Summarizing Online User Reviews Using Bicliques

Azam Sheikh Muhammad[1], Peter Damaschke[2], and Olof Mogren[2]

[1] College of Computing and Informatics, Saudi Electronic University (SEU), Riyadh 13316, Kingdom of Saudi Arabia, m.sheikh@seu.edu.sa
[2] Department of Computer Science and Engineering, Chalmers University, 41296 Göteborg, Sweden, [ptr,mogren]@chalmers.se

Abstract. With vast amounts of text being available in electronic format, such as news and social media, automatic multi-document summarization can help extract the most important information. We present and evaluate a novel method for automatic extractive multi-document summarization. The method is purely combinatorial, based on bicliques in the bipartite word-sentence occurrence graph. It is particularly suited for collections of very short, independently written texts (often single sentences) with many repeated phrases, such as customer reviews of products. The method can run in subquadratic time in the number of documents, which is relevant for the application to large collections of documents.

1 Introduction

Extractive summarization, i.e., selection of a representative subset of sentences from input documents, is an important component of modern information retrieval systems. Existing methods usually first derive some intermediate representation, then score the input sentences according to some formula and finally select sentences for the summary [16].

The field of automatic summarization was founded in 1958 by Luhn [14], who related the importance of words to their frequency of occurring in the text. This importance scoring was then used to extract sentences containing important words. With the advent of the World Wide Web, large amounts of public text became available and research on multi-document summarization took off. Luhn's idea of a frequency threshold measure for selecting topic words in a document has lived on. It was later superseded by $tf \times idf$, which measures the specificity of a word to a document and has been used extensively in document summarization efforts.

Radev et al. pioneered the use of cluster centroids in summarization in their work [18], with an algorithm called MEAD that generates a number of clusters of similar sentences. To measure the similarity between a pair of sentences, the authors use the cosine similarity measure where sentences are represented as a weighted vector of $tf \times idf$ terms. Once sentences are clustered, a subset from each cluster is selected. MEAD is among the state-of-the-art extractive summarization techniques and frequently used as baseline method for comparing new algorithms.

Summarization is abstractive when new content is generated while summarizing the input text. Ganesan et al. [8], considered online user reviews, which are typically short (one or two sentences) and opinionated. They presented an abstractive approach that used parts of the input sentences to generate the output. For evaluations they provided the Opinosis dataset, consisting of user reviews on 51 different topics. Their system performed well evaluating generated summaries against those written by human experts. They also compare their results with the aforementioned MEAD [18] method.

Bonzanini et al. [4] introduced an iterative sentence removal procedure that proved good in summarizing the same short online user reviews from the Opinosis dataset. Usually, an extractive summarization method is focused on deciding which sentences are important in a document and considered for inclusion in the summary. The sentence removal algorithm by Bonzanini et al., [4] would instead iteratively choose sentences that are unimportant and remove them, starting with the set of all sentences in the input. The process continues until the required summary length is reached.

SumView [19] is also specialized on collections of short texts and uses feature extraction and a matrix factorization approach to decide on the most informative sentences. Besides the aforementioned work we refer to an extensive survey [16] discussing different approaches to sentence representation, scoring, and summary selection, and their effects on the performance of a summarization system.

Our contribution:
We propose a novel, purely combinatorial approach aimed at extractive summarization of collections of sentences. Since we will consider online user reviews as input, that typically are single sentences from independent authors, we speak of *sentences* rather than *documents* from now on. The idea is simply to find the key combinations of words appearing in several sentences. We work with a bipartite graph where the two vertex sets correspond to sentences and words, respectively, and edges exist between words and the sentences where they appear. Then, finding sets of sentences that share the same combination of words is equivalent to finding the bicliques in this graph. (Formal definitions follow in Section 2.) Finally we select bicliques for the summary according to further criteria. Leveraging recent advances in fast algorithms for determining the most similar sets, the approach is also computationally fast. Altogether this enables us to present a method for extractive summarization of short independent texts that should be attractive due to its conceptual simplicity and direct interpretability of the output. We show that it performs well on the benchmark Opinosis dataset [8]. It also outperforms state-of-the-art systems achieving the best precision, F_1 and F_2 measures. (See definitions and results in Section 6.2.)

