Topic Detection Based on the PageRank's Clustering Property

Mario Kubek, Herwig Unger
Faculty of Mathematics and Computer Science
FernUniversität in Hagen
Hagen, Germany
Email: kn.wissenschaftler@fernuni-hagen.de

Abstract: This paper introduces a method to cluster graphs of semantically related terms from texts using PageRank calculations for use in the field of text mining, e.g. to automatically discover different topics in a text corpus. It is evaluated by providing empirical results of tests by applying this method on real text corpora. It is shown that this application of the PageRank formula realizes suitable clustering such that the mean similarity between the terms in the clusters reaches a high level. A special state transition in the mean term similarity is discussed when analysing texts with stopwords.

1 Introduction and Motivation

In the field of text mining clustering has gained growing importance over the last years e.g. for the automatic detection and tracking of topics. Many authors [Al02][Kl02][Wa06] have contributed to this specific research area, for instance with novel methods to identify new and relevant terms in text streams. A topic in this sense can be regarded as a set of terms that significantly co-occur in a text corpus or in single texts, whereby the latter can also cover many topics (main topics and subtopics) themselves. In this paper, a clustering method is introduced to identify different topics in single texts with the help of PageRank (PR) calculations. PageRank is an algorithm developed by Brin and Page [Pa98] that measures the importance of a web page based on the importance of its incoming links. Popular web search engines like Google return search results according to the PageRank of indexed web pages. The PageRank includes factors to cover the behavior of a random web surfer. A user follows a random link on a web page with a probability of around 0.90 (damping factor η in formula 1) and visits another page not linked by this page with a probability of around 0.10. The PageRank of a page corresponds to the number of page visits of users. The more users visit a page the more important it is. For the PageRank calculation the web pages are represented by nodes and the links between them are represented by directed edges, forming a directed graph. Besides other possibilities the PageRank PR_i of a page i can be iteratively calculated by applying the following formula:

$$PR_i = (1-\eta) + \eta \sum_{j \in J} \frac{PR_j}{|N_j|} \qquad (1)$$

The Set J represents the pages linking to page i and $|N_j|$ is the out-degree of page j. In the last years, many solutions for distributed PageRank computations have been published [Zh05][Sa03][Is09] in order to address the shortcoming that originally the whole network graph needs to be considered. In [So10] extended methods for distributed PageRank calculation including network parameters based on random walks are discussed and empirically evaluated.

Herein, the PageRank formula (1) is applied on graphs of semantically related terms of texts in order to identify strongly connected terms in them. It is shown, that if these terms are iteratively removed from such a graph, it is separated into clusters (components) containing semantically similar terms. Moreover, these clusters represent the diverse topics covered within the text.

Therefore, the paper is organized as follows: the next section explains the methodology used whereby basics on text mining and methods of statistical co-occurrence analysis are given. Then the clustering algorithm as the main contribution of this paper is presented and the expected results are discussed. Section three focuses on the conducted experiments applying this algorithm while discussing the various input parameters used and elaborating on the results such as the number of clusters, cluster sizes (number of terms in a cluster) and term similarities within the clusters during the iterations of the algorithm. Section four concludes the paper and provides a look at options to employ this algorithm in applications for topic detection and tracking.

2 Description of Methodology

2.1 Text Mining

Text mining comprises a set of methods to analyse mostly unstructured text data by linguistic and statistical means in order to detect semantic structures within them. Text clustering for instance is a typical task in text mining [He06]. Its aim is to take a set of elements like terms or documents and detect subsets (clusters) that contain similar elements but which are dissimilar to elements of other subsets. A first step to determine similar terms in a text is usually the detection of semantic relations between terms by applying methods for statistical co-occurrence analysis.

