Data Fusion: Boosting Performance in Keyword Extraction

Conference Paper · April 2013

DOI: 10.1109/ECBS.2013.12

2 authors:

Thomas Bohne

6 PUBLICATIONS **20** CITATIONS

SEE PROFILE

Uwe M. Borghoff

University of the Bundeswehr Munich

314 PUBLICATIONS **1,947** CITATIONS

SEE PROFILE

Data Fusion : Boosting Performance in Keyword Extraction

Thomas Bohne and Uwe M. Borghoff
Computer Science Department
Universität der Bundeswehr München
Werner Heisenberg Weg 39
München, Germany
Email: {thomas.bohne,uwe.borghoff}@unibw.de

***Abstract*—In a time when volatile data is in constant growth, the importance of keyword extraction becomes particularly evident. Keywords can quickly identify, structure and reveal potentially worthwhile information. The quality of automatically extracted keywords reflects the individual characteristics of the various retrieval approaches that may be used for extraction. A combinatorial approach using multiple heuristic keyword extraction algorithms may enhance the quality of the results significantly, though it may also compound the inherent limitations. In our paper we compare different ranking aggregation and data fusion methods for single documents. Furthermore we apply principal component analysis to determine an optimal selection of retrieval algorithms for combination with respect to the use-case. To validate our approach, we provide a statistical evaluation with real-world examples.**

***Keywords*-keyword extraction, algorithm combination, principal component analysis**

I. Introduction

The colossal growth of online documents emphasizes the demand for quick analysis, indexing and retrieval. A list of keywords may provide an interpretation of the content of a text to the user, though single keywords may not provide a meaningful summary [1]. Furthermore, keywords are used for index generation, categorization, or browsing document collections.

The majority of online available documents do not have author-assigned keywords or any keywords at all. Also, in some cases author-assigned keywords do not fit the content of the documents because they have been assigned based on other - sometimes organizational or social - constraints. Automatic keyword extraction is a powerful and convenient method to process the large volume of documents to make them useful and organized. This method helps to summarize the content of an individual document or a section of it, while its biggest challenge is to extract the most meaningful words contained in the text.

Over the years many different keyword extraction methods have been proposed, including probabilistic models [2], language-based models [3], and learning-based approaches [4], [1]. Other methods generate term rankings based on structural relationships or term distributions [5], [6]. Furthermore, graph-based ranking algorithms such as PageRank [7] can be utilized to generate rankings for graph-based document representations [8]. One of the most often used and widely spread heuristic weighting formula is the Term Frequency - Inverse Document Frequency (TF-IDF)-weighting formula [9].

In this paper, we focus on heuristic retrieval methods for keyword extraction that perform best on single documents and documents that are between one and several hundred pages. Heuristics can be performed without the need for huge databases, web access, nor a learning process – which accounts for low runtimes so long as the text is within a certain size limit. However, a drawback of heuristics is evident in the quality – defined as meaningfulness to the source text – of the extracted keywords. Another limitation is that retrieval heuristics do not meet all desirable constraints (see Section IV of an optimal algorithm [10]).

Heuristics used in combination have been shown to have supplementary properties that improve retrieval results [11], [6]. A method of combining retrieval results is feature ranking aggregation [12], [13]. Prati investigated four different ranking aggregation methods, and demonstrated the compelling performance of combinations towards single heuristics. We mention his most successful approaches in Section IV and further discuss their performance in Section VI. Another promising approach presented by Li et al. suggests the application of Principal Component Analysis (PCA) on the result space of term weights [14]. With PCA they create a weighed linear combination of retrieval heuristics that results in a final weight. A similar application of Singular Value Combination is Latent Semantic Analysis (LSA) [15] but LSA is not part of this work.

The main contributions of our work are the following: We compare the PCA-based approach with state-of-the-art combination and ranking aggregation methods using real world examples of limited size. We are not aware that such a comparison has been previously performed in the context of keyword extraction for single documents. Furthermore we utilize PCA to select the optimal combination of two or more heuristics depending on specific characteristics of the document to be analyzed, and found this approach to be flexible and parameter-free.

978-0-7695-4991-0/13 $26.00 © 2013 IEEE
DOI 10.1109/ECBS.2013.12

This paper is structured as follows: In Section II we describe the document model and the necessary preprocessing steps for analysis. In Section III we detail a selection of retrieval heuristics that we intend to combine in Section V. Different combination methods and data fusion techniques are presented in Section IV and analyzed in Section V. Finally, we discuss the performance of the proposed combination methods in Section VI.