As a delimitation, due to its very idea the method cannot be expected to extract good summaries of single complex texts, but this type of application is beyond the scope of this work.

2 Preliminaries

The following notation will be used throughout this paper. We consider any sentence as a bag of words, that is, as the set of distinct words, disregarding order and multiplicity of words. The ith sentence is denoted s_i. and $|s_i|$ is the number of distinct words, after removal of multiple occurrences and stopwords. The given set of sentences is T, and a summary is denoted $S \subset T$. Note that an extractive summary is merely a selection of the most informative sentences; no new text is generated. The length of a summary in terms of the number of sentences is denoted by ℓ.

Symbol $M_{s_i;s_j}$ stands for a sentence similarity measure. Several measures for sentence similarity have been explored for extractive summarization. The simplest ones are called the surface matching sentence similarity measures [15], because they do not require any sophisticated linguistic processing, and the similarity value is merely based on the number of words that occur in both of the sentences. In the present paper we are considering two of them:

- the *match* measure $M_{s_i;s_j} = |s_i \cap s_j|$,
- the *cosine* measure $M_{s_i;s_j} = |s_i \cap s_j| / \sqrt{|s_i| \times |s_j|}$.

Note that the cosine measure can be viewed as the match measure normalized by the sentence size.

Given a collection of sentences, we consider the *word-sentence occurrence graph*, that is, the bipartite graph $B = (T, W; E)$ with T and W being the set of sentences and words, respectively, where $sw \in E$ is an edge if and only if word w occurs in sentence s. Here the number of occurrences is disregarded, that is, edges are not weighted.

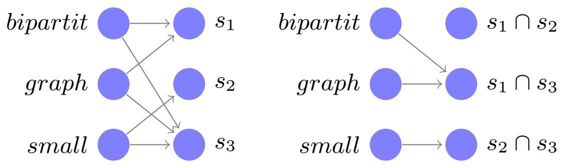
We use the standard notation $N(v)$ for the set of neighbored vertices of a vertex v in a graph. It naturally extends to vertex sets V, by defining $N(V) := \bigcup_{v \in V} N(v)$. A bipartite clique, *biclique* for short, is a pair (X, Y) of sets $X \subseteq T$ and $Y \subseteq W$ that induces a complete bipartite subgraph of B, that is, $N(X) \supseteq Y$ and $N(Y) \supseteq X$. A maximal biclique (not to confuse with a biclique of maximum size) is a biclique not contained in other bicliques. We call a subset Y of W of the form $Y = N(s_i) \cap N(s_j)$, with $s_i, s_j \in T$, a *2-intersection*. It corresponds to a biclique (X, Y) with $|X| = 2$.

3 Overall Idea

Customer reviews and similar text collections consist of sentences written independently by many authors. Intuitively, combinations of words appearing in several sentences should be important for a summary. Obviously they are bicliques in the word-sentence occurrence graph.

The problem of enumerating the maximal bicliques (and hence, implicitly, all bicliques) in a graph is well investigated, in particular, several output-sensitive algorithms with different running times are known [1, 6, 9, 11]. However, in general

Fig. 1. Biclique Summarization. Example sentences s_1: "This is a bipartite graph", s_2: "Look how small it is", s_3: "This bipartite graph is very small". After stemming and stopwords removal: s_1: "bipartit graph", s_2: "small", s_3: "bipartit graph small".



a graph has very many bicliques, and we need to select those which appear most relevant for a small extractive summary. We found that the large 2-intersections (see definitions above) are good candidates, by the following reasoning.

The word set Y of a biclique (X, Y) with $|X| = 2$ has been used by at least two independent authors, hence it has not only occurred by chance, in one sporadic statement. Moreover, Y is a maximal word set with this property. More repetitions of less specific word sets, that is, bicliques with $|X| > 2$ but smaller Y, do not seem to add much to a summary. The restriction to $|X| = 2$ also allows the use of standard pairwise similarity measures in the heuristic rules that afterwards select the sentences to be put in the summary.