The occurrence of two terms in a text section next to each other is called co-occurrence or syntagmatic relation [He06]. Co-occurrences that appear above chance are called significant co-occurrences. The most prominent kinds of co-occurrences are term pairs that occur as immediate neighbours and term pairs that occur together in a sentence. The following considerations will focus on the latter ones. There are several well-established measures to calculate the statistical significance of such term pairs by assigning them a significance value. If this value is above a preset threshold the co-occurrence can be regarded as significant and a semantic relation between the involved terms can often be derived from it. Rather simple co-occurrence measures are for instance the frequency count of co-occurring terms and the similar Dice and Jaccard coefficients [Bu06].

More advanced formulas rely on the expectation that two terms are statistical independent which is a usually inadequate hypothesis. They then calculate the deviation from their observation of real corpus data to their expectation. A significant deviation leads therefore to a high co-occurrence value. Co-occurrence measures based on this hypothesis are for instance the mutual information measure [Bu06], the poisson collocation measure [Qu02] and the log-likelihood ratio [Du94]. Stimulus-response experiments show that co-occurrences found to be significant by these measures correlate well with term associations by humans. The co-occurrences of a text can be considered a graph of semantically related terms (with the terms as the nodes and the significance values as the edges) that has the small-world property, meaning, it comprises of definable groups of strongly connected nodes, whereby a member of such a group can also be a member of other groups of that kind. Such co-occurrence graphs are the basic input for the algorithm described herein.

The set of significant co-occurrences of a term can be represented as a vector that can be regarded as its semantic context. For clustering purposes it is further necessary to determine the semantic similarity between all pairs of terms in a cluster. The comparison of these co-occurrence vectors is therefore a sensible possibility to gain the similarity for all pairs of terms. This approach is based on the assumption that similar terms have similar contexts. To calculate this term-term-similarity, measures that operate on vectors such as the Euclidian distance, the dot product (for normalized vectors) or the cosine similarity can be applied. The latter is defined in formula 2 and is used in the following considerations to obtain the similarity between term *a* and term *b* by comparing their co-occurrence vectors $\overrightarrow{t_a}$ and $\overrightarrow{t_b}$:

$$sim_{cos}(\overrightarrow{t_a}, \overrightarrow{t_b}) = \frac{\sum_{l=1}^{n} sig(t_a, t_l) * sig(t_b, t_l)}{\sqrt{\sum_{l=1}^{n} sig(t_a, t_l)^2} * \sqrt{\sum_{l=1}^{n} sig(t_b, t_l)^2}} \qquad (2)$$

The vector $\overrightarrow{t_a}$ contains all significant co-occurrences $sig(t_a, t_l)$ of term *a*. The same applies to term *b*. These values have to be calculated for all term pairs in order to obtain a matrix that contains similarity values for each term pair in the text. As this formula takes the contexts of terms into account, the calculated similarity value is more meaningful than just the co-occurrence value which of course cannot be calculated for each term pair. Particularly, terms could have a high similarity to each other even if they do not co-occur in the text. The resulting term-term-matrix contains the term-term-similarities in the range of 0 and 1 for all term combinations in the text, whereby values near 0 indicate a low and values near 1 a high similarity. With the help of these values it is also possible to determine the mean term-term-similarity inside a cluster of terms.

2.2 The Algorithm

In this section, the basic algorithm to cluster co-occurrence graphs based on PageRank calculations is presented. It is a hierarchical clustering algorithm that tries to separate components of co-occurrence graphs and can therefore be regarded as a divisive clustering algorithm.

It is possible to implement this algorithm using random walkers in distributed systems.

1. *Remove stopwords and apply stemming algorithm on all terms in the text. (Optional)*
2. *Calculate the significant co-occurrences for all terms in the text and save them in an adjacency matrix in order to gain the initial co-occurrence graph.*
3. *Calculate the term-term-matrix containing the term-term-similarities for all term combinations in the text based on formula 2.*
4. *Determine all separate components in the (remaining) co-occurrence graph and print the terms contained in them.*
5. *If there are components with more than 1 term in them continue, otherwise terminate the algorithm.*
6. *Calculate PageRanks for all terms in the co-occurrence graph using formula 1.*
7. *For all terms i in the co-occurrence graph check if the PageRank of term i is greater than the PageRank of their neighbours. If yes, mark term i for removal.*
8. *Remove all marked terms from the co-occurrence graph.*
9. *Go to step 4.*

