Keeping Keywords Fresh: A BM25 Variation for Personalized Keyword Extraction

Margarita Karkali
Athens University of
Economics and Business
karkalimar@aueb.gr

Vassilis Plachouras
Presans
vplachouras@acm.org

Constantinos Stefanatos
Athens University of
Economics and Business
justcureious@gmail.com

Michalis Vazirgiannis
Telecom Paris-Tech, LIX -
Ecole Polytechnique, Athens
University of Economics and
Business
mvazirg@aueb.gr

ABSTRACT

Keyword extraction from web pages is essential to various text mining tasks including contextual advertising, recommendation selection, user profiling and personalization. For example, extracted keywords in contextual advertising are used to match advertisements with the web page currently browsed by a user. Most of the keyword extraction methods mainly rely on the content of a single web page, ignoring the browsing history of a user, and hence, potentially leading to the same advertisements or recommendations.

In this work we propose a new feature scoring algorithm for web page terms extraction that, assuming a recent browsing history per user, takes into account the *freshness* of keywords in the current page as means of shifting users interests. We propose BM25H, a variant of BM25 scoring function, implemented on the client-side, that takes into account the user browsing history and suggests keywords relevant to the currently browsed page, but also fresh with respect to the user's recent browsing history. In this way, for each web page we obtain a set of keywords, representing the time shifting interests of the user. BM25H avoids repetitions of keywords which may be simply domain specific stop-words, or may result in matching the same ads or similar recommendations. Our experimental results show that BM25H achieves more than 70% in precision at 20 extracted keywords (based on human blind evaluation) and outperforms our baselines (TF and BM25 scoring functions), while it succeeds in keeping extracted keywords fresh compared to recent user history.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; M.7 [**Knowledge Retrieval**]: Miscellaneous

General Terms

Algorithms, Experimentation, Performance

Keywords

fresh keywords, keyword extraction, web personalization

TempWeb '12, Apr 16-17 2012, Lyon, France
ACM 978-1-4503-1188-5/12/04..

1. INTRODUCTION

Keyword extraction is an important part of various text mining tasks on the web, including user profiling recommender systems and contextual advertising. Extracted keywords are typically optimized for relevance to the topics of interest of a user in profiling systems, or to the web page's topic in the case of contextual advertising. Even if the extraction considers the user's topics of interest to bias keyword extraction, keywords are selected independently of any previously selected ones. Hence, when we consider the browsing history of a user, there is likely to be a significant overlap in the keywords selected for similar web pages, leading to the same recommendations. As a result, the repetition of recommendations may be boring or even annoying for the user, and it may reduce the effectiveness of the recommender system, regardless of the quality of the extracted keywords, because the selected keywords have been employed before. In popular existing browser plug-ins for recommendations, such as Chrome's 'Google Similar Pages' and Firefox's 'SimilarWeb' [1], the recommended pages are chosen independently of the browsing history of the user.

In this paper we propose a method for keyword extraction and recommendation that aims to select not only relevant but also *fresh* keywords, that is, relevant keywords which have not been selected for the recently browsed web pages. We consider both the browsing history of the user and the temporal distribution of term occurrences in a sequence of viewed web pages. More specifically, we assume a temporally evolving corpus, constructed from a user's browsing history and we introduce *temporal document frequency*, a novel function to compute the importance of keywords not only according to their relevance to the currently browsed page but also their freshness to browsing history. We employ the temporal document frequency to define a temporal version of *IDF* and propose BM25H, a variation of BM25 scoring function based on the new temporal document frequency. The new scoring function BM25H aims to rank relevant but also fresh keywords for the web pages that a user currently visits. The proposed method allows us to capture the shifting interests of the user and to discover new relevant keywords, which can lead to diversified recommen-

[1] `bit.ly/dIUkBw`, `bit.ly/ePxcH8`

dations. The method we propose needs access to browsing history of the user and thus, it should be implemented on the client side (*e.g.* as a browser plug-in).

We perform a comprehensive experimental evaluation with blind user evaluation and data from Wikipedia and news web sites. The evaluation results show that BM25H outperforms in terms of relevance a simple term frequency-based keyword selection as well as BM25 and at the same time it returns fresh keywords.

The remainder of the paper is organized as follows. In Section 2, we introduce the notation we use through-out the paper and we define the problem of fresh keyword extraction. In Section 3, we introduce the proposed keyword weighting formulas. Section 4 describes the experimental setting and the development of the dataset used in the evaluation process. In Section 5 we describe the experimental results and show the effectiveness of the proposed methods. In Section 6 we discuss the issue of parameter tuning for BM25H. Section 7 provides a review of related works. Finally, Section 8 closes this work with our conclusions.

2. PROBLEM DEFINITION

We study the problem of extracting and weighting relevant and fresh keywords for a web page a specific user is currently visiting. The extracted keywords are based on her browsing history in the following setting. We assume that a user is browsing web pages. For a web page d_x that is browsed at time x, we aim to extract a set of informative keywords, which are both relevant to the topic of the web page as well as *fresh*, in order to match personalized advertisements, or to make recommendations about similar web pages that may be of interest to the user. We choose to extract only unigrams. We state that a phrase extraction mechanism is not necessary, based on the results of the recent work of Broder et al [3], who experimentally showed that the use of phrases do not improve the performance of either semantic or the syntactic matching. The proposed method BM25H can also be applied to phrase extraction tasks. A fresh keyword is a keyword which has not been selected for any of the M previously browsed web pages $d_{x-M}, d_{x-M+1}, \ldots, d_{x-1}$.