II. Single Document Processing

In this paper we focus on keyword extraction for single documents and texts of medium length. Heuristics can only show their advantage of short reaction time with documents of short lengths (see Section III).
In this section we define our few preprocessing steps and present a window-based approach for document segmentation that is essential for the weighting algorithms.

A. Pre-processing

For pre-processing, punctuation symbols are removed, all relevant terms of the document are down-cased, and special characters are excluded. This procedure can be performed without prior knowledge of the structure and language (based on the Latin alphabet) of the single document. We furthermore exclude all terms with non-alphabetic characters, URLs, and numbers because we experienced that most numbers are used as page numbers or indices (e.g. in the appendix). A common preprocessing procedure is the removal of stop words. We do not remove stop words because it is language-specific and affects the precision of some of the presented heuristics, which is contrary to our evaluation in Section VI. Besides that, some commonly used words may be of particular importance in a specific context, such as the name of the British rock band "The Who".

B. Document Segmentation

A *term* can be a single word, a compound, or a phrase within a document – we define single words as terms. A *document* is a structured sequence of terms that can appear in forms such as e-mail, website, or text document.

The aim of document segmentation is to identify topic boundaries within documents in order to subdivide into subtopic segments [16]. The document segmentation technique utilized in this work is based on *windows*. The windows $w_1, ..w_N$ in the document d consist of sequences of single terms, are non-overlapping and contain all terms in d. We propose three different criteria to determine the optimal window composition:

1) Number of extracted keywords per window
2) Number of terms that compose a window
3) Logical document structure

We claim that a well-performing segmentation process leads to better keywords in the outcome. The number of extracted keywords strongly depends on the performed retrieval heuristic and the size of the windows that compose the underlying document. Since some weighting algorithms do not exclude any terms but create an exhaustive ranking, the first criteria is rendered inapplicable. Usage of document structure information to create an appropriate window is the preferred criteria because the risk of topic-overlapping windows is reduced while preserving contextually related sections.

Here we presume the lack of an external corpus of documents that could provide useful domain-specific information for the segmentation process. Linguistic features can provide worthwhile information about the author and the content of a text, but they are language-specific and domain-dependent. Since document segmentation is not the main focus of this work, we solely account for structural data of documents and texts based on paragraphs. Each paragraph in the source document is represented by a window $w \in \{w_1, ..., w_N\}$ for our application. This method is used in numerous studies and is applicable to most available texts.

III. Heuristic Retrieval Algorithms

Various heuristic retrieval methods have been proposed and analyzed over the past years, but only a few of them have been successfully applied and were used widely. In this section we introduce a selection of state-of-the-art heuristic retrieval algorithms that comprise specific features such that a combination of them would be expected to yield better extracted keywords.

A. Term Frequency - Inverse Document Frequency (TF-IDF)

The measure of *term specificity* proposed by Karen Spärck Jones in 1972 is one of the most often used term weighting factors [9]. This famous measure is now known as TF-IDF and consists of the two components *term frequency* and *inverse document frequency*, multiplied with each other.

$$tf - idf(t) = tf(t, w_i) * idf(t, d) = x_t^{w_i} * log\frac{|w|}{|d_t|}$$

In our case with a single document, the TF component x_t^w of a term t in a window w represents the number of occurrences of t in w. The IDF component of a term t is obtained by the quotient of the total number of windows and the number of windows in the document containing t – the elite set of windows. A large number of variants of TF-IDF is known in the field of text data mining because this weighting scheme has proven to be extremely simple, robust and successful.

Despite its success there are inherent limitations to TF-IDF that can be overcome with a combination with other heuristics, that e.g. account for other additional document characteristics. We furthermore provide a feedback method to determine the optimal combinations.

