PostRank: A New Algorithm for Incremental Finding of Persian Blog Representative Words

Mohsen Sayyadiharikandeh
Sharif University of Technology
Web Intelligence Laboratory
Azadi Avenue,Tehran
0098-2166166698
M_sayyadi@ce.sharif.edu

Mohammad Ghodsi
Sharif University of Technology
and Institute for Research
in Fundamental Sciences (IPM)
Theory Group Laboratory
Azadi Avenue, Tehran
0098-2166166625
Ghodsi@sharif.edu

Mohammad Naghibi
Sharif University of Technology
Department of Computer Science
Azadi Avenue, Tehran
0098-2166166628
M_nagihbi@ce.sharif.edu

ABSTRACT

Dimension reduction techniques for text documents can be used for in the preprocessing phrase of blog mining, but these techniques can be more effective if they deal with the nature of the blogs properly. In this paper we propose a novel algorithm called PostRank using shallow approach to identify theme of the blog or blog representative words in order to reduce the dimensions of blogs. PostRank uses a graph-based syntactic representation of the weblog by taking into account some structural features of weblog. At the first step it models the blog as a complete graph and assumes the theme of the blog as a query applied to a search engine like Google and each post as a search result. It tries to rank the posts using Markov chain model like PageRank in Google. We used the ranking model under the assumption that top ranked nodes contain blog best representative words. Then it tries to identify post groups according to their scores. Finally this algorithm analyzes the first group using statistical methods(like TF-IDF) to identify blog representative words. Other groups are candidates of having blog theme after occurring change of theme to the blog. By arriving new instances of posts we try to update the blog graph by setting the initial scores of old nodes in the Markov chain to their final score from last run and continue the PostRank iterations until reaching convergence point. If half of the representative words have changed we would say that theme of the weblog has been changed.

We evaluated our method on the Persianblog dataset and obtained promising results. The blogs have been assigned to ten representative words by human beings and the results of PostRank have been compared to them and results of old related algorithms in this area.

Keywords

Dimension Reduction-Incremental-Markov Chain

1. INTRODUCTION

Blog space on the Web, like other Web pages, can be considered as a huge source of information to be used as text resources in data mining Issues. Number of blogs has been raised tremendously in recent years and still keeps growing. So we can't skip such great amount of information. On ther other hand We can't see the blogs as just a Web page and apply the web pages tools on the blogs because blog has its own characteristics which should be taken into account. In order to extract information from such a big data collection automatic analysis and discovery process which have been customized for the blogs is needed. Considering the mining of the blogs as an automatic tool to discover interesting knowledge from blogs, an appropriate preprocessing can increase the quality of gained knowledge as well as decrease the complexity of the mining process.

There are several dimension reduction techniques for text documents which can be used to reduce the vector size of a text document in vector space model [1][2]. These techniques model a document in a low-dimensional space using statistical and analytical techniques. The drawback of these techniques is that they don't deal with the structure of the text; relation of a word to main topic of document; and the various themes in a document which is a usual phenomenon in Web pages and blogs especially. We believe that considering mentioned factors can help us to lead more accurate results. On the other hand blog owners usually add new post in their blogs over and over and learning the blog model using batch approach (traversing all the posts from oldest to newest) involves too much time which sometimes does not worth to try. So updating the blog model upon arriving new instances is desired.

In this paper we propose a shallow method for identifying blog representative words capable of incremental learning. This method could be used as a preprocessing step for blog mining. When developing a blog-specific dimension reduction method, one has to deal appropriately with specific characteristics of the blogs including posts time-stamps, "about" note, comments, posts self-similarity and link structure. In our method we take benefits of first four items.

Three below main reasons motivated us for developing such method:

1- Dimension reduction techniques that are applied to blogs should be developed according to special characteristics and dynamics of blogs. The related work to the same problem although takes into account some of blog features but skips some important ones like comments, "About" note and posts self-similarity.

2- The old algorithm in this context uses batch approach which involves analyzing the old posts and beginning from the first step.

