ISSN No. 0976-5697

Volume 2, No. 2, Mar-Apr 2011

International Journal of Advanced Research in Computer Science

CASE STUDY AND REPORT

Available Online at www.ijarcs.info

Efficient Blog Mining using Summarizing and Clustering Techniques

Emmanuel.Dinakar Babu R*
M.tech in information and technology[networking]
School of Information Technology and Engineering
VIT University
Vellore, India
emmanuel.dinakar@gmail.com

Saravavanakumar K
Assistant Professor
School of Information Technology and Engineering
VIT University
Vellore, India
ksaravanakumar@vit.ac.in

Abstract: In the Blogosphere the blogs have become pervasive media in these years, the number of bloggers has increased at an exponential rate. Manually monitoring and analyzing the blogs is a labour-intensive and time-consuming task. Intuitively, you could apply existing text and Web mining techniques to blog analysis and mining , so mining the blog content has necessitated intensive research in the area of automatic text summarization and clustering semantically similar blog content from different blogs. This survey intends to investigate some of the most relevant approaches in the areas of text summarization and clustering giving special emphasis to empirical methods and extractive techniques. Some approaches that concentrate on details of the summarization problem are also discussed.

Keywords: Information Retrieval, Blog mining, text summarization, text summarization, clustering

I. INTRODUCTION

Blogs are frequently updated Internet journal that has become a growing Internet subculture. Blogs have received much attention, because they provide users a place to post their thoughts easily and thus become a mass movement. Nowadays, Blog becomes the popular place where people can use to express their opinions. The popularity of blogs opened up a lot of new opportunities in social study. While in the past, researchers in social study had to spent huge efforts in collecting data. For Example, we can collect public opinions non-intrusively from the blogosphere by querying articles and comments on a specific topic during a specific time period. However, the large volume of returned documents requires analysis and summarization to help us understand public opinions.

In this paper, we are going for an unsupervised approach which selects topic-related words and searches the related topic in the blogosphere. This topic related searches proposed in [1] use the number of topic-related words in a paragraph as an indicator for the strength of containing a reason. Each reason is then judged by the system for its semantic orientation. In our architecture it undergoes different steps for extraction of fact and summarized information from blogs, the Blog parser which extracts the content of the blog. The Blog analyzer generates the semantic dependency graph between the sentences in the blog. The three main tasks in Blog analyzer are reason extraction, sentiment classification, and reason clustering. For product reviews, the features extracted are generally noun phrases [1]. However, a reason can be expressed in many different ways. Sometimes, a reason can span across sentences to explain. That is why we use paragraphs as the unit for recognizing reasons. Sentiment classification, or polarity identification, is more difficult than that for a single sentence because there are more ways to express positions. In [1], lexicons used in General Inquirer with Turney's internet-based approach for sentiment classification. The purpose of clustering is to provide a good visualization for the reasons extracted from blogs toward a topic.

Summarizer module scores each sentence in the blog and picks top 30 percent of high scored sentences. In [2] paradigms proposed for extracting salient sentences from text using features like word and phrase frequency position in the text and key phrases to summarize scientific documents. Many approaches addressed the problem by building systems depending of the type of the required summary. In [2] While extracting summarization is mainly concerned with what the summary content should be, usually relying solely on extraction of sentences, abstractive summarization puts strong emphasis on the form, aiming to produce a grammatical summary; this usually requires advanced language generation techniques.

A crucial issue that will certainly drive future research on summarization is evaluation. During the last fifteen years, many system evaluation competitions like TREC, 1 DUC2 and MUC3 have created sets of training material and have established baselines for performance levels. However, a universal strategy to evaluate summarization systems is still absent.

Fact Extractor module processes each sentence and evaluates whether it contains any fact. For clustering module in blog mining, clustering is an important method for searching and extracting useful knowledge from massive blog text data in a huge number of the blog websites spread in the world [3]. A blog document consists of three text blocks i.e. title body and comments. Importantly, they play different roles in presenting the topics and opinions of blog pages. For a blog data, the features groups are formed through extracting the features from each text block of the three. In clustering the features groups should be treated differently to reflect their roles in page characterization. We collect the output of the above module and cluster the obtained information from different blogs.