We introduce keyword freshness in order to handle keyword extraction based on the keywords distribution within user's history. Figure 1 shows a demonstrative example of the term occurrences of four terms in the last fifteen pages from a user's browsing history. The horizontal axis represents time. Document d_x with $x = 15$ is the most recently browsed document. Assuming that in the next web page all four terms co-occur and are equally important to the web page, we need to rank them in a way that termA, which has not been seen recently, will be ranked high, termB will be ranked lower than termA, as it has been seen in the near past, and termC will be ranked lower than termA and termB because of its recurring appearance in the most recent web pages. In addition, termD, which re-occurs in every third document in the example, will be penalized if we take into account its past occurrences, or it will be ranked high only when it appears to be very important for the current page.

The keyword extraction and weighting is performed at the client-side, for example within the user's browser. This setting limits the complexity and the processing requirements of the algorithms we can employ. In addition, the keyword extraction and weighting must be performed online at the same time that the user is browsing web pages. In this setting, we assume that we can keep statistics from the N most recently browsed web pages $d_{x-N+1}, d_{x-N+2}, \ldots, d_x$. The deployment on client-side surpasses the potential privacy concerns as it allow us to access and use personal data locally.

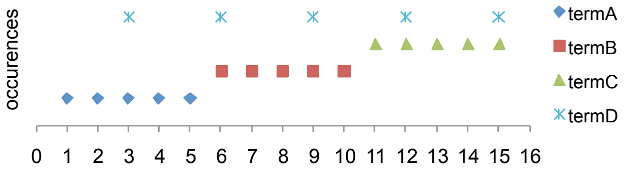

Figure 1: Illustrative example of keyword occurrence patterns in a user's last 15 browsed documents

Our setting is different from contextual advertising in two ways. First, the extracted keywords do not depend only on the currently browsed Web page, but also on the user's browsing history, enabling personalized advertising and recommendations. Second, we extract keywords from any web page, and not only those pages which are enabled for displaying ads.

3. TEMPORAL KEYWORD WEIGHTING

A central issue in our design is to capture the shifting interests of the user in terms of keyword/terms that appear in the browsed web pages. If we observe repetition in the extracted keywords from each new page, we understand that this reaffirms the continuing interests of users. On the other hand, terms that appear for the first time or after a long time indicate a new interest.

In the remainder of this section, we first define a temporal document corpus from the recently browsed documents of a user (Section 3.1). Next, we introduce temporal document frequency, which adjusts the document frequency of a term with respect to its occurrence patterns (Section 3.2). Finally, we introduce a new weighting function for extracting keywords from web documents, using the temporal patterns of term occurrences in the browsing history of a user (Section 3.3).

3.1 User History as Temporal Corpus

The use of TF-IDF weights in feature extraction is effective to eliminate common words in a corpus and select terms that describe a document well. The TF-IDF weight requires a predefined corpus to extract the required values, most importantly the document frequency (DF) of term t, which is the number of documents in a corpus that contain t. To introduce the concept of TF-IDF weights on keyword extraction, while the user browses the web, we must define a corpus. This corpus must be stored locally and be used to extract statistics for the terms of the page user is currently visiting.

To take advantage of TF-IDF weighting functions for the task of keyword extraction we introduce the concept of the user's recent browsing history as a corpus. We define a sliding window of the last N most recently browsed web pages, from which we compute the document frequencies and any other statistics needed, such as the average document length used in BM25. In this way, terms that tend to occur frequently in the browsing history will be penalized and will not be recommended as keywords. Examples of such terms are the common words in a specific domain. For example, in

Wikipedia articles it is common to have words such as `retrieve`, `edit`, `Wikipedia` and `encyclopedia` as a suggested keyword, because e.g. the word `retrieve` in many articles occurs in every reference in the corresponding section. By using the browsing history as the corpus to compute TF-IDF weights from, we can eliminate such domain-specific stopwords.

The use of IDF weights results in the gradual reduction of weights of keywords that occur in many documents, and hence, their elimination from the set of recommended keywords. This filtering can be a desirable effect depending on the application that employs the keywords. For example, in a recommender system, the gradual adjustment of keywords, which were ranked high at their first occurrences, will result in refreshing ads with new content, still relevant to the current interests of the user, but different from the recently matched ones. In this case, although a user may keep visiting web pages of a certain topic, the recommendations will still be fresh, offering a better user experience. The above observations lead us to use a TF-IDF base scoring function in order to extract keywords by taking into account the user browsing history. We chose BM25 ranking function [16], and we define as corpus the last N visited web pages.

BM25 uses an evolving corpus to compute the statistical measures used in its formula. Thus, when a term t occurs for the first time in user's history, the corresponding DF value is equal to 1. While the term t continue to re-occur in user's browsing history, its DF value continue to increase. When a web page exits the sliding window of the last N web pages the DF values of all terms in the page are reduced by one. Using BM25 in keyword extraction tasks will result in penalization of terms that reoccur in the last N web pages, regardless to their distance from the current web page. BM25 with browsing history as corpus involves the concept of freshness. As BM25 penalize terms that are common in the corpus used, when the corpus is time-evolving, the uncommon terms are also the fresh ones.