B. The Bernoulli Model of Randomness

A number of statistical language models are available to approximate the term distribution in a document. For each term t, the term weight is inversely proportional to the probability of the term occurrences x_t generated with the probability model of choice. We apply the classic approach of modeling the presence or absence of a term by using a binomial distribution. The general assumption of the Bernoulli Model of Randomness is: a term t is spread across a document d with N windows $w_1, .., w_N$ according to the binomial law [11]:

$$B(t) = \binom{x_t^d}{x_t^w} p^{x_t^w} q^{x_t^d - x_t^w} \tag{1}$$

The probabilities p and q are defined as: $p = \frac{1}{|d|}$ and $q = \frac{|d|-1}{|d|}$. Contrary to the TF-IDF heuristic, the Bernoulli model does not take into account the *document frequency* of t. In order to generate a weight and to reduce the computational costs, we follow the proposal of Amati et al. and determine the *informative content* for each term in the document:

$$inf(t) = -logB(t) \tag{2}$$

Accordingly the terms with the highest informative content (weight) are considered the most probable keywords in the documents.

C. The Γ-Metric

Most frequency-based heuristics consider a document as a *bag of words* model. A bag of words model is a simplified representation of a text that does not take into account word order and grammatical structure. Zhou and Slater [5] developed the Γ-metric, which is a computationally simple and robust measure that accounts for word order and long-range relations in text by analyzing the word distribution in a text. This metric is based on the assumption that important words are locally concentrated in a limited area and do not spread equally distributed across the entire document.

The Γ-metric is based on an imaginary time-series of term occurrences $\{\tau_0, \tau_1, ..., \tau_{x_t}, \tau_{x_t+1}\}$, whereas x_t accounts for the number of occurrences of the term t. The space before the first term occurrence is τ_0 and after the last term occurrence τ_{x_t+1}. Zhou and Slater define the *separation* around term occurrence t_i with $sep(t_i) \equiv \frac{\tau_{i+1} - \tau_{i-1}}{2}$, with $1 \leq i \leq x_t$. The separation represents the median distance of a single occurrence in the document. The mean waiting time $\hat{\mu}$ is defined as follows:

$$\hat{\mu} = \frac{|d|}{x_t^d},$$

The term occurrence at position i is defined as a cluster point $\gamma(t_i)$ if the separation of t_i is less than the mean waiting time $\hat{\mu}$:

$$\Gamma(t_i) = \begin{cases} \frac{\hat{\mu} - sep(t_i)}{\hat{\mu}} & \textit{if } t_i \textit{ is a cluster point} \\ 0 & else. \end{cases}$$

where $|d|$ is the total number of terms in the document and x_t^d is the number of occurrences of term t in the document d. The individual cluster points $\gamma(t)$ represent the spread of t across the whole document d. Hence a normalized average of γ can be used to weigh the terms in the document:

$$\Gamma(t) = \frac{1}{x_t^d} \sum_{i=1}^{x_t^d} \gamma(t_i)$$

D. The Laplace Law of Succession

If the term t has not occurred for a long time and suddenly t appears once, the expectation to find further occurrences of t rises. The effect of rising expectation is called the *aftereffect* of future sampling and is similar to the notion of *burstiness* [17]. The probability of an observed term t contributing to the discrimination of a window is assumed to correlate to the probability of another appearance of t [11]. This probability is obtained by the conditional probability $p(m+1|m, w)$ of the term t by the aftereffect model. This probability is only related to the *elite* set of windows - the set that contains the term t.

Amati et al. applied the Laplace Law of Succession to account for burstiness in their proposed algorithm [11]. They first surveyed the performance of several aftereffect models, and found the Laplace Law of Succession provided reasonable results despite its simplicity and linear complexity. We include this method in the set of selected algorithms for combination because it accounts for term frequency of a term t across the whole document and differs significantly from the Γ-score.

It is assumed that $x_t^d + 1$ appearances of t have been observed in d. Based on the assumption of term independence and Laplace's Law of Succession, the probability of $x_t + 2$ appearances is defined as $\frac{x_t^d + 1}{x_t^d + 2}$. Assuming that $x_t^d - 1$ occurrences have been observed, the additional appearance is approximated with the following equation [11]:

$$laplace(t) = \frac{x_t^d}{x_t^d + 1} \tag{3}$$

Laplace's Law of Succession models the aftereffect of the appearance of a term for the elite set of windows.

IV. Combination of Weighting Algorithms

The performance of keyword extraction algorithms is closely related to the properties of the retrieval heuristics used to determine the keywords. Fang et al. formally defined and characterized a set of desirable constraints that retrieval heuristics should meet [10]. They concluded that none of their analyzed retrieval formulas satisfies their formal constraints which comprise characteristics related to:

1) *term frequency*
2) the effect of *document frequency*
3) *length normalization*

The work of Clinchant and Gaussier extends the formal retrieval constraints defined by Fang et al. [18] by providing a formal definition of

4) *burstiness*

where a *burst* is defined as a sudden unexpected rise of term frequency in a short period of time.

