight function $g((w_a, w_b))$ can be introduced to represent the frequency of a co-occurrence in a document, while usually only co-occurrences of a high significance $g((w_a, w_b)) > 0.5$ are taken into account. For this filtering, the used weight function must yield values between 0 and 1 (both inclusive) as it is the case with e.g. the Dice coefficient [5]. In a next step, a distance must be defined in G. Two words are close, if $g((w_a, w_b))$ is high. If $(w_a, w_b) \in E$ (i.e. the words involved are co-occurrents) their distance $d(w_a, w_b)$ is easily to be defined as

$$d(w_a, w_b) = \frac{1}{g((w_a, w_b))}. \tag{1}$$

Otherwise, the shortest path $p = (w_1, w_2), (w_2, w_3), \ldots, (w_k, w_k + 1)$ must be considered with $w_1 = w_a$, $w_{k+1} = w_b$ and $w_i, w_{i+1} \in E$ for all $i = 1(1)k$ such that

$$d(w_a, w_b) = \sum_{i=1}^{k} d((w_i, w_{i+1})). \tag{2}$$

If there is no path between any two words w_a and w_b, $d(w_a, w_b) = \infty$ shall be set. The definition of the centroid term $\chi(D)$ of a document D uses all N words $w_1, w_2, .., w_N \in D$, which can be reached from any term t in G. Therefore, the average distance $d(D, t)$ of all words in D to the term t can be obtained by

$$d(D, t) = \frac{\sum_{i=1}^{N} d(w_i, t)}{N}. \tag{3}$$

The centroid term $\chi(D)$ is defined to be the term with

$$d(D, \chi(D)) = MINIMAL. \tag{4}$$

Note, that $\chi(D)$ does not necessarily occur in D. Let χ_1 be the centroid term of D1, and χ_2 the centroid term of D2, then $d(\chi_1, \chi_2)$ can be understood as the distance of the two documents D1 and D2.

3 Properties of Centroids

3.1 Background

The authors have always argued that the centroid-based computation is close to thinking mechanisms in the human brain. Mostly the feeling about similarities of words and documents and their sorting within ontological categories is learnt in a long process. Hereby, every new text source can not only be classified but is also used again to refine the knowledge of the individual. First rough working approximations are learnt fast and seem then to be stable for long times. It is observed that a few known keywords are enough to classify even completely unknown sources in the right manner. From the authors' point of view, semantic

relations among words and their associated senses are the reason for these effects. Starting with WordNet [6], those relations have been put into graph-based models. Later, different forms of co-occurrence graphs have been found to be a good approximation for the human's intuition for word and term associations, confirmed by stimulus-response experiments [7]. The following experiments shall give some more justification for those thoughts. For all of the exemplary experiments (many more have been conducted) discussed herein, 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 [5]. The particularly used sets of documents to create G and to calculate the centroid terms will be described in the respective subsections[1].

3.2 Stability of the Co-occurrence Graph

The first experiment shall confirm the fast convergence and stability of the co-occurrence graph which is a prerequisite for its use as a dynamic knowledge base of the individual or local computing node. A co-occurrence graph may be constructed from a text corpus in an iterative manner by successively adding co-occurrences from one document after another and finally removing all non-significant co-occurrence relations. During this learning process, the number of nodes and edges added to the co-occurrence graph for each new incoming document is converging. As especially high-weight edges representing significant co-occurrences are of interest for the centroid determination, Fig. 1 shows that the number of these edges converges quickly when a well-balanced corpus is used to construct G. The effect is even stronger for a topically focussed corpus as the terminology used in it does not vary greatly. The topically well-balanced corpus from dataset 3.2.1 used in this experiment contains 100 randomly chosen online news articles from the German newspaper "Süddeutsche Zeitung" from the months September, October and November of 2015 and covers 19 topics. The topically focussed corpus from the same dataset contains 100 articles on the European migrant crisis (a hotly discussed topic in late 2015) from the same newspaper and the same period.

[1] Interested readers may download these sets (1.3 MB) from: `http://www.docanalyser.de/cd-properties-corpora.zip`

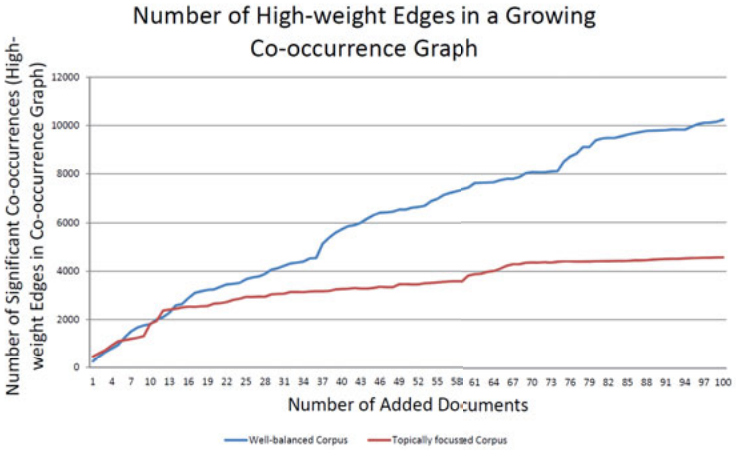


Fig. 1: Convergence of the number of high-weight edges in a growing co-occurrence graph

During the learning process, the probability that new words (nodes) are included is drastically decreasing, also the nodes' ranks (according to their outdegrees) only seldom changes. The node with rank 1 has the highest connectivity. Fig. 2 shows that most of the rank changes occur at the nodes with higher ranks (in this case, rank changes have been determined after 100 documents have been added to the collection) only. These are usually nodes with low connectivity and often have been added to the co-occurrence graph just recently.

For this experiment, the topically focussed corpus from dataset 3.2.1 has been used again. Centroid terms –– as defined in the previous section –– do not change frequently, even when the co-occurrence graph is growing. For this experiment, only one document has been added to the co-occurrence graph in each time step. As it can be seen in Fig. 3, the 'movement' of a reference document's centroid term (calculated after each time step) stabilises quickly. However, the convergence time depends on the order in which documents are added to the co-occurrence graph. Here, the topical orientation of and similarity between them plays an important role.

If similar documents to the reference document are added first (blue curve) and other, topically dissimilar documents afterwards, then the centroid term changes rarely (almost never) during their addition. If, however, the documents are added randomly (orange curve), then the convergence time increases. The

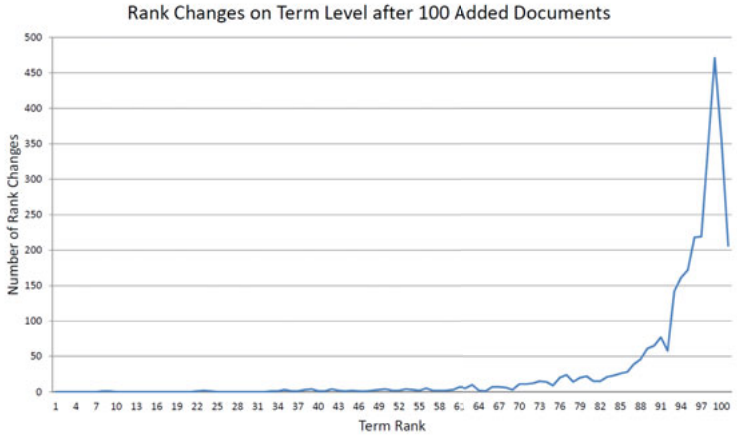


Fig. 2: Stability of term ranks in a grown co-occurrence graph

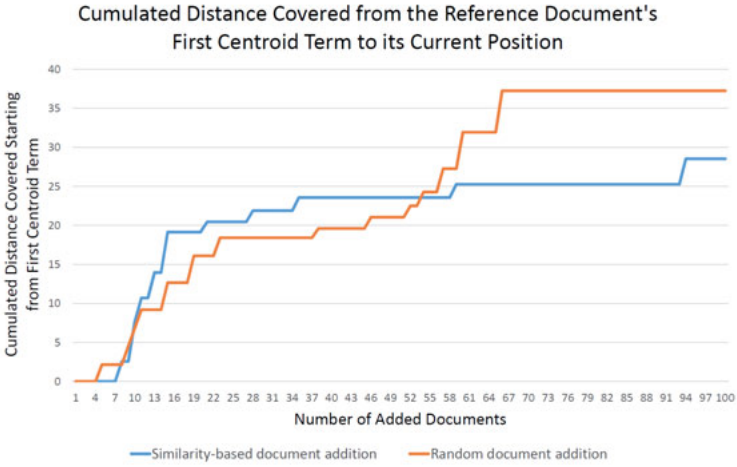


Fig. 3: Changes of document centroids in a growing co-occurrence graph

reason for this observation is that topically similar documents (which mostly influence the centroid term's position) can be added at any time. Thus, the probability that the centroid term changes at any time is increased, too.

The corpus from dataset 3.2.3 used for this experiment contains 100 newspaper

articles from "Süddeutsche Zeitung" which cover three topical categories 'car' (34 articles), 'finance' (33 articles) and 'sports' (33 articles). The reference document used was 'Schmutzige Tricks' (an article on the car emissions scandal). Therefore, it is not surprising that mainly in the first 34 time steps during the similarity-based document addition (in which the 34 car-related documents are added) the centroid term's position of this article is changed. Even so, in both cases, the centroid term's jumping distance between two consecutive 'positions' in the co-occurrence graph is low. However, it is still possible that the centroid term (usually just slightly) changes when the co-occurrence graph significantly grows as this process changes the distances between all nodes as well.

3.3 Uniqueness of Centroid Terms

Unfortunately, it is not easy to generate the centroid term of two parts of text from knowing their separate centroids. It costs once an effort of $\mathbf{O}(\mathbf{W^3})$ to construct the distance matrix of the co-occurrence graph G and then an additional $\mathbf{O}(\mathbf{W})$ to determine for every document any possible centroid candidate; an effort which must be definitely reduced in the future (although the calculation must be carried out only once or – at least – not very often for every document). However, since the co-occurrence graph's stability is quickly reached, centroids usually need to be calculated only once when documents first appear in the corpus or after larger time periods, in which significant changes of the underlying knowledge have occurred due to incoming sets of new documents.

In fact, it might happen in a given co-occurrence graph that one or more terms have the same, minimal average distance to all terms of the text or document. This would mean that the centroid term is not uniquely defined and more than one term could represent the document. In particular, this complies with reality – some documents, especially interdisciplinary ones – may not by clearly assigned to the one or another category. However, the subsequently explained practical experiences justify that this case in fact might only appear extremely rarely.

In Fig. 4, it is shown that in general there is a significant distance between the best (the actually chosen one) centroid term and the next 150 potential centroid candidates closest to it. This experiment has been conducted using 500 randomly selected sentences from the mentioned Wikipedia corpus for which their respective centroid terms have been determined while avoiding a topical bias. The results show that the mean distance from the best centroid to the potential centroid candidates gradually increases, too.

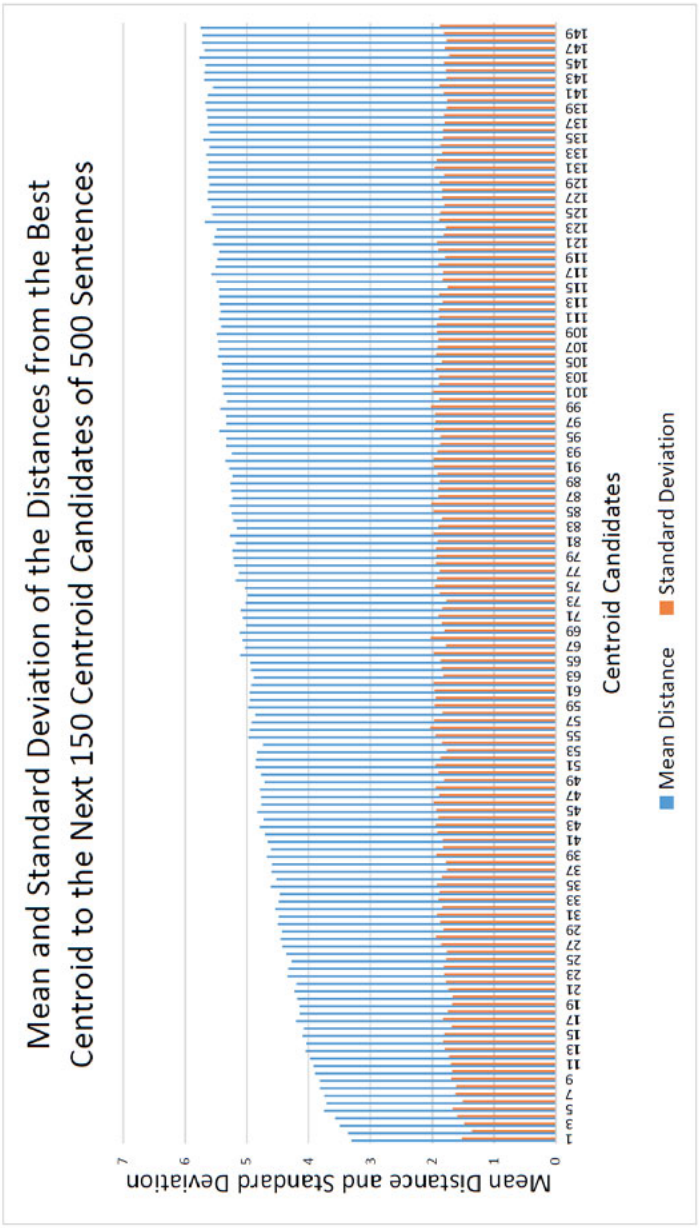


Fig. 4: Distances between centroid candidates

Although the standard deviation is relatively large, it stays constant. However, even when taking this value into account as well, the mean distance in the co-occurrence graph between the best centroid and its e.g. 10 closest centroid candidates is still large enough to come to the conclusion that the determined centroid (its position) is in general the best choice to represent a given textual entity. Further research needs to be carried out to find out, if and how much the centroid candidates' topical orientation or focus generally differs from the centroid term's one.

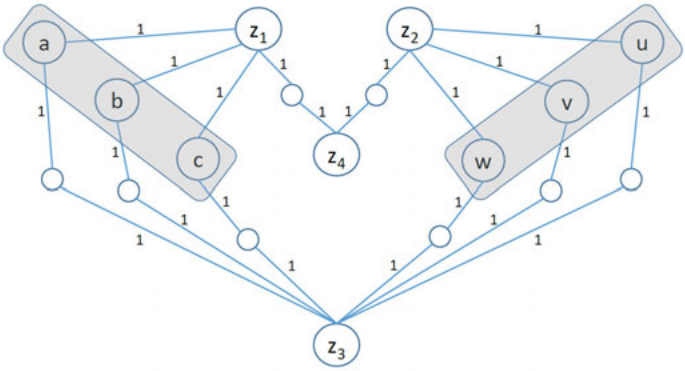


Fig. 5: Counterexample for the calculation of centroids

At this point, it must be mentioned that the well-known superposition principle may not be applied to text centroids, i.e., if $\chi(D_1)$ and $\chi(D_2)$ are the centroids of two pieces of text (or documents) D_1 and D_2, the following usually holds:

$$\chi(D_1 \cup D_2) \neq \chi(\chi(D_1) \cup \chi(D_2)).$$

Fig. 5 illustrates one respective counterexample. Given the presented graph, the nodes Z_1 and Z_2 are the centroids for the sets of nodes {a, b, c} and {u, v, w} respectively. These sets could represent the terms contained in two sentences. The following distances between the nodes can be extracted: $d(Z_1, x) = 1$ for $x \in \{a, b, c\}$; $d(Z_2, x) = 1$ for $x \in \{u, v, w\}$; $d(Z_4, Z_1) = d(Z_4, Z_2) = 2$; $d(Z_3, Z_1) = d(Z_3, Z_2) = 3$; $d(Z_3, x) = 2$ for $x \in \{a, b, c, u, v, w\}$.

The centroid of Z_1 and Z_2 is node Z_4 and not Z_3 which is, however, the centroid of all single nodes contained in these sets. In particular, this fact contradicts

with the hope to reduce the needed effort to calculate the centroid of a set of documents in an easy manner. It shows once more that there are significant differences between the text centroids and their physical analogon, the centre of mass, due to the discrete character of the co-occurrence graph.