3.2 Temporal Document Frequency

The drawback of the technique described in Section 3.1 is that regardless of their last occurrence in browsing history, frequent terms in the defined window will be penalized in the same way. For example, assume that during the first half of the window a user browses web pages on a specific topic with a dominant word t. During the second half of the window the user browses web pages on a different topic with a dominant word t'. If terms t and t' co-occur in the following web page, they will be treated in the same way and will be both penalized, even though t is relevant to the current topic and its last occurrence was about $N/2$ web pages ago. In Figure 1 different term distributions in a window are illustrated. All the four presented terms are treated in the same way with BM25.

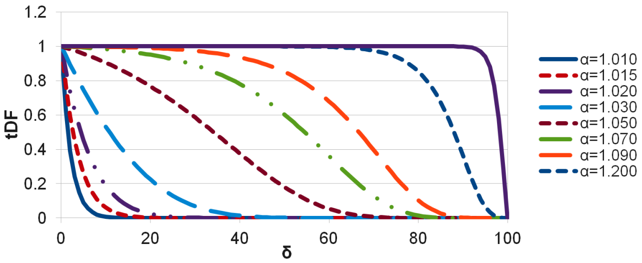

Figure 2: tDF evolution of a term that apeared only once in browsing history, for different values of α parameter for $N = 100$.

To address the issue described above, we introduce the temporal dimension in the corpus statistics. We create a pseudo document frequency, which relates to both the document frequency of a term and its distribution of occurrences over time. Every time a new web page arrives at the client, the document frequency is reduced by a percentage related to its last occurrence in the sequence of browsed web pages. The concept of the sliding window corresponds to the number of times the document frequency of a term will be reduced before it is set equal to zero and be considered unseen. We define the temporal document frequency $tDF_x(t)$ as follows:

$$tDF_x(t) = \big(tDF_{x-1}(t) + O_x(t)\big) \times (1 - \alpha^{(\delta - N)}) \qquad (1)$$

where $tDF_x(t)$ is the document frequency of term t, after the insertion of the currently browsed web page (with sequence number x) in the corpus. δ denotes the distance of the last occurrence of term t to the currently browsed web page. $O_x(t)$ can be 0 or 1 and denotes the occurrence of term t in document with sequence number x. The window size is denoted by N, and α stands for the reduction parameter, which controls the rate with which document frequency is reduced.

The decay function is capable of handling the distribution of terms in user's history in the way described in Section 2. Consider the terms' distributions presented in Figure 1. If in the sixteenth web page all four terms co-occur, the tDF of termA, termB and termC will be approximately 1, 2.5 and 4 respectively (for N=100 and α=1,025), which means that using any TF-IDF scoring function termA will be ranked higher as its tDF is lower than the other important terms.

When a term keeps showing up in browsing history, its tDF will keep increasing. As tDF is designed to be used in TF-IDF scoring functions, it must have a maximum value equal to corpus length, to prevent a negative value to the logarithm in IDF equation 3. We define an upper bound equal to N. If a term reaches a document frequency equal to N it is not increased further. Reaching the maximum value of tDF implies that the term appeared in all web pages of the corpus.

Although the window length is the parameter which defines when the tDF of a term will become equal to zero through repeated reductions, a small value of α can result in earlier tDF elimination, *i.e.* in narrower window. Moreover, the rate of change of tDF affects the balance between relevance and freshness. In other words, as the user browses the web, a high value of α may result in high penalization of relevant keywords because of their multiple occurrence some time in the past. The tuning parameters process will be discussed later in section 6. Figure 2 presents tDF evolution of a term that is not appearing any more, for different values of α with $N = 100$. Notice that the reduction curve will not change, for different initial values of tDF. In Figure 2 we have an initial value $tDF_x(t) = 1$ where x is the sequence number of term t last occurrence. It can be easily seen that as α parameter increases, tDF becomes zero only when it reaches the end of the window. This is because, for high values of α the decay function results in marginal reduction of tDF for x<N and $tDF = 0$ when $x = N$.

3.3 BM25H Scoring Function

Next, we introduce BM25H, which is based on BM25 term weighting function. In BM25H, we replace document frequency with the temporal document frequency $tDF_x(t)$, presented in the former subsection. The term weighting function BM25H assignes a weight to a term t based on its relevance to the document d, and is defined as follows:

$$BM25H(t,d) = \frac{tIDF(t) \times TF(t,d) \times (k_1+1)}{TF(t,d) + k_1 \times (1 - b + b\frac{|d|}{avgdl})} \qquad (2)$$

where $|d|$ is the number of tokens in document d, $avgdl$ is the average document length of the N most recently browsed web pages. The parameters b and k_1 correspond to the same parameters of BM25, controlling the document length normalization and term frequency saturation, respectively. Last, $tIDF(t)$ is defined as follows:

$$tIDF(t) = log\frac{N - tDF_x(t) + 0.5}{tDF_x(t) + 0.5} \qquad (3)$$

In order to replace the traditional statistic measure DF, with tDFs described in section 3.2, we must ensure that tDF takes values in the range of 0 and N. As described in section 3.2, tDF cannot take negative values and we upper bound it by N. In this way, the use of tDF in the classic IDF formula is ensured to give rational scoring values.

BM25H enables the control of the importance of a term occurrence with respect to the position of the corresponding document in the window of the last N browsed documents. A measure using IDF, such as BM25, down-weights frequently occurring keywords as their document frequency increases, hence favoring the extraction of fresh keywords. However, there is no way to tune the weighting function to return more relevant or fresher keywords. BM25H overcomes this limitation by employing $tDF_x(t)$ and setting parameter α to an appropriate value, as described earlier.