WIMS'12, June 13-15, 2012 Craiova, Romania

3- Persian text processing generally suffers from low quality and this is because of the lack of a comprehensive solution for the stemming of verbs and nouns. Hence, vector space in Persian text mining is too large and consequently the process is very time consuming and leads to weak results. We believe that our method can enormously decrease the size of the vector space and effect the Persian text mining process positively.

The rest of this paper is organized as follows. Section 2 addresses some related works. In section 3 we go through the details of proposed algorithm assumption and explain steps of our algorithm in detail. In Section 4 we analyze the incremental accuracy of proposed Algorithm. Section 5 includes details of our experimental evaluation which briefly describes how data collection was prepared, how experiments were leaded, and what results were achieved. We conclude the paper and present future works in section 6.

2. Related Works

Till now several different methods have been proposed for weblog summarization using shallow approach. Maximum Marginal Relevance Multi Document (MMR-MD)[3] and MEAD[4] are two methods that are related to our works. The authors analyze clusters of a document to summarize it. MMR-MD summarization is a purely extractive summarization method that is based on Maximal Marginal Relevance concept proposed for information retrieval. It can accommodate a number of criteria for sentence selection such as content words, chronological order, query/topic similarity, anti-redundancy and pronoun penalty. MEAD is a sentence level extractive summarizer that takes document clusters as input. Documents are represented using term frequency-inverse document frequency(TF-IDF) of scores of words. IDF value is computed based on the entire corpus. The summarizer takes already clustered documents as input. Each cluster is considered a theme. The theme is represented by words with top ranking TF-IDF scores in that cluster. Sentence selection is based on similarity of the sentences to the theme of the cluster.

Also much work has been done on Web page summarization recently. A great system in this domain is OCELOT[5] which is a prototype system for automatically summarizing Web pages. It uses probabilistic models to generate the "gist" of a Web page. The models used are automatically obtained from a collection of human-summarized Web pages. Sun et al used LSA and Luhn's sentence selection methods to summarize a Web page using clickthrough data[6]. They suppose that clickthrough data provides some human understanding about Web pages. Also some research has been done to enhance the level of accuracy of Web page classification based on Web summarization[7].

Also blogs have received much attention from some researchers in recent years. There already exist several techniques for spam blog post detection, blog posts tagging and opinion mining. Hu and Liu proposed an opinion summarization of products, categorized by the opinion polarity[8]. They then illustrated an opinion summarization of bar graph style, categorized by product features [9].

But to the best of our knowledge, very few studies on blog post summarization have been reported. Zhou et al viewed a blog post as a summary of online news articles it linked to, with added personal opinions [10]. A summary is generated by deleting sentences from the blog post that are not relevant to its linked news articles. Comments associated with blog posts were however not used. There are also some existing researches based on comments. In [11] a new solution is proposed which first derives representative words from comments and then selects sentences which include those words. They believe that reading comments have serious affects on one's understanding about a blog post. The authors of [12] see the blog as perfect graph and try to analyze posts self-similarity in order to recognize blog temporal dynamics.

In [13] we proposed an algorithm which pays attention to some characteristics of weblogs like body-title composition, Post Time-stamp. The main drawback of this algorithm is that it uses the batch approach rather than incremental approach. In this paper we propose a novel algorithm which can learn incrementally and pays attention to several blog characteristics.

3. Proposed Algorithm

Based on the related works in this area we decided to propose a novel algorithm which assigns individual score to each post by considering the different characteristics of weblogs. The mentioned scores denote how much one post is related to the theme of the weblog. We believe that not all the posts are related enough to be analyzed for identifying the theme of the weblog. Consider the case in which weblog owner writes some posts about some daily events and not related to the weblog theme. After scoring all the posts we try to rank and partition the posts according to the scores. In other words we see the problem of finding representative words of a weblog as a problem of ranking at the first stage. It is similar to the problem of ranking search results in search engine as we see the theme of the weblog as query applied to a search engine and posts as search results. Also we can assume that the blog owner is like a search engine who retrieved the related documents. As the search engine rank the search results after retrieving the related documents we also try to rank the posts here to identify the most relevant posts. Note that we see each post a search result because we can assume that each post is related to the theme because the blog owner once wrote it.