An example of the concept is visualized in Figure 1. The bipartite graph is shown on the left while all possible 2-intersections are on the right. Since s_3 is present in both of the nonempty 2-intersections, it may be regarded as representative of the other two and thus selected for the summary.

4 Implementation Details

The given set of sentences T undergoes a preprocessing before passing it to the main algorithm. This involves stopwords removal and stemming. Words such as "am", "are", "by", "is", are called stopwords. They are common to many sentences and usually do not carry meaningful information when comparing texts in the context of summarization. Stemming reduces a word to its stem, base or root form. Stemming prevents mismatch between two words which apparently differ but are actually grammatical variations of the same word, such as singular and plural. In many cases stemming is achieved by chopping off the ends of words. Details about the stopwords list and stemming technique used in our implementation (which, however, do not affect the core ideas of the method) are described in Section 6.2.

In Algorithm 1 we give a pseudocode description of the method. As said in Section 3, in a biclique (X, Y), the Y part is the word set and X with the restriction $|X| = 2$ (because we need 2-intersections only), is the set containing i, j corresponding to a pair of sentences (s_i, s_j), $i \neq j$, in T. Our implementation using the subroutine $FindBiclique(s_i, s_j, sim)$ collects bicliques (X, Y) with $|Y| \geq 2$, that is, there are at least two words common in the corresponding pair of sentences (s_i, s_j). For every such pair we also compute the value of the similarity measure specified by the parameter `sim` which is either equal to `match` or `cosine`; see Section 2. We find all such bicliques and this concludes the first part of the algorithm. There are two more major parts in our algorithm.

Algorithm 1 Biclique Summarization

```
Input: $sim, top, \ell, T$
Output: $S$
 1: Part-1: Find Bicliques:
 2: $nBicliques \leftarrow 0$
 3: repeat
 4:     Choose a new pair $(s_i, s_j)$, $i \neq j$, in $T$
 5:     $[X, Y, simValue] \leftarrow FindBiclique(s_i, s_j, sim)$
 6:     $Bicliques.add(X, Y, simValue)$
 7:     $nBicliques \leftarrow nBicliques + 1$
 8: until $NoMoreBicliques$
 9: Part-2: Filter Important Bicliques:
10: $Bicliques \leftarrow SortBySimValue(Bicliques)$
11: $max \leftarrow \lceil \frac{nBicliques \times top}{100} \rceil$
12: $impBicliques \leftarrow Select(Bicliques[1...max])$
13: Part-3: Summary Selection:
14: $SentenceIDs \leftarrow UniqueSortedByFreq(impBicliques.X)$
15: $S \leftarrow GetSentences(T, SentenceIDs[1...\ell])$
16: return $S$
```

In the second part we filter out important bicliques, making use of the parameter `top` which defines percentage of the bicliques that should be kept. Typical values used are `10`, `30` and `50`.

First, the bicliques are sorted with respect to their decreasing similarity value and then we select best `top%` of the entire bicliques collection. In this way, the most informative bicliques are kept while the rest are discarded. For example when `top=10`, we select top `10` percent of the bicliques from the sorted list.

Finally, to select a summary, we need a ranking of the sentences appearing in the filtered important bicliques. In our implementation we simply count the occurrences of sentences in the filtered bicliques and take the ℓ most frequent sentences.

5 Processing Time

The time complexity is dominated by the time needed to determine the sentence pairs (2-intersections) with top similarity scores. This subproblem is known as *top-k set similarity joins* (more precisely: self-joins), where our k is the number of bicliques to keep. A basic implementation as displayed in Algorithm 1 loops through all pairs of sentences and therefore costs quadratic time in the number of sentences. However, the top-k set similarity joins problem is well studied for different standard similarity measures, and fairly simple heuristic algorithms as proposed in [20, 2, 7] and related work have experimental running times far below the naive quadratic time bound. They can replace Part 1 and 2 of Algorithm 1. These heuristics rely on the very different word frequencies in texts, which essentially follow power laws [10]. Some further theoretical analysis of time bounds is given in [5].