To implement BM25H on a keyword extraction system there is no need of storing the whole corpus, just the length of the last N web pages and the *tDF* for the terms occurred in browsing history with *tDF>0*. The procedure may be described as follows. When a new web page reaches the client, the HTML content is preprocessed by removing stopwords and applying the Porter's stemming algorithm. For each term found in the content, a TF value is calculated. Next, using the information stored in the aforementioned structures a BM25H score is calculated for each term. Based on this scores the top K keywords can be selected. For each term in the HTML content with no *tDF* record, a new one is created setting *tDF* equal to one. Finally, for all terms stored in *tDF* structure, the new *tDF* value is updated applying the decay function from Equation 1.

4. EXPERIMENTS

To evaluate the proposed method, we have developed two datasets, one with articles from Bloomberg[2] and a second one with articles from the English version of Wikipedia[3], from which we extract and weight keywords, that have been assessed for their relevance to the articles topic by assessors. In the remainder of this section, we describe the two datasets (Section 4.1), the parameter setting of the methods we compare (Section 4.2) and the assessment procedure (Section 4.3).

4.1 Datasets

The evaluation of the proposed method is based on two synthetic datasets, generated by simulating a user who browses web pages from different topics, in order to better control the parameters of the evaluation and examine the dependence of our method to different browsing patterns. In particular, we control how the topics of the browsed web pages change to assess the freshness of selected keywords.

In Bloomberg dataset we have collected articles from the news website bloomberg.com. We selected Bloomberg articles for two reasons. First, they allows us to simulate the interchange of topics in a user's browsing history, by considering the categories to which the articles belong. In addition, it facilitates the evaluation process as human assessors may spend less time understanding the main topic of a recent news article. As most news websites, Bloomberg categorizes its articles by wide topics such as *economics*, *science* and *sports*. We have created the dataset using articles from the categories of *Business* and *Arts & Culture*. The dataset comprises a sequence of 100 news articles that were randomly selected. We use the Bloomberg dataset in order to tune the parameter of BM25H, α and N, and also to examine the dependence of the method to different browsing patterns. In order to do the later, we have created four versions of the Bloomberg dataset. In each version there are groups of different session lengths of articles that interchange. We choose groups of five, ten, twenty and forty news articles.

In Wikipedia dataset all web pages are from its English version. To create the dataset, we have selected articles from two categories, namely *Information Technology* and *Music*. The dataset comprises a sequence of 150 Wikipedia articles that correspond to three groups of articles related to information technology and three groups of articles related to music. Each group comprises 25 Wikipedia articles. In this way, we are able to observe the results of the keyword extraction process during browsing web pages on a certain topic, as well as when there is a transition from one topic to another.

4.2 Methods

In order to compare our baseline methods (plain text TF and BM25) with BM25H, which employs temporal document frequency, we applied all three methods on the sequences of articles of the datasets. For each method we take the top twenty ranked keywords as recommended keywords from the corresponding method. Keywords recommended from at least one of the three methods were collected in a set of keywords in alphabetical order for each article.

We use a sliding window with length $N = 100$ pages. This length was selected based on the collected statistics recently published by Kumar *et al.* [10], who report that 90% of the users do not visit more that 113 pages per session. We selected a window length of 100 web pages in order to approximate actual web behavior. The BM25 parameters used are $k_1 = 3$ and $b = 0.75$. The value of the reduction parameter α, employed in temporal document frequency (Equation 1) and used in BM25H, is 1.02. We select this value empirically by checking the number of reductions needed to eliminate the term from the corpus. A part of this process is shown in

[2] http://www.bloomberg.com

[3] http://en.wikipedia.org/wiki

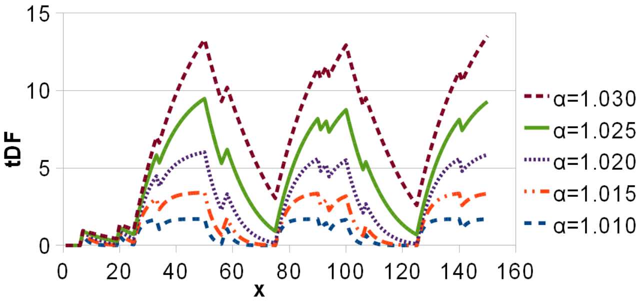

Figure 3: tDF evolution through time for term `music` and different values of parameter α, in Wikipedia dataset.

Figure 3 which illustrates the evolution of tDF('music') for five different values of α. In agreement with the evaluation datasets used, we chose the value of α so that the $tDF(t)$ of term t will be significantly reduced after approximately 25 pages where term t does not occur.

In addition to comparison with the baselines, we also examine the impact of different browsing patterns to the method performance. To do that we use the four versions of Bloomberg dataset as described in Section 4.1. Moreover, we discuss the potential of parameter tuning using Bloomberg dataset. We apply different values of α and N to select the optimal values for a certain browsing pattern.

4.3 Evaluation Process

We evaluate our method by measuring its performance in terms of relevance and freshness.

To measure relevance of proposed keywords, we have developed an online assessment system and we quantified relevance using the precision at top k recommended keywords, for different values of k, and mean average precision at the top twenty keywords for each web page. For the Wikipedia dataset we constructed eight sets of URLs, by sampling from the sequence, such as we have the same representation of information technology and music articles, and a normal distribution of samples in the sequence. Each set contains six URLs. For the Bloomberg dataset we have ten sets of URLs, with five URLs each. Assessors were invited to assess one set of URLs at a time. The assessors could label a keyword as *relevant* when it was related to the topic of the article, and *irrelevant* when it did not relate to the topic of the article.