According to what we discussed earlier We can exploit the idea of proposed algorithms in the area of ranking search results here in our problem. The search engine Google uses the algorithm PageRank after retrieving the related documents and websites. PageRank tries to identify the authority of each website by assigning its score. It identifies the score by one contstant term and one dynamic term. PageRank defines a Markov chain and tries to identify the score by multiplying Markov chain matrix until reaching convergence point. We can use the same idea here in the ranking of posts. We define the Markov chain by considering all posts and "about " note. Each post has at least a constant value score named d. Also we consider a dynamic term which denotes the authority of one post among other posts. The most similarity one post has with other pots the most score it gets. We update the score of the post in each iteration until the convergence point. Top post is the one which has high similarity to the other high score posts like in PageRank where the top node is the one who has more links from other high score nodes. The difference is that here we have text similarity instead of link.

The proposed algorithm is called PostRank which includes four general steps. The overall PostRank algorithm is portrayed in figure 1. We describe each step in details in further sections.

3.1 Extraction of Baseline Bag of Words

First in PostRank we consider the whole weblog as a simple text document without any structure and try to identify the most

important words by using statistical methods like TF-IDF. We call the output of this step Baseline. Although the problem we are solving here is related to Persian weblogs but since we use the statistical methods we can claim that it is also suitable for other languages. The idea behind this step is that we use the importance of some words in calculating the similarity between two posts. When two posts share a word located in baseline we consider a coefficient (more than one) for that word.

Figure 1. Four Steps of PostRank

The interesting fact about baseline is that if blog owner writes the posts fairly (by fairly we mean consider the theme in all posts) the baseline is a good candidate of weblog theme. But if we have some kinds of unrelated posts in the blog merging the posts does not lead in good representative words (especially in case of long unrelated posts with repeating some keywords).

3.2 Define Markov Chain and Correspondent Graph

In this step We define the Markov chain by considering each post as one state and the similarity between posts as transition probability. So weblog nodes are states (S1,…, S_N) of Markov chain. In order to make the Markov chain regular we eliminate some nodes which can't be accessed from the other nodes. This make the Markov chain a regular one which helps us to reach the convergence point according to the Markov Chain Fundamental Limit Theorem [14].

As discussed earlier in the related work section, authors of [12] try to model the weblog as a graph considering each post as one node. We use this model here by adding "about" note as one extra node to the graph. The resulted graph is a perfect graph with one edge between each pair of nodes. The graph of a blog with four nodes is shown in figure 2. The number next to each node denotes number of comments it has.

Figure 2 – Example of Blog Graph

Transition matrix M is defined as the following:

"i,j'th entry denotes the similarity between I'th post and j'th post."

3.3 Solve Ranking Problem

As mentioned earlier PostRank multiplies the transition matrix again and again until it converges like PageRank. So we have N steps to solve the ranking problem. The score of a node like V in step N+1 is calculated as below:

$$S_{t+1}(v) = S_t(v) + \sum_{v' \in V-\{v\}} (f(c,t) W_{vv'} S(v'))$$

f(c,t) in the formula determines an appropriate coefficient based on post time-stamp and number of comments it has.

After reaching convergence point we consider the converged output as the final score of weblog posts. The first node with the highest score is the best candidate for finding blog representative words.

3.4 Posts Grouping and Analyzing First Group

After identifying posts' scores the algorithm tries to assign each post to one group. By defining a threshold value as group radius, PostRank starts from the first post and puts the posts with score difference less than the threshold to the same group.

After grouping the posts each group is the target of having one theme. First group which has the biggest average score is the first candidate of including blog most important theme. So PostRank analyzes this group by merging posts and using TF-IDF in order to retrieve top M words as blog representative words. We call these representative words theme of the weblog.

4. Analysis of PostRank Incremental Performance

As we discussed earlier, the most important weakness of the similar algorithm [13] in finding the weblog's theme is that it does not support incremental approach. Thus we attempted to propose a method that improves the efficiency of the algorithm and produces the appropriate result by incremental learning and updating correspondent model.