For the sake of completeness we briefly outline these techniques. The first main idea is to process the words by increasing frequencies f and collect the pairs of sentences where any common words have been detected. A word appearing in f sentences trivially appears in fewer than $f^2/2$ of the 2-intersections. Since most words have low frequencies, the total number of such sentence pairs does not grow much in the beginning. The second main idea is called prefix filtering, which is actually a branch-and-bound heuristic. It bounds for every considered sentence pair the maximum possible value of the similarity measure, in terms of the number of (frequent) words not yet considered. This allows early exclusion of many pairs of sentences that cannot be among the top k pairs any more. Building on these principles, the "segment bounding algorithm" for the overlap measure [3] divides the set of words into segments according to their frequencies. Then the largest 2-intersections within the segments are computed by subset hashing, and finally the partial results are combined following some simple priority rules, until the top k pairs are established for a prescribed k. The authors of [3] report that their algorithm runs faster than those from [20] especially on large datasets. Finally, for the largest 2-intersections found, one can also filter those where other similarity measures are maximized.

Thanks to these techniques, our method can be implemented to run in subquadratic time. By way of contrast, this is apparently not possible for other summarization approaches like sentence removal [4] which is intrinsically more than quadratic.

6 Experimental Results

In this section, we present an empirical evaluation of the proposed method.

6.1 Dataset

Considering large amounts of highly redundant short text opinions expressed on the web, our experimental study focuses on assessing the performance of the proposed method on such a dataset which includes users stating their single line opinions about products or services, or commenting on some hot topics or issues on certain discussion forums and social media sites.

The Opinosis dataset [8] is particularly relevant to our purpose because it contains short user reviews in 51 different topics. Each of these topics consists of between 50 and 575 one-sentence user reviews made by different authors about a certain characteristic of a hotel, a car or a product (e.g. *"Location of Holiday*

Inn, London" and *"Fonts, Amazon Kindle"*). The dataset includes 4 to 5 gold-standard summaries created by human authors for each topic. The length of the gold-standard summaries is around 2 sentences.

6.2 Evaluation Method and Baseline Selection

Following standard procedure, we use ROUGE [12] for evaluation. ROUGE compares the system-generated summary with the corresponding gold-standard summaries for a topic and reports the assessment in terms of quantitative measures: recall, precision and F-measure. Recall is the number of words in the intersection of system-generated and the gold-standard summaries divided by the number of words in the gold-standard summary. Precision is the number of words in the intersection divided by the number of words in the system-generated summary.

F-measure, also called F_1 score, is a composite measure defined as the harmonic average of precision and recall. Sometimes, in order to emphasize the importance of recall over precision, another F-measure called F_2 score is also computed. On our benchmark dataset, Opinosis, F_2 scores are also reported in state-of-the-art results.

The general definition of F-measure for positive real β is:

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}$$

ROUGE works by counting n-gram overlaps between generated summaries and the gold standard. Our results show ROUGE-1, ROUGE-2 and ROUGE-SU4 scores, representing matches in unigrams, bigrams and skip-bigrams respectively. The skip-bigrams allow four words in between.

The experiments are aligned (in terms of ROUGE settings etc.,) with those of Bonzanini et al. [4], which provide state-of-the-art results on extractive summarization on the Opinosis dataset. As mentioned in Section 1, they use a sentence removal (SR) algorithm. They used ROUGE to evaluate their methods, and MEAD [18], an extractive multi-document summarizer (see Section 1, too), as a baseline.

There are two different versions of the SR algorithm in [4], one based on similarity (SR_{SIM}) and the other one based on diversity (SR_{DIV}). We compare our method to both of them.

Summary length ℓ was fixed at 2 sentences. This matches the supplied gold-standard summaries and is also necessary to align our results to [4].

In addition to MEAD [18], they [4] use a brute-force method as a baseline which, for any given topic, enumerates all combinations of 2 sentences and chooses the pair that optimizes on the same scoring functions as used in their sentence removal algorithm.

Our implementation maximizes ROUGE scores: We consider all possible pairs of sentences within each topic, compute ROUGE-1, ROUGE-2 and ROUGE-SU4 scores (of recall, precision, F_1, and F_2) and choose the sentence pair with highest value. Choosing such pairs for all topics in the dataset gives us the maximum scores, denoted with OPTIMAL, in our evaluations. These are the ideal scores attainable by an extractive summarization algorithm on this dataset.