Since no retrieval heuristic meets all the constraints defined by [10] and [18], a combination of multiple heuristics may give improved results. Below we describe five of the most successful combination approaches used to combine retrieval heuristics. In Section VI, we apply these methods on sample texts in order to evaluate their performance for keyword extraction.

A. The Divergence from Randomness Framework

Amati et al. proposed a term weighting function that combines two heuristics and consists of the following two probabilities [11]:

$$weight = -log_2 Prob_1 \cdot (1 - Prob_2) \tag{4}$$

The first component is defined as *informative content* and refers to a model of randomness with probability distribution $Prob_1$. Whereas the informative content of a term t depends on the chosen model of randomness and can meet the constraints 1), 2), or 3), the second probability $Prob_2$ incorporates the aspect of 4) burstiness and is called the *aftereffect of future sampling*. The quality of the results of Equation 4 depends solely on the choice of the models for $Prob_1$ and $Prob_2$.

B. Rank Aggregation

For improved results and a more robust weighting algorithm, ranking aggregation methods can be utilized for combining the results of multiple heuristics. In principle, each heuristic produces a ranked list of terms and the number of extracted top-ranked keywords is determined by the user. The advantages of ranking include simplicity, scalability and good empirical success [12], though they do not take into account the actual term weights and statistical properties of the retrieval results. Even though absolute values might be in the same absolute scales, their relative scales might differ and therefore represent a diverse relevance. In this paper we utilize three different ranking aggregation methods to keyword extraction for single documents:

1) Minimum Ranking Method: The Minimum Ranking Method was proposed by Louloudis et al. [13], and despite it's simplicity it outperformed most of the state-of-the-art ranking aggregation methods. The final ranking score of a term t_i is the minimum rank position on all the different retrieval rankings $\rho_1..\rho_n$:

$$rank(t_i) = \min_{0 \le k \le n} (\rho_k(t_i))$$

2) Borda Count (BC): The Borda Count (BC) determines the rank of a term t_i by its number of points according to its position in the different retrieval rankings $\rho_1..\rho_n$. The term with the highest term weight receives the highest number of points, which is $m-1$ – the maximum number of positions decreased by one. The final rank of t_i represents its mean position over all the rankings $\rho_1..\rho_n$:

$$BC(t_i) = \sum_{j=1}^{n} (m - \rho_j(t_i))$$

3) Schulze Method: The Schulze method is a voting system that has been developed by Markus Schulze in 1997 and is based on the *Condorcet* method [19]. The Condorcet algorithm performs pairwise comparisons of the ranks of all candidates (here: terms $t_1..t_m$). The winner of a Condorcet method is the candidate which is preferred over all other candidates – the candidate that wins most of the pair wise comparisons.

The Schulze method first counts how many times the rank of term t_i ranks higher than the rank of t_j and vice versa. If t_i ranks higher than t_j in one comparison, then t_i wins this comparison, and the number of wins $d[t_i, t_j]$ is increased by one. The results of all comparisons can be stored in an $m \times m$ matrix or in a directed graph, where the terms represent the nodes. If $d[t_i, t_j] > d[t_j, t_i]$ (t_i defeats t_j) we draw an edge from the node t_i to the node t_j. The output of the Schulze method is then determined by computing the *strongest path* p between all candidate pairs in the graph [19]. The strongest path $p[t_i, t_j]$ between the nodes t_i and t_j is an ordered set of candidates $C(1)..C(n)$ with the following properties:

1) $t_i = C(1)$ and $t_j = C(n)$
2) $\forall k \in [1, ..., n-1]$,
   $d[C(k), C(k+1)] > d[C(k+1), C(k)]$
3) $\forall k \in [1, ..., n-1]$,
   $p[t_i, t_j] =_{def} max(d[C(k), C(k+1)])$

If there does not exist a path between the nodes t_i and t_j the strongest path equals zero, otherwise the strongest path represents the maximum value, such that there is a path of the accumulated number of wins from t_i to t_j. The term t_i is a *Schulze winner* iff $p[t_i, t_j] > p[t_j, t_i]$ for every other term t_j:

$$t_i \ wins \iff p[t_i, *] \geq p[*, t_i]$$

C. Principal Component Analysis

PCA is a statistical method used to structure data and identify patterns in a multivariate dataset by decomposing it into orthogonal components. The PCA-procedure is defined

in such a way that the first of those resulting components comprises the largest amount of variance - the *principle component* of the data set. Each subsequent component has the next highest variance and is orthogonal to all preceding components. Another advantage of PCA is that these patterns allow for dimension reduction, e.g. if three out of four components already reveal the internal structure of the data to a sufficient degree, the dimensionality of the data can be reduced to only the first three components.