The evaluation procedure resulted in a total of 48 assessed URLs from the Wikipedia article and 50 URLs from the Bloomberg dataset. For the assessed URLs, 2142 and 2226 unique keywords, from Wikipedia and Bloomberg respectively, were extracted by the three scoring algorithms presented earlier. Assessors marked 4368 unique terms with an average of 44.6 unique keywords per article. From these keywords 2375 were marked as relevant to the corresponding article, 1875 as irrelevant and 272 as incomprehensible. The maximum number of relevant keywords per URL were 28, of irrelevant 32 and of incomprehensible 8.

We define freshness as the number of the top k keywords that did not appear in the top k keywords in the past M web pages divided by k ($F_M@k$). We compute freshness for different values of k and M. The measure of freshness we have selected is consistent to its definition (Section 2). Small values of freshness mean that we have high overlap with keywords recommended in the near past. In addition, low freshness at the top keywords increases the possibility of recommendations repetition as keywords of higher weights may affect with higher possibility the proposed recommendations.

To measure the performance of the proposed system we use the harmonic mean H of the measures described above, precision and freshness.

$$H_k = \frac{2 \times P@k \times F_M@k}{P@k + F_M@k} \quad (4)$$

A high value of H_k shows a good performance in both relevance and freshness.

5. RESULTS

We propose BM25H as a scoring function to extract keywords from web pages. The evaluation of our function includes the measurement of the relevance of extracted keywords and their freshness in comparison to the recently visited web pages.

5.1 Relevance

Figure 4 (a) shows $P@k$ keywords for the Wikipedia dataset. The four diagrams correspond to the precision at 5, 10, 15 and 20 extracted keywords. From the diagrams, we can see that the different scoring functions achieve similar levels of precision, when a small number of keywords is extracted. As the number of keywords increases, the precision of BM25H is higher than the precision of TF and BM25. For P@5, the real values per scoring function are 3.98/5 for TF, 3.71/5 for BM25 and 3.9/5 for BM25H. The differences at top 5 keywords are small, but, as we will discuss later, the freshness is not the same. The differences between the three scoring functions are quite important for precision at 10, 15, and 20. Indeed, P@20 is 0.587 for simple TF, 0.652 for BM25 and 0.72 for BM25H. BM25H remains at the top of the four scoring functions on the relevance. Not only it outperforms the TF-based functions, but also the BM25 which includes the temporal dimension. It is interesting to note that BM25 had presented a high rate of incomprehensible keywords compared to BM25H. Over 10% was the percentage of term marked as `'cannot tell'` for BM25, in contrast to 7.5% for BM25H.

We can see similar findings in Figure 4 (b), which contains the results for the Bloomberg dataset. BM25H dominates the two other methods also in precision at five.

Except from precision we also measure the mean average precision at top twenty keywords ($MAP@20$) of TF, $BM25$ and $BM25H$ for both the datasets. MAP allows us to observe not only the dominance of BM25H in extracting relevant keywords, but also in ranking the relevant keywords better than the other methods. The exact values of MAP are 0.79 for TF, 0.72 for BM25 and 0.86 for BM25H in bloomberg dataset and 0.75 for TF, 0.77 for BM25 and 0.8 for BM25H in Wikipedia dataset.

5.2 Freshness

To count freshness we use $F_M@k$ as described in section 4.3. We have measured freshness for $k = 5, 10, 15, 20$, and $M = 5, 10, 15$. Tables 1 and 2 show the value of freshness for all combinations of the aforementioned parameters for the Wikipedia and Bloomberg datasets, respectively. The freshness in TF scoring function, which is lower than 0.7 in most cases, strongly indicates the significant problem of keyword repetition when browsing through relevant pages. Keywords

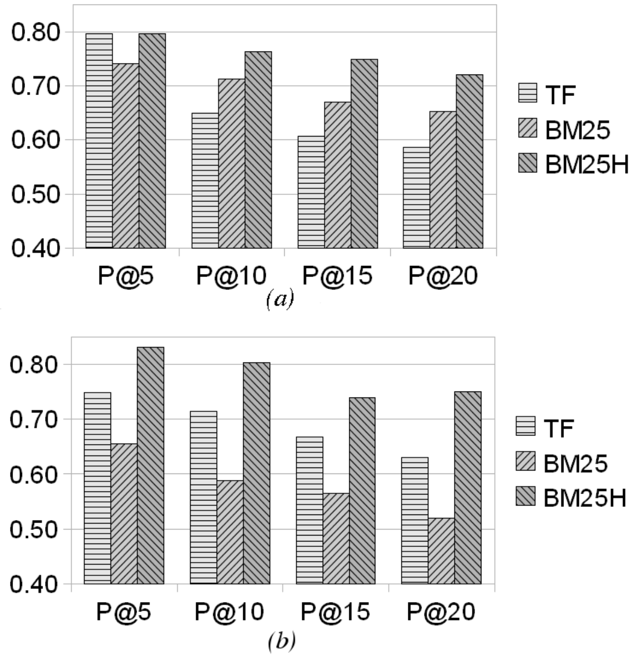

Figure 4: Precision at 5, 10, 15 and 20 keywords for the scoring functions TF, BM25 and BM25H in (a) the Wikipedia (b) the Bloomberg dataset

that are more likely to reoccur are, as the experiments denote, domain specific and topic specific keywords. Some of the domain specific keyword for the Wikipedia dataset are the terms `edit`, `free`, `retrieve` and `isbn`, and for the Bloomberg dataset `story` and `bloomberg`. In our method, the domain specific keywords do not reoccur with scoring function BM25H as their tDF continue to increase through time, while the user browses through articles. On the contrary, topic specific keywords reoccur as topics interchange. For example, in the Wikipedia dataset, term `music` appears for the first time in extracted keywords of all scoring functions on the 26^{th} web page, the first of the new topic. On the 75^{th} web page where the topic changes again, the keyword `music` appears again in keywords extracted from BM25H, but not in BM25, as the real DF of term `music` is still high, because of its appearance in the last 100 web pages.

