Clustering of Blogs with Enhanced Semantics

A. K. Singh
Department of Electronics
and Computer Engineering
Indian Institute of Technology
Roorkee, Uttaranchal, India

R. C. Joshi
Department of Electronics
and Computer Engineering
Indian Institute of Technology
Roorkee, Uttaranchal, India

ABSTRACT
Blogs are among the fastest growing space among the user generated content over the internet. It is fast becoming the tool for information dissemination, and communication. Blogs provide a platform for information sharing, discussions, and expression of reader's reactions to the blog post. Clustering of blogs greatly simplify blog searching and browsing by organizing them into similar groups. The Blogs are generally organized using tags. In this paper, we have studied the effect of considering other relevant neighborhood contexts and adding the extracted information to the original tag set carried by the blog. The added semantics is extracted by disambiguating all the synsets for the important terms/ or key phrases within the blog. This work reports the study of measuring similarity, on enhanced blog features and subsequently grouping of all blog articles based on the semantics of the tags they carry. We propose to include the semantics extracted from the title, body, and comments of a blog post to its original tagset in clustering blog documents and evaluate the hypothesis that adding extracted semantics from these blog constituents improves the cluster quality. For clustering k-means algorithm is used. The experimental results obtained confirm our hypothesis that adding the semantics improves better clusters. The approach first extracts the relevant features from the target blog corpus, title and comments. The other senses represented by the relevant keywords are discovered by using a general purpose semantics extractor. All the synsets of the relevant keywords are extracted from the WORDNET. The extracted keyword senses are then appended to the base tagsets. A semantic similarity measure is used for computing the semantic similarity among the documents. Clusters are obtained based on it. The two clusters output are compared.

General Terms
Clustering, blog mining, blog, text mining, semantic similarity

Keywords
tagset, VSM (vector space model), k-means clustering, blog clustering, semantic simialrity

1. INTRODUCTION
Blogs are significantly used as a digital diary that is available on web, by bloggers with freedom of expression and sharing their real world, and web based experiences. Blogs are characterized with more focus on publishing, and with a strong sense of community. Although the definition of blogs is not necessarily definite, blogs are generally understood to be personal web pages authored by a single individual and made up of a sequence of entries of the author's thoughts, opinion, or beliefs etc., that are arranged chronologically. Blogs content is dynamic and frequent changes to the blogs are observed. Blogs include links to other blogs. The content and purpose of blogs vary greatly, from links and commentaries about other web sites, to news about events / individual, and comments it receives from the readers. Blog update frequency depends upon the availability of blogger and readers interest. Most often the blogs see a short span of life in terms of active users' interest. The blogs became popular just before a decade with the emergence of build-your-own blog providers (e.g., blogger.com, blogspot.com etc.) and many easy-to-use blog publishing tools. Over the past few years, there has been an exponential growth in the number of blogs [1]. *Technorati* publishes an article, regarding state of blogosphere, yearly. In 2010, it says that "*nearly 50% of all bloggers believe that they will be getting the news, entertainment from blogs in next five years than from the traditional media*". Recently there is a significant growth in the mobile blogging. The *Blog Herald* reports the estimate of the size of the blogosphere as 150 million based upon the count of websites tracked by Blog Pulse that are identified as blogs.

The task of blog clustering facilitates the task of managing blogs, searching, and subsequent retrieval. In text clustering domain there is a challenge of high dimensionality of features. Humans find difficulty in intuitive understanding of high dimensional data. The better understanding a user has about the data in hand the more likely the user will succeed in accessing true class structure [2]. The domain information can be utilized for improving quality of feature extraction, similarity computation, grouping, and cluster representation [3].

Blogs are user generated content thus have the characteristics of decentralized and independent development. Since most of the blog articles are written by people who have little domain knowledge. There can be spelling mistakes, new words, and moreover different bloggers may use different vocabulary for same contexts. These all makes blog mining difficult. Users may categorize the post under the category he feels it belongs solely based on his perception. Thus different bloggers could categorize the similar content in different categories.

Blogs articles use the tagging approach as another means of indexing and properly organizing blog articles. Tags are collections of keywords that are attached to blog entries, with the intension to help describe the blog post. Tagging is a widely used feature in the latest web application. Tagging allows users to define their own terms to arrange items [4]. Although usability is enhanced with tagging, precision and recall may drop. While tagging has become very popular, and tags can be found on many popular blogs, there has not been any standard tagging system (to our knowledge).