Nowadays, as we mentioned above, extensive mass of information has caused many systems and algorithms to suffer from low speed. In fact, one should say that batch learning approach has arisen this problem. Therefore, algorithms with ability of incremental learning can bring us better performance. In this section, we evaluate our claim and inspect whether the proposed algorithm shows promising in incremental learning or not. We begin with a brief introduction.

PostRank algorithm is based on Regular Markov Chain, which according to Markov Chain Fundamental Limit Theorem, its iterations will converge eventually. According to convergence definition, scores of posts will be equal in N^{th} and $N+1^{th}$ iterations. Also, sum of scores of all posts should be one. In order to accomplish this, we divide all elements of the transition probability matrix by sum of elements of one of the rows. Thus, we have N variables and N equations and so we have a unique result.

In our problem, when we inspect a weblog with V posts (including "about" post), the transition probability matrix would

have V columns and V rows. We would have a system of V equations and V unknown variables that can be solved easily. After adding M new posts, we will have V+M equations and V+M unknown variables, and we will again have a unique result. So by adding new instances of posts, we will still be able to prove convergence of Markov Chain. However, we claim that the scores will converge after less iterations that would be needed normally by using incremental approach. This fact has been asserted by our experimental results. As an intuitive interpretation of this fact, we can say after adding new posts, the initial scores of Markov Chain elements would be the same as converged scores of previous step. If the new posts have similarity to the old ones, the scores of other nodes will be injected to them during first iterations. In other words, if a post has the potential to gain a high score and lie into the first group of posts, it gains the needed score during initial iterations. Vice versa, if it does not have the potential (due to weak similarity to high scored posts), it will gain a low score in each iteration to be accumulated with previous scores. Hence, it cannot find a path to the first group.

The remarkable point is that although we need to solve a system of much more equations and variables to prove convergence of the algorithm, we do less iterations when we simulate it by computer. We prove this important claim by experimental results in next section. Here we give an example in which our claim is illustrated, and later in experimental results section, we verify this claim by running our algorithm on a large number of blogs.

To verify the incremental nature of our algorithm, we first consider a weblog with some of its nodes, and compute the required number of iterations for convergence by forming the Markov Chain matrix. Then we consider all of the nodes and using results from the previous section, we obtain new number of iterations and figure the reduction in iterations. If the reduction is considerable, we conclude that the algorithm has an acceptable incremental operation. It is evident that to prove this claim by experimental results we have to run the algorithm on a large number of blogs. Also, regarding convergence of numbers, we look at them with 8 floating points, even though there is no necessity for this precision in the problem definition.

Here is an example run of our algorithm. Suppose below is the transition probability matrix of a hypothetical blog in unnormalized version, which at first have 4 nodes. By executing the algorithm and multiplying the matrix by itself, we discover that it converges after 18 iterations.

$$\begin{bmatrix} 1/4 & 23/40 & 34/50 & 3/20 \\ 1/14 & 20/45 & 1/11 & 31/40 \\ 3/24 & 23/45 & 3/10 & 31/50 \\ 7/24 & 1/15 & 6/23 & 1/4 \end{bmatrix}$$

Now, we add another node into the Markov Chain graph. Here is the 5 by 5 matrix in batch mode.

$$\begin{bmatrix} 1/4 & 23/40 & 34/50 & 3/20 & 0.1 \\ 1/14 & 20/45 & 4/44 & 31/40 & 0.3 \\ 3/24 & 23/45 & 6/20 & 31/50 & 0.1 \\ 7/24 & 3/45 & 6/23 & 3/12 & 0.2 \\ 0.1 & 0.3 & 0.1 & 0.2 & 0.2 \end{bmatrix}$$

By running Markov Chain iterations in this batch mode matrix we find out this matrix will converge after 17 iterations. Now we form the 5 by 5 matrix in incremental mode. To do so, instead of calculating all of the elements from beginning, we use the converged 4x4 matrix.

$$\begin{bmatrix} 0.23944864 & 0.23944865 & 0.23944864 & 0.23944865 & 0.1 \\ 0.2968103 & 0.29681027 & 0.29681027 & 0.29681027 & 0.3 \\ 0.27734095 & 0.27734092 & 0.27734092 & 0.27734092 & 0.1 \\ 0.24767202 & 0.24767202 & 0.24767202 & 0.24767202 & 0.2 \\ 0.1 & 0.3 & 0.1 & 0.2 & 0.2 \end{bmatrix}$$

By executing the incremental algorithm, we see that the Markov Chain will converge after only 9 iterations, using the same size 5x5 matrix. Compared to converging in 17 iterations with batch mode execution, we conclude that incremental learning used in PostRank method can significantly reduces running costs, which witnesses the performance of our proposed algorithm.