However, experiments have shown, that $\chi(D_1 \cup D_2)$ and $\chi(\chi(D_1) \cup \chi(D_2))$ are not too far from each other. As it can be seen in Table 1 for an example using the Wikipedia-article 'Measles', those centroids have an average low distance between 2 and 3 while the maximum distance of two terms in the co-occurrence graph used is 18. The distance between the centroids of all sentence centroids in a fixed section of the article and the direct centroid of this section (in this case, its sentence boundaries have not been considered and its terms have been directly used to determine the centroid) is shown for all 11 sections.

Table 1: Distances of specific centroids of sections in the Wikipedia-article 'Measles'

Number of section	Centroid of all sentence centroids in section	Centroid of section	Distance of both centroids
1	measles	treatment	3.63
2	virus	measles	2.31
3	HIV	HIV	0
4	infection	diagnosis	3.40
5	aid	measles	3.24
6	infection	risk	2.52
7	infection	symptom	3.44
8	net	prevention	1.53
9	malaria	prevention	2.23
10	aid	interaction	1.51
11	research	health	2.23

Summarising, it can be noted that

- the centroid of a single term is the term itself, the centroid of two terms is usually a node close to the middle of the shortest path between them,
- the centroid is usually not the most frequent or most central term of a document,

- usually, the centroid is uniquely defined although two or more terms may satisfy the condition to be the centroid,
- the centroid of a query or text document can be a term which is not contained in its set of words and
- the centroid of two or more centroid terms is usually not a node on any shortest path among them or a star point with the shortest distance to them.

Nevertheless, finding a representing term to pieces of text also brings with it significant advantages, which shall be discussed in the following section.

3.4 Hierarchies of Centroids

Although – differing from semantic approaches – the assigned centroid terms may not represent any semantic meaning of the given text, they are in each case a formally calculable, well-balanced extract of the words used in the text and their content relations. This approach has been used to define the distance of documents [1] and to determine cluster hierarchies [2], too. Additionally, centroids may be used to detect topical shifts, i.e. subsequent changes in sections, paragraphs or sections of texts may be analysed, where usually classic methods offer only a pairwise comparison of the similarity of text fragments and do not take additional structural information of the given texts into account.

Fig. 6 and Fig. 7 show for the two structurally similar Wikipedia-articles 'Measles' and 'Chickenpox' the obtained dendrograms of centroid terms if the centroids of sentences are set in relation with those of paragraphs, sections and the whole documents. The centroids of those text fragments have been calculated by directly taking all terms contained in them into account and not by computing the centroids of the centroids of the respective next lower structural level. As an example for the article 'Measles', 11 sections are contained in it while the first section's centroid is 'treatment', the second section's centroid is 'infection' and so on. Furthermore, the sixth section (on the treatment of measles with the centroid 'risk') contains five paragraphs with up to 5 sentences in them. For each of those paragraphs and each of the sentences contained in them, the computed centroids are presented as well.

The section on the treatment of chickenpox (also the sixth section) contains in contrast to the article 'Measles' 10 paragraphs. As it can be seen in the depicted lists of sentence centroids for both articles and even on paragraph level, the terms such as 'paracetamol', 'vitamin', 'drug' and 'dipyridamole' are more

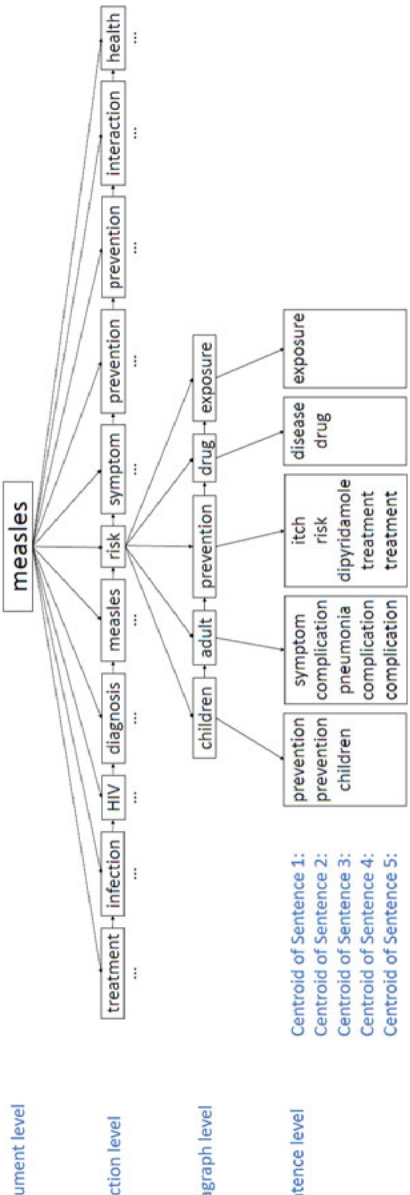


Fig. 6: Hierarchy of centroids obtained from sentences, paragraphs, sections and the entire Wikipedia-article 'Measles'

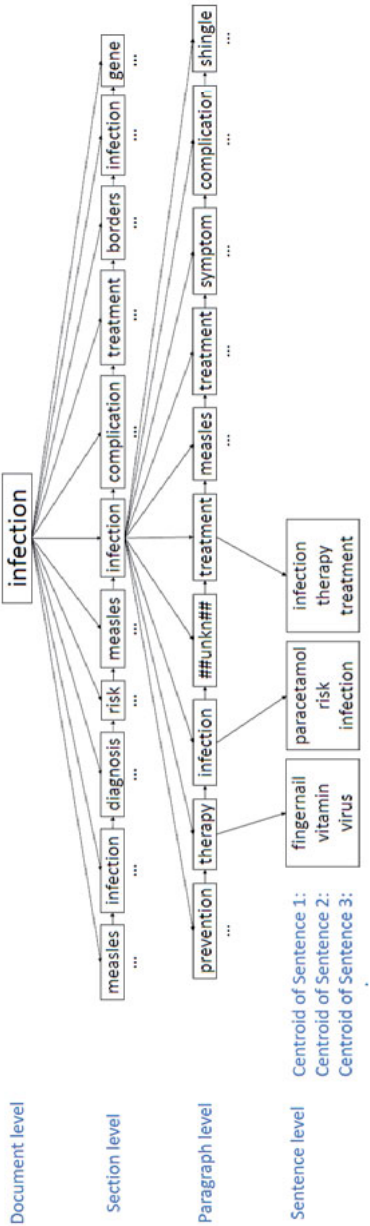


Fig. 7: Hierarchy of centroids obtained from sentences, paragraphs, sections and the entire Wikipedia-article 'Chickenpox'

specific than on section level. The centroid term of the fourth paragraph in the treatment section of the article 'Chickenpox' could not be properly determined ('##unkn##') as the one sentence in this paragraph contains only one term (the noun 'antiviral') which is, in addition to it, not existing in the given co-occurrence graph as it does not co-occur with any other term in the used Wikipedia corpus. Alternatively, as stated in the previous section, the term 'antiviral' could have been chosen as the centroid term instead.

However, it can be noticed from this example and similar cases that the determination of centroid terms is partly difficult and their quality is reduced when the textual context used for this purpose is small, the co-occurrence graph does not contain required terms or isolated and small clusters in it are addressed. In practice, the handling of exceptional cases like this must be specified. When comparing both dendrograms (and the centroid terms in them), it becomes obvious that both articles exhibit a similar topical structure. Even on section level, it is recognisable that the articles first deal with the general description of the diseases, cover usual diagnostic methods and the treatment of the diseases afterwards and then discuss their epidemiology. The interesting aspect is not only the decomposition of segments into sub-topics but the traces obtained from the left-to-right sequence of centroid terms and topics on the same level of the tree. Thus, it can be concluded that

- the diseases covered in the selected articles are similar,
- the articles exhibit the same structural composition,
- the centroid terms in the lower structural levels are more specific than in the upper levels (the number of terms in the sentences and paragraphs used to calculate them is of course lower and their centroids are determined by smaller, more topically specific contexts),
- a distance calculation of equally-ranked centroids on the same structural level will result in an estimation of how semantically close the respective descriptions (in this case those diseases) are to each other (as another finding, the authors detected the similarity of two diseases, whose English names are similar but greatly differ from e.g. German ones, i.e. 'Measles' and 'German Measles' (Rubella)) as well as
- a continuous distance check of paths or sequences of section- or paragraph-based centroid terms of similar documents can show where exactly their semantic or topical differences lie.

Therefore, it is sensible that future cluster building solutions take into account these findings. Also, they present a new direction to compute the centrality [8, 9] of words in a co-occurrence graph in order to find proper generalising terms for contents and to perform further semantic derivations from the position of centroids and their traces in the co-occurrence graph.

4 Conclusion

As the behaviour and changes of centroids as well as their determining context, the co-occurrence graph, are hard to derive in a theoretic manner, a set of experiments in well-defined environments have been conducted. The results justify the practicability and usability of centroid terms as text representatives. It could also be demonstrated that the used context behaves in a stable manner and especially its extension in a knowledge learning process does not influence the situation. Further publications will investigate mechanisms to reduce the effort needed to determine centroids by utilising neighbourhood effects in co-occurrence graphs.

References

[1] 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.

[2] M. Kubek and H. Unger. Towards a librarian of the web. In *Proceedings of the 2nd International Conference on Communication and Information Processing*, ICCIP '16, pages 70–78, New York, NY, USA, ACM, 2016.

[3] G. Salton, A. Wong, and C. S, Yang. A Vector Space Model for Automatic Indexing. In *Communications of the ACM*, vol. 18, issue 11, pages 613—620, New York: ACM, 1975.

[4] S. Deerwester et al. Indexing by latent semantic analysis. In *Journal of the American Society of Information Science*, vol. 41, no. 6, pages 391—407. Hoboken: Wiley-Blackwell, 1990.

[5] L. R. Dice. Measures of the amount of ecologic association between species. In *Ecology*, vol. 26, no. 3, pages 297-–302, July 1945.

[6] G. A. Miller. WordNet: A Lexical Database for English. In *Communications of the ACM*, vol. 38, issue 11, pages 39-–41, Nov. 1995.

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

[8] G. Erkan and D. R. Radev. LexRank: graph-based lexical centrality as salience in text summarization. In *Journal of Artificial Intelligence Research*, vol. 22, issue 1, pages 457–-479, Palo Alto: AAAI Press, 2004.

[9] I. Cantador, D. Vallet, D., and J. M. Jose. Measuring Vertex Centrality in Co-occurrence Graphs for Online Social Tag Recommendation. In *Proceedings ECML/PKDD Discovery Challenge 2009 (DC09)*, vol. 497, pages 17—33, Aachen: CEUR-WS, 2009.

Towards a Librarian of the Web

Mario M. Kubek and Herwig Unger

Chair of Communication Networks, University of Hagen, Germany

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.

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 Fig. 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.

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

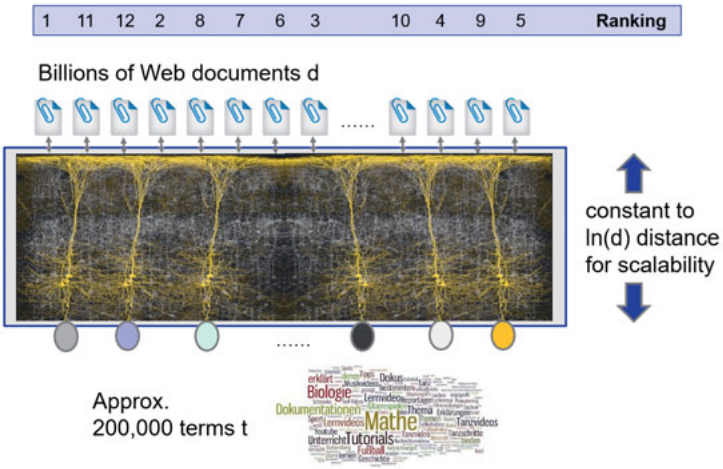


Fig. 1: The dimensionality problem of the WWW

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:

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

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

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

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

$$\zeta(D_1, D_2) = d(t_1, t_2). \tag{3}$$

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.

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

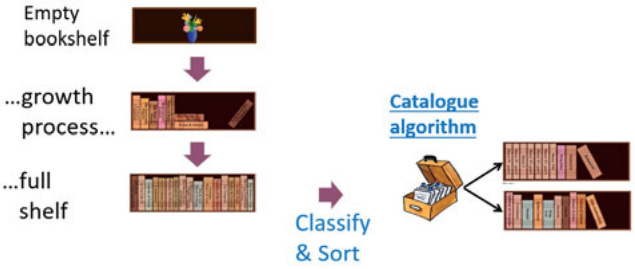


Fig. 2: Growth and division of a book collection

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.

[1] Interested readers may download these sets (1.3 MB) from: `http://www.docanalyser.de/cd-clustering-corpora.zip`

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 application 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 Fig. 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. Fig. 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

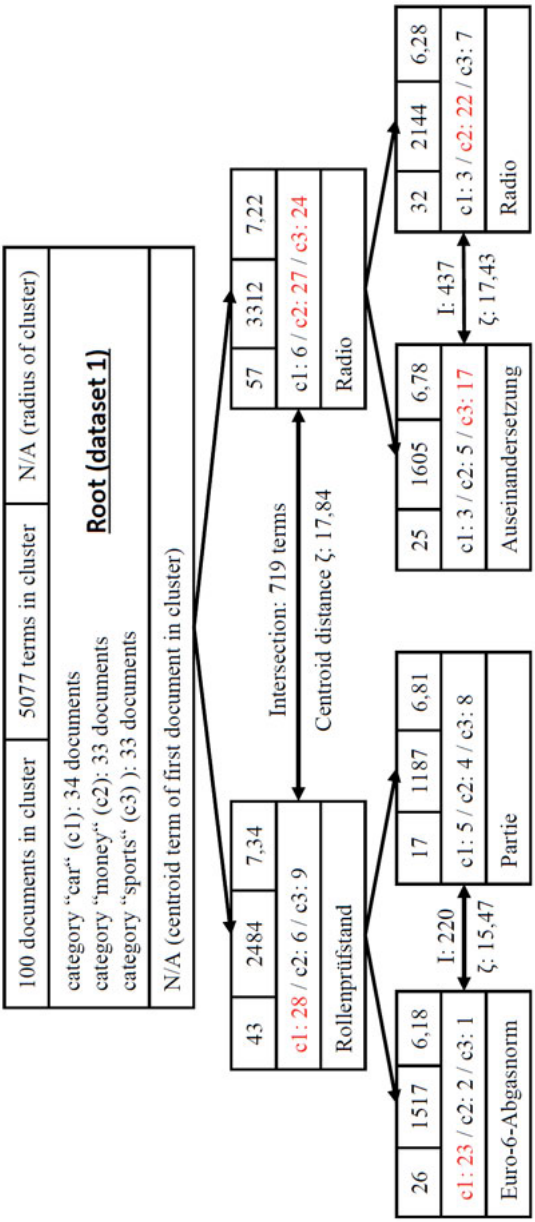


Fig. 3: Basic approach: clustering using antipodes

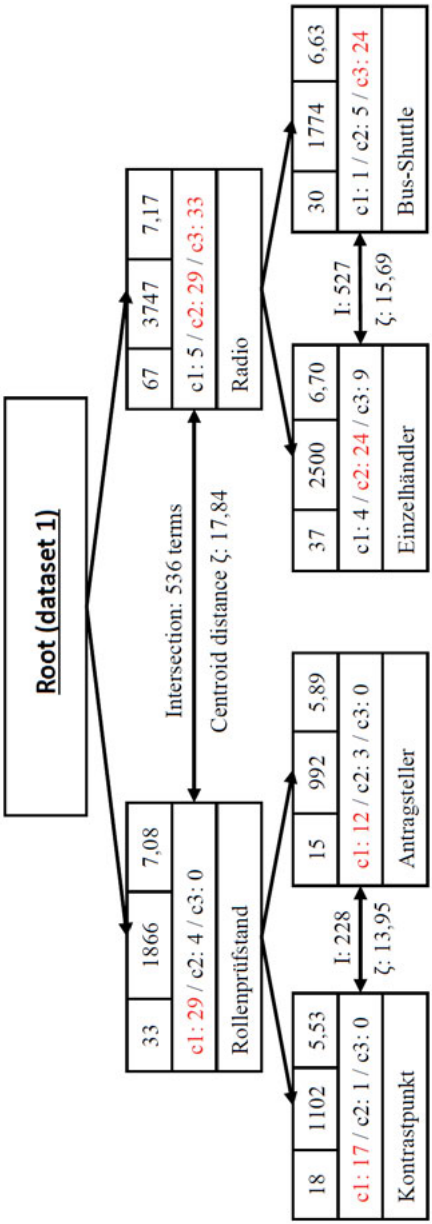


Fig. 4: Centroid-based clustering

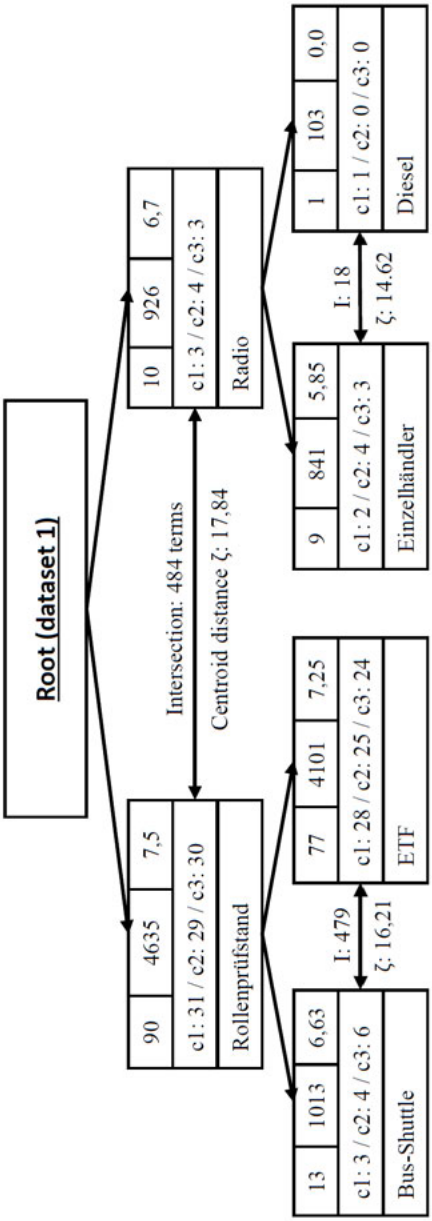


Fig. 5: Clustering using a Naive Bayes classifier

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. Fig. 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 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.