The task of blog Clustering is broken down into two stages. The first stage is to preprocess the blogs, i.e. transforming the blogs into a suitable and useful data representation. Preprocessing the blogs is probably at least as important as the choice of an algorithm, since an algorithm can only be as good as the data it works on. The second stage is to analyze the prepared data and group them into clusters, i.e. the application of clustering algorithm. This work explores the use of semantics extracted from the blog content, title, and comments appended to the blog tagset, for improving the effectiveness of clustering. The approach studies the potential benefits of appending all important synonyms obtained from WordNet [5], corresponding to the important key terms/ key phrases retrieved from the blog title, body , and comments which are extracted from the articles using popular TFIDF score, and applying a Word Sense Disambiguation (WSD) before appending to the base blog tagset. It also explores the semantic similarity computation between relevant words for overall computation of semantic similarity of documents for clustering.

2. RELATED WORK

Clustering approaches are reviewed in [6], It gives a taxonomy clustering approaches. They are primarily sub grouped into partitional, or hierarchical approaches. A detailed review of the document clustering approaches is provided in [7]. A comparision of the agglomerative and partitional clustering algorithm can be found in [8].

Hotho et al. [9] have analysed the benefits of using WordNet synonyms and up to five levels of hypernyms for document clustering (using the bisecting k-means algorithm) for text clustering. Moreover they have also applied the use of ontology for the purpose of text clustering [10]. The use of wordnet synsets for text clustering is explored in [11], and is found to be effective.

Tags are generally freely chosen by the user. Blog post indexing by the blogging softwares is based on the tags. The blog tagging site www.technorati.com, and Blog pulse tracks millions of tags. These tags are not part of a controlled vocabulary. They are aggregated into a hierarchy and are referred as a "folksonomy" [12]. Folksonomies differ from established classification methods e.g., in library science; that rely on controlled vocabularies, established taxonomic organization. They appear to be superior for fast-changing domains with unclear boundaries.

There is lot of other research activities in Blogosphere. The detailed comparison among key knowledge discovery techniques for blogs is made in [13]. The authors compare them in terms of effectiveness in combating present challenges of blog mining.

A feature weighting k-means algorithm for text document mining that applies k-means on weighted features is given in [14].

Since the blogs are huge. It is not clear how to make effective use of an unstructured plethora of tags. Moreover the spam blogs are beginning to threaten tag validity.

3. PREPROCESSING AND CLUSTERING

3.1 Preprocessing

A clustering task typically involves feature extraction and/or feature selection. It follows the task of defining a similarity measure appropriate to the data domain, then applying the clustering algorithm, and finally the evaluation of the clustering output. This process is explained in figure 1.

Like any other clustering task, here also the first step involves the preprocessing of target blog corpus for feature extraction. The pre processing phase involves many sub tasks.

The first step in the pre processing phase is to remove all the "stopwords". Stopwords are words with high frequency of occurrence, in the documents. They are considered as non–descriptive within a bag–of–words approach. If a word appears in all the document classes evenly (regardless of class distribution) it is unlikely to be useful in clustering. They typically comprise prepositions, articles, etc. Since frequency of keyword decides weight of that keyword in the document, all stopwords are removed using a standard stopwords list.

This is followed by stemming stage where blog documents are processed using the Porter stemmer for grouping words that has same conceptual meaning. The stemming reduces feature space, for example "computer" and "computers" would be same word.

Here we use the vector space model (VSM), proposed in [15]. VSM is based on tf-idf for computing the term weights in a document, in which a document is represented as a vector or 'bag of words', i.e., by the words it contains and their frequency, regardless of their order.

Definition 1: A feature vector (FV) is the data object that typically consists of measurements over all the dimensions for all the features. $X = (x_1, x_2, ..., x_d)$, where individual scalar component x_i are called feature of a pattern X, and d is the dimensionality.

Definition 2: A pattern set is denoted as $\mathbf{X} = (\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_n})$ where i[th] pattern in **X** is

$x_i = (x_{i,1}, x_{i,2}, ..., x_{i,d})$. A pattern set to be clustered is n X d pattern matrix.

A cluster is a set of patterns whose distribution in feature space corresponds to probability density function specific to the class.

The initial feature vector is obtained for each blog document in the target corpus. The length of the resulting vectors is given by the number of different stemmed terms in the text corpus. Instead of using the original terms in the documents, the frequency of stemmed terms is computed. Normalized TFIDF score is computed for each term. Stemming, stopwords removal, and pruning all aim to improve clustering quality by removing noise, i.e. irrelevant data for clustering. They all lead to a reduction in the number of dimensions in the feature space.