The expected results of this algorithm mainly depend on the input parameters given. It can be assumed that very common terms like stopwords will receive a high PageRank. Therefore, it is very likely that these terms will be ruled out of the co-occurrence graph first. Based on this assumption, the initial co-occurrence graph is unlikely to be separated into clusters during the first iterations of the algorithm, because those terms are related to a large number of other terms in the given text. Therefore, the mean term-term-similarity inside the first gained cluster(s) should not be high and the number of terms in these clusters will be quite high. When the number of clusters increases the number of terms in the clusters should decrease and their mean term-term-similarity should increase. In case that the initial co-occurrence graph does not contain stopwords it is likely that it is already separated in clusters. In this basic algorithm, clusters will be separated until there are only clusters containing one term left. Therefore, the mean term-term-similarity will be near 1 when the number of terms in the clusters decreases.

In [Pa07] an approach for word sense disambiguation has been presented that also relies on PageRank calculations and is based on the HyperLex [Ve04] algorithm. The method presented in there is similar to the algorithm in this paper in the sense, that terms with a high PageRank, called hubs, are identified. This is performed in one step of the algorithm to acquire the main meanings of a term from a text corpus. In this paper however these hubs are identified for removal in every iteration of the algorithm to separate components in the co-occurrence graph.

3 Experiments

3.1 Setup

In this section, empiric evaluations of the introduced algorithm will be presented. The main goal of the experiments was to prove the hypothesis that the PageRanks of nodes in co-occurrence graphs can be used to separate clusters of semantically similar terms.

Besides this aim, the experiments will show for some example text documents in which iteration of the algorithm clusters will be separated from the co-occurrence graph and how the number of clusters and their average number of terms changes during the execution of the algorithm. Also, the convergence time of the mean term-term-similarity is analysed.

In the first and second experiments five texts with 260 to 450 unique words from a German newspaper corpus [LeCo] have been used for the analysis of the introduced algorithm. The following data have been collected for all the texts during each iteration of the algorithm:

- the mean term-term-similarity and its standard deviation in the found clusters,
- the number of clusters with more than one term,
- the mean number of terms in all clusters and
- the number of clusters with a high mean term-term-similarity.

No stopwords have been removed and no stemming has been applied in the first experiment. In the second experiment however all stopwords have been removed from the texts and stemming has been applied. The third experiment discusses the influence of text length on the mean term-term-similarity.

3.2 Simulation Results

First experiment without stopword removal and without stemming:

Figure 1 shows the mean term-term-similarity and its standard deviation in the calculated clusters per iteration for all of the texts. At the start of the algorithm the mean term-term-similarity is at around 0.6 for all documents and increases to almost 1 when the algorithm terminates. Between the 10th and 20th iteration a significant increase in term-term-similarity can be recognized that correlates with the increased number of clusters during these iterations as depicted in figure 2. Also the standard deviation of the mean term-term-similarity in all clusters increases first when the number of clusters rises and confirms this significant state transition in the mean term-term-similarity. Moreover, it

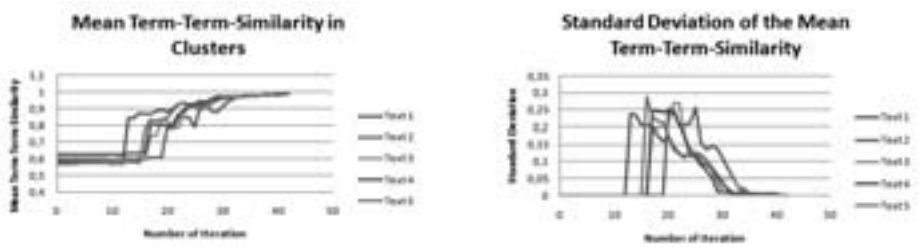

Figure 1: Mean term-term-similarity in clusters and its standard deviation

can be seen (as expected) that the mean term-term-similarity stays constant during the first iterations, as the number of clusters stays the same because stopwords with a high PageRank need to be removed first. In figure 2 the mean number of terms in all clusters is shown for each iteration. This number slowly decreases in the first iterations because of the same reason and significantly drops when the number of clusters increases.