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.

A Concept Supporting a Resilient, Fault-tolerant and Decentralised Search

Herwig Unger and Mario M. Kubek

Chair of Communication Networks, University of Hagen, Germany

Abstract: Decentralised search engines usually replace huge databases by distributed information connected via respective structures. These structures must be generated and maintained with a low overhead and shall come along with a high scalability and fault tolerance. The article will show that random walkers are able to establish suitable hierarchic, *tree-like* structures. Due to their composition mechanism they exhibit self-maintaining and -healing properties as well as a high adaptivity to changing size and needs of the stored information.

1 Introduction and Motivation

In previous works, the fundamental functionality of centralised search engines has been criticised. It was figured out that nowadays the establishment of a more or less indexed copy of the web is no longer a future-proof concept [1].

Therefore, the main concept for a decentralised web search engine has been worked out. It is based on a peer-to-peer (P2P) system similar to [4] or [3] which is realised as an extension of existing web servers [1]. Hereby, the bootstrap problem is solved through the use of the existing hyperlinks in the webpages (Fig. 1).

Additionally, centroid terms where defined and used to characterise and compare any documents on the basis of a single, representing term [5]. The properties of these centroids have been considered in detail by [7]. Furthermore, centroids have been used to derive a hierarchical document clustering method [6].

Nevertheless, the immediate application of the hierarchical clustering algorithm presented before requires the election of an initial first node, since subsequent merging activities may be difficult. This problem can be overcome by a novel bottom-up construction of the document hierarchy, which will be described below, although its origins go back to a contribution in 2002 [2]. Using methods

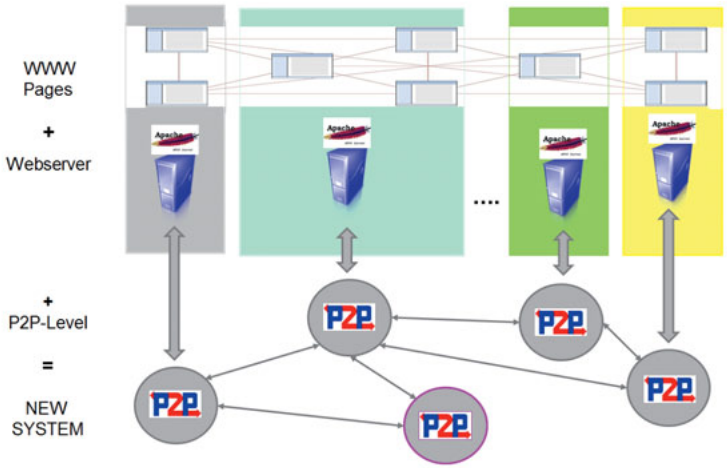


Fig. 1: First concept of a decentralised World Wide Web

of self-organisation and self-healing, the developed approach may even significantly contribute to the fault tolerance of the solution as, on the downside, the previously used classical tree-like architecture is characterised by its weak connectivity, too.

2 Handling of Random Walkers

Now, a P2P-system is considered, which is built by the web server extension as described before in [1]. Every peer P with the IP address $IP(P)$ of this system has its own neighbourhood warehouse $N(P)$ containing initially the neighbours derived from the links of pages hosted by the web server. It is later updated by the known, standard P2P ping-pong protocol.

For the intended hierarchy building procedures, the random walker concept shall be used. In the following considerations, the terms 'random walker' and 'agent' are used interchangeably. The needed information structure of each random walker

$$RW = (IP[], D[], W[], c_{max}, c, z, ptr)$$

consists of

- an array $[IP_1, IP_2, \ldots, IP_{c_{max}}]$ of positions to store IP addresses of peers belonging to a local cluster on a given level; whereby IP_1 always denotes and

is initialised with the IP-address $IP(P)$ of the owning peer, i.e. the peer which has generated the random walker,

- a second array $[D_1, D_2, \ldots, D_{c_{max}}]$ which contains the respective, remaining path information (locator) to identify the document on the corresponding peer (needed if a peer offers more than one document). Position D_1 is again reserved for the locator of the document of the owner/generator of the random walker.

- a third array $[W_1, W_2, \ldots, W_{c_{max}}]$ containing an ASCII content description of the document represented by the locator in the corresponding positions of $IP[] : D[]$ (and subsequent documents or sub-tree) managed by this random walker[1]. In the course of this elaboration, the centroid term of the document will be used for this purpose.

- c_{max} the overall number of positions in the address array field which may be either a global constant or determined depending on the network conditions for each walker in an adaptive manner (e.g. depending on circulation times).

- c a counter indicating the number of filled positions in the IP array (initially assigned to 1). Note, that $c \leq c_{max}$ must be always fulfilled.

- z the number of additional, randomly to be determined positions.

- ptr the index of the current position of the walker with $1 \geq ptr \geq (c_{max} + z)$ and initialised to 1.

In the beginning, one random walker is generated for each document available from the respective web server by the corresponding peer. Every document will be made identifiable besides $IP(P)$ by a local document locator D_h. An empty set $X(D_h)$ is stored, which will later contain IP addresses of peers with similar contents, i.e. which will form a completely connected sub-cluster.

Furthermore, each peer belonging to a web server executes the following algorithm to process the generated and/or arriving random walkers:

1. $LOOP_1$**:**
   Receive a new random walker $R_n = recv()$.

[1]Note, the random walker may be more light-weighted, if this information is not a part of the random walker but retrieved from the peer of origin

2. Determine randomly the earliest departure time $\tau(R_n)$ for the random walker.
3. Use a set C to keep all pairs of random walkers at the computer and let $C = \emptyset$ at the start time of the algorithm.
4. Let R be the set of random walkers, recently visiting the considered peer. Initialise $R := R_n$.
5. $LOOP_2$**:**
   Check $\forall i := 1 \ldots |R|$:
   IF $\forall j := 1 \ldots |R| \wedge (i \neq j)$
   - $(R_i, R_j) \notin C$ and
   - $(R_j, R_i) \notin C$ and
   - the recent time $t > \tau(R_i)$

   THEN
   - Set $R := R \setminus \{R_i\}$.
   - If $(ptr \leq c)$ then set $X(D_{ptr}) = \{IP_1, IP_2, \ldots, IP_c\}$.
   - Increase $ptr := ((ptr + 1) mod(c_{max} + z)) + 1$.
   - Send out R_i to IP_{ptr} by $send(IP_{ptr}, R_i)$, if $ptr \leq c$ and otherwise to a randomly chosen successor out of the set of location in the neighbourhood set of the peer $IP_{next} \in N$ by $send(IP_{next}, R_i)$, if $c < prt \leq (c_{max} + z)$.
6. If another random walker R_s is received
   a) Determine randomly the earliest departure time $\tau(R_s)$ for this random walker.
   b) Set $C := C \cup (\{R_s\} \times R)$.
   c) Update $R := R \cup \{R_s\}$.
7. If $C \neq \emptyset$
   a) Take any $(R_i, R_j) \in C$.
   b) $Compute(R_i, R_j)$.
   c) Set $C := C \setminus \{(R_i, R_j)\}$
8. If $R = \emptyset$ GoTo $LOOP_1$ otherwise GoTo $LOOP_2$.

In order to ensure fault tolerance, any document D_i which is not checked by its random walker within a given timeout period T_{out} may take activities to synchronise a new random walker using a standard election mechanism with all nodes within $X(D_i)$.

3 Computation of Random Walker Data

This section describes the needed computation $Compute(R_i, R_j)$ for any pair of random walkers, which are meeting on any node in the P2P-system. To distinguish both data areas, they are denoted by a leading first index in the formulas.

For the computation, two cases can be distinguished.

1. $c_i + c_j \leq c_{max}$
   i.e. the two random walkers may be merged into a single one. This case mostly appears at the start of the system or if a set of new documents and/or peers is added.

   The following updates are carried out for merging:

   a) $\forall k = (c_i + 1) \ldots (c_i + c_j) : IP_{i,k} = IP_{j,k-c_j}$

   b) $\forall k = (c_i + 1) \ldots (c_i + c_j) : D_{i,k} = D_{j,k-c_j}$

   c) $c_i = c_i + c_j$

   d) $c_{max,i}$, ptr_i and z_i remain unchanged

   e) The random walker R_j is cancelled by

   - $\forall (j := 1 \ldots |R|) \wedge (i \neq j)$
     let $C := C \setminus (R_i, R_j)$ and
     let $C := C \setminus (R_j, R_i)$.
   - $R := R \setminus \{R_j\}$

2. $c_i + c_j > c_{max}$
   i.e. the two random walkers met cannot be merged but the attached document links shall be sorted such that a maximum similarity is reached. Therefore the documents addressed via the set of URL's $(IP_n, m : D_n, m)$

$$U = \bigcup_{m=1}^{c_i} \{(IP_{i,m} : D_{i,m})\} \cup \bigcup_{m=1}^{c_j} \{(IP_{j,m} : D_{j,m})\}$$

are considered. Any clustering method or dichotomy building algorithm using document centroids in $W[]_i$ and $W[]_j$ as described before in [6] shall be used to generate two subsets U_i and U_j from U with $|U_i| < c_{max,i}$ and $|U_j| < c_{max,j}$. The co-occurrence graph used for this purpose can be either obtained from the existing sub-cluster i or j or be a (temporary) combination of both.
The following updates are carried out for merging R_i and R_j using U_i and U_j:

a) $\forall k = 1 \ldots |U_i| : IP_{i,k} = IP_{i,k}(U_i)$

b) $\forall k = 1 \ldots |U_j| : IP_{j,k} = IP_{j,k}(U_j)$

c) $\forall k = 1 \ldots |U_i| : D_{i,k} = D_{i,k}(U_i)$

d) $\forall k = 1 \ldots |U_j| : D_{j,k} = D_{j,k}(U_j)$

e) $c_{max,i}$ and $c_{max,j}$ remain unchanged.

f) Set $c_i = |U_i|$ and $c_j = |U_j|$

g) z_i and z_j remain unchanged.

h) $ptr_i = c_i + z_i$ and $ptr_j = c_j + z_j$ to ensure that the updates are executed in the fastest manner starting at the first node of the cycle in the next step.

In both cases, the notification of all participating nodes on the performed changes is sent out. Also, an update of the local clusters is executed within the next circulation of the random walker.

In addition, the co-occurrence graphs of all nodes related to the newly merged random walker are merged as well and the respective resulting centroid terms of all documents are determined under the management of the peer owning the random walker (i.e. which is on $IP[1]$). Since this procedure as well as the future use of the owner (first peer in the sequence) of a random walker requires some computational performance of the respective machine, changes in the order of $IP[]$, $D[]$ as well as $W[]$ might be indicated and useful.

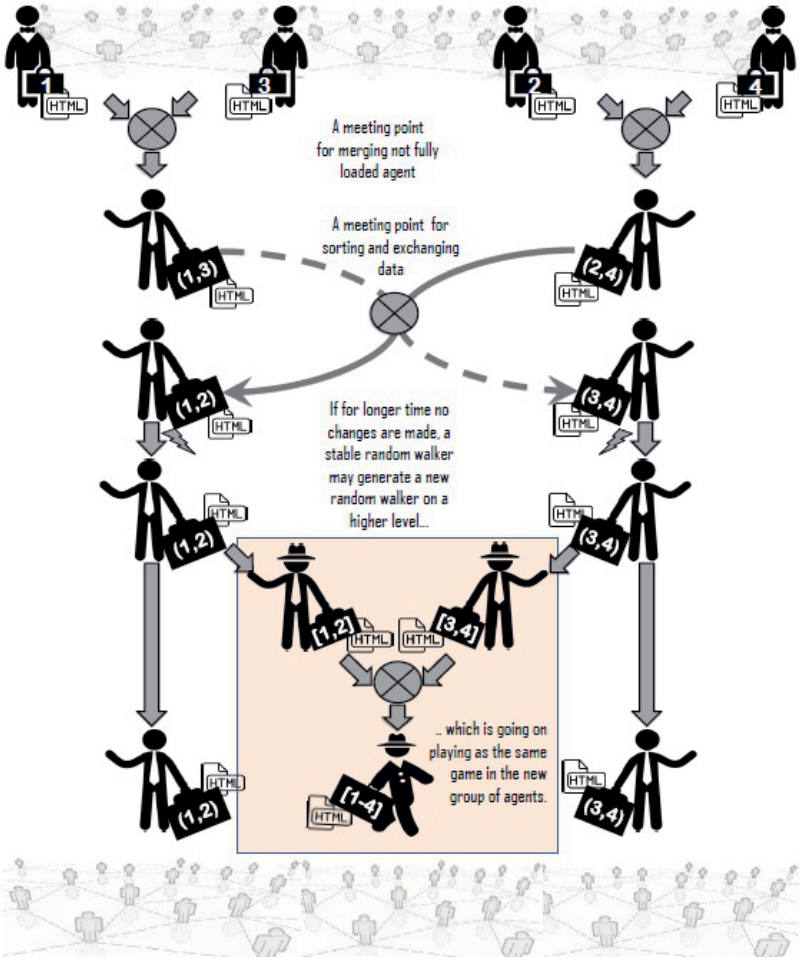


Fig. 2: The agents' activities to build hierarchical, tree-like structures

4 Building the Hierarchy

4.1 Structural Hierarchies

The methodology introduced in the previous sections is used to generate completely connected clusters of the most similar documents, which are still isolated substructures. Now, they shall be connected into a tree-like, hierarchical structure of completely connected sub-graphs.

Therefore, an addition must be made to the definition of random walkers

$$RW = (IP[], D[], W[], c_{max}, c, z, ptr)$$

by changing it to

$$RW = (IP[], D[], W[], c_{max}, c, z, ptr, lev)$$

whereby lev represents the level of the tree, on which the respective random walker is acting. Note that the leaves will have – differing from usual enumerations – level 1.

For the further processing steps, the following updates are necessary:

1. In the beginning, i.e. at the starting point of the above described algorithm, $lev = 1$ represents the lowest level (the document level).

2. In W, the centroid term of the associated document is indicated, when random walkers are merged. It is replaced by the centroid term of all documents represented by the random walker.