In Wikipedia dataset, BM25 has the highest freshness for all measurements as expected. This is because BM25 keeps in DFs all occurrences of terms in the window. As a result, it is unlikely to rank a term high when it has been seen in the past N documents. In Bloomberg dataset the freshness is not that high most probably due to the small size of articles that limits the choices for fresh keywords. BM25H performs very well with respect to freshness.

The high values of freshness compared to TF combined with the higher relevance, confirms our claims that BM25H extracts fresh and relevant keywords. Its small lack in freshness compared to BM25 denotes the successful recurrence of keywords that occurred in distance in the past and are relevant to the current web page. The harmonic mean between precision and freshness strengthens this statement. Figure 5 shows the harmonic mean for TF, BM25 and BM25H for the two datasets. BH25H achieves a higher H_{20} in both cases.

Considering the precision and freshness results together, we can draw one important conclusion. As mentioned earlier, the precision at top 5 keywords is almost the same for all scoring functions. But these keywords at top ranks are not the same ones for all functions. This can be confirmed by the overlap rate of BM25H which is much higher for the TF measures. For example, for the article on *computer insecurity* in Wikipedia, simple TF ranks as top 5, the keywords `{securiti, computer, system, attack, edit}` with relevance 4/5, and BM25H `{insecuriti, attacker, backdoor, worm, eavesdrop}` with 5/5 relevance.

Table 1: Keywords Freshness in Wikipedia Dataset

Pages	Terms	**TF**	**BM25**	**BM25H**
Past 15	Top 5	0.73	1.00	0.97
	Top 10	0.65	0.99	0.95
	Top 15	0.61	0.99	0.92
	Top 20	0.59	0.99	0.91
Past 10	Top 5	0.77	1.00	0.98
	Top 10	0.69	1.00	0.96
	Top 15	0.74	0.99	0.96
	Top 20	0.63	0.99	0.93
Past 5	Top 5	16.00	1.00	0.99
	Top 10	0.76	1.00	0.98
	Top 15	0.73	1.00	0.97
	Top 20	0.71	1.00	0.96

Table 2: Keywords Freshness in Bloomberg Dataset

Pages	Terms	**TF**	**BM25**	**BM25H**
Past 15	Top 5	0.74	0.96	0.94
	Top 10	0.65	0.92	0.91
	Top 15	0.64	0.90	0.88
	Top 20	0.63	0.90	0.87
Past 10	Top 5	0.76	0.96	0.94
	Top 10	0.67	0.92	0.92
	Top 15	0.66	0.90	0.89
	Top 20	0.64	0.90	0.88
Past 5	Top 5	0.81	0.96	0.94
	Top 10	0.71	0.94	0.94
	Top 15	0.71	0.92	0.92
	Top 20	0.71	0.92	0.91

Table 3: Precision, MAP, and Freshness of Keywords in Bloomberg Dataset for different session lengths (T)

T	**5**	**10**	**20**	**40**
P@20	0.73	0.74	0.75	0.74
MAP	0.87	0.86	0.86	0.85
F_{15}@20	0.86	0.88	0.88	0.86

6. PARAMETERS TUNING

The two parameters introduced in BM25H function through tDF (Equation 1) need to be examined for their impact to BM25H performance. In order to measure the performance of BM25H for different parameter values we use a version of Bloomberg dataset with a topic length of twenty web pages.

As mentioned in section 3.2, α is the reduction parameter and N is the window length. The N parameter defines how long, at most, it takes to eliminate the temporal document frequency of a term when it does not show up anymore. The parameter α control the reduction rate in the defined window. A very small α value will lead to fast elimination, before the end of the window. Thus different combinations of parameters α and N can lead to very similar results.

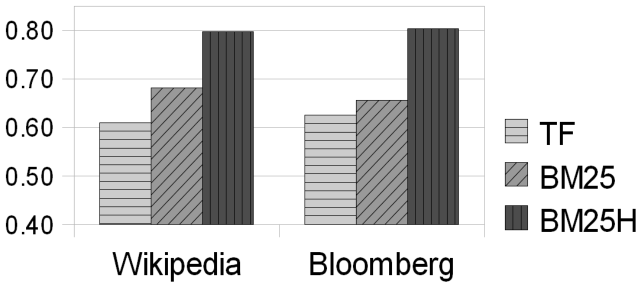

Figure 5: Harmonic mean between precision and freshness for the scoring functions TF, BM25 and BM25H in the Wikipedia and Bloomberg datasets.

Given the aforementioned facts, the parameter tuning process was the following. Using a high value of parameter $N = 100$ we measure the performance of BM25H in terms of both relevance and freshness. We try different values for α and choose the best reduction rate. Then, we search for substitute values of N and α that correspond to the same rate and examine the effect in BM25H performance.

