Clustering Blogs Using Document Context Similarity and Spectral Graph Partitioning

Ramesh Kumar Ayyasamy[1], Saadat M. Alhashmi[1], Siew Eu-Gene[2], and Bashar Tahayna[1]

[1] School of Information Technology, [2]School of Business
Monash University, Malaysia
{ramesh.kumar,alhashmi,siew.eu-gene,bashar.tahayna}@monash.edu

Abstract. Semantic-based document clustering has been a challenging problem over the past few years and its execution depends on modeling the underlying content and its similarity metrics. Existing metrics evaluate pair wise *text similarity* based on text content, which is referred as *content similarity*. The performances of these measures are based on co-occurrences, and ignore the semantics among words. Although, several research works have been carried out to solve this problem, we propose a novel similarity measure by exploiting external knowledge base-Wikipedia to enhance document clustering task. Wikipedia articles and the main categories were used to predict and affiliate them to their semantic concepts. In this measure, we incorporate context similarity by constructing a vector with each dimension representing contents similarity between a document and other documents in the collection. Experimental result conducted on TREC blog dataset confirms that the use of context similarity measure, can improve the precision of document clustering significantly.

Keywords: Blog Clustering, Bipartite graph, Similarity Measures, Wikipedia.

1 Introduction

Clustering is an unsupervised learning technique that organizes similar members into groups or clusters and dissimilar members into other clusters. Clustering helps to discover the patterns and correlation in large data sets [2]. Document clustering received a significant attention in the last few years in the area of machine learning and text mining applications, such as webpages and blogs [4]. The entire text collection could be represented as *term by document* matrix, where each term is used as features for representing documents. Most traditional clustering systems are based on *bag-of-word* (BOW) representation and disregards semantic information and word order [7]. This BOW representation is restricted in calculating only term frequency in a document. Blogs - a new form of webpages, which consists of group of blogposts and each blogpost could be written on various topics, and BOW assumption cannot perform well [18]. There is no suitable way to identify and categorize blogposts; hence, it is an open problem for research [1]. Clustering similar posts based on different categories helps the user identify or search the particular topic of interest. Recently, blogs became the subject of research as the information content is large and diverse, and creates a need for automated organization of blog pages. Traditional clustering cannot find the hidden relationship between heterogeneous objects. Many clustering algorithms have been proposed in the various contexts of literature. Berkhin [3] investigated the applications of clustering algorithms for data mining. In most of the works, partitioned clustering algorithm [4] is suited for large data set clustering due to their low computational requirements. Baker et al. [11] used the distributional clustering method, which clusters together those terms that tend to indicate the presence of the same category, or group of categories. Xu et al. [10] presents a simple application using *Non-negative Matrix Factorization* (NMF) for document clustering. As *K-means* clustering algorithm cannot separate clusters that are non-linearly separable in input space, Dhillon et al. [13] used Jensen-Shanon divergence to cluster words in K-means fashion in text clustering. Despite widespread adoption on clustering tasks, only few studies have investigated using Wikipedia as a knowledge base for document clustering [5 - 7]. Gabrilovich [5] have applied structural knowledge repository-Wikipedia as feature generation technique. Their work confirmed that background knowledge based, features generated from Wikipedia can help text categorization. Huang et al. [6] have clustered documents using Wikipedia knowledge and utilized each Wikipedia article as a concept. This work extracted related terms such as synonyms, hyponyms and associated terms. Hu et al. [7] have used Wikipedia concept feature for document clustering. Development in data mining applications have demanded for clustering highly inter-related heterogeneous objects, such as *documents* and *terms* in a text corpus. Clustering each type of objects independently might not work well, since one type of objects can be defined by other type of objects. This paved the way for many researchers to co-cluster two or more heterogeneous data. The authors in [17], [22], expanded the traditional clustering algorithms and proposed bipartite spectral graph partitioning algorithm to cluster documents and terms simultaneously. Similar bipartite techniques were applied in medicine [8], image [20] and video processing [21]. Gao et al. [9] have suggested consistent bipartite graph co-partitioning (CBGC) by treating the tripartite graph as two-bipartite graphs. This work proved that consistent partitioning provides the optimal solution using positive Semi-Definite Programming (SDP). To co-cluster triplet data [9], terms, category and documents (Web pages) were used. This work utilizes, terms from web pages and does clustering based on it. Like bipartite graph partitioning, it has limitations that the clusters from different types of objects must have one-to-one associations and it fails to consider the surrounding text and context information. On the contrary, each blogpost consists of various features and a framework is needed beyond the using of terms representing documents. With the above motivation in mind, in this paper we constructed a tripartite spectral graph clustering, using *concepts*, *document*, and *document contexts* to cluster similar topical blogposts based on Wikipedia categorical index.