II. RELATED WORK

Blog analysis can be done by three main tasks reason extraction, sentiment classification, and reason clustering. [3]

Shows the *Reason Extractor:* In some way, reason extraction is more similar to the subjective or objective classification problem for editorial reviews. The density of topic related words in a paragraph can then be used as a measure of how strong a paragraph contains a reason. A measure is used as an indicator for how closely the word is to the topic we are interested. *Sentiment classification*: The lexicons in General Inquirer are compiled by experts, with each word labeled with tags, like positive, negative, hostile, strong, etc. A total of 11788 words are included in the vocabulary, while only 4206 words are labeled with positive or negative tags. To extend the dichotomy of words to more grade level, we use Turney's point wise mutual information (PMI) as the similarity measure of two words. *Reason Clustering*: The purpose of clustering is to provide a good visualization for the reasons extracted from blogs toward a topic. We use frequent item set based hierarchical clustering to cluster those paragraphs containing reasons. Each paragraph is represented by bag of words. We use either topic related words or all words as the bag. This comparison would be used to verify the effectiveness of topic related words identified. [6] describes the *Price related searching*: Most of the users generally decide their budget first and then start searching for a product so price related querying will be a major improvement. *Verbs and Conjunctions*: From the examples taken into consideration, we observed that a big role is being played by the conjunctions and verbs. Conjunctions can easily change the point of view of the author where as verbs used by the author can often be misleading. *Processing comments on the blog*: A very useful source of information which we have missed out till now is the comments which the users leave on the blog. This plays a very important role in judging about the prestige of the blog as well as the rating of the product. Some of the comments which are harsh should lead towards the decline in the rating of the product and vice-versa should also hold true.

For the special characters of blog pages we present blogger's role based retrieval model by mining the blogger role to represent semantic meaning between blogger and query. [5] Describes to extract the blog role and role features from semantic dictionary WorldNet then we compute document relevance probability by the blogger role and the classical retrieval by iterative algorithm. [4] Describes the strategies for knowledge discovery in blogs are *Matrix Factorization*: Matrix factorization is another common tool researcher's use for knowledge discovery in blogs. This technique involves decomposing a matrix into some canonical form. Many different matrix decompositions exist, such as LU decomposition (which writes a matrix as the product of a lower triangular matrix and an upper triangular matrix), singular value decomposition (SVD), Cholesky decomposition, and QR decomposition, and each is useful for particular problems.

A user might choose to drive knowledge discovery in blogs by specifying criteria for ranking retrieved blog entries. Ranking blogs is quite similar to ranking Web pages Page Rank and Hypertext Induced Topic Selection (HITS) are two popular techniques for Webpage ranking that exploit the link structure between such pages. These algorithms focus on a directed graph setting that describes resources via nodes and hyperlinks. Link popularity-based algorithms, however, might not work well for blog mining because blog pages aren't well linked and bloggers might try to exploit such a system to boost their rank.

Clustering assigns a set of observations to subsets, referred to as clusters, such that observations in the same cluster are similar according to pre specified criteria. Partition algorithms typically determine all clusters at once but can also act as divisive algorithms in hierarchical clustering. K-means clustering and quality threshold (QT) clustering are both partitioned clustering algorithms. Clustering algorithms might require users to specify the number of clusters the algorithm should produce in the input data set. An important step in clustering is to select a distance measure, which will determine how the algorithm calculates two elements' similarity.

III. TASK DEFINITIONS AND ALGORITHMS

Figure 1 shows the flowchart of the blog analysis system. The three main tasks are word extraction, sentiment classification, and clustering. Other modules include preprocessing step and irrelevant blog filtering. Given a topic q (e.g. abortion), we use the popular search engines like goggle's blog search engine (http://blogsearch.google.com/) to query the blog entries.

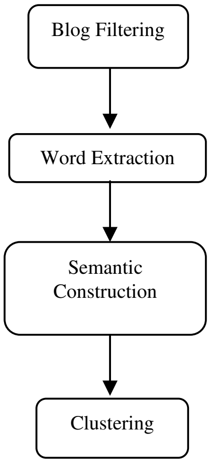

Figure 1

The pre-processing step contains paragraph segmentation, sentence segmentation, part-of-speech tagging, stemming and paragraph segmentation. We use tags such as <div>, <h>, <p> and consecutive <br> as the rules for segmenting documents into paragraphs, since they are the common tags used by blog service providers. If somehow the blogger didn't divide his article into paragraphs, then we will not be able to divide it. In practice, almost all bloggers use paragraphs to organize their thought and usually a paragraph contains less than 10 sentences. For sentence segmentation, we use the tool from [13]; to carry out stemming, we use the famous Porter stemmer [15]; for part-of-speech tagging we use GeniaTagger [16]