3. The owner of the random walker, i.e. usually the peer represented in the first position of $IP[]$, observes all changes in the random walker.
   If for a fixed, longer time interval Δ

   - no changes are observed in the $IP[]$- and $D[]$-array of the random walker and

   - $1 < c \leq c_{max}$,

   the $IP[1]$ peer is allowed *to launch another, new random walker*.

4. By doing so, it may happen that on any level a single random walker may not be connected to the community, since all existing other random walkers have filled exactly all c_{max} positions.
   To avoid this case, position $c_{max} + 1$ is used as a temporary (emergency) position. If $c = c_{max}$, it may merge with a random walker with $c = 1$ but must release this position at the next possible solution regardless any content aspects, i.e. as soon as either a random walker with $c < c_{max}$ is met or another random walker with $c_{max} + 1$ filled positions is found (in this case a new random walker at the same level is created).

5. This *new random walker* follows exactly the rules for the initial settings, as described above for the level 1-random walker, with the following updates:

   - The respective level information is derived from the level information of the generating walker and increased by 1.

   - W contains the centroid term which is calculated from the centroid terms of all participating (represented) documents (level 2) or random walkers (in the higher levels), which are usually stored on the generating peers.

6. Also, the behaviour of the new random walker follows exactly the rules described above. However, a computation, merging and sorting will only take place for random walkers having the same level lev.

7. For reasons of fault tolerance, the $IP[]$- and $D[]$-array may contain instead of single information a sub-array, containing a small number (i.e. two or three) of information from other nodes of the represented sub-cluster. In such a manner, an unavailability of the first peer in the list may be tolerated by using a replacement peer.

8. The described hierarchy building mechanism is successively repeated for all higher levels $2, 3, \ldots$ and stops automatically, if the conditions formulated in 3 cannot be fulfilled any more.

Fig. 2 presents the activities carried out by the agents in the system.

4.2 Hierarchies of co-occurrence graphs

Any decision making is based on co-occurrence graphs stored locally. On the lowest level 1, the respective co-occurrence graphs are built from one document. As soon as the positions $1 \ldots c_{max}$ of each random walker are filled, temporary

and bigger co-occurrence graphs may (will) be built and used for decision making but must eventually be re-organised, since every document may only represent its set of co-occurrences in one co-occurrence graph $R_{c,lev}$ on each level lev.

With the generation of a random walker for $lev + 1$, the co-occurrence graph of the respective represented region on level lev shall be built and be stable. At least with the complete compositions of the random walkers on level $lev + 1$, also the next level co-occurrence graph $Rc, lev + 1$ shall be available and will be used in the same manner to assemble the next hierarchy level $lev + 2$. In addition, $Rc, lev + 1$ will be handed down to all random walkers until level 1.

Of course, the respective (but seldom necessary) updates may cause repeated evaluations of the similarity of documents (sets of documents in different regions) combined in a random walker. Since the remaining z randomly chosen positions in each random walker will allow meetings on any peer at any time, these discrepancies will automatically be recognised and the structure will be adapted. Exactly the same process will result in an automatic inclusion of newly appearing documents and their random walkers.

Note that any change in a random walker requires a new calculation of the co-occurrence graph on the respective level with the corresponding re-calculations of its upper and lower levels. However, this process is relatively seldom needed (last but not least to the stability of co-occurrence graphs). Due to its size, all Rc, lev shall never be a part of a random walker but solely be kept in a graph database of the represented nodes, regions or sub-clusters. The complete connection of those structures (as described above) may make those updates even more simple.

5 Conclusion

The algorithmic fundamentals of a fully decentralised P2P-based web search engine have been presented. It is relying on the classification of documents as well as search requests by so called centroids, i.e. single descriptive terms. Furthermore, an agent-based method has been devised to contruct hierarchic, tree-like structures using local information which allow for a routing of search requests without broadcasts. In addition, it is scalable and fault tolerant as well.

References

[1] Eberhardt, R., Kubek, M., Unger, H.: *Why Google Isn't the Future. Really Not.* In: H. Unger and W. Halang: Proceedings der Autonomous Systems 2015, Fortschritt-Berichte VDI, Series 10: Informatik/Kommunikation, VDI, Düsseldorf, (2012)

[2] Unger, H., Wulff, M.: *Cluster-building in P2P-Community Networks.* In: Journal on Parallel and Distributed Computing Systems and Networks, Vol. 5(4), pp. 172–177, (2002)

[3] Christen, M. et al.:*YaCy: Dezentrale Websuche*, Online Documentation, (2017) `http://yacy.de/de/Philosophie.html`

[4] Faroo Limited: *FAROO: Distributed Search.* White Paper, (2017) `http://www.faroo.com/hp/p2p/whitepaper.html`

[5] Kubek, M., Unger, H.: *Centroid Terms as Text Representatives.*, In: DocEng '16, Proceedings of the 2016 ACM Symposium on Document Engineering, Vienna, Austria, ACM, (2016)

[6] Kubek, M., Unger, H.: *Towards a Librarian of the Web.* In: Proceedings of the 2nd International Conference on Communication and Information Processing (ICCIP 2016), ACM, Singapore, (2016)

[7] Kubek, M., Böhme, T., Unger, H.: *Empiric Experiments with Text Representing Centroids.* In: Proceedings of the 6th International Conference on Software and Information Engineering (ICSIE 2017), Singapore, (2017)

An Associative Ring Memory to Support Decentralised Search

Herwig Unger and Mario M. Kubek

Chair of Communication Networks, University of Hagen, Germany

Abstract: Centroid terms are single words that semantically and topically characterise text documents and thus can act as their very compact representation in automatic text processing and mining. In order to make them a useful tool in textual search tasks as well, a concept for a novel associative memory with a ring-like structure storing and managing these terms and their associated contents is proposed. As this memory is designed to be applied in decentralised search systems, centroid terms will inherently support the efficient routing and forwarding of queries to matching nodes this way. Besides providing remarks on its implementation, necessary load balancing activities as part of the memory's management functions are discussed as well.

1 Motivation

Text-representing centroids [12] have been introduced to foster the categorisation of text documents and multi-word queries using a single, descriptive term. These terms can be used to compute of the similarity of documents [15] and to build document clusters as well as hierarchic structures to support the search of documents in the Internet [14, 17].

The practical application of centroid-based methods was made possible by a new graph-based method [16] for the fast calculation of centroid terms for texts of arbitrary length while relying on preferably large co-occurrence graphs as a knowledge base. Last but not least, the derived measure *diversity* of a given document or query indicates how general or detailed and topic-oriented the analysed content is [16].

In particular, multiple keywords of a search query and whole (longer) texts are represented by single centroid classifiers. Those single terms can be easily matched or compared [13, 14]. Thus, complex programs to combine a set of partial results for multi-keyword queries (like MapReduce of Google [5]) are no longer needed.

Additionally, the calculation of centroid terms can make use of personal co-occurrence graphs. Therefore, centroids may be obtained reflecting the recent knowledge and experience of the user. Last but not least, centroids can be lexically (alphabetically) ordered. The obtained order and relation ($<, =, >$) between any two centroid terms allows for the creation of well-ordered structures (like lists, rings, or trees), which can be efficiently searched by standardised algorithms. Only the expected, extremely high number of entries will require a distributed storage of such structures, preferably on peers of a flexible peer-to-peer (P2P) system.

After a short discussion of related works, a new associative ring structure shall be introduced, which enables a fast, fault-tolerant management of an almost unlimited number of entries and supports a fast textual search.

2 Review of Related Works

After the rise of classical, client/server-based content delivery systems, several approaches have been proposed to manage a large amount of contents within decentralised, P2P systems [2]. In order to avoid flooding, a slow access or replication of files, several approaches based on dynamic hash tables (DHT) have been introduced (for an overview see [3]). Chord [7][1] seems to be the most practicable and widely used approach since it reduces additional overhead and problems, especially in case of an unexpected leave of peers, to a minimum. A set of projects is available, using Chord and/or offering libraries to setup a Chord system, e.g. Open Chord (`http://open-chord.sourceforge.net/`).

Two major problems could be identified using a standard Chord implementation: first, no load balancing is available. This may result in a concentration of entries on a single peer, if the hash value of many keys/entries is the same, but also if a concentrated access to a few, dedicated content items takes place. Second, due to the used hash function, similar content is usually not placed on neighbouring peers on the Chord ring. The latter said requires that a user must know the exact key (category) in order to locate and access the wanted content on the Chord ring. Since similar document keys would not be placed in close proximity, a document search by human users would become more difficult.

[1] In order to avoid misinterpretations, 'Chord' is written with a capitalised 'C', if we mean the DHT-system, while 'chord' with a non-capitalised, first letter denotes the connection of two vertexes in a ring graph/circle.

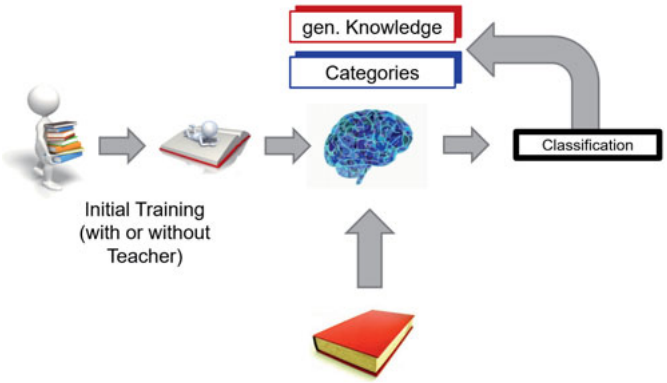


Fig. 1: Human learning and application of categories and classification

Humans usually learn about categories and their descriptors in a longer, supervised or unsupervised learning process (Fig. 1). Starting with the research on WordNet [9], there is a consent that the relation between those categories can be appropriately modelled and approximated by lexical graphs such as the so-called co-occurrence graphs. Different co-occurrence graphs for each individual may therefore reflect the state of knowledge of that person and are subject to a permanent change depending on the person's private experiences. As discussed in [10], these personal experiences are poorly considered in the search procedures of the big Internet search providers.

A data structure to be developed must take the above discussed facts and processes into account and support the following requirements:

1. Search must be understood as process, rather than a single event. Each document read and each search carried out qualifies the user.

2. The user's knowledge and experience must be represented in a data structure, which is – due to its personal character and for privacy reasons – stored on the user's machine, only.

3. Similar information shall be stored in a close proximity in the data structure to support a fast location.

4. The number of results returned shall be adjustable.

5. The used structure must be adaptable to a growing or huge number of data. Absolute addressing shall be avoided in order to support a hardware-independent, flexible work.

In the next sections, a respective data and communication structure shall be described and evaluated.

3 Design of Data- and Access Structures

3.1 Concept

Since the major concern is the support of search in the World Wide Web (WWW), it is intended to build up the data and access structures for the peer-to-peer (P2P) system, which has already been provided in form of a web server extension, described in [11] (Fig. 2).

Here, every web server also generates an own peer running in parallel to the web server system but is able to exchange data with it. Each peer obtains an initial neighbourhood derived from the links of pages hosted by the web server. This neighbourhood is continuously updated by the known, standard P2P PING-PONG protocol (what also allows for the inclusion of non-webserver peers in a more advanced stage of the system).

In addition, every peer may index the locally offered webpages and may therefore immediately answer any incoming queries matching those documents. With the standard functions of a P2P-(file-)sharing system, a simple, decentralised search engine is already made available.

Fig. 3 shows the design of the proposed system. Its main component is a ring structure of peers, which are running on different machines. It initially contains a single peer, only. A respective management functionality ensures that new, participating peers may be added and – if not needed anymore – be removed.

The ring of peers hosts a ring list of entries, whereby every peer can store a larger number of entries (partial list). Every entry represents one HTML document by a pair consisting of a key and a link (i.e. an URL) to the document. As searchable key, the corresponding centroid of the HTML document is used. In such a manner, the $i-$th entry in the ring list has the form $[centroid_i, URL_i]$.

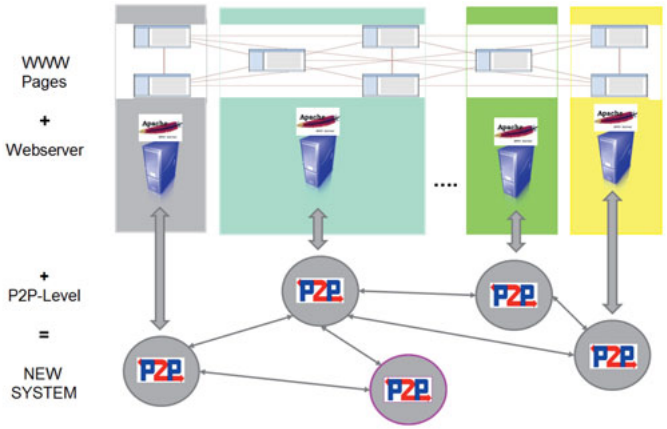


Fig. 2: Generating a P2P-system with the Apache Tomcat web server

Seen from the first entry $[centroid_1, URL_1]$, all items until the last one are lexically ordered[2], i.e. $centroid_i \leq centroid_j, \forall i, j$ with $i < j$.

Differing from Chord, as no fixed position can be given for any entry and depending on the number of entries and peers (as well as the load), this position is flexible. Queries as well as update operations of items (search, add and remove) can be directed to any peer participating in the ring structure and will be forwarded along the ring to the respective place, where the operation can be executed.

Furthermore, two types of chords are added to each node of the ring:

- F random chords with chosen destinations from all peers of the ring, shortening the access to any item, similar to the fingers in Chord.
- S chords connecting the peer with its K nearest neighbours for fault tolerance reasons. Therefore, each peer must mirror the contents of those K peers as a backup copy in case one of these peers suddenly leave the ring without carrying out a proper exit procedure.

Last but not least, every user may access the ring structure by a respective peer service. This peer service includes

[2]Note that for a position (x, y, λ) representing an evolving centroid $x \leq y$ is required ensuring that for every position a unique representation is given and a lexical ordering by x, y and last but not least followed by the value of λ will be possible.

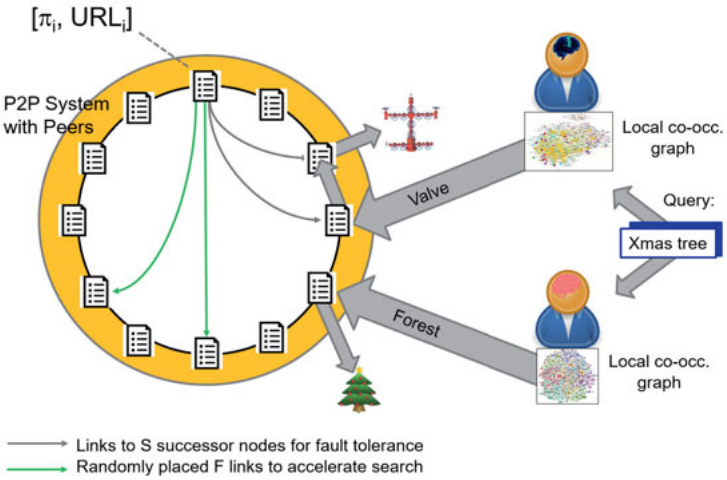


Fig. 3: The general concept

- an individual co-occurrence graph, which allows for the calculation of a centroid to handle a query or document, is built depending on the experience and history of the user and possibly his or her files on the local system. Following some considerations on the stability of co-occurrence graphs, in [15], no highly experienced user or librarian is needed to add a new document to the ring.
- a (bounded) broadcast mechanism (TTL 3..4) to locate at least one peer of the ring.
- functions with the respective communication protocol to execute the respective operations on the ring.

With these preliminaries, the functionalities and operations for the ring list memory can now be described.

3.2 Operations

The functionalities may be divided into the following three groups: user operations, ring management function and load balancing activities.

1. **User Operations**
   include the data operations *add*, *remove* and *search* for items on the ring.

Therefore, a *lookup* function allows to locate any peer of the ring structure in the entire P2P-system. With the known address, any data operation may be send to this peer, while the ring itself provides the forwarding functionality to transfer and execute the requested operation on the right peer. Therefore, the given key $(centroid_r)$ must be routed to the peer p with $centroid_i$ such that $centroid_i < centroid_r \leq centroid_j$ for $j = i + 1$.
(Depending on security and privacy needs, every entry may contain an owner and access right information in addition to the above defined version.)

2. **Ring Management Operations**
adapt the size of the ring to the needed one depending on the relation between available and used memory and keep the system of chords.