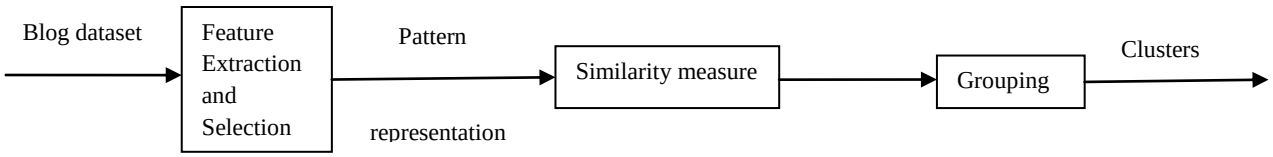

Fig1: Steps in the blog clustering

A feature vector that distributes proper weight to the features that are keywords here; appearing in a document could be represented as,

$[w_{1d,}w_{2d,},...,w_{nd,}]^T$,where d ∈ D, and n is the number of terms in the document d.

$$w_{ti,dj} = tfidf(t_i, d_j) = tf(t_i, d_j) * \log \frac{|D|}{|\{d \in D \mid t \in d\}|}$$

$$= tf(t_i, d_j) * \log \frac{N}{N(t_i)}$$

where N is total number of documents, and N(t_i) is the number of documents conating t.

Term weighting [17] is concerned with the estimation of the importance of individual terms. The weights are assigned to the terms based on the statistical properties of the terms in the documents. Term Frequency (tf) is the measure of how many times a term has appeared in the document. The tf weighing scheme considers all the terms equally important, but this is always not true. Certain terms have no discriminating power but have more occurrences in the documents. Inverse Document Frequency (idf) takes care of this by assigning less weight to frequently appearing terms and high weight to rarely appearing terms. In practice not all documents are of equal length. Long documents generally contain certain terms repeated more frequently, therefore the tf scores may be large for features extracted from long documents than the shorter ones. As a result long documents pretend to be more relevant to the words that occur more frequently, though it may not be the case in reality. Both tf and idf suffer with this drawback. A normalized tfidf score easily compensates this effect. The normalized version could be represented as,

$$w_{i,j} = \frac{tfidf(t_i, d_j)}{\sqrt{\sum_{k=1}^{|T|} tfidf(t_k, t_j)^2}}$$

All the top score words are taken as the most relevant keyphrases for each document. These keyphrases are used for computing the similarity between the documents. There are various similarity measures in the literature. Similarity measures in the text mining domain could primarily be grouped into Eucledian similarity measures; that are based on computing the distances between the coordinates represented in space corresponding to the data objects, and non Euclidian similarity measures. Popular non-Euclidian similarity measures are Jacard similarity, cosine similarity, and edit distance based similarity measures. Eucledian distance for high dimensional data like text data that is sparse is less effective. Here we have used cosine similarity measure for calculating the similarity between two blog posts d1, d2, which is defined as cosine of the angle between the vectors $\vec{d_1}, \vec{d_2}$ as:

$$sim(d_1, d_2) = \cos(\vec{d_1}.\vec{d_2}) = \frac{\vec{d_1}.\vec{d_2}}{|\vec{d_1}|.|\vec{d_2}|}$$

$$= \frac{\sum_{i=1}^{N} w_{i,d_1} \times w_{i,d_2}}{\sqrt{\sum_{i=1}^{N} {w_{i,d_1}}^2} \times \sqrt{\sum_{i=1}^{N} {w_{i,d_2}}^2}}$$

3.2 Semantic Similarity

The tfidf score is purely based on the statistical pattern distribution measure irrespective of the location and context. It has no consideration of the semantic meaning of the keyword. A word could have different senses/meaning in the different contexts. The methodologies for measuring semantic similarity among words could primarily be classified into graph based methods, and content based methods. Graph based method use a predefined taxonomy of the domain, and enumerates the similarity between the words based on the distance between the corresponding concepts in that hierarchy. Information content measures rely on the comparison of information content.

In an effort to measure the semantic similarity between documents, we have used a measure given in [18].

Definition 3: For concepts c_1, c_2 from wordnet, corresponding to any two relevant keywords or their synsets; the nearest common ancestor of the two concepts is some concept node NCA(c_1,c_2). Let h be the height of node NCA(c_1,c_2) from the root concept node, and l is the length of shortest distance path from c1,c2.