We evaluate an implementation of the Biclique algorithm with sentence similarity measures *match* and *cosine*. For the *cosine* measure we let `top` vary between `10` and `30`. For *match* we evaluate with `top` fixed at `50`, which gave us the highest scores. Accordingly, the methods are abbreviated $Biclique_{Cosine1}$, $Biclique_{Cosine3}$, and $Biclique_{Match5}$, respectively.

The systems are evaluated with ROUGE version 1.5.5 using the following options: `-a - m -s -x -n 2 -2 4 -u`. For F_2, alongside the previous option, we also add: `-p 0.2`.

For preprocessing we make use of a Porter stemmer [17]. Stopwords were removed using the stopword list distributed as part of the ROUGE evaluation system [12].

6.3 Results

In Table 1 we show results for the experiments. R-1, R-2 and R-SU4 represent scores of ROUGE-1, ROUGE-2 and ROUGE-SU4 respectively. The best results (with the exception of the brute-force optimal scores) are shown in bold. The best scores among biclique alternatives are shown in italic. The brute-force optimal scores with respect to the evaluated measure in Table 1, marked in gray in the first row in each sub-table, are the maximum attainable scores. Baselines are shown at the bottom of each sub-table.

Recall:
With respect to recall, $Biclique_{Match5}$ has attained a ROUGE-1 value of 43.54 and ROUGE-SU4 of 16.36. This is better than the corresponding values (37.46 and 13.80) of the baseline method SR_{SIM} making an improvement of 16.23% and 18.55%, respectively. Similarly, with respect to ROUGE-2 value, $Biclique_{Match5}$ has attained 8.91 compared to the value 8.67 of the baseline method SR_{DIV}. Comparing ROUGE-1 and ROUGE-SU4, SR_{DIV} has attained better scores, and $MEAD$ has overall higher recall scores. However, it should be kept in mind that both of the latter are biased towards recall and do not perform well on precision compared to our method $Biclique_{Match5}$ in all three ROUGE settings.

Precision:
$Biclique_{Cosine1}$ is the best system for precision in all three settings of ROUGE, increasing the best scores among the baseline methods by over 85% for the ROUGE-1, over 90% for ROUGE-2, and over 223% for ROUGE-SU4.

$\mathbf{F_1}$:
The performance of $Biclique_{Cosine1}$ is consistent when we consider the measures F_1 and F_2, showing that the high precision is also combined with a high recall. For example, using $Biclique_{Cosine1}$, the best F_1-scores of the best performing baseline methods are improved by at least 34%, 35%, and 120% for ROUGE-1, ROUGE-2, and ROUGE-SU4, respectively.

Recall

	R-1	R-2	R-SU4
OPTIMAL	57.86	22.96	29.73
$Biclique_{Cosine1}$	30.48	7.62	12.40
$Biclique_{Cosine3}$	31.94	8.13	13.40
$Biclique_{Match5}$	*43.54*	*8.91*	*16.36*
MEAD	**49.32**	**10.58**	**23.16**
SR_{DIV}	46.05	8.67	20.10
SR_{SIM}	37.46	9.29	13.80

Precision

	R-1	R-2	R-SU4
OPTIMAL	57.35	22.07	36.07
$Biclique_{Cosine1}$	***36.85***	***9.88***	***17.58***
$Biclique_{Cosine3}$	33.29	9.07	15.03
$Biclique_{Match5}$	11.70	2.26	3.36
MEAD	9.16	1.84	1.02
SR_{DIV}	9.64	1.77	1.10
SR_{SIM}	19.87	5.18	5.44

F_1

	R-1	R-2	R-SU4
OPTIMAL	46.57	19.49	23.76
$Biclique_{Cosine1}$	***32.67***	***8.41***	***13.93***
$Biclique_{Cosine3}$	31.85	8.35	13.46
$Biclique_{Match5}$	17.93	3.50	5.40
MEAD	15.15	3.08	1.89
SR_{DIV}	15.64	2.88	2.03
SR_{SIM}	24.38	6.23	6.31