   - Since the ring starts with a single peer only, new peers must be *added* and correctly linked in the ring structure, if the number of entries exceeds a given maximum depending on the size of the entire ring (due to the subsequently described load balancing, it is not usually necessary to consider a single peer only).

   - In the same manner, a peer can be *removed* if the number of entries on the ring is getting so low that (more than) one peer can be detached without exceeding the remaining capacity. After finding a respective candidate for removal, any remaining items must be moved to the remaining peers before the removal can be executed. It is self-evident, that the links of the successor and predecessor node must be adapted, if peers are added or removed from the ring.

   - Last but not least, chords to other nodes in the ring shall be periodically updated; the S links for fault tolerance in shorter periods, the F finger links supporting long-distance jumps after longer periods.
   For doing so, a *peer-lookup*-protocol is used, which consists of *peer-lookup*-operations and -messages. Of course, all peers in the ring must support the execution of all *lookup*- operations as well as the generation and forward of needed communications in the ring.

In order to stabilise the system, all activities shall be executed only, if the respective conditions are fulfilled for a (randomly chosen) period. In case the management does not prevent the existence of several ring structures after system initialisation, a mechanism must allow the survival of the older

structure and the entry-by-entry inclusion of the newer one; a ring identifier with a respective time stamp sent to all participating peers is useful for doing so.

3. **Load Balancing Activities**
represent the most significant difference of this new approach to the classic Chord rings. It is therefore intended to use the physical analogue of a drop of water, wetting a surface (see Fig. 4).

Fig. 4: A drop of water dissolving on a surface

The effects of the different acting powers and surface tension are modelled by the communication of each peer in the ring with its direct neighbours. If a peer contains β percent more entries than the average of its neighbours, it sends entries to both of its neighbours to equalise the powers and to level the height (Note that this realises a sender-based load balancing).

By doing so, it must be ensured that the right lexical order of the entries is kept. To simplify communication procedures, the levelling only involves the one neighbouring peer with the highest difference of items. For an example, see Fig. 5. Load balancing activities can also be initiated, if a peer experiences a significant access load, i.e. an extremely high volume of communication must be handled.

As a result of this load balancing, no item is assigned to any fixed peer (and respectively no fixed IP address). Moreover, this position usually will change over time. Therefore, any item can be addressed only by its key, i.e. the realised memory is an associative one.

In the following section, the working principles of the described associative ring memory shall be considered.

Time	Peer1	Peer2	Peer3	Peer4	Peer5
0	20	0	100	20	20
1	20	50	50	20	20
2	35	35	35	35	20
3	35	35	35	28	27
4	31	35	32	31	31
5	33	33	32	31	31
6	32	33	32	31	32

Fig. 5: Example: balancing the number of entries on a ring with 5 peers

4 Remarks on Implementation

4.1 Ring Management

The algorithm for the ring management shall ensure that enough memory is available to keep and insert all entries on request but also avoid that to many peers are involved and resources remain unused. Therefore ring management operations can be executed resulting in an addition to or removal of peers from the ring structure. Of course, frequent changes of the ring size (i.e. oscillations) shall be avoided, i.e. without those changes, the size of the ring shall converge to a constant one.

To avoid central instances and allow a fully decentralised ring management, a token-/(random walker-) based procedure carried out by all participating peers is suggested. The set of all tokens represent the state of the system. The system strives to combine tokens and derive ring management activities from the obtained information.

Therefore, the content of each token stands for the balance of a part of the system: a value of 0 represents a fully balanced system having enough resources to add new data items to the list, a positive value denotes (especially with increasing numbers) that the system is getting filled up more and more. Negative values describe an underused system having too much unused resources (peers) employed.

Now, the system can work and adapt its size in a suitable manner by following the subsequent rules on each peer:

1. Every token i stands for an integer value $T(i)$.

2. Inserting into and removing an entry from the ring list result in the creation of a token with the value $T(i) = 1$ and $T(i) = -1$, respectively,

   - after the operation was successful and

   - by the peer hosting the respective item.

3. Tokens are forwarded in a randomly chosen direction clockwise or counterclockwise along the doubly connected peer ring and remain for a randomly chosen time on each peer. This time will be extended if more than one token are on a peer or if an entry is added or removed from the ring list at the respective peer.

4. Two tokens i and j meeting on one and the same peer are merged into a single token by adding their values, i.e. $T(i) := T(i) + T(j)$ and token j will be removed.

5. Let $INS(p)$ be a constant corresponding to each peer p describing its memory capacity (usually 80% of the peer's memory capacity, which is made available to the ring). If a token with a value $T(i) \geq INS(p)$ is recognised on a peer p at any time, this peer inserts a new peer p' to the ring list. After doing so, the token value is reduced by $T(i) := T(i) - INS(p')$, where $INS(p')$ is representing the space on the new peer p', which can be different from those on p (Note, that by doing so, the token value may become negative, if peers with different resources (memory sizes) are used).

6. Let $DEL(p)$ be a (negative) constant, which is equal to $-(Capacity\_of\_Peer)$ p. If a token value is $T(i) \leq DEL(p)$, the peer p on which the token is located organises the its removal from the ring list. The token value is increased by $T(i) := T(i) + DEL(p)$.
   Note that

   - the last peer (i.e. a peer pointing on itself as predecessor and successor) can not be removed and

   - the token will be forwarded without any operation, if the data items of the peer cannot be successfully moved to its neighbouring peers within a given time limit.

7. Any token with $T(j) = 0$ will be removed immediately.

8. *Emergency rule:* If the capacity of any peer is used and an entry insertion is requested, a new peer p'' will be immediately added. In addition, a token k with the value $T(k) = -INS(p'')$ is created.

The described token game is susceptible to token losses, however, the S fault tolerance chords can be used to avoid problems from lost tokens. Therefore, the following handling and protocol is suggested (as a simplification of the procedures published in [1] for the use on ring structures).

1. Every token gets an unique Token ID in addition to its value.
2. The S predecessor peers $P_{i-1}..P_{i-|S|}$ of peer P_i keep a log of the passed token in a special registry.
3. After moving a token from peer P_i to peer P_{i+1}, P_i sends a message to $P_{i-|S|}$ to cancel the entry about this token in its registry and adds the token to its own.
4. If token in any registry gets older than a deadline T_{out}, the last node in the chain (i.e. the peer whose predecessor does not contain an entry on that token) re-creates that token.
   If $|S| \geq 2$, the following nodes confirm the re-creation of the token and adjust their registries in an appropriate manner.

It is easy to be seen that $|S|$ nodes must fail at the same time, in order to cause faults in the token management. Therefore, with the above described methods, a stable, fault-tolerant execution shall be achievable.

As a matter of last resort, selected administrator nodes shall be allowed to

1. collect and stop all incoming tokens as well as process them as described in the basic token procedure,
2. initialise a special status token, which counts the available (free) memory space while completing one round in the ring and
3. generate an adjusted token after processing the status token and all stopped token on the administrator peer.

When needed, more administrative tasks may be subsequently added to this status token protocol, which anyhow realises a decentrally working, yet central control of the ring memory structure.

4.2 Finger Management

To ensure a fast, scalable search, Chord implements a method based on finger tables on each node containing up to m entries, where m denotes the number of bits in the hash key. Hereby, the i^{th} entry of node n will contain a pointer to the successor $((n + 2^{i-1}) mod (2^m))$ of n (see Fig. 6). A stabilisation protocol is used to update all finger links and generates additional overhead. [8] showed that a random distribution of the fingers will not significantly influence efficiency.

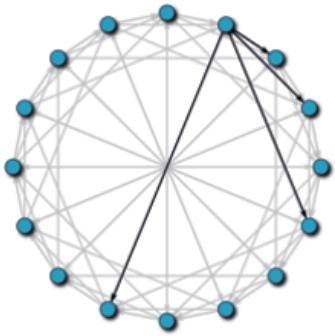


Fig. 6: The Chord Ring with its finger assignment

The overhead may also be generally avoided, if two not yet considered circumstances are used:

1. An unstructured but well-connected P2P-system is running underneath the described ring structure. It makes use of frequent *PING-PONG* messages to keep the network connected.

2. The items on the peers along the ring are lexically ordered and the distribution of letters is known (for a large number of entries), see [6] and Fig. 7.

To increase the speed of search in the ring, especially the long-distance fingers are important, i.e. those connecting a peer to other peers having distances of $1/2, 1/4, 1/8, \ldots$ of the ring length (number of peers in the ring) to the origin.

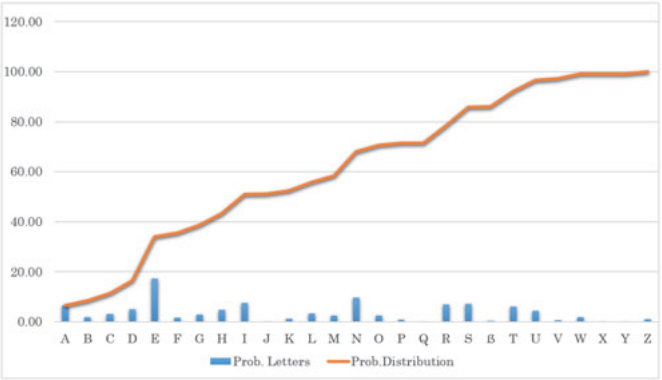


Fig. 7: Probability of letters and the respective distribution

It shall be avoided, that the corresponding end peers of the chords shall be determined by (frequently repeated) 'counting'-procedures along the ring.

Therefore, let us consider the probability of letters as well as the corresponding probability distribution in Fig. 7. Lets assume that l_i, l_j are two letters with $l_j > l_i$. In this case

$$p = F(l_j) - F(l_i)$$

determines the percentage p of words starting with initial letters between l_i and l_j. Thus, finding a peer with an approximate distance of $1/2, 1/4, 1/8, \ldots$ of the total ring length starting from the origin l_i means to divide the set of words in fractions with the size of $1/2, 1/4, 1/8, \ldots$ and then, finally, find the peer having a key starting with the letter l_j such that $p = 1/2, 1/4, 1/8, \ldots$ and so on. Let us consider for instance the first peer in the ring storing the very first key with the initial letter **A**. From the probability distribution in Fig. 7, it can be derived that 50% of all words will have initial letters between **A** and **I**. Consequently, the chord from the first peer to the opposite side of the ring must end (approximately) on a peer having the key **I** on it.

If finally the message content (payload) of the *PONG*-reply messages in the underlying P2P-system is extended to $[IP, First\_key\_on\_peer, Last\_key\_on\_peer]$, the (same) peers participating in the ring may obtain the IP-addresses of the needed fingers without any additional overhead by simply listening to the already existing communication.

5 Conclusion and Outlook

The concept of an associative memory with a ring-like structure to be applied in P2P-systems has been introduced. The entire memory structure is managed in a completely decentralised manner by the participating peers and can adapt itself to changing needs of the user as well as to a changing system environment. Anyhow, only the basic concept has been presented so far. Thus, many refinements may make the proposed operations more efficient and at the same time increase the stability of the structures. These adaptations will be elaborated on in future contributions.

References

[1] Unger, H., Böhme, T.: A probabilistic money system for the use in P2P network communities, In: *Proceedings of the Virtual Goods Workshop*, Ilmenau, pp. 60–69, 2003

[2] Lv, Q., Cao, P., Cohen, E., Li, K., Shenker, S.: Search and replication in unstructured P2P networks, In: *Proceedings of the 16th International Conference on Super Computing*, ACM, pp. 84–95, 2002

[3] Castro, M., Costa, M., Rowstron, A.: Peer-to-peer overlays:structured, unstructured, or both, *Technical Report MSR-TR-2004-73*, Microsoft Research, System and Networking Group, Cambridge (UK), 2004

[4] Castro, M.; Druchel, P.; Hu, Y.C.; Rowstron, A.: Topology-aware routing in structured peer-to-peer overlay networks, *Technical Report MSR-TR-2002*, Microsoft Research, 2002

[5] Lämmel, R.: Google's Map Reduce programming model – Revisited, In: *Science of Computer Programming*, Elsevier, Vol.70(1), pp. 1–30, 2008

[6] Meier, H.: Deutsche Sprachstatistik, In: *Olms Paperbacks 31*, 2nd edition, Olms, Hildesheim, 1967

[7] Stoica, I., Morris, R., Karger, D., Kaashoek, M. F., Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for internet applications, In: *ACM SIGCOMM Computer Communication Review*, 31(4):149, 2001

[8] Trompeter, M.: Evaluierung der Auswirkung von gleichverteilten Fingertabellen bei Chord, Masterthesis, University of Hagen, Chair of Communication Networks, Hagen, 2014

[9] Fellbaum, C.: WordNet and wordnets, In: *Brown, Keith et al. (eds.): Encyclopedia of Language and Linguistics*, 2nd edition, Oxford: Elsevier, pp. 665–670, 2005

[10] Kubek, M., Unger, H., Loauschasai, T.: A Quality- and Security-improved Web Search using Local Agents, In: *Intl. Journal of Research in Engineering and Technology (IJRET)*, Vol. 1, No. 6, 2012

[11] Eberhardt, R., Kubek, M., Unger, H.: Why Google Isn't the Future. Really Not., In: *H. Unger and W. Halang: Proceedings der Autonomous Systems 2015*, Fortschritt-Berichte VDI, Series 10: Informatik/Kommunikation, VDI, Düsseldorf, 2015

[12] Kubek, M., Unger, H.: Centroid Terms as Text Representatives, In: *Proceedings of the 2016 ACM Symposium on Document Engineering*, DocEng '16, New York, NY, USA, ACM, pp. 99–102, 2016

[13] Kubek, M., Unger, H.: Centroid Terms and their Use in Natural Language Processing, In: *Autonomous Systems 2016*, Fortschritt-Berichte VDI, Reihe 10 Nr. 848, VDI-Verlag Düsseldorf, pp. 167–185, 2016

[14] Kubek, M., Unger, H.: Towards a Librarian of the Web, In: *Proceedings of the 2nd International Conference on Communication and Information Processing (ICCIP 2016)*, New York, NY, USA, ACM, pp. 70–78, 2016

[15] Kubek, M., Böhme, T., Unger, H.: Empiric Experiments with Text Representing Centroids, In: *Lecture Notes on Information Theory*, Vol. 5, No. 1, pp. 23–28, 2017

[16] Kubek, M., Böhme, T., Unger, H.: Spreading Activation: A Fast Calculation Method for Text Centroids, In: *Proceedings of the 3rd International Conference on Communication and Information Processing (ICCIP 2017)*, New York, NY, USA, ACM, 2017

[17] Kubek, M., Unger, H.: A Concept Supporting Resilient, Fault-tolerant and Decentralised Search, In: *Autonomous Systems 2017*, Fortschritt-Berichte VDI, Reihe 10 Nr. 857, VDI-Verlag Düsseldorf, pp. 20–31, 2017

The WebEngine – A Fully Integrated, Decentralised Web Search Engine

Mario M. Kubek and Herwig Unger

Chair of Communication Networks, University of Hagen, Germany

Abstract: This paper presents a basic, new concept for decentralised web search which addresses major shortcomings of current web search engines. Its methods are characterised by their local working principles, making it possible to employ them on diverse hardware configurations. The concept's implementation in form of an interactive, librarian-inspired peer-to-peer software client, called 'WebEngine', is elaborated on in detail. This software extends and interconnects common web servers creating and forming a decentralised web search system on top of the existing web structure while – for the first time – combining modern text analysis techniques with novel and efficient search functions as well as approaches for the semantically induced P2P-network construction and its flexible management. This way, an alternative, fully integrated and powerful web search engine under the motto 'The Web is its own search engine.' is built making the web searchable without any central authority.

1 Introduction

It is definitely a great merit of the World Wide Web (WWW, web) to make the world's largest collection of documents of any kind in digital form 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 base or library of mankind in the age of information technology. Google (`https://www.google.com/`), as the world's largest and most popular web 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.

Even so, in the authors' opinion, Google and Co. are just the mechanistic, brute force answer to the problem of effectively managing the complexity of the WWW and handling its big data volumes. As already discussed e.g. in [1], a copy of the web is established by crawling it and indexing web content

in big reverse index files containing for each occurring word a list of files in which they appear. Complex algorithms try to find those documents that contain all words of a given query and closely related ones. Since (simply chosen) keywords/query terms appear in millions of (potentially) matching documents, a relevance ranking mechanism must avoid that all of these documents are touched and presented in advance to the user (see Fig. 1).