As we utilize n algorithms, we generate term weights for each term as described in Section III. To perform PCA, we normalize the weights for each dimension $\delta_1..\delta_n$ and then subtract the mean $\bar{\delta}$ from each single value – the mean is the average across the corresponding dimension. The resulting data set has a mean of zero. In the next step we have to create a covariance matrix:

$$C^{n\times n} = (c_{i,j}, c_{i,j} = cov(\delta_i, \delta_j))$$

The square matrix $C^{n\times n}$ contains n rows and n columns. With the covariance matrix it is possible to calculate the eigenvectors and the eigenvalues $\lambda_1..\lambda_n$ of $C^{n\times n}$:

$$\begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ & \vdots & & \\ 0 & \dots & 0 & \lambda_n \end{pmatrix}$$

The eigenvalues $\lambda_1..\lambda_n$ account for the amount of variation in the dimensions and therefore give an indication of the significance of the individual weighting algorithms.

Li et al. [14] took the eigenvalues into account to determine the *contribution rate* α:

$$\alpha_i = \frac{\lambda_i}{\sum_{k=1}^{n} \lambda_k}$$

The contribution rate α is then used to create a weighted combination of keyword weights for terms $t_1..t_m$ in the dimensions $\delta_1..\delta_n$:

$$weight_{t_i} = \alpha_1\delta_{1,t_i} + \alpha_2\delta_{2,t_i} + ..\alpha_n\delta_{n,t_i}$$

In our approach we do account for all eigenvalues $\lambda_1..\lambda_n$.

V. Selections for a Successful Combination

Numerous works have demonstrated that the combination of keyword extraction methods is more successful than single heuristics [11], [6], [12], [13]. In this section we analyze the result space of the presented heuristics (see Section III) and present a simple method to determine optimal candidates for a successful combination procedure.

The properties of the retrieval heuristics play a crucial role for their combination and the quality of the retrieved results. In addition to these properties, the document to be analyzed may also influence the outcome: document length, structure, and writing style of the author do affect the behavior of heuristic algorithms. We apply PCA to the result space of six different combinations of retrieval heuristics to determine potential candidates for a successful combination. Here, the objective of using PCA is not to reduce one of the input components. Quite the contrary: a balanced set of components accounts for a low correlation rate of the data and therefore for a diverse set of results. We believe that a highly diverse set of results embraces more individual characteristics of the underlying document and is therefore desirable. Since PCA exclusively considers the result space, it accounts for all effects that have influenced the result generation and does not require any adjustment of additional parameters.

Based on the basic properties of the heuristics, presented in Section III we have selected the following combinations for evaluation:

1) Γ-Score and TF-IDF
2) Γ-Score and the Bernoulli Model
3) TF-IDF and the Bernoulli Model
4) Laplace and Γ-Score
5) Laplace and TF-IDF
6) Laplace and the Bernoulli Model

We only combine two algorithms at a time in order to emphasize the effect of their individual properties on the result space. A successful selection of retrieval algorithms should comprise the defined constraints with respect to the source text (Section IV) and may require a set of three or more retrieval algorithms.

To determine the variance of the result set for each of the six algorithm pairs, we applied them to three different texts of different genres, lengths and structures: the top 14 ECBS paper abstracts of 2012, US-President Obama's State of the Union Address in January 2012 and the English book Treasure Island by Robert Louis Stevenson.

Scientific research paper abstracts are very suitable for keyword extraction because the length of the abstracts is very similar, the scientific texts contain a lot of potential keywords, and the number of windows is relatively low. We treat each abstract of the top 14 ECBS papers from 2012 as a window and the collection of windows as a document. In contrast, the presidential speech represents a larger document of a different writing style. It is divided into 104 consecutive paragraphs of different length, representing the individual windows. Our largest sample – the book Treasure Island – is the biggest document of the three test cases and consists of 5758 paragraphs of different size.