*A. **Blog Filtering***

Although blogosphere provides a non-intrusive method for collecting public opinions, the following problem is the diversified, miscellaneous and complicated opinions which touch many aspects of the query topic, which are way out from the reasons that we want to extract. For example, of the returned blogs for the “gene cloning”topic, some report the pass of bills for using cloning for stem cell research, some are personal profiles but containing the opinion toward cloning without reasons. The mixed contents with irrelevant blogs make the extraction of reasons difficult for our specific study.

To filter such irrelevant blogs, we adopt density based approach which calculates the percentage of sentences containing at least one topic word. If the density is below a threshold, we simply discard that blog.

B. *Word extraction*

We observe that, when people write their reasons, topic words are usually included in the same paragraph. Although, topic words themselves may be few, they can still be the clues for us to find other topic related words. The density of topic related words in a paragraph can then be used to measure the strength of a paragraph. We use logarithm of odds ratio (LODR) score to find topic related words. This measure is then used as an indicator for how closely the word is to the topic we are interested.

To evaluate how related a word t is to the topic words, we calculate the probability that t occurs in the circumstance that topic words show up as well as the same event when the topic words disappear. We then take the logarithm of the odds ratio of these two probabilities for our first measure. Formally, Let p be the probability of the word, t, co-occurs with any of the topic words.

$$P = \Pr(t \mid \text{Paragraph containing topic words})$$
$$= \frac{(\#\ of\ paragraph\ containing\ t\ and\ any\ topic\ word)}{(\#\ of\ paragraph\ containing\ any\ topic\ word)}$$

Also, let q be the probability of t, co-occurs with no topic words. That is, the occurrence probability of the Word t when no topic word shows.

$$q = Pr(t \mid paragraph\ not\ containing\ any\ topic\ word)$$
$$= \frac{(\#\ of\ paragraph\ containing\ t\ an\ no\ topic\ word)}{(\#\ of\ paragraph\ not\ contaoining\ any\ topic\ word)}$$

The logarithm of the odds ratio is then defined as:

$$LODR(t) = log\frac{p/(1-p)}{q/(1-q)} = log\frac{p(1-q)}{q(1-p)} \qquad (1)$$

This is also the difference of the logs of the two probabilities p and q.

With the weight for a word, we can calculate the average strength of a paragraph containing a reason toward the topic by:

$$ATS(p) - \sum_{t\epsilon p} \mathrm{LODR}(t) * f\left(\frac{t}{P}\right) \qquad (2)$$

Where f(tlP) denotes the frequency of word t in P. We use the above average topic strength as an indicator of whether a paragraph contains a reason or not. Generally, the higher the strength, the stronger the paragraph contains a reason

C. *Semantic construction:*

In this paper, we combine the lexicons used in General Inquirer [17] with Turney’s internet-based approach [11] for semantic construction. The lexicons in General Inquirer are compiled by experts, with each word labeled with tags, like positive, negative, hostile, strong, etc. A total of 11788 words are included in the vocabulary, while only 4206 words are labeled with positive or negative tags. To extend the dichotomy of words to more grade level, we use Turney’s point wise mutual information (PMI) as the similarity measure of two words. The semantic orientation of a word is then given by the difference between the PMI of a word to “excellent” and “poor”. Let hits (query) be the number of hits returned by search engine, such as AltaVista or Google. The semantic orientation of a word can be calculated as follows:

$$SO(phrase) = \log_2 \frac{hits(phrase\ NEAR\ excellent)hits\ (poor)}{hits(phrase\ NEAR\ poor)hits(excellent)}$$

The word “excellent” and “poor” are chosen for Turney’s study because they are mentioned often in product reviews. With the scores for all words in semantic dictionary, we sum up the scores of the words in the paragraph which contain a reason. If the score is greater and equal than 0, the reason is positive, otherwise, the reason is negative.

$$ASO(P) = \sum_{t\epsilon P \cap D} \mathrm{SO}(t) * f(\frac{t}{P})/|P \cap D|$$