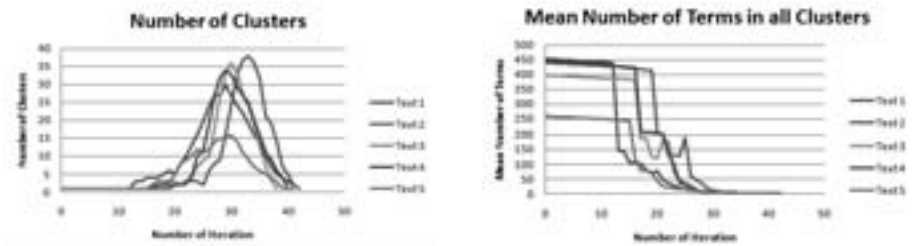

Figure 2: Number of clusters and mean number of terms in all clusters

Figure 3 shows the number of clusters with a high mean term-term-similarity (>0.85) for each iteration. The depicted courses of the curves are very similar to the courses of the curves for the number of all clusters in figure 2. This observation is very interesting because it shows that each term removal from the co-occurrence graph that separates clusters is also likely to instantly produce clusters with a high mean term-term-similarity.

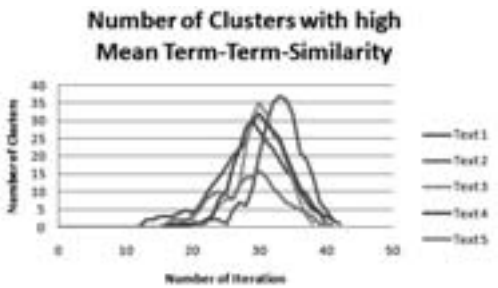

Figure 3: Number of Clusters with a high mean term-term-similarity (>0.85)

Second experiment with stopword removal and applied stemming:

In the second experiment a pre-processing with stopword removal and stemming has been applied on all of the five texts of the first experiment. Therefore, fewer terms with a low semantic relevance and the same meaning in different word forms had to be analysed in the co-occurrence graph. The main result of this experiment is that the mean term-term-similarity in the calculated clusters starts at about 0.6 and gradually rises to almost 1 right from the start of the algorithm. Around 20 iterations were needed to reach this value. This is shown in figure 4. The standard deviation of the mean term-term-similarity decreases at the same time. In comparison to the first experiment also the number of clusters gradually rises right from the start of the algorithm (figure 5) because of the applied stopword removal that mainly influences the initial number of clusters. Moreover, this number was greater than 1 for all of the five used texts at the start of the

algorithm, meaning that the initial co-occurrence graph was already separated into clusters. The stopword removal therefore also influences the convergence time of the mean term-term-similarity as well.

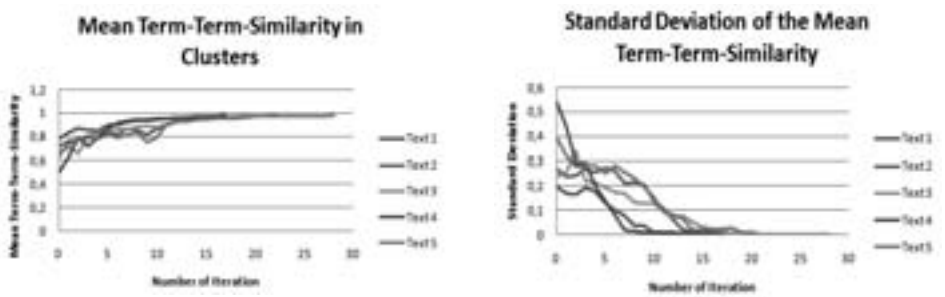

Figure 4: Mean term-term-similarity in clusters and its standard deviation

The mean number of terms in all clusters drops rapidly in the first five iterations for all texts due to the increasing number of clusters. This can also be observed in figure 5.