For each of the combinations 1) to 6), we performed the algorithms sequentially and subsequently analyzed the results with PCA. The fraction of the individual eigenvalues of the sum of the eigenvalues of all components as defined in Equation IV-C is depicted in Figure 1. The single bar charts depict the contribution of the results of each individual retrieval algorithm to the variance of the results of the combination for the three sample texts. The results for combination 1) and 2) show that the combination of

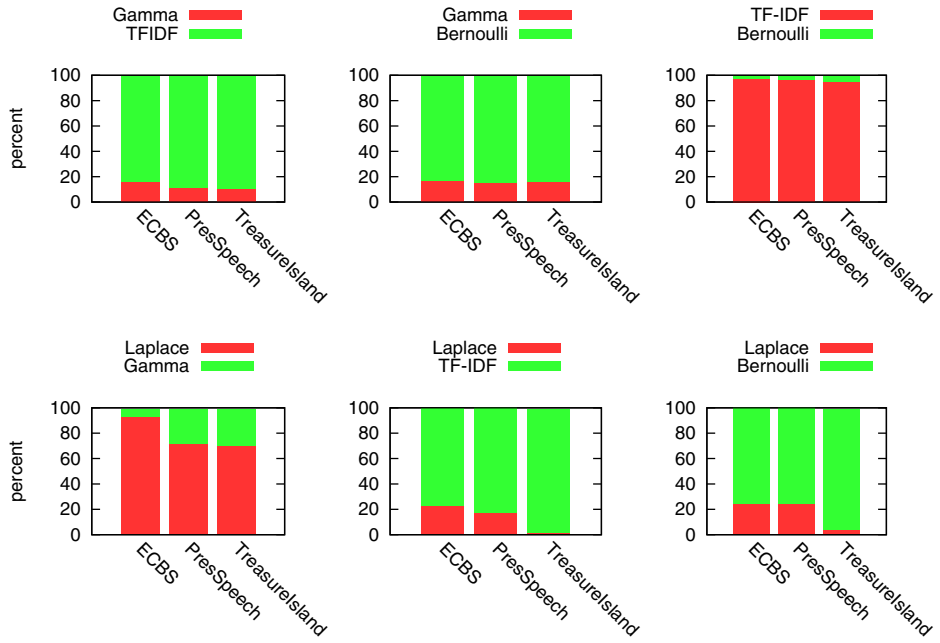

Figure 1. The PCA eigenvalue proportions for six different heuristic retrieval combinations show the individual contribution of variance to the result set.

a frequency-based heuristic with a term-distance measure appears to be beneficial, although the contribution of the γ-score is significantly lower than the contribution of TF-IDF/ the Bernoulli Model. It is clearly visible that the two frequency-based measures TF-IDF and the Bernoulli Model do not contribute to richer results when combined with each other. The bottom row graphs indicate that the test documents affect the results of the Laplace measure because the contribution of the Laplace measure varies significantly throughout the test sets. This is due to the fact that its contribution of variability to the result set decreases with larger document size. This behavior is not surprising since the Laplace Law of Succession is solely based on term frequency. In fact, this indicates that the γ-score is a better candidate for modeling the burstiness for larger documents.

The results of the PCA-analysis account for the properties of the retrieval heuristics as well as for the characteristics of the test documents. We propose to apply PCA to the set of results of the individual heuristics before the final combination procedure in order to select the most beneficial retrieval heuristics for combination.

VI. Comparison Data Fusion Methods

In this section we provide a comparison of the state-of-the-art data fusion methods for keyword extraction. We show that the heuristics perform well for single documents by applying them to a real-world-example.

The evaluation of the performance of keyword extraction algorithms is challenging because it is a very subjective task. Whereas a term appears to be a keyword for one reader, it might not be a meaningful term for another one. Besides, single terms may only become meaningful in combination with another term, or they may change its meaning in another context. For this reason we decided to choose a telling real-world example as a source document and evaluated the results by hand. In addition, we treat the keyword extraction as a classification process where the extracted term is either a keyword or not. This allows the use of the classic evaluation methods: *precision* and *recall* (*accurracy* is not considered due to the high number of negatives). Precision is the fraction of extracted keywords that are actually keywords:

$$precision = \frac{|keywords\ extracted \cap keywords\ identified|}{|keywords\ extracted|}$$

Not all rankings exclude terms from the set of potential keywords but provide a ranked list of potential keywords, we select the top ten keyword candidates for the precision determination. Recall is the fraction of extracted keywords out of all keywords in the whole document. As it is very subjective to determine the exact amount of potential keywords in a document, we do not consider recall in this evaluation.