D. *Clustering*

We extend the vector space model (VSM) to structured vector space model to encode the blog data. In the extended VSM, we divide vectors into three groups of sub vectors, $V_{t,}$ V_b and $V_{c,}$ contain the features of the title, body and comments block, respectively. The elements of a sub vector are the frequency of the corresponding terms in that block. As such a blog page is represented as a vector shown $V=[V_t,V_b,V_c]$, Where V_t, V_b and V_c are the sub vectors for three blocks

Let W = $\{w_t,w_b,w_c\}$ denote the weights to the three features groups. The word frequency matrix of blog data can be defined as follows

$$V = (v_{lji})3\times(m1+m2+m3)\times n$$

Where v_{lji} denote the frequency of the word of the jth feature in the lth group in blog document I, The three blocks of the title, body and comments are ordered as group 1, group 2 and group 3 respectively. The numbers of the features in these groups are denoted as $m_{1,}m_2$, and m_3

1) Similarity measure between sentences

Definition 1: Word Form Similarity The word form similarity is mainly used to describe the form similarity between two sentences, is measured by the number of same words in two sentences. It should be getting rid of the stop words in the computation. If S1 and S2 are two sentences, the word form similarity is calculated by the formula (1).

Sim1(S1,S2)=2*(SameWord(S1,S2)/(Len(S1)+Len(S2))) (1)

Here SameWord(S1,S2) is the number of the same words in two sentences, Len(S) is the word number in the sentence S.

Definition 2: Word Order Similarity

The word order similarity is mainly used to describe the sequence similarity between two sentences can be presented by many kinds of style, the different sequence of the words stand for different meanings. Here we describe the sentence as three vectors as follows:

V1= {d11, d12… d1n1}

V2= {d21, d22… d2n2}

V3= {d31, d32… d3n3}

Here the weight d1i in vector V1 is the tf-idf value of the words; the weight d2i in vector V2 is the bi-gram whether occur in the sentence (0 stands for no-occurring, 1 stands for occurring); the weight d3i in vector V3 is the tri-gram whether occur in the sentence. The word order similarity between S1 and S2 is:

Sim2(S1,S2)=λ1*Cos(V11,V21)+λ2*Cos(V12,V22)
+λ3*Cos(V13,V23) (2)

Here λ1+λ2+λ3=1. λi stands for the ratio of each part.

Definition 3: Word Semantic Similarity

The word semantic similarity is mainly used to describe the semantic similarity between two sentences. Here the word semantic similarity computing (Jiang Min, 2008) is based on the HowNet [1]. Based on semantic similarity among words, we define Word-Sentence Similarity (WSSim) to be the maximum similarity between the word w and words within the sentence S. Therefore, we estimate WSSim(w,S) with the following formula:

WSSim(w,S)=max{Sim(w,Wi)|Wi∈S, (3)

Where w and Wi are words

Here the Sim(w,Wi) is the word similarity between w and Wi. With WSSim(w,S), we define the sentence

Definition 4: Sentence Similarity

The sentence similarity usually described as a number between zero and one, zero stands for non-similar, one stands for total similar. The larger the number is, the more the sentences similar. The sentence similarity between S1 and S2 is defined as follows:

Sim(S1,S2)=λ1*Sim1(S1,S2)+λ2*Sim2(S1,S2)
+λ3*Sim3(S1,S2) (5)

Here λ1, λ2, λ3 is the constant, and satisfied the equation: λ1+λ2 +λ3=1. Where λ1=0.2, λ2=0.1, λ3=0.7.

2) *Estimating the number of clusters*

Determination of the optimal number of sentence clusters in a text document is a difficult issue and depends on the compression ratio of summary and chosen similarity measure, as well as on the document topics. For clustering of sentences, customers can't predict the latent topic number in the document, so it's impossible to offer k effectively. The strategy that we used to determine the optimal number of clusters (the number of topics in a document) is based on the distribution of words in the sentences:

$$k = n\frac{|D|}{\sum_{i=1}^{n}|si|} = n\frac{|\bigcup_{i=1}^{n} si}{\sum_{i=1}^{n}|si|}$$

Where |D| is the number of terms in the document D, |Si| is the number of terms in the sentence Si, n is the number of sentences in document D. Here we analyze the property of this estimation by two extreme cases, please references the (Ramiz M. Aliguliyev, 2008),

3) *Sentences Clustering*

Once determinates the number of sentences clusters, we can use the K-means method to cluster the sentences of the document. This algorithm can be described as follows:

Input: n sentences
K: the number of clusters
Output: the sentences clusters

Step1: Random select K sentences into K clusters respectively, these sentences represent the initial cluster central sentences.