This incremental approach used in PostRank, is the important factor that had been neglected in old 3-Step algorithm[13]. In that algorithm, there was always need to cluster all of posts and re-score them every time we add new posts. But PostRank overcomes this problem by taking previous results into account.

Another important point is that in incremental mode we can evaluate the concept of "theme change". In this paper we give a simple definition of change theme based on our definition of theme. We say theme of a blog is changed whenever at least half of representative words of that blog have been changed. To be able to detect change in theme, we should specify the origin time from when we want to compare the keywords. For instance, suppose we inspect a weblog until year 2007 and extract ten representative words from its posts. If we incrementally add its new posts until 2011 and from the set of new representative words, six of them have been changed, we can say the theme of weblog has been changed since 2007.

5. Experimental Results

In this paper we introduced a novel method to find representative words for weblogs which considers special properties of weblogs that ordinary web pages lack, like posts self-similarity, body-title correspondence, timestamp of each post and number of comments. To evaluate our method, we ran it on dataset of PersianBlog[1] and compared the results with both the previous similar method (which is discussed above) and manually selecting representative words. The following steps have been taken for evaluation. In the preprocessing step we tried to eliminate Persian stopwords introduced in [15] and used the stemming algorithm proposed in [16] for reducing the size of the document. After that first we tried to select those weblogs which are appropriate for evaluating our algorithm. In this context, those weblogs are appropriate that have an acceptable number of posts and a distinct theme. For instance, the well-known blog "Professors Against Plagiarism" fits our requirements, since it has a specific theme most of its posts are about plagiarism in academic society. We do not mean that a blog's title must be relative to blog's theme. Let us explain the meaning of an appropriate weblog in this context. If we ask someone to manually extract ten representative words from "Professors Against Plagiarism", the result would almost contain plagiarism, university, student, professor, etc. and that mean this blog has a distinct theme. On the other hand, suppose a weblog in which its owner writes about his or her daily thoughts

[1] Persianblog.ir, the first free blog service in Persian.

about diverse subjects. This blog does not have a specific theme, and hence it is not appropriate for our evaluation.

5.1 Finding Appropriate Weblogs

To select befitting blogs, a large number of blogs from the designated dataset have been read by two human beings and finally, 200 blogs were intended for evaluation. For each of these blogs, two readers manually extract ten keywords as a theme for that blog. There has been a big effort for these keywords to be as relative as possible to content of the blogs. In fact, these are the ideal set of representative words that our algorithm should search for. By providing these measures, we try to evaluate both the old and the new algorithms and find out which one is better.

5.2 Applying PostRank and Old Algorithm

After preparing the dataset and manual measures, we ran PostRank on the designated blogs. We also ran the old 3-Step method and compared the results using F-measure.

As stated previously, our algorithm can be categorized as a dimension reduction method and to verify its performance, we should compare our results with well-known dimension reduction methods like LSI. Experimental results in [13] testified that 3-State algorithm performs dimension reduction on Persian weblogs more efficiently than LSI does. That was because LSI does not care about unique characteristics of weblogs and treats them like unstructured texts. Since in [13] PersianBlog had been used as dataset, all we have to do is to compare performance of PostRank and 3-Step method. If we get better results, it can be deduced that PostRank will perform better than LSI.

We know that both PostRank and 3-Step algorithms have parameters that control their behavior. We evaluated these two methods with variant values for these parameters. It was a controversial issue to assign parameters for comparison; however, we tried to compare them in a way that the number of output groups of PostRank would approximately be equal to the number of clusters in 3-Step method. Threshold values for the old algorithm are the same values as were used in [13] experimental results. We also know that by decreasing the threshold value of PostRank, the number of groups increases.