F_2

	R-1	R-2	R-SU4
OPTIMAL	48.41	20.45	24.72
$Biclique_{Cosine1}$	31.16	7.88	12.86
$Biclique_{Cosine3}$	***31.73***	***8.17***	***13.28***
$Biclique_{Match5}$	26.91	5.36	8.66
MEAD	26.27	5.43	4.34
SR_{DIV}	25.39	4.70	4.16
SR_{SIM}	29.92	7.54	8.08

Table 1. ROUGE scores for $Biclique_{Dice}$, $Biclique_{Cosine}$ - Biclique (with Match and Cosine sentence similarity, respectively) obtains the highest Precision, as well as F_1 and F_2 scores. *OPTIMAL* scores (gray) contain the corresponding score for an optimal summary for each cell. SR_{SIM}, SR_{DIV} - Bonzanini et al (2013).

$\mathbf{F_2}$:
Here the best results are achieved by our method $Biclique_{Cosine3}$. We consistently improve on the best scoring method among the baselines by at least 6%, 8% and 64% for ROUGE-1, ROUGE-2, and ROUGE-SU4, respectively.

6.4 Discussion

Empirical evaluation of the method proposed in this paper suggests that we have a clear improvement over state-of-the-art extractive summarization results on the Opinosis dataset. Our method has shown substantial improvement compared to the existing results, especially for precision, F_1, and F_2 on all ROUGE settings. The only result we cannot beat is the recall scores of the baseline methods MEAD and SR_{DIV}, which achieve the high recall at the expense of a sharp drop in precision (explained by the fact that these methods tend to choose larger sentences which results in high recall only).

Generally our method provides a balance between the two metrics, recall and precision, which is clear from the F_1-scores (Table 1). Still the biclique method has the flexibility of optimizing a certain metric, e.g., $Biclique_{Match5}$ is obtained using a parameter setting favoring the recall.

To conclude, supported by the best scores for all ROUGE settings for precision, F_1, and F_2 on the Opinosis dataset, our biclique method should be a good addition to the existing multi-document summarization systems, and it is particularly well suited to summarizing short independent texts like user reviews. With all its strengths, the method should also be appealing because of its simplicity and good time complexity. However, it is not expected to perform equally well on datasets of more complex texts which are not in the focus of our study.

7 Conclusions

We have proposed a novel method for extractive multi-document summarization, and showed with empirical results that it outperforms state-of-the-art summarization systems. Our method is based on the detection of bicliques in a bipartite graph of sentences and their words. To keep it simple and to highlight the strength of the main idea, we have evaluated only a basic version of the method which is already better than existing top-performing systems. The technique is also flexible as it can be easily adapted for a higher recall, a higher precision, or a balance between the two metrics. Considering the time efficiency, our proposed biclique algorithm offers, for standard similarity measures, the possibility of subquadratic running time, as opposed to the at least quadratic running time of the baseline sentence removal method.

We believe that more elaborate versions making use of deep similarity measures and combining with ideas from other methods, such as MEAD, can further enhance the performance. A natural extension of the preprocessing would be to cluster semantically related words (synonyms, etc.) and to replace the words from each cluster with one representative. As mentioned in Section 6.2 we use stopwords from ROUGE that also include negations. As the method does not rely on the meaning of words this is not an issue, still one could study the effect of different stopword lists.

Acknowledgments

This work has been supported by Grant IIS11-0089 from the Swedish Foundation for Strategic Research (SSF), for the project "Data-driven secure business intelligence". We thank our former master's students Emma Bogren and Johan Toft for drawing our attention to similarity joins, and the members of our Algorithms group and collaborators at the companies Recorded Future and Findwise for many discussions.