Step2: Assign each sentence to the cluster that has the closest central sentence.

Step3: When all sentences have been assigned, recalculate the central sentence of each cluster. The central sentence is the one which own the lowest accumulative similarity.

Step4: Repeat Steps 2 and 3 until the central sentence no longer move. This produces a separation of the sentences into K clusters from which the metric to be minimized can be calculated.

4) *Topic Sentences Extraction*

Based on the result of section C, assume the sentences clusters is: D = {C1, C2, … , Ck}. First, determinates the central sentence μi of each cluster based on the accumulative similarity between the sentence Si and other sentences, then calculates the similarity between the sentence Si and the central sentence μi. Assume that the similarity of central sentence μi as 1, sorts the sentences based on its similarity weight, and chooses the high weight sentences as the topic sentences. At the same time, considering the recall rate of the text summarization, the text summary should include every 168cluster sentences according to the principle of priority extract clusters in the process of extracting sentences.

IV. CONCLUSION

In this paper, we suggest the application of blogosphere on summarization and clustering of blogs. We stress the importance of reasons for each given topic. We solve the problem of blog summarization by four subtasks: blog filtering, word extraction, semantic construction and clustering. The main contribution of this work is two folds: We propose unsupervised approaches for the above subtasks, which make them easy to apply to different topics. We suggest four new measures for clustering. Meanwhile, other sophisticated approaches can be devised to better improve the performance

V. REFERENCES

[1] K. Dave, S. Lawrence, and D.M. Pennock.Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews.WWW' 03, pp. 519-528..

[2] B. Fung, K. Wang and M. Ester. Hierarchical Document Clustering Using Frequent Itemsets.SDM' 03, pp. 59-70.

[3] M. Hu and B. Liu. Mining and Summarizing Customer Reviews. SIGKDD' 04, pp. 168-177..

[4] S.-M. Kim and H. Eduard. Determining the Sentiment of Opinions. Coling' 04, pp.1367-1373.

[5]] L. W. Ku, Y. T. Liang and H. H. Chen. Opinion extraction, summarization and tracking in newsand blog corpora. AAAI-2006 Spring Symposium on Computational Approaches to Analyzing Weblogs, pp. 100-107.

[6] B. Liu, M. Hu and J. Cheng. Opinion observer: analyzing and comparing opinions on the Web. WWW' 05, pp. 342-351.

[7] H. Ohno, Y. Kusumura, Y. Hijikata and S.Nishida, Social summarization method for feedback comments in online auction. Syst.Comput. Japan 37, 8 (Jul. 2006), 38-55.

[8] B. Pang and L. Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. ACL' 04, pages 271-278.

[9] E. Riloff and J. Wiebe. Learning extraction patternsfor subjective expressions. Proceedings of the 2003 Conference on EMNLP, pages 105-112. 2003.

[10] H. Takamura, T. Inui and M. Okumura. Extracting Semantic Orientations of Phrasesfrom Dictionary. NAACL-HLT' 07, pp. 292– 299.

[11] P. Turney. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. ACL' 02, pp.417-424.

[12] H. Yu and V. Hatzivassiloglou. Towards Answering Opinion Questions: Separating Facts from Opinions and Identifying the Polarity of Opinion Sentences. EMNLP' 03, pp. 129– 136.

[13] L. Zhuang, F. Jing and X.Y. Zhu, Movie review mining and summarization. CIKM' 06, pp.43-50.

[14] Sentence segmentation tool from Cognitive Computation Group in UIUC

[15] Barzilay, Elhadad, 1997. Using lexical chains for text summarization. Proceedings of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization.

[16] Berger and Mittal, 2000. Query-relevant summarization using faqs. Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics.

[17] Carbonell, Goldstein, 1998. The use of MMR, diversity-based reranking for reordering documents and producting summaries [A], In: Proceedings of the 21st ACM-SIGIR International Conference on Research and Development in Information Retrieval [C], Melbourne, Australia.

[18] Jiang Min, Xiao Shi-bin, Wang Hong-wei et al, 2008. An improved word similarity computing method based on HowNet[J].Journal of Chinese information processing.

[19] Goldstein, Kantrowitz, Mittal, and Carbonell, 1999. Summarization text documents: Sentence selection and evaluation metrics. Proceedings,SIGIR.