In figure 6 one can see the evolution of precision at $20(P@20)$, freshness of top 20 keywords in past 15 web pages ($F_{15}@20$), and the harmonic mean of these two measures for different values of α, between 1.01 and 1.1, for $N = 100$. We need to select our parameter values in a way we assure that we have good performance in both relevance and freshness. Based on this diagram we can conclude to a value of 1.02 for parameter α as the optimum value, because with this value we achieve the highest H_{20}.

For the optimum tuning of $\alpha = 1.02$ for $N = 100$, we want to examine BM25H performance for different N values, trying to keep a similar reduction rate. We found that the reduction rate curves match for $\alpha = 1.02$ with $N = 100$, $\alpha = 1.025$ with $N = 80$, $\alpha = 1.35$ with $N = 60$,$\alpha = 1.053$ with $N = 40$ and $\alpha = 1.13$ with $N = 20$. We performed these experiments to examine the impact of choosing different N values and observed that the performance for all the aforementioned combinations is the same.

The overall conclusion of the parameter tuning process is that in the given setting we have better results when the reduction curve tends to zero for δ larger than 20. We must emphasize that this is independent of the topic length T, as shown in the results section 5.

7. RELATED WORK

In this section, we provide an overview of related works for keyword extraction and temporal weighting of keywords. One of the most known weighting schemes is TF-IDF and its variations [17], where the weight of a term depends on its within-document frequency and its document frequency in a corpus. BM25 [16] is a stable and well-performing weighting scheme, which employs the length of documents to normalize term and document frequencies. BM25F [21] extends BM25 to the case of documents with fields or zones. We extend BM25 with a temporal IDF computed from the inter-arrival distances of terms in the browsing history of a user. Our method can also be applied in the case of BM25F.

In addition to weighting keywords based on frequencies of terms in documents and collections of documents, keyword extraction has been investigated using machine learning [20][7] and graph-based approaches [15]. Our proposed method can be applied in both machine learning and graph-based settings as a feature for terms.

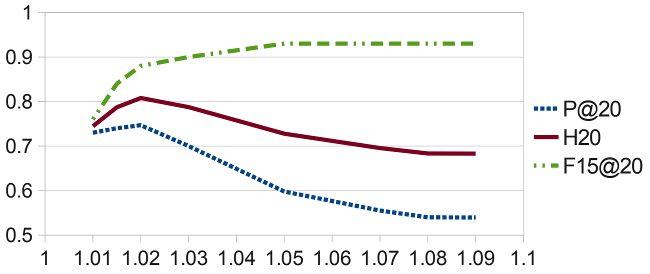

Figure 6: precision at $20(P@20)$, mean average precision(MAP) and freshness of top 20 keywords in past 15 web pages ($F_{15}@20$), for different values of α, between 1.01 and 1.1, for $N = 100$.

Keyword extraction has also been applied to contextual advertising, where ads are shown when a user browses a specific web page. Ribeiro-Neto *et al.* [15] added keywords to a web page from other similar web pages using a Bayesian Inference Network. Yih *et al.* [20] train a system to learn to extract keywords for contextual advertising using features such as TF-IDF, web page's metadata, and search engine query logs. Anagnostopoulos *et al.* [2] addressed the issue of high latency and computational cost for placing contextual advertisements in dynamic pages in real-time by using summaries of web pages. Differently from our method, the described techniques do not consider the browsing history of the user who is going to view the placed ads.

Keyword extraction and weighting algorithms may also employ evidence from a user's browsing history to enhance the relevance of the extracted keywords. Matsuo [13] treated the user's browsing history as a set of keywords and weights keywords in new documents according to their relevance to the user's browsing history. Kondo *et al.* [9] proposed to extract keywords from the whole browsing history of a user, by first extracting keywords contained in Wikipedia article titles, and then applying HITS algorithm to compute the authority of these articles. Our approach is different compared to the works by Matsuo and Kondo *et al.* in that we consider the inter-arrival distances of terms in the user's browsing history.

The term weighting schemes described above assume that collections of documents are static and do not change over time. As a consequence, term weights are independent of any temporal patterns in their occurrences in a document collection as a whole, or in the documents read by an individual user. This is a limiting assumption, if we consider that the contents of web pages change over time [6][14], but also that the topics of interest for users may change over time.

Lappas *et al.* [11] modeled the burstiness of terms in collections of time-stamped sequences of documents. Kleinberg [8] modeled the burstiness of topics in streams of data using infinite-state automata. Vlachos *et al.* [19] detected bursts and periodicities in query activity from search query logs using spectral analysis and Fourier coefficients. In this work, we do not directly model the burstiness of term occurrences in a stream of documents, but we use the pattern of term occurrences to improve the term weighting.

Recently, there has been a body of work on estimating the importance of terms in the presence of several snapshots of a document, such as web page crawled at different timestamps, or a Wikipedia page that is being edited. Jawowt *et al.* [7] proposed to compute the importance of terms as the sum of term frequencies in each snapshot of a web page multiplied by the time period between two consecutive snapshots of the document containing the term. Aji *et al.* [1] employed the history of revisions from Wikipedia pages to compute improved term frequency counts and incorporate them in BM25 and Language Models. Elsas and Dumais [5] examined the stability of document scores as dynamic documents get updated over time. They proposed to use different classes of terms according to the length of time during which a term occurs on a particular web page and a mixture language model with term counts for each of the different classes.