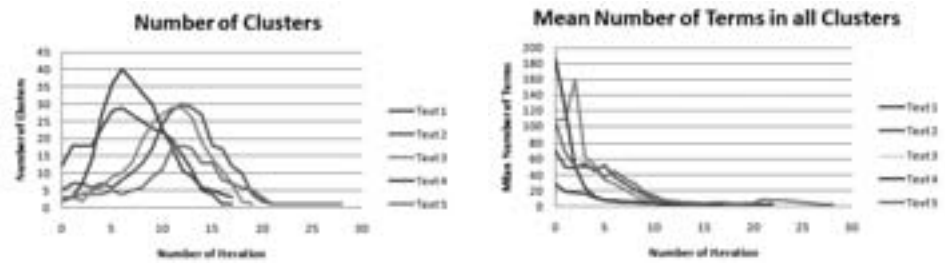

Figure 5: Number of clusters and mean number of terms in all clusters

Figure 6 shows the number of clusters with a high mean term-term-similarity (>0.85) for each iteration. It can be seen that this number is greater or equal to 1 for some of the texts because the initial co-occurrence graph is already separated due to the applied stopword removal. However, this number is always lower than the number of all clusters (figure 5) at the start of the algorithm.

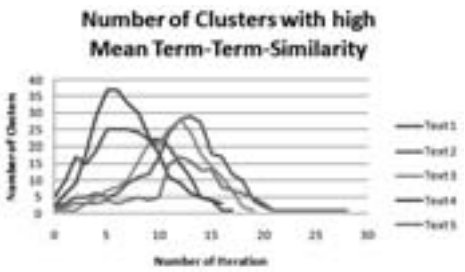

Figure 6: Number of Clusters with a high mean term-term-similarity (>0.85)

Third experiment to test the influence of text length on the state transition point in mean term-term-similarity:

The aim of the third experiment was to determine the number of iterations needed to reach the state transition point in mean term-term-similarity that has been found in the first experiment when varying the text length. Another five texts of different length (500 to 5000 unique words) have been used for this experiment, whereby stopwords have been maintained and no stemming has been applied. In figure 5 it can be seen, that the occurrence of this state transition point in fact depends on the text length, mainly because larger texts also contain more stopwords but also other common terms that are assigned a high PageRank and that will be ruled out first before the graph falls apart.

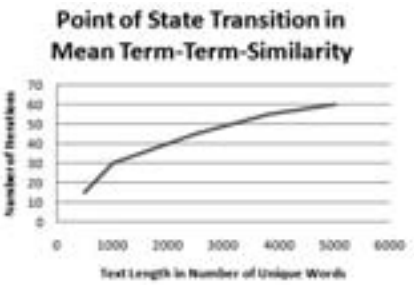

Figure 7: Point of state transition depending on text length

3.3 Discussion

The results above demonstrate that clustering based on PageRank computations can be used to detect topically homogeneous sets of terms within text documents. In particular, the following results could be gained from the experiments:

1. Clustering of co-occurrence graphs based on PageRank calculations is possible.
2. Clusters of high mean term-term-similarity could be gained by this method.
3. A special state transition in mean term-term-similarity is observed when dealing with graphs containing stopwords. The mean term-term-similarity increases significantly when these common terms have been ruled out.

Yet there are still plenty of further options to refine the algorithm:

1. The clustering could be accelerated if terms would not mark themselves for removal in each iteration but would "determine" the term with the highest PageRank in their neighbourhood to be removed.
2. A global PageRank threshold for the terms to be removed is also an option.
3. The PageRank calculation could also yield more meaningful results when the co-occurrence significances would be taken into account. These values could be interpreted in the same way as the bandwidth parameters in [So10] and could therefore influence the PageRank values significantly.