The semantic similarity measure between c1, c2 is defined as;

$$sim(c_1, c_2) \cong \frac{2h}{l+2h}$$

3.3 Evaluation of Clusters

For evaluating the quality of clusters two different evaluation measures are used in this work, namely purity and entropy. Entropy is an internal measure that measures the distribution of each class of documents with the obtained clusters. The smaller the entropy value is the better is the cluster output. The entropy E of a cluster $\mathbf{C_r}$ is computed as,

$$E(C_r) = \frac{1}{\log_2 k} \sum_{i=1}^{k} \frac{n_r^i}{n_r} \log_2 \frac{n_r^i}{n_r}$$

Where n_r^i is the number of documents in the i[th] class that are assigned to the r[th] cluster. The combined entropy for all the clusters is computed as,

$$E = \sum_{r=1}^{k} \frac{n_r}{n} E(C_r)$$

The purity measures homogeneity of a cluster, i.e., the extent to which each cluster contains documents from primarily one class. The purity of a cluster C_r is computed as,

$$P(S_r) = \frac{1}{n_r} \max_i (n_r^i)$$

and, the overall purity for the obtained clusters can be computed as,

$$P = \sum_{r=1}^{k} \frac{n_r}{n} P(C_r)$$

4. EXPERIMENTS AND RESULTS

Our fundamental approach is to group documents that share tags, into clusters and then compare the similarity of all documents within a cluster for evaluating the cluster quality. Since the tagsets are the primary features used here for clustering, documents that share a tagset t are contextually similar than the documents which have different tagsets. As a next step to this approach, the features extracted from the wordnet; synsets of all the relevant keywords are computed, and are appended to the original features.

Table 1: Cluster Purity and Entropy with enhanced semantics on using tfidf score.

Features	Purity	Entropy
X	.352	.89
X_t	.275	.903
X_{tc}	.394	.756
X_{all}	.654	.65

This is the enhancement of the feature set. For the blog data used for experimentation; 4 categories of blogs are collected. The blogs were searched from the blogosphere using "food", "sports", "health", and "music" as key words in google blog search. Each entry has at least one reader comment. Each category has 60 files for a total of 240 blog articles.

Table 2: Cluster Purity and Entropy with enhanced semantics on using semantic similarity

Features	Purity	Entropy
X	.381	.912
X_t	.294	.953
X_{tc}	.473	.826
X_{all}	.752	.703

In total four set of blog feature configurations are prepared for the experiments. The Blogs has initial Feature matrix X that contains the original tagsets the blog carries. Another two feature matrices Xt, Xtc has the additional features that are extracted from the blog title, and blog content appended, respectively. Further, one more feature matrix Xtcn that has those additional features that are extracted from the users' comment to the blog.

The entropy and purity computed for the experiments done using the two similarity measures discussed in the paper are tabulated in table 1 and table 2 respectively. The comparison is plotted in figure 2, and figure 3.

5. DISCUSSION AND FUTURE WORK

Table 1 proves our hypothesis of including the semantics extracted from the body, title, and user comments. By enhancing the feature set the blogs can be better grouped. Figure 1and 2 show the comparison of the two approaches. It can be seen that enhancement of feature set combined with semantic similarity computation improves over the cluster quality. The comments are very dynamic for highly active blogs. They change too frequently. The clustering method shall be adaptable to the incremental changes in the comments. The use of synsets gige limited semantics to the semantic similarity evaluation. Other forms needs further work.

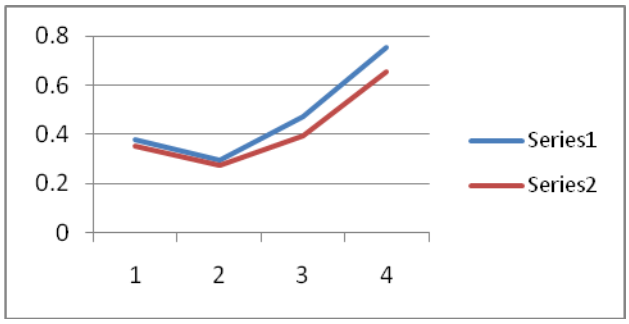

Fig 2: A Comparison of Cluster Purity for semantic similarity and tfidf based approach.