Liebscher [12] introduced temporal term weighting, where a document collection evolving over a period of time is partitioned in time slices and the IDF is computed for each partition separately. Efron [4] suggested that informative terms appear in patterns, which are hard to predict, and developed a weighting scheme, where the importance of a term is inversely proportional to how well its occurrences can be predicted from its past occurrences. Uehara *et al.* [18] introduced a modified TF-IDF weighting where the term frequency factor depends on both the occurrences of a term in a document as well as the frequency of updates in the document. The above models relate to our work in the sense that they propose time-aware versions of IDF weights, but they do not consider the problem of weighting terms based on the browsing history of a particular user.

8. CONCLUSIONS

Keyword extraction from web pages is a key process task for several text mining tasks, such as contextual advertising, user profiling and personalization. A limitation of existing approaches to keyword extraction, however, is that they select keywords for each web page independently of the sequence of browsed web pages by a user. Consequently, while the user browses through web pages on the same topic, extracted keywords may overlap through time causing repetition of information displayed to the user. We introduced a novel variation of BM25 scoring function, BM25H, which weights and ranks the terms of an HTML document, based on their importance in the document and their freshness in recent browsing history of a user. To achieve this effect, we introduce a temporal document frequency measure, *tDF*, which decreases while the corresponding term does not occur in the web pages visited by the user.

Experimental evaluation, realized by human evaluators, shows that BM25H achieves over 0.70 P@20 and over 0.80 MAP at 20 keywords for both datasets, outperforming the baselines presented in the paper. Overall, our experimental results show that BM25H returns a high number of relevant keywords and, at the same time, reduces the repetition in the returned keywords compared to approaches based only on within-document term frequencies. Hence, the introduced temporal document frequency better captures the significance of terms in corpora of documents based on a user's browsing history.

9. ACKNOWLEDGMENTS

The research of M. Karkali has been co-financed by the EU (ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the NSRF - Heracleitus II. The research of V. Plachouras and M. Vazirgiannis was partially financed by the DIGITEO grant LEVETONE in France.

10. REFERENCES

[1] A. Aji, Y. Wang, E. Agichtein, and E. Gabrilovich. Using the past to score the present: extending term weighting models through revision history analysis. CIKM '10, pages 629–638, 2010.

[2] A. Anagnostopoulos, A. Z. Broder, E. Gabrilovich, V. Josifovski, and L. Riedel. Just-in-time contextual advertising. CIKM '07, pages 331–340. ACM, 2007.

[3] A. Broder, M. Fontoura, V. Josifovski, and L. Riedel. A semantic approach to contextual advertising. SIGIR '07, pages 559–566, 2007.

[4] M. Efron. Linear time series models for term weighting in information retrieval. *J. Am. Soc. Inf. Sci. Technol.*, 61:1299–1312, 2010.

[5] J. L. Elsas and S. T. Dumais. Leveraging temporal dynamics of document content in relevance ranking. WSDM '10, pages 1–10, 2010.

[6] D. Fetterly, M. Manasse, M. Najork, and J. Wiener. A large-scale study of the evolution of web pages. WWW '03, pages 669–678, 2003.

[7] A. Jatowt, Y. Kawai, and K. Tanaka. Visualizing historical content of web pages. WWW '08, pages 1221–1222, 2008.

[8] J. Kleinberg. Bursty and hierarchical structure in streams. KDD '02, pages 91–101, 2002.

[9] M. Kondo, A. Tanaka, and T. Uchiyama. Search your interests everywhere!: wikipedia-based keyphrase extraction from web browsing history. HT '10, pages 295–296, 2010.

[10] R. Kumar and A. Tomkins. A characterization of online browsing behavior. WWW '10, pages 561–570, 2010.

[11] T. Lappas, B. Arai, M. Platakis, D. Kotsakos, and D. Gunopulos. On burstiness-aware search for document sequences. KDD '09, pages 477–486, 2009.

[12] R. A. Liebscher. Temporal context: applications and implications for computational linguistics. In *ACL 2004 Workshop on Student research*, ACLstudent '04, 2004.

[13] Y. Matsuo. Word weighting based on user's browsing history. In P. Brusilovsky, A. Corbett, and F. de Rosis, editors, *User Modeling 2003, LNCS 2702*, pages 145–145. 2003.

[14] A. Ntoulas, J. Cho, and C. Olston. What's new on the web?: the evolution of the web from a search engine perspective. WWW '04, pages 1–12, 2004.

[15] B. Ribeiro-Neto, M. Cristo, P. B. Golgher, and E. Silva de Moura. Impedance coupling in content-targeted advertising. SIGIR '05, pages 496–503, 2005.

[16] S. E. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. SIGIR '94, pages 232–241, 1994.

[17] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.*, 24:513–523, 1988.

[18] M. Uehara, N. Sato, and Y. Sakai. Adaptive calculation of scores for fresh information retrieval. ICPADS '05, pages 750–755, 2005.

[19] M. Vlachos, C. Meek, Z. Vagena, and D. Gunopulos. Identifying similarities, periodicities and bursts for online search queries. In *SIGMOD*, pages 131–142, 2004.

[20] W.-t. Yih, J. Goodman, and V. R. Carvalho. Finding advertising keywords on web pages. WWW '06, pages 213–222, 2006.

[21] H. Zaragoza, N. Craswell, M. J. Taylor, S. Saria, and S. E. Robertson. Microsoft cambridge at trec 13: Web and hard tracks. In *TREC*, volume Special Publication 500-261. NIST, 2004.