Based on the results of Section V we select the most favorable algorithm pairs

1) Γ-Score and TF-IDF and
2) Γ-Score and the Bernoulli model

for combination. The top 14 ECBS-paper abstracts of 2012 were used as a source document, where the single abstracts represent the windows. The ECBS paper abstracts allow a more accurate categorization of the results because each

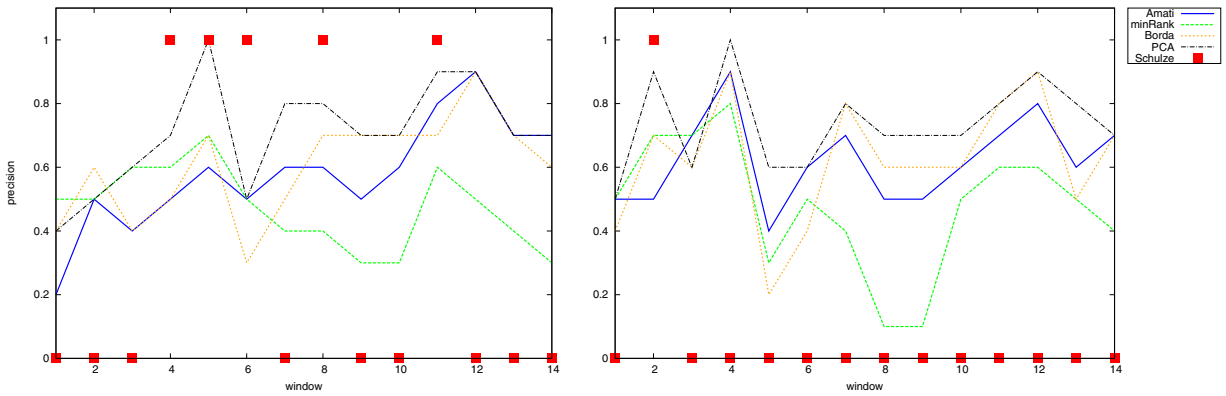

Figure 2. The graph shows the precision of the extracted keywords from 14 ECBS paper abstracts 2012. The data fusion methods have been applied to the results of Γ-Metric / TF-IDF and Laplace Law of Succession / Bernoulli model.

abstract comes with author assigned keywords, a headline, and comprises only a single topic – the subject of the paper. Figure 2 shows the results of applying all presented data fusion methods to heuristic algorithm selection 1) (left) and 2) (right). The two graphs show the precision rate for each fusion method for each of the 14 abstracts (windows). Since the Schulze method only determines the winner candidate for both heuristics, the resulting term is either categorized as a keyword (100% precision) or not (0% precision). Since we did not perform extensive preprocessing nor filter stop-words from the source text, we expect a relatively low precision for the extracted keywords.

Overall, the PCA-weighted combination has the highest average precision whereas the precision of Divergence from Randomness (Amati) and minRank rank lower but the extracted keywords of these methods appear to be subjectively more compelling, whereas the quality of keywords can not be represented with the chosen metrics. Even though the order-based results are not scale-sensitive, they perform surprisingly well for our example. Due to its single keyword output rate, the Schulze method performs not very well.

The results show that the presented combination methods perform well with the chosen retrieval heuristics and extract correct keywords from the sample texts. Extensive preprocessing was not necessary. The PCA-based combination method outperformed all other methods. Since the combination processes are computationally expensive, it is not desirable to apply them to large document collections.

VII. Conclusion

Single retrieval heuristics fail to encompass all information of a text that is potentially relevant for a keyword extraction process. We introduced state-of-the-art retrieval combination and ranking aggregation methods to combine retrieval heuristics that works best with single documents. We evaluated these combination methods with real-world-examples for different combinations of retrieval heuristics. The most successful combination method for keyword extraction in single documents is PCA due to the continuously high precision rate in our tests.

In this paper we utilized PCA as a parameter-free and effective method for determining an optimal selection of retrieval heuristics for combination. This approach accounts for the properties of the heuristics as well as for the characteristics of the analyzed text. We validated this approach with the presented heuristics, applied to texts of different size and different genres.