References

1. Alexe, G., Alexe, S., Crama, Y., Foldes, S., Hammer, P. L., Simeone, B.: Consensus Algorithms for the Generation of all Maximal Bicliques. Discr. Appl. Math. 145, 11–21 (2004)
2. Arasu, A., Ganti, V., Kaushik, R.: Efficient Exact Set-Similarity Joins. In: Dayal, U. et al. (Eds.) VLDB 2006. pp. 918–929, ACM (2006)
3. Bogren, E., Toft, J.: Finding Top-k Similar Document Pairs – Speeding up a Multi-Document Summarization Approach. Master's thesis, Dept. of Computer Science and Engin., Chalmers, Göteborg (2014)
4. Bonzanini, M., Martinez-Alvarez, M., Roelleke, T.: Extractive Summarisation via Sentence Removal: Condensing Relevant Sentences into a Short Summary. In: Jones, G.J.F. et al. (Eds.), SIGIR 2013. pp. 893–896, ACM (2013)
5. Damaschke, P.: Finding and Enumerating Large Intersections. Theor. Comp. Sci. 580, 75–82 (2015)
6. Dias, V.M.F., de Figueiredo, C.M.H., Szwarcfiter, J.L.: On the Generation of Bicliques of a Graph. Discr. Appl. Math. 155, 1826–1832 (2007)
7. Elsayed, T., Lin, J., Oard, D.W.: Pairwise Document Similarity in Large Collections with MapReduce. In: ACL-08: HLT, Short Papers (Companion Volume), pp. 265–268, Assoc. Comp. Ling. (2008)
8. Ganesan, K., Zhai, C., Han, J.: Opinosis: a Graph Based Approach to Abstractive Summarization of Highly Redundant Opinions. In: Huang, C.R., Jurafsky, D. (Eds.) COLING 2010, pp. 340–348, Tsinghua Univ. Press (2010)
9. Gely, A., Nourine, L., Sadi, B.: Enumeration Aspects of Maximal Cliques and Bicliques. Discr. Appl. Math. 157, 1447– 1459 (2009)
10. Li, W.: Random Texts Exhibit Zipf's-law-like Word Frequency Distribution. IEEE Trans. Info. Th. 38, 1842–1845 (1992)
11. Li, J., Liu, G., Li, H., Wong, L.: Maximal Biclique Subgraphs and Closed Pattern Pairs of the Adjacency Matrix: A one-to-one Correspondence and Mining Algorithms. IEEE Trans. Knowl. Data Eng. 19, 1625–1637 (2007)
12. Lin, C.Y.: ROUGE: A Package for Automatic Evaluation of Summaries. In: Moens, M.F., Szpakowicz (Eds.) ACL Workshop "Text Summarization Branches Out", pp. 74–81 (2004)
13. Lin, H., Bilmes, J.A.: A Class of Submodular Functions for Document Summarization. In: Lin, D., Matsumoto, Y., Mihalcea, R. (Eds.) ACL 2011, pp. 510–520, Assoc. Comp. Ling. (2011)
14. Luhn, H.P.: The Automatic Creation of Literature Abstracts. IBM Journal 2, 159–165 (1958)
15. Manning, C.D., Schütze, H.: Foundations of Statistical Natural Language Processing. MIT Press (1999)
16. Nenkova, A., McKeown, K.: A Survey of Text Summarization Techniques. In: Aggarwal, C.C., Zhai, C. (Eds.) Mining Text Data. pp. 43–76, Springer (2012)
17. Porter, M.F.: An Algorithm for Suffix Stripping. Program 14, 130–137 (1980)
18. Radev, D.R., Allison, T., Blair-Goldensohn, S., Blitzer, J., Celebi, A., Dimitrov, S., Drábek, E., Hakim, A., Lam, W., Liu, D., Otterbacher, J., Qi, H., Saggion, H., Teufel, S., Topper, M., Winkel, A., Zhang, Z.: 2004. MEAD - a platform for multidocument multilingual text summarization. In: LREC (2004)
19. Wang, D., Zhu, S., Li, T.: SumView: A Web-based Engine for Summarizing Product Reviews and Customer Opinions. Expert Systems with Appl. 40, 27–33 (2013)
20. Xiao, C., Wang, W., Lin, X. Haichuan Shang, H.: Top-k Set Similarity Joins. In: Ioannidis, Y.E., Lee, D.L., Ng, R.T. (Eds.) ICDE 2009, pp. 916–927, IEEE (2009)