In figure 3, we illustrate comparison of F-measure in both methods. The x dimension in the chart indicates algorithm parameters. The right-side number is value of PostRank's parameter, and the left-side number is the number of clusters in the old method. and dimension y shows resulted F-Measure value.

We can conclude from figure 3 that by increasing radius of groups, quality of PostRank output and F-measure decreases. The reason is that when we increase the radius, all of posts tend to lie into one big group. It is like we treat all the weblog as a unstructured text and it reduces the performance.

The other test we conducted to verify PostRank algorithm was evaluating convergence of the algorithm when adding new instances to verify incremental nature of it. Pursuing this goal, we

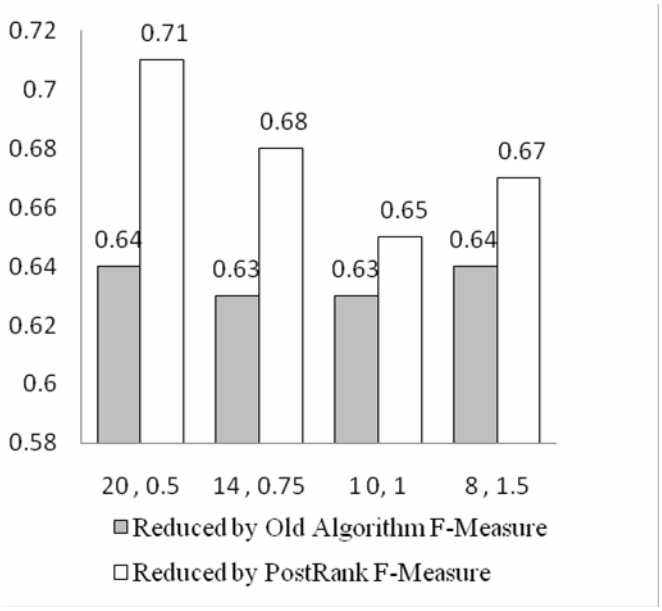

Figure 3- Comparing F-Measure of Two Algorithms

divided posts of each weblog into N sections, and ran our algorithm on some of those sections. Then we saved output of algorithm as well as number of iterations into a database. Then we added remaining posts in both batch and incremental mode, ran the algorithm and saved the results. For instance, if N=3, we first ran our algorithm on two third of posts. Then we added remaining one third of posts in incremental method and we fed the matrix with output of algorithm in the first run to calculate the output for all of posts. We then ran the batch-mode algorithm (in which matrix elements are calculated directly from post similarities) until it converged, and compared the results with incremental method.

In figure 4, you can review the output of incremental algorithm with different values of N. The x dimension indicates N as the number of groups and dimension y shows number of iterations until convergence. We executed the algorithm with N=2 and N=3. By comparing number of iterations required for these algorithms to converge, we discovered that using the incremental method, this number is reduced by 47 percent when N=2, and by 60 percent when N=3. It should be noticed that we calculated our number (with four floating points). For example, in "Professors Against Plagiarism" blog with 18 posts, the Markov Chain converges after 15 iterations. However, in incremental method (adding 6 posts to previous 12 posts), it will converge after only 5 iterations. Also, when adding 9 posts to previous 9 posts, N=2, the Markov Chain will converge after 8 iterations. The summary of our results is depicted in following lustration

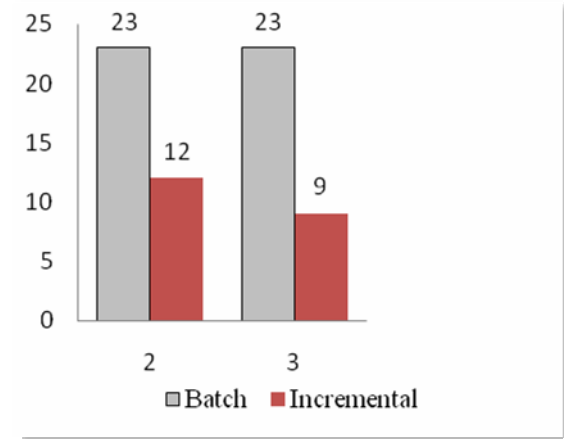

Figure 4 – Comparing Batch and Incremental Learning

6. Conclusion and Future works

In this paper we proposed a novel incremental algorithm for finding blog representative words in which markov chain is used for modeling of the blog. The main contribution of this algorithm is that in can work incrementally. In order to enhance the effectiveness of PostRank we should consider following items in our algorithm:

- Assuming some conditions for deleting some edges between the graph nodes in order to make the corresponding matrix sparse.
- Considering self-similarity of comments and similarity between posts and comments.
- Considering some special circumstances like deleted posts in incremental mode of our algorithm.
- Comparing our algorithm with famous keyword extraction algorithms by using English Datasets.

7. REFERENCES

[1] Tang B, Shepherd M, Milios E, Heywood M (2005) Comparing and combining dimension reduction techniques for efficient text clustering. Proceeding of SIAM International Workshop on Feature Selection for Data Mining:17-26.

[2] Molina LC, Belanche L, Nebot A (2002) Feature selection algorithms: a survey and experimental evaluation. Proceeding of ICDM'02:306-313.

[3] Carbonell, J., and Goldstein, J. : The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. In: SIGIR'98. Melbourne, Australia (1998)

[4] Radev, D., Allison, T., Blair-Goldensohn, S., Blitzer, J.C., Elebi, A., Dimitrov, S., Drabek, E., Hakim, A., Lam, W., Liu, D., Otterbacher, J., Qi, H., Saggion, H., Teufel, S., Topper, M., Winkel, A., and Zhang, Z.: MEAD - a Platform for Multidocument Multilingual Text Summarization. In: LREC. Lisbon, Portugal (2004)

[5] Berger, A.L., Mittal, V.O.: OCELOT: a System for Summarizing Web Pages. In: 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 144--151, Athens, Greece (2000)

[6] Sun, J.T., Shen, D., Zeng, H.J., Yang, Q., Lu, Y., Chen, Z.: Web-page Summarization Using clickthrough Data. In: SIGIR'05, pp. 194--201, Salvador, Brazil (2005)

[7] Shen, D., Chen, Z., Yang, Q., Zeng, H.J., Zhang, B., Lu, Y., Ma, W., Y.: Web-page Classification through Summarization. In: 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Sheffield, United Kingdom (2004)

[8] Minqing H, Bing L: Mining and summarizing customer reviews. Proceeding of SIGKDD'04:168-177((2004)).

[9] Ku, L.W., Liang, Y.T., Chen, H.H.: Opinion Extraction, Summarization and Tracking in News and Blog Corpora. In: AAAI-CAAW'06, Stanford, CA, USA (2006)

[10] Zhou, L., Hovy, E.: On the Summarization of Dynamically Introduced Information: Online Discussions and Blogs. In: AAAI-CAAW'06, Stanford, CA, USA (2006)

[11] Hu, M., Sun, A., Lim, E.P.: Comments-Oriented Blog Summarization by Sentence Extraction. In: CIKM '07, pp. 901--904, Lisbon, Portugal (2007)

[12] Lin, Y.R., Sundaram, H.: Blog antenna: summarization of personal blog temporal dynamics based on self-similarity factorization. Proceeding of International Conference on Multimedia and Expo (ICME'07):540-543 , Beijing, China(2007)

[13] Jafari-Asbagh, M. ,Sayyadiharikandeh, M. ,Abolhassani, H.: Blog Summarization for Mining Persian Blogs, SNPD (2009).

[14] Manning, C. D. ,Raghavan, P. , Schtze, H.: Introduction to Information Retrieval, i0521865719, 9780521865715, Cambridge University Press(2008)

[15] Sharifloo, A.A. and Shamsfard, M.: A bottom up approach to Persian stemming', IJCNLP, Hyderabad, India(2008)

[16] Taghva, K. , Beckley, R. , Sadeh, M.: A List of Farsi Stopwords. Technical Report, 2003-01, Information Science Research Institute, University of Nevada, Las Vegas (2003)