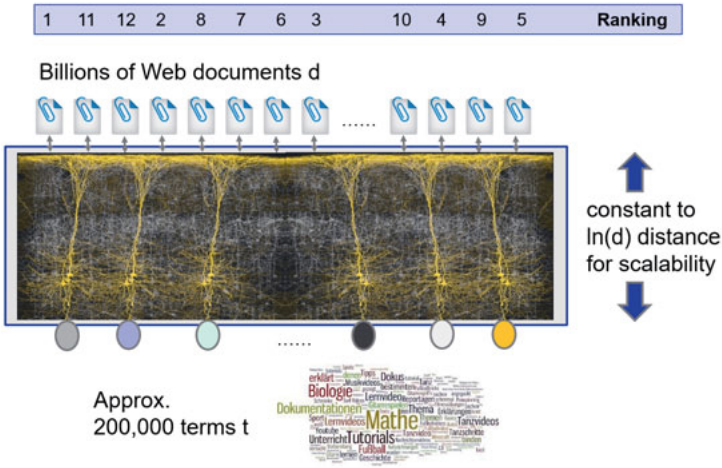


Fig. 1: The dimensionality problem of the WWW

In the ranking process, the content quality and relative position of a document in the web graph as well as the graph's linking structure are taken into account as important factors. In addition to organic search results obtained from this process, advertisements are often presented next to them which are related to the current query or are derived from personalisation efforts and detected user's interests. In both cases, web search engines do not take into account probably existing (local) user knowledge. To a certain extent, this procedure follows a top-down approach as this filtering is applied on the complete index for each incoming query in order to return a ranked list of links to matching documents. The top-ranked documents in this list are generally useful.

However, due to the sheer amount of data to handle and in contrast to the

bottom-up approach of using a library catalogue or asking a librarian or human expert in their role as active intermediaries between resources and users for guidance in order to find (more) actually relevant documents, the search engines' approach is less likely to return useful (links to) documents at an instant when the search subject's terminology is not fully known in advance. Furthermore, the web search results are not topically grouped, a service that is usually inherently provided by a library catalogue. Therefore, conducting research using web search engines means having to manually inspect and evaluate the returned results, even though the presented content snippets provide a first indication of their relevance.

From the technical point of view, the search engine's architecture carrying out the mentioned procedures has some disadvantages, too: In order to generate a refindable connection between contents and their locations and to be able to present recent results, the crawlers must frequently download any reachable web pages and thus make and store (multiple) copies of the entire web in their indexes. To achieve a high coverage and actuality (web results should cover contents that have been updated in the last 24 hours with a probability of at least 80 percent), they cause avoidable network load. Problems get bigger, once the hidden web (deep web) is considered besides regularly accessible HTML (Hypertext Markup Language) pages (surface web), too. Modern web topology models (like the evolving web graph model [2]) emanate from the fact that there are linear as well as exponential growth components, if the overall number of websites is considered. The constant crawling of these components causes especially high network load and their archiving needs a huge amount of storage capacity, too.

This brute-force method of making the web searchable is therefore characterised by a significant overhead for maintaining and updating the indexes. Furthermore, the used technical components like servers and databases are potential targets for cyber-attacks and pose a threat for the system's safety and security as well as for data protection.

As it is necessary to properly address all these problems of centralised web search engines, this paper introduces a new concept along with its technical solutions and infrastructures for future, decentralised web search relying on peer-to-peer (P2P) technology. In order to show that P2P-technology is actually useful in information retrieval tasks, the following section discusses several approaches in this regard first before deriving the respective requirements for this concept.

2 P2P Information Retrieval

When it comes to using P2P-systems for the purpose of information retrieval, one has to keep in mind that – in contrast to the use case of content delivery – replica of (relevant) documents often do not exist. Thus, it is needed to find the few peers that actually can provide them. Therefore, efficient routing mechanisms must be applied to forward a query to exactly those matching peers and to keep network traffic at a low level. Thus, a suitable network structure must be set up and adapted in a self-organising manner as well. At the same time, such a network must be easily maintainable.

Some of the most important results in the field of P2P information retrieval (P2PIR) have been obtained in the SemPIR projects [3]. Their goal was to make search for information easier in unstructured P2P-networks. In order to reach this goal, a self-organising semantic overlay network using content-depending structure building processes and intelligent query routing mechanisms has been built. The basic idea of the approach applied therein is that the distribution of knowledge in society has a major influence on the success of information search. A person looking for information will first selectively ask another person that might be able to fulfil her or his information need.

In 1967, Milgram [4] has shown that the paths of acquaintances connecting any two persons in a social network have an average length of six. These so-called small-world networks are characterised by a high clustering coefficient and a low average path length. Thus, the mentioned structure building processes conceived modify peer neighbourhood relations such that peers with similar contents will become (with a high probability) direct neighbours. Furthermore, a certain amount of long-distance links (intergroup connections) between peers with unrelated contents is generated. These two approaches are implemented in order to keep the number of hops needed (short paths) to route queries to matching peers and clusters thereof low. This method is further able to reduce the network load.

In order to create those neighbourhood relations, a so-called 'gossiping' method has been invented. To do so, each peer builds up its own compact semantic profile (following the vector space model) containing the k most important terms from its documents which is periodically propagated in the network in form of a special structure-building request, the gossiping-message. Receiving peers compare their own profiles with the propagated one and

1. put the requesting peer's ID and profile in the own neighbourhood list and

2. send the own profiles to the requesting peer if the profiles are similar to each other.

Also, the requesting peer can decide based on the received profiles which peers to add to its neighbourhood list. Incoming user queries (in the form of term vectors as well) are matched with the local profile (matching local documents will be instantly returned, too), the profiles of neighbouring peers and are forwarded to the best matching ones afterwards. This mechanism differs from the mentioned approach in real social networks: in the technical implementation, the partaking peers will actively route queries from remote peers. In real social networks, people will likely just give the requesting person some pointers on where to find other persons that have the required knowledge instead of forwarding the requests themselves.

In doing so, a semi-structured overlay P2P-network is built which comprises of clusters of semantically similar peers. Additionally, each peer maintains a cache of peers (egoistic links) that have returned useful answers before or have been successfully forwarded queries to matching peers. Furthermore, the network's structure is not fixed as it is subject to dynamic changes based on semantic and social aspects.

Further approaches to P2P-based search engines are available, too. *YaCy* (`https://yacy.net/de/index.html`) and *FAROO* (`http://www.faroo.com/`) are the most famous examples in this regard. However, although they aim at crawling and indexing the web in a distributed manner, their respective client-sided programs are installed and run on the users' computers. They are not integrated in web servers or web services and thus do not make inherent use of the web topology or semantic technologies for structure-building purposes. Especially, they do not take into account semantic relationships between documents.

3 Conceptual Approach

This section introduces the new concept for decentralised web search mentioned in the introduction. Beforehand, important requirements for its realisation are derived from the previous considerations.

3.1 Requirements

Based on and in continuation to the foregoing considerations and identified shortcomings of current web search engines, the authors argue that a new kind of decentralised search engine for the WWW should replace the outdated, more or less centralised *crawling-copying-indexing-searching* procedure with a scalable, energy-efficient and decentralised *learn-classify-divide-link&guide* method, that

1. employs a learning document grouping process based on a successive category determination and refinement (including mechanisms to match and join several categorisations/clusters of words (terms) and documents) using a dynamically growing or changing document collection (the local knowledge base),

2. is based on a fully decentralised, document management process that largely avoids the copying of documents and therefore conserves bandwidth,

3. allows for search inquiries that are classified/interpreted and forwarded by the same decision process that carries out the grouping of the respective target documents to be found,

4. ensures that the returned results are 100 percent recent,

5. returns personalised results based on a user's locally kept search history yet does not implicitly or explicitly propagate intimate or personal user details to any centralised authority and therefore respects data privacy and contributes to information security and

6. returns results without any commercial or other third-party influences or censorship.

Differing from the approaches cited above, the authors intend to build and maintain a P2P-network whose structures are directly formed by considering content- and context-depending aspects and by exploiting the web's explicit topology (links in web documents). This way, suitable paths between queries and matching documents can be found for any search processes. In the next subsections, the respective concept is presented.

3.2 Preliminary Considerations

In the doctrine of most teachers and based on the users' experience, the today's WWW is considered a client/server system in the classical sense. Web servers offer contents to view or download using the HTTP protocol while every web browser is the respective client accessing content from any server. Clicking on a hyperlink in a web content means to be forwarded to the content, whose address is given in the URL (Uniform Resource Locator) of the link. This process is usually referred to as surfing the web.

Nevertheless, any web server may be regarded as a peer, which is connected to and therefore known by other peers (of this kind) through the addresses stored in the links of the hosted web pages. In such a manner, the WWW can be regarded as a P2P-system (with quite slow dynamics with respect to the addition or removal of peers). However, this system only allows users to surf from web document to web document by following links. Also, these restricted peers lack client functionalities (e.g. communication protocols) offered by web browsers such that there is usually no bidirectional communication between those peers possible (simple forwarded HTTP requests neglected which are mostly initiated by web browsers in the first place).

Moreover, as an integrated search functionality in the WWW is missing so far (the aforementioned restrictions might have contributed to this situation), centralised web search engines have been devised and developed with all their many shortcomings discussed before. These problems will be inherently addressed by the subsequent implementation concept and its implementation.

3.3 Implementation Concept

In order to technically realise the mentioned decentralised web search engine, common web servers shall be significantly extended with the needed components for automatic text processing (clustering and classification of web documents and queries), for the processes of indexing and searching of web documents and for the P2P-network management. A general architecture of this concept can be seen in Fig. 2.

The concept scheme shows that a P2P-component is attached to standard web server. Its peer neighbourhood is induced by the incoming and outgoing links of local web documents. By this means, a new, fully integrated and decentralised web search engine is created.

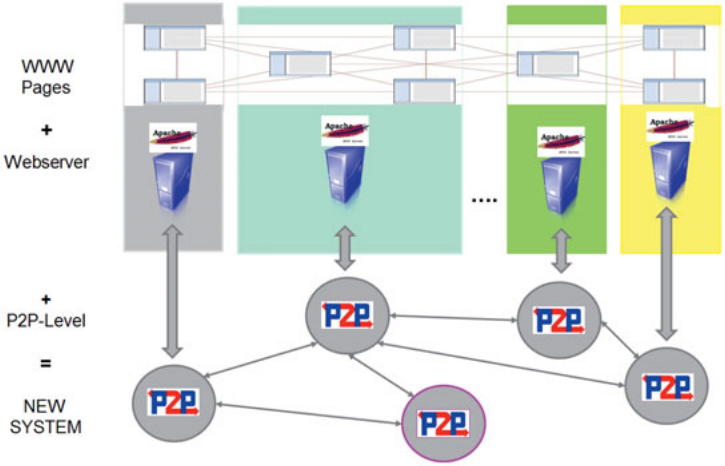


Fig. 2: First concept of a decentralised, integrated web search system

In the following implementation-specific elaborations, the P2P-client software 'WebEngine', which follows this concept, is described.

4 Implementation

As a prototype for this concept, the Java-based P2P-plug-in 'WebEngine' for the popular Apache Tomcat (`http://tomcat.apache.org/`) servlet container and web server with a graphical user interface (GUI) for any standard web browser has been developed. Due to its integration with the web server, it uses the same runtime environment and may access the offered web pages and databases of the server with all related meta-information. The following key points are addressed:

1. A connected, unstructured P2P-system is set up. Initially, the links contained in the locally hosted web pages of the Apache Tomcat server are used for this purpose. Other bootstrap mechanisms as known from [5] and the *PING/PONG*-protocol from *Gnutella* and other P2P-systems may be applied at a later time, too. Note, that

   - HTTP (HTTPS if possible) is used as frame protocol for any communication between the peers.

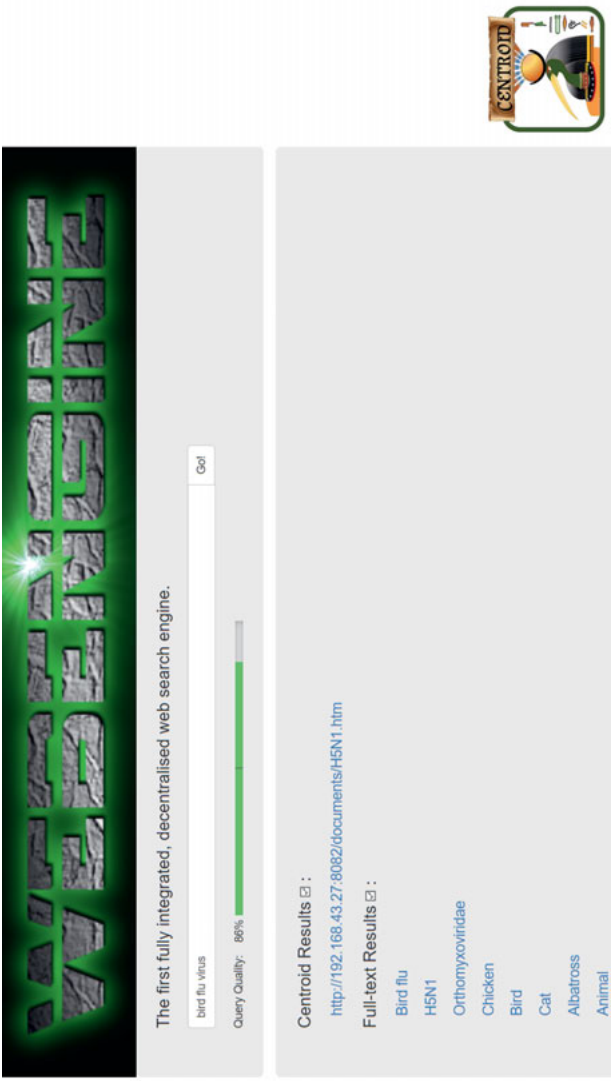


Fig. 3: The graphical user interface of the WebEngine

- A fixed number of connections between the peers will be kept open (although more contactable neighbours are locally stored).
- Furthermore, a time-to-live (TTL) counter is used to limit the number of forwarded messages.

2. All hosted web documents will be indexed in separate index files after applied stopword removal and stemming[1]. The index is updated after every change in one of the hosted web documents. It acts as a cache to answer incoming queries in a fast manner. However, it would be sufficient – as shown in [6] – to only store and use the centroid terms (single descriptive terms found in preferably large co-occurrence graphs to represent queries and whole texts alike) of local documents and the neighbouring peers' document centroids (their topical environment) in order to be able to route and answer queries properly.
3. The plug-in is able to provide a graphical user interface. In particular, a suitable search page for the requesting user (see Fig. 3) is generated.
4. Search results will be generated through a search in the local index files. Queries will also be sent via flooding to all opened connections to neighbouring peers. As they contain a unique message ID, incoming duplicates are discarded. As mentioned before, a TTL counter is applied to limit the number of hops of a query in the network. Responding peers will return their results directly to the originating peer. Multi-keyword search is possible as well.
5. Proliferation mechanisms in the plug-in are integrated to support the distribution of the WebEngine-software over the entire WWW. The P2P-client is able to recognise the peer software on other web servers addressed and offer the download of its own program, in case the peer is not running at the destination yet.

The authors hope that the specified system rapidly changes the way of how documents are accessed, searched for and used in the WWW. The P2P-network may slowly grow besides the current WWW structures and make even use of centralised search engines when needed but may make them more and more obsolete. In this manner, the manipulation of search results through commercial influences will be greatly reduced.

[1] In the first version of the P2P-plug-in, indexing is limited to nouns and names as the carriers of meaning.

4.1 The Software Components

In the previous section, the general architectural concept of the WebEngine has been outlined and depicted in Fig. 2. In a more detail manner, Fig. 4 shows the software components of the WebEngine-client. The blocks in the upper half of the scheme depict the functionality of currently running WebEngine prototype (basic implementation) presented so far with a particular storage facility to maintain the addresses of neighbouring peers.

The *Search Unit* is responsible to index local documents as well as to locally answer, forward and handle search requests issued by users. As mentioned above, in the WebEngine prototype, queries will be sent via flooding to all opened connections to neighbouring peers. However, a replacement of this basic procedure by a single-message, non-broadcasting, universal search protocol (USP), which forwards the search requests based on the centroid distance measure [7] to the target node(s), is currently being integrated.