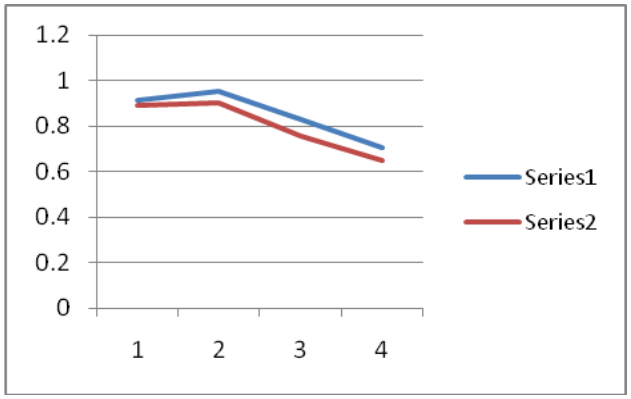

Fig 3: A Comparison of Cluster Entropy for semantic similarity and tfidf based approach

6. REFERENCES

[1] Jain, A.K. and Dubes, R. C. 1988, Algorithms for Clustering Data, Prentice-Hall advanced reference series, Prentice-Hall, Inc., Upper Saddle River, NJ.

[2] Murty, M.N. and Jain, A. K. 1995, Knowledge-based clustering scheme for collection management and retrieval of library books, Pattern Recognition, 28, pp. 949–964.

[3] Mishne, G. 2006, AutoTag: A collaborative approach to automated tag assignment for weblog posts, In Proc. of WWW2006, pp. 953–954.

[4] Haveliwala, T., Gionis, A., Klein, D., and Indyk, P. 2002, Evaluating strategies for similarity search on the web, In Proceedings of the Eleventh International World Wide Web Conference, Honolulu, Hawaii, pp. 432–442.

[5] G. A. Miller. 1995, Wordnet: A lexical database for English, Communications of the ACM, 38(11), pp. 39–41.

[6] Jain, A. K., Murty, M. N., and Flynn, P. J. 1999, Data Clustering a review, ACM Computing Surveys, Vol. 31, No. 3.

[7] Michael Steinbach and George Karypis and Vipin Kumar 2000, A comparison of document clustering techniques, In KDD Workshop on Text Mining, Boston, MA, pp. 109-111.

[8] Y.Zhao and G.Karypis 2002, Comparison of agglomerative and partitional document clustering algorithms, Technical Report #02-014, University of Minnesota.

[9] Andreas Hotho, Steffen Staab, and Gerd Stumme 2003, Wordnet improves Text document Clustering, In Proc. of the SIGIR 2003 Semantic Web Workshop, pp. 541-544.

[10] A.Hotho, A.Maedche and S.Staab 2003, Ontology-based text document clustering, Proc. of the Conf. on Intelligent Information Systems.

[11] Julio Gonzalo, Felisa Verdejo, Irina Chugur, Juan Cigarrin 1998, Indexing with WordNet : synsets can improve text retrieval, pp. 38—44.

[12] Rujiang Bai, Xiaoyue Wang, and Junhua Liao 2009, Folksonomy for the Blogosphere: Blog Identification and Classification, Computer Science and Information Engineering, 2009 WRI World Congress on , vol.3, no., pp.631-635.

[13] Lakshmanan G.T., and Oberhofer M.A. 2010, Knowledge Discovery in the Blogosphere: Approaches and Challenges, Internet Computing, IEEE , vol.14, no.2, pp.24-32.

[14] Liping Jing, M.K. Ng, J. Xu, and J.Z. Huang 2005, Subspace clustering of text documents with feature weighting k-means algorithm, Proc.of PAKDD, volume 3518 of Lecture Notes in Computer Science, pp. 802-812.

[15] G. Salton, A. Wong, and C. S. Yang 1975, A Vector Space Model for Automatic Indexing, Communications of the ACM, vol. 18, no.11, pages 613-620.

[16] Salton G., and Buckley C. 1988, Term Weighting Approaches in Automatic Text Retrieval, Information Processing and Management 24(5), pp. 513-523.

[17] Baeza Yates R., and Ribeiro-Neto B. 1999, Modern information retrieval, Addison Wesley Longman Publishing Co. Inc., Boston, MA, USA.

[18] X. Wu, M. McTear, and P. Ojha 1993, Word sense disambiguation by a higher order connectionist net based on distributed representations, in Proceedings of TENCON '93. IEEE Region 10 International Conference on Computers, Communications and Automation, NY, USA, pp. 893-897.