4. The quality of the resulting clusters could be enhanced if these significances would be asymmetric in the sense that a term A does not necessarily have to have the same relation to a term B, as term B has to term A.
5. It is also sensible to define a threshold to stop the further separation of a cluster in order to get a usable group of topically related terms. This threshold should primarily depend on the mean term-term-similarity in a cluster. However, the experiments in this section do not consider such a threshold because the main focus of this paper is set on the PageRank's clustering property in general but it will be a task for future research.

Therefore, it can be seen that there is enough room for further research to enhance the introduced algorithm.

4 Conclusion

In this paper, an algorithm to cluster co-occurrence graphs using PageRank computations has been presented. It could be shown that this approach can be used to detect topically homogeneous clusters of terms by ruling out terms in these graphs that have a higher PagaRank than their co-occurring terms. This method can also be used to filter out very common terms from the co-occurrence graph. After these terms are ruled out a significant increase in the mean term-term-similarity is observable due to the increased number of separated graph components. Besides the mentioned options to enhance the algorithm an interesting research question could be if it is possible to use this method to identify characteristic and discriminating terms of texts in order to automatically build queries for searching similar documents in large corpora or even the World Wide Web? The benefit of using the PageRank algorithm to identify these terms is that there is no need to consult a reference corpus to determine a term's significance in a text. In the area of topic detection and tracking the presented method could have some impact on how to automatically identify different topics in single texts with minimal effort.

Bibliography

[Al02] Allan, J.: Introduction to Topic Detection and Tracking. Kluwer Academic Publishers, chapter 1, 1-16, 2002.

[Kl02] Kleinberg, J.: Bursty and Hierarchical Structure in Streams. Proc. 8th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining, 2002.

[Wa06] Wang, X.; McCallum, A.: Topics over Time: A Non-Markov Continuous-Time Model of Topical Trends. In: Conference on Knowledge Discovery and Data Mining (KDD), 2006.

[Pa98] Page, L.; Brin, S.; Motwani, R.; Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[Zh05] Zhu Y.; Ye S.; Li X.: Distributed PageRank computation based on iterative aggregation-disaggregation methods. In Proc. ACM Int. Conf. Information and knowledge management, pp. 578-585, 2005.

[Sa03] Sankaralingam, K.; Sethumadhavan, S.; Browne, J. C.: Distributed pagerank for P2P systems. In Proc. IEEE Int. Symp. High Performance Distributed Computing, pp. 58-68, 2003.

[Is09] Ishii, H.; Tempo R.: Distributed pagerank computation with link failures. In Proc. The 2009 American Control Conf., pp. 1976-1981, 2009.

[So10] Sodsee S.; Komkhao M.; Meesad P.; Unger H.: An Extended Pagerank Calculation Including Network Parameter. Computer Science Education: Innovation and Technology (CSEIT 2010) Special Track: knowledge Discovery (KD 2010), 2010.

[He06] Heyer, G.; Quasthoff, U.; Wittig, Th.: Text Mining – Wissensrohstoff Text. W3L Verlag Bochum, 2006.

[Bu06] Buechler, M.: Flexibles Berechnen von Kookkurrenzen auf strukturierten und unstrukturierten Daten. Master's thesis, University of Leipzig, 2006.

[Qu02] Quasthoff, U., Wolff, Chr.: The Poisson Collocation Measure and its Applications. In: Proc. Second International Workshop on Computational Approaches to Collocations, Wien, 2002.

[Du94] Dunning, T.: Accurate methods for the statistics of surprise and coincidence. Computational Linguistics 19(1):61–74, 1994.

[Pa07] Palta, E.: Word Sense Disambiguation. Master's thesis, Kanwal Rekhi School of Information Technology, Powai, Mumbai, 2007.

[Ve04] Veronis, J.: Hyperlex: lexical cartography for information retrieval. Computer Speech & Language, 18(3):223–252, 2004.

[LeCo] Website of the Leipzig Corpora Collection, http://corpora.informatik.uni-leipzig.de/