In the near future we plan to conduct further studies with a larger number of heuristics in order to develop a flexible combination approach for different extraction scenarios. The PCA-based approach for algorithm selection can be utilized to determine an adaptable configuration of retrieval heuristics for keyword extraction.

References

[1] P. D. Turney, "Learning algorithms for keyphrase extraction," *Information Retrieval*, vol. 2, pp. 303–336, May 2000.

[2] N. Fuhr, "Probabilistic models in information retrieval," *The Computer Journal*, vol. 35, no. 3, pp. 243–255, 1992.

[3] T. Tomokiyo and M. Hurst, "A language model approach to keyphrase extraction," in *Proceedings of the ACL 2003 workshop on Multiword expressions: analysis, acquisition and treatment - Volume 18*, ser. MWE '03. Stroudsburg, PA, USA: Association for Computational Linguistics, 2003, pp. 33–40.

[4] A. Munoz, "Compound key word generation from document databases using a hierarchical clustering ART model," *Intelligent Data Analysis*, vol. 1, no. 1-4, pp. 25–48, 1997.

[5] H. Zhou and G. W. Slater, "A metric to search for relevant words," *Physica A: Statistical Mechanics and its Applications*, vol. 329, no. 12, pp. 309 – 327, 2003.

[6] C.-H. Lee, C.-H. Wu, and T.-F. Chien, "Burst: A dynamic term weighting scheme for mining microblogging messages," in *Advances in Neural Networks ISNN 2011*, ser. Lecture Notes in Computer Science, D. Liu, H. Zhang, M. Polycarpou, C. Alippi, and H. He, Eds. Springer Berlin / Heidelberg, 2011, vol. 6677, pp. 548–557.

[7] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," in *Proceedings of the seventh international conference on World Wide Web 7*, ser. WWW7. Amsterdam, The Netherlands, The Netherlands: Elsevier Science Publishers B. V., 1998, pp. 107–117.

[8] J. Wang, J. Liu, and C. Wang, "Keyword extraction based on pagerank," in *Proceedings of the 11th Pacific-Asia conference on Advances in knowledge discovery and data mining*, ser. PAKDD'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 857–864.

[9] K. S. Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of Documentation*, vol. 28, pp. 11–21, 1972.

[10] H. Fang, T. Tao, and C. Zhai, "A formal study of information retrieval heuristics," in *Proc. of the* 27^{th} *ann. int. conf. on Research and development in information retrieval - SIGIR '04*. New York, NY, USA: ACM, Jul. 2004, p. 49.

[11] G. Amati and C. J. Van Rijsbergen, "Probabilistic models of information retrieval based on measuring the divergence from randomness," *ACM Trans. Inf. Syst.*, vol. 20, pp. 357–389, October 2002.

[12] R. Prati, "Combining feature ranking algorithms through rank aggregation," in *Neural Networks (IJCNN), The 2012 International Joint Conference on*, june 2012, pp. 1–8.

[13] G. Louloudis, A. Kesidis, and B. Gatos, "Efficient word retrieval using a multiple ranking combination scheme," in *Document Analysis Systems (DAS), 2012 10th IAPR International Workshop on*, march 2012, pp. 379 –383.

[14] C.-J. Li and H.-J. Han, "Keyword extraction algorithm based on principal component analysis," in *Intelligent Computing and Information Science*, ser. Communications in Computer and Information Science, R. Chen, Ed. Springer Berlin Heidelberg, 2011, vol. 135, pp. 503–508.

[15] T. K. Landauer, P. W. Foltz, and D. Laham, "An Introduction to Latent Semantic Analysis," *Discourse Processes*, no. 25, pp. 259–284, 1998.

[16] M. A. Hearst, "TextTiling: Segmenting Text into Multi-paragraph Subtopic Passages," *Computational Linguistics*, 1997.

[17] W. Feller, *An Introduction to Probability Theory and Its Applications, Vol. 2*. Wiley, 1967.

[18] S. Clinchant and E. Gaussier, "Retrieval constraints and word frequency distributions a log-logistic model for ir," *Information Retrieval*, vol. 14, no. 1, pp. 5–25, Feb. 2011.

[19] M. Schulze, "A new monotonic, clone-independent, reversal symmetric, and condorcet-consistent single-winner election method," *Social Choice and Welfare*, vol. 36, pp. 267–303, 2011.