Analogously, in its lower half, Fig. 4 depicts the components which are yet to be integrated and tested at the time of writing this paper (extension). As it is planned to turn the WebEngine into a powerful 'Librarian of the Web' [6], more sophisticated, centroid-based methods for the local management of document collections (their cataloguing, classification and topical clustering), the semantically induced query interpretation and targeted forwarding to neighbouring peers as well as the decentralised construction and maintenance of hierarchical library structures usually comprising a large number of connected peers have been devised and are carried out by these components. The following components are currently being integrated:

- As the decentralised library management is – in contrast to the top-down algorithm presented in [6] – carried out using random walkers, a particularly structured data unit circulating in the P2P-network, in the actual implementation of the WebEngine, a special *RW-Management* unit is added that carries out a special random walker protocol (RWP). Its working principles and methods are described in detail in [8]. Also, the mentioned USP will be additionally extended such that random walkers will not only be used to generate the tree-like library but to perform search operations in it as well. For this purpose, special random walkers with query data as their payload will be sent out. This payload is matched and exchanged with other random walkers in the network when appropriate until matching documents are found.

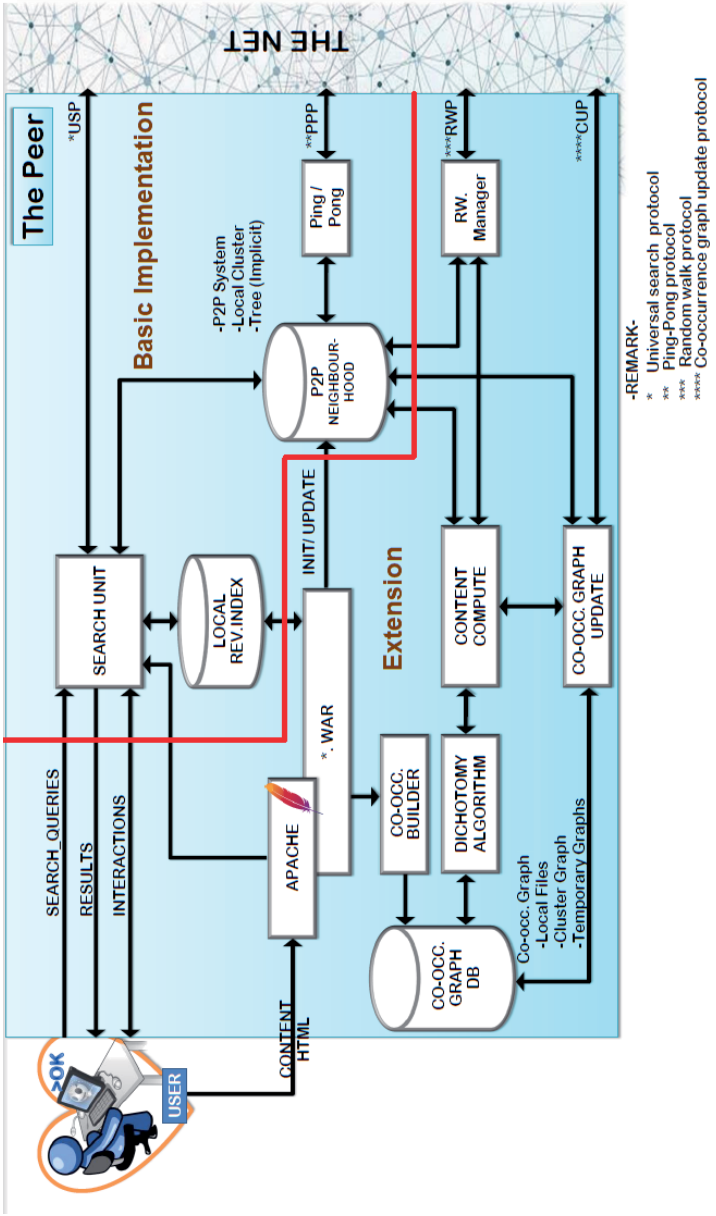


Fig. 4: The WebEngine's internal structure

- The processing of random walker data is performed by the *Content Compute* unit, which needs to access the term co-occurrence graph databases (e.g. constructed and stored using the graph database system Neo4j (`https://neo4j.com/`)), which in turn contain the co-occurrence graph data of
  1. each web document offered by the local WWW server,
  2. the local term cluster which the node is responsible for,
  3. temporary operations of the random walkers to build or update the hierarchical library structures.
- The remaining units support various operations on co-occurrence graphs:
  - In order to construct co-occurrence graphs, a co-occurrence graph builder is implemented in a separate unit.
  - The needed document clustering (see [6]) is carried out by the *Dichotomy Algorithm* unit.
  - the exchange of co-occurrence information between peers is supported by the co-occurrence update protocol (CUP) controlled by a respective separate unit.

As the WebEngine makes heavy use of graph databases for the storage and retrieval of co-occurrences, their role is discussed in the next subsection, too.

4.2 Graph Databases

When taking a look at the WebEngine's architecture and functionalities from a technological point of view, it becomes obvious that it is necessary to be able to manage large graph structures efficiently and effectively. Graph database systems such as Neo4j (`https://neo4j.com/`) are specifically designed for this purpose. Also, they are well-suited to support graph-based text mining algorithms. This kind of databases is not only useful to solely store and query the herein discussed co-occurrence graphs with the help of its property graph model, nodes (terms) in co-occurrence graphs can be enriched with additional attributes such as the names of the documents they occur in as well as the number of their occurrences, too. Likewise, the co-occurrence significances can be persistently saved as edge attributes. Graph databases are thus an immensely useful tool to realise the herein presented technical solutions. Therefore, the WebEngine makes especially use of embedded Neo4j graph databases for the storage, traversal and clustering of co-occurrence graphs and web documents.

5 Conclusion

This paper presented the concept of a novel, decentralised web search engine as well as its P2P-based implementation, called the 'WebEngine'. Its features and software components have been elaborated on in detail. Specifically, it utilises existing web technologies such as web servers and links in web documents to create a decentralised and fully integrated web search system. As an extension to the well-known Apache Tomcat servlet container, the WebEngine is easy to install and maintain for administrators. Internally, it makes use of modern graph-based text analysis techniques. Therefore, a decentralised web search system is created that – for the first time – combines state-of-the-art text analysis techniques with novel, effective and efficient search functions as well as methods for the semantically oriented P2P-network construction and management. The basic implementation of the WebEngine is currently being greatly enhanced by numerous additions which will turn it into a modern 'Librarian of the Web'.

References

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

[2] A. Broder et al. Graph Structure in the Web: Experiments and Models. In *Computer Networks: The International Journal of Computer and Telecommunications Networking*, pages 309–320, Amsterdam, The Netherlands, 2000.

[3] Website of the DFG-project 'Search for text documents in large distributed systems'. `http://gepris.dfg.de/gepris/projekt/5419460`, 2009.

[4] S. Milgram. The Small World Problem. In *Psychology Today*, 2:60–67, 1967.

[5] P. Kropf, J. Plaice, and H. Unger. Towards a Web Operating System. In *Proc. of the World Conference of the WWW, Internet and Intranet (WebNet'97)*, pages 994–995, Toronto (CA), 1997.

[6] M. M. Kubek and H. Unger. Towards a librarian of the web. In *Proc. of the 2nd International Conference on Communication and Information Processing*, ICCIP '16, pages 70–78, New York, NY, USA, ACM, 2016.

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

[8] M. M. Kubek and H. Unger. A Concept Supporting Resilient, Fault-tolerant and Decentralised Search. In *Autonomous Systems 2017*, Fortschritt-Berichte VDI. 10(857):20–31, VDI-Verlag Düsseldorf, 2017.

On Evolving Text Centroids

Herwig Unger and Mario M. Kubek

Chair of Communication Networks, University of Hagen, Germany

Abstract: Centroid terms are single words that semantically and topically characterise text documents and thus can act as their very compact representation in automatic text processing tasks. In this paper, a novel brain-inspired approach is presented to first simplify the determination of centroid terms and second to generalise the underlying concept at the same time. As the precision of the centroid-based text representation is improved by this means as well, new applications for centroids are derived, too. Experimental results obtained confirm the validity of this new approach.

1 Motivation and Definitions

From the literature, a plentitude of approaches to compare and categorise text documents are known [1, 2]. Regarding their exactness and computational costs they differ significantly. Their major disadvantage is not to utilise the general knowledge a human reader may have. Furthermore, these methods are usually not able to process and consider documents as sequences of words, whereby different sequences may significantly determine a text's meaning as well as quality. To solve these problems, the concept of text centroids as defined in [3] will be extended to refine the capabilities of text categorisation. To this end, some fundamentals are needed and shall be summarised shortly.

Let X be the set of pairwise different terms as found in a text corpus, and E the set of tuples (x_i, x_j) whose two terms x_i, x_j appear together in a sentence or paragraph of the text corpus. The connected subgraph G of the graph (X, E) is then called *co-occurrence graph* of the corpus. The number of a tuple's (x_i, x_j) co-occurrences in the corpus is denoted by $sig(x_i, x_j)$. It indicates a weight of the corresponding edge in G, whereas its reciprocal $d(x_i, x_j) = 1/sig(x_i, x_j)$ is used to define a distance between the terms x_i, x_j [3].

Given a text document D in form of an ordered set of not necessarily different words $w_1, w_2, \ldots, w_n$ with all $w_i \in X$. In [3], for D a centroid χ was defined as its node with the lowest average distance to any other node reachable in the

subgraph of G induced by the elements of D. Properties and applications of centroids were discussed in [4–6]. It was found that, due to the discrete character of G, a 'centre of words' may not coincide with an element of D, just as the 'centre of mass' of a multi-body mechanical system does not necessarily coincide with one of the bodies. In order to improve the precision of text representation by centroids, in the sequel their position relative to the co-occurrence graph will be generalised.

This is motivated by the observation that for a simple document with just two co-occurring words the centroid's position would be expected to lie exactly at the centre of the edge connecting them. The original definition from [3] does not allow for this position, but assigns one of the two words to be the centroid. This effect causes imprecisions and misinterpretations in bigger settings.

Furthermore, the approach of [3] (and other methods known from literature) requires a text's complete processing within a single step. This contradicts the way in which humans read a text word by word from its very beginning and incrementally acquire the information contained. This is the reason why text documents D are considered here as ordered sets, and their centroids are constructed by a gradual process along the given orderings.

2 Centroids of Text Documents

2.1 Centroid Positions

As indicated above, generalised positions π of centroids in a graph shall be introduced, which either coincide with a node or lie somewhere on an edge connecting two nodes.

A **position** π in a graph $G = (X, E)$ is defined as a tripel

$$\pi = (x, y, \lambda)$$

with $x, y \in X$, $(x, y) \in E$ and the displacement λ, $0 \leq \lambda < d(x, y)$, of π from x into the direction of y (see also Fig. 1). For the special case of $x = y$ and, thus, $\lambda = 0$ the position $\pi = (x, x, 0)$ may be denoted just by x. Note that a position may be described by two different expressions, viz. $\pi = (x, y, \lambda) = (y, x, (d(y, x) - \lambda))$.

Distances $d(\pi_1, \pi_2)$ between any two positions $\pi_1 = (x_1, y_1, \lambda_1)$ and $\pi_2 = (x_2, y_2, \lambda_2)$ may now be defined and calculated using the following rules:

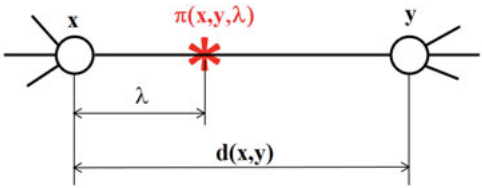


Fig. 1: Position π in a graph

1. **Both positions coincide with nodes.**
   $d(\pi_1, \pi_2) = d(x_1, x_2)$ as $(x_1 = y_1) \wedge (x_2 = y_2) \wedge (d_1 = d_2 = 0)$

2. **Both positions are located on a common edge, with the displacements oriented in equal directions.**
   $d(\pi_1, \pi_2) = \mid \lambda_1 - \lambda_2 \mid$ as $(x_1 = x_2) \wedge (y_1 = y_2)$

3. **Both positions are located on a common edge, with the displacements oriented in opposite directions.**
   $d(\pi_1, \pi_2) = \mid d(x_1, x_2) - \lambda_1 - \lambda_2 \mid$ as $(x_1 = y_2) \wedge (y_1 = x_2)$

4. **The two positions are located on different edges.**
   Then, the distance is the length of the shortest path between π_1 and π_2 along the $s \geq 1$ intermediate nodes $p_1, p_2, \ldots, p_s \in X$:

$$d(\pi_1, \pi_2) = d(\pi_1, p_1) + \sum_{k=1}^{s-1} d(p_k, p_{k+1}) + d(p_s, \pi_2)$$

With these basic definitions, the concept of centroids can now be generalised.

2.2 Evolution of Centroids

A **generalised centroid** ξ_D of a text document D is introduced as the position $\xi_D = (x, y, \lambda)$ in G, which has the minimum average distance to all nodes representing the document's words $w_1, w_2, \ldots, w_n \in D$. It is easy to see that the definition of ξ refines the one of χ, since ξ may not only be situated on a particular node of G.

Nevertheless, at first glance its seems to be difficult to determine generalised centroids by a fast algorithm as shown, for instance, in [7]. The situation becomes easier to deal with, however, when an evolutionary approach is employed, i.e. when a document's centroid is calculated in an iterative manner by taking one word $w_i \in D$ after the other into account.

Such an **evolution of centroids** $\xi_D^{(i)}$, $i = 1, 2, \ldots, \mid D \mid$, can be carried out by the following algorithm:

1. *Initialisation*
   Set $i := 1$ and $\xi_D^{(1)} := w_1$ (or respectively $(w_1, w_1, 0)$).
2. **REPEAT**
   Consider the next word w_{i+1} of D.
   Calculate the shortest path $\{\xi_D^{(i)}, p_1, p_2, \ldots, p_s\}$ with $p_1, \ldots, p_s \in X$ and $p_s = w_{i+1}$ and determine its length $d(\xi_D^{(i)}, w_{i+1})$.
3. Determine $\delta = f(i) \cdot d(\xi_D^{(i)}, w_{i+1})$ with a damping factor f depending on i in order to weigh the influence of the $(i+1)$-st word in relation to the text processed so far.
4. Let $\xi_D^{(i)} = (x_b, x_e, \lambda_i)$.
   Search along the shortest path $\{\xi_D^{(i)}, p_1, p_2, \ldots, p_s\}$ with $s \geq 1$, $p_1, \ldots, p_s \in X$ and $p_s = w_{i+1}$ for the smallest k where for the first time it holds $d(\xi_D^{(i)}, p_k) \geq \delta$.
   Execute the alternative applicable:
   a) If $(s = 1) \wedge (x_a = x_b)$, then set $\xi_D^{(i+1)} := (x_a, p_1, \delta)$.
   b) If $(s = 1) \wedge (p_1 = x_b)$, then set $\xi_D^{(i+1)} := (x_b, x_e, (\lambda_i - \delta))$.
   c) If $(s = 1) \wedge (p_1 = x_e)$, then set $\xi_D^{(i+1)} := (x_b, x_e, (\lambda_i + \delta))$.
   d) If $s > 1$, then set $\xi_D^{(i+1)} := (p_{k-1}, p_k, (\delta - d(\xi_D^{(i)}, p_{k-1})))$.
5. Increment $i := i + 1$.
6. **UNTIL** $i > n$

Fig. 2 shows the positions of the centroids evolving when reading and processing one word of a text document D after the other in the order given. Therefore, these generalised centroids can be called *evolving centroids* as well. Use and properties of the *evolving centroids* shall now be considered in detail.

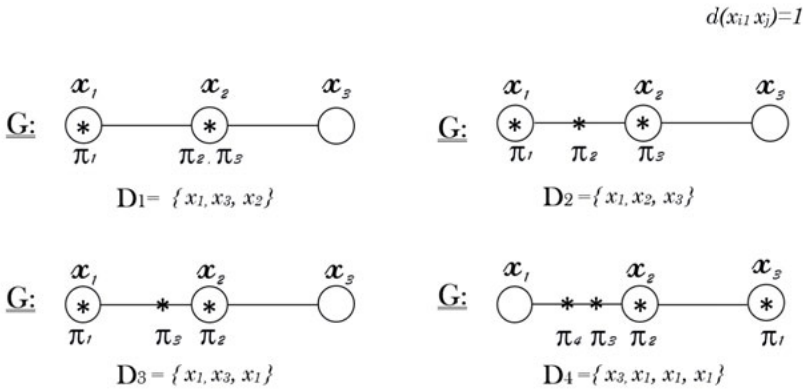


Fig. 2: Positions of centroids π_s resulting from word-by-word processing of documents D_s containing several permutations of three words x_1, x_2, x_3

2.3 Interpretation of Centroid Traces

Applying the above described algorithm results in a consecutive calculation of a text-representing centroid. While in former publications like [3] only the centroid itself was important, now the consecutive, intermediate states can and shall be considered, too. It is obvious to call

$$\{\xi_D^{(1)}, \xi_D^{(2)}, \ldots, \xi_D^{(n-1)}\}$$

the **centroid trace of document** D. Each $\xi_D^{(i)}$ on this trace might be interpreted as the centroid of a partially read document, which – by definition – points to one or two words from the set of words $\{w_1, \ldots, w_i\} \in D$.

It is expected that the length and direction of the intermediate steps from $\xi_D^{(1)}$ to ξ_D are quite randomly distributed and that a selection of this information for a certain number of steps might (possibly) be employed as a fingerprint of the analysed document. Further use cases of centroid traces are discussed in the next section.

3 Experimental Results and Discussion

For all of the exemplary experiments discussed herein, 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), co-occurrences on sentence level have been extracted. Their significance values have been determined using the Dice coefficient [8]. For the document analysis, 30 English Wikipedia articles[1] from an offline English Wikipedia corpus downloaded from `http://www.kiwix.org` have been used. The corpus used to create the reference co-occurrence graph G consisted of 100 randomly selected Wikipedia articles (including the mentioned 30 ones) from the same source.

3.1 Centroids of Wikipedia Articles

In order to get a first impression of the generalised centroids' quality in terms of whether they are actual useful representatives of documents, Table 1 presents those centroid terms of the mentioned 30 English Wikipedia articles. The damping factor f depending on the i-th read word of these articles has been set to $1/\sqrt{i}$. Furthermore, this table also lists the previously determined centroid terms [4] of the same documents using the initially described method of computation [7] inspired by the 'centre of mass'. They are referred to as classic centroids. This way, a direct comparison between these types of centroids is possible. In both cases, the articles' 25 most frequent terms have been taken into account for the centroid computation.

It can be seen that almost all centroids properly represent their respective articles. In case of the herein presented generalised centroids, it can be noted that they are in fact related to the classic centroids. Yet, the generalised centroids are more specific (e.g. in case of the article 'Malaria') and meaningful due to the way they have been computed.

Unless such a centroid resides directly on a node, there are in each case two distinct terms (connected in G) to characterise it along with the displacement λ. In future contributions, the influence of different damping factors f on the resulting centroids will be investigated. Here, logarithmic or other non-linear functions could be factored in.

[1] Interested readers can download these articles (1.5 MB) from: `http://www.docanalyser.de/ec-articles.zip`

Table 1: Centroids of 30 Wikipedia articles

Title of Wikipedia Article	Generalised Centroid	Classic Centroid Term
Art competitions at the Olympic Games	[medal, artist, 3.92]	sculpture
Tay-Sachs disease	[carrier, gene, 6.08]	mutation
Pythagoras	[influence, Pythagoras, 3.16]	Pythagoras
Canberra	[Canberra, capital, 3.17]	Canberra
Eye (cyclone)	[cycle, eyewall, 6.87]	storm
Blade Runner	[Ridley Scott, film, 0.40]	Ridley Scott
CPU cache	[memory, cache, 0.08]	cache miss
Rembrandt	[Amsterdam, Rembrandt, 0.72]	Louvre
Common Unix Printing System	[print, system, 1.73]	filter
Psychology	[psychology, behavior, 0.59]	psychology
Religion	[religion, belief, 1.15]	religion
Universe	[universe, form, 2.03]	shape
Mass media	[media, term, 2.48]	database
Rio de Janeiro	[Rio, zone, 0.50]	sport
Stroke	[stroke, factor, 2.51]	blood
Mark Twain	[fiction, Twain, 1.16]	tale
Ludwig van Beethoven	[Beethoven, life, 2.37]	violin
Oxyrhynchus	[republic, Egypt, 2.47]	papyrus
Fermi paradox	[paradox, civilization, 0.57]	civilization
Milk	[milk, cow, 0.52]	dairy
Corinthian War	[Greece, Sparta, 3.48]	Sparta
Health	[WHO, health, 4.48]	fitness
Tourette syndrome	[tic, child, 0.71]	tic
Agriculture	[crop, production, 5.51]	crop
Finland	[island, Finland, 2.21]	tourism
Malaria	[malaria, parasite, 1.32]	disease
Fiberglass	[fiber, process, 5.04]	fiber
Continent	[continent, Europe, 1.85]	continent
United States Congress	[Senate, member, 1.07]	Senate
Turquoise	[crystal, property, 3.56]	turquoise

3.2 Centroid Traces of Wikipedia Articles

In Table 2, the centroid trace of the article 'Milk' is presented. The trace clearly shows that while reading the text word by word (those words must be among the 25 most frequent terms of the article) the centroid dynamically changes.

Table 2: Centroid trace of article 'Milk'

[[milk, milk, 0.0] → [milk, milk, 0.0] → [milk, glass, 4.48] → [milk, glass, 0.06] → [milk, glass, 0.03] → [milk, glass, 0.02] → [milk, glass, 0.01] → [milk, source, 2.75] → [milk, food, 1.74] → [milk, food, 1.19] → [milk, food, 0.83] → [milk, protein, 0.97] → [milk, protein, 4.00] → [protein, parasite, 1.71] → [protein, milk, 0.13] → [protein, milk, 1.45] → [protein, milk, 0.34] → [protein, milk, 1.54] → [protein, milk, 2.44] → [protein, milk, 3.11] → [protein, milk, 4.62] → [milk, product, 0.69] → [milk, product, 0.55] → [milk, product, 0.43] → **[milk, cow, 0.52]**]

It is also noteworthy that the centroid is almost always related to the respective article's main topic but touches several relevant thematic aspects as it evolves. These subtopics are of interest when it comes to suggesting related search words for the expansion of queries in information retrieval. A similar outcome is to be seen in Table 3 for the article 'Fermi paradox'.

As depicted in these examplary results, the centroid traces inherently reflect the topical organisation of the respectively analysed texts. This makes it possible to compare two texts regarding their structural and semantic similarity in a new way: The distance of two intermediate centroids indicates the degree of their semantic closeness. If the position of these centroids in the centroid traces is taken into account as well when doing so, a structural similarity of those texts can be detected. For this approach to work, the overlap of their respective sets of words can be low or even empty.

Table 3: Centroid trace of article 'Fermi paradox'

[[Fermi, Fermi, 0.0] → [Fermi, paradox, 0.77] → [Fermi, paradox, 0.32] → [Fermi, paradox, 0.71] → [paradox, civilization, 2.36] → [paradox, civilization, 0.95] → [paradox, civilization, 0.59] → [paradox, civilization, 2.42] → [paradox, evidence, 0.18] → [paradox, civilization, 1.69] → [paradox, equation, 1.80] → [equation, planet, 2.54] → [equation, paradox, 1.04] → [equation, paradox, 1.83] → [equation, paradox, 2.40] → [paradox, civilization, 1.05] → [paradox, evidence, 0.50] → [paradox, probe, 0.96] → [paradox, emission, 1.27] → [paradox, emission, 0.74] → [paradox, life, 1.05] → [paradox, life, 0.59] → [paradox, life, 0.47] → [paradox, evidence, 0.72] → **[paradox, civilization, 0.57]**]

3.3 Discussion

Based on these findings and as suggested before, a document's centroid trace might be used as its (individual) fingerprint in further text processing tasks, e.g. as a means for plagiarism detection. Also, a recommender system like DocAnalyser [9] could make use of these traces to find and suggest similar and related web documents. Furthermore, the visualisation of these traces in form of curves along with the used reference co-occurrence graph G will give interested users another possibility to explore and examine the structural and semantic closeness of text documents. Even though two traces might not even share one intermediate centroid, (at least partly) side-by-side running curves and similar curve shapes could suggest an existing text similarity, too. These application scenarios will be investigated in more detail in future contributions.

4 Conclusion

A novel, brain-inspired method for the computation of text-representing centroids has been presented. At the same time, the classic concept of centroid terms has been generalised such that a much more precise determination of centroids is possible. The derived concept of centroid traces allows for their usage in manifold interesting applications in the field of automatic text processing.

Foremost, these traces can be applied to determine the degree of semantic similarity of text documents in a new way while considering their structure, too. Future research will investigate and evaluate this approach in greater detail.

References

[1] Paaß, G.: Document Classification, Information Retrieval, Text and Web Mining. In: Handbook of Technical Communication. pp. 141–188, Walter de Gruyter (2012)
[2] Mihalcea, R., Radev D.: Graph-based Natural Language Processing and Information Retrieval. Cambridge University Press (2011)
[3] Kubek, M., Unger, H.: Centroid terms as text representatives. In: Proceedings of the 2016 ACM Symposium on Document Engineering DocEng '16. pp. 99–102, ACM, New York, NY, USA (2016)
[4] Kubek, M., Unger, H.: Centroid terms and their use in natural language processing. In: Autonomous Systems 2016. Fortschritt-Berichte VDI, Reihe 10 Nr. 848, pp. 167–185, VDI-Verlag Düsseldorf (2016)
[5] Kubek, M., Unger, H.: Towards a librarian of the web. In: Proceedings of the 2nd International Conference on Communication and Information Processing (ICCIP 2016) pp. 70–78, ACM, New York, NY, USA (2016).
[6] Kubek, M., Böhme, T., Unger, H.: Empiric experiments with text representing centroids. Lecture Notes on Information Theory 5(1), 23–28 (2017)
[7] Kubek, M., Böhme, T., Unger, H.: Spreading activation: A fast calculation method for text centroids. In: Proceedings of the 3rd International Conference on Communication and Information Processing (ICCIP 2017). ACM, New York, NY, USA (2017).
[8] Dice, L. R.: Measures of the amount of ecologic association between species. Ecology 26(3),297–302 (1945)
[9] Kubek, M.: DocAnalyser - Searching with web documents. In: Autonomous Systems 2014. Fortschritt-Berichte VDI, Reihe 10 Nr.835, pp.221–234, VDI-Verlag Düsseldorf (2014)

Addendum

WebEngine Search Results

This addendum presents and explains four exemplary search results generated by the WebEngine, the introduced, decentralised and librarian-inspired web search engine. Here, the focus is set on the description of the visible components of the respective result page (presented in form of screenshots). The documents searched for stem from an offline English Wikipedia corpus downloaded from `http://www.kiwix.org`. The individual (on each WebEngine-peer) co-occurrence graphs for query evaluation and centroid determination have been constructed using these articles as well.

Query 1: 'bird flu'

Fig. 1 depicts the results of the query: 'bird flu'. The indicator 'query quality' underneath the query input field suggests the user that the query is with 87% of high quality. The bar's green color underlines this evaluation in form of a meaningful visual feedback. This evaluation is based on the measure *diversity* of a given document or query (see Chapter 3) that indicates how general (rather unspecific) or topic-oriented (rather specific) the analysed content is. The centroid term of the query is 'bird'. Therefore, the search for documents whose centroid term is 'bird' returns under 'Centroid Results' the listed two links to the highly relevant articles 'bird' and 'bird flu'. Also, the list of the top-ranked full-text results is shown which contains those two results as well. Furthermore, the user has the possibility to expand and collapse the two result categories.

Fig. 1: WebEngine search results of the query: 'bird flu'

Query 2: 'bird flu virus'

For query 2, the previous query 1 has been expanded with the term 'virus' and therefore made more precise. Fig. 2 depicts the results of this expanded query 2: 'bird flu virus'. The query quality almost stayed the same, but its centroid changed to 'H5N1'. Accordingly, the article 'H5N1' has been returned as the only and higly relevant centroid search result (due to the topical closeness of the query, it is to be expected that the number of centroid search results decreases). The list of top-ranked full-text results still contains links to matching and likewise relevant articles.

Fig. 2: WebEngine search results of the query: 'bird flu virus'

Query 3: ‘cat crop’

Fig. 3 depicts the results of the topical unspecific query: ‘cat crop’. As its intentionally chosen terms are semantically unrelated, the determined query quality is with only 43% rather low as well. The bar’s red color intuitively suggests the user to reformulate the query, too. Accordingly, no centroid search results have been found and the list of full-text results contains merely a mixture of links to articles that are relevant to either ‘cat’ or ‘crop’.

Fig. 3: WebEngine search results of the query: 'cat crop'

Query 4: 'khmer rouge'

Fig. 4 depicts the results of the query: 'khmer rouge'. The query quality is with 92% very high. The only centroid search result (article 'Cambodia') suggests that the determined query's centroid is likewise 'Cambodia'. Taken into account the historical origins of the Khmer Rouge, this centroid is the perfect representative for both the query and the returned article alike. The list of top-ranked full-text results contains links to topically relevant articles as well.

Fig. 4: WebEngine search results of the query: 'khmer rouge'

Thomas Böhme is a professor for mathematics at the Institute for Mathematics of the Technische Universität Ilmenau (TU Ilmenau). He obtained his PhD with a work on spatial representations of finite graphs from the Technische Hochschule Ilmenau in 1988 and his habilitation with a work on cycles in embedded graphs from TU Ilmenau in 1999. His main research focus is on game theory, especially learning in repeated games. Second, he is conducting research in the field of graph theory. Also, he is interested in distributed algorithms.

Mario M. Kubek is a researcher at the FernUniversität in Hagen, from which he received his PhD with a thesis on locally working agents to improve the search for web documents in 2012 and his habilitation with a work on how to create a fully integrated, decentralised and librarian-inspired web search engine in 2018. His research focus is on natural language processing, text mining and semantic information retrieval in large distributed systems. He is also interested in mobile computing environments and contextual information processing.

Herwig Unger received his PhD with a work on Petri Net transformation in 1994 from the Technische Universität Ilmenau and his habilitation with a work on large distributed systems from the University of Rostock in 2000. Since 2006, he is a full professor at the FernUniversität in Hagen and the head of the Chair of Communication Networks. His research interests lie in the areas of self-organization, adaptive and learning systems, Internet algorithms, simulation systems as well as information retrieval in distributed systems.

INGENIEUR.de
TECHNIK - KARRIERE - NEWS

powered by VDI Verlag

Das TechnikKarriereNews-Portal für Ingenieure.

Testen Sie Ihr Gehalt.

Mit dem Gehaltstest für Ingenieure überprüfen Sie schnell, ob Ihr Einkommen den marktüblichen Konditionen entspricht. Er zeigt Trends auf und gibt Ihnen Orientierung, z. B. für Ihr nächstes Gehaltsgespräch. Und Ihre individuelle Auswertung können Sie jederzeit bequem aktualisieren.

JETZT KOSTENFREI TESTEN UNTER:
WWW.INGENIEUR.DE/GEHALT

Die Reihen der Fortschritt-Berichte VDI:

1 Konstruktionstechnik/Maschinenelemente
2 Fertigungstechnik
3 Verfahrenstechnik
4 Bauingenieurwesen
5 Grund- und Werkstoffe/Kunststoffe
6 Energietechnik
7 Strömungstechnik
8 Mess-, Steuerungs- und Regelungstechnik
9 Elektronik/Mikro- und Nanotechnik
10 Informatik/Kommunikation
11 Schwingungstechnik
12 Verkehrstechnik/Fahrzeugtechnik
13 Fördertechnik/Logistik
14 Landtechnik/Lebensmitteltechnik
15 Umwelttechnik
16 Technik und Wirtschaft
17 Biotechnik/Medizintechnik
18 Mechanik/Bruchmechanik
19 Wärmetechnik/Kältetechnik
20 Rechnerunterstützte Verfahren (CAD, CAM, CAE CAQ, CIM ...)
21 Elektrotechnik
22 Mensch-Maschine-Systeme
23 Technische Gebäudeausrüstung

ISBN 978-3-18-386310-5