The outline of this paper is structured as follows: Section 2 defines, the different terms used in our paper. Section 3 explains our CSSGP framework which uses similarity based clustering on *content similarity*, *context similarity* and our *Concept frequency measure*. Section 4 describes the experiments and results in detail. Finally, section 5 concludes with the summary of the work and future research.

2 Definition of Terms

In this paper, we use these following terms:

- *"Document (D)"* refers to single blogpost from the collection. Let χ be the data set, where $\chi = \{D_1, D_2, ..., D_n\}, n \in \mathbb{Z}^+$
- *"Term(T)"* refers to a meaningful word from a particular blogpost. Let D be the Document which consists of set of terms, where $D = \{T_1, T_2, ..., T_m\}, m \in \mathbb{Z}^+$
- *"Document context(Dc)"* refers to document co-occurrence information and are constructed against the set of documents.
- *"Concept(C)"* refers to a Wikipedia article title. Let φ be the set of concepts, where $\varphi = \{C_1, C_2, ..., C_i\}, i \in \mathbb{Z}^+$
- *"Categories"* refers to Wikipedia's 12 main categories, where $\mathrm{T} = \{\varphi_1, \varphi_2, ..., \varphi_{12}\}$
- *"T-D-Cat"* refers to tripartite clustering of *Terms*, *Documents* and *Categories*
- *"C-D-Dc"* refers to tripartite clustering of *Concepts*, *Documents* and *Document Contexts*
- *"T-D"* refers to bipartite clustering of *Terms*, and *Documents.*

3 Clustering Blogs Using Context Similarity and Spectral Graph Partitioning Framework- CSSGP

We present our blog clustering framework using Context similarity and Spectral graph partitioning to deal with the above observations (See Figure 1).To illustrate the CSSGP framework effectiveness, we have conducted an extensive experiment aiming at document clustering.

Our contribution comes in two folds: First by using Wikipedia, we perform an explicit semantic based topic analysis by converting terms to concepts. We calculate surrounding text similarity using *conf.idf.* Second we used Content Similarity and surrounding text similarity to calculate Document Context Similarity. We used *document context, concepts,* and *document*s to produce a tripartite spectral graph, which helps for efficient document clustering. There is a general notion that the combination of terms and the BOW approach hold the greatest promise in text clustering. The fundamental challenges in bridging the gap between the syntactic and semantic elements is the design of distance function that measure the perceptual similarity between text features. To evaluate the content-based text similarity, we utilize the term weighting function, which is one of the most widely used metrics due to its simplicity and effectiveness [15]. Adding to this document-based text similarity (in section 3.1), we compute the concept-based similarities that two documents have in common (i.e., *the surrounding text*). Thus, we further compute this kind of similarity, which we denote as context similarity.

3.1 Document Context Similarity

Content Similarity Measure.
When two documents contain similar topics, there are higher chances for existence of many common terms. After pre-processing, we represent the stemmed words as

vectors for each document in a vector space model. The term weighting function *tf-idf* [15] been defined as;

$$tf.idf(t_k, D_j) = \#(t_k, D_j).\log\frac{|T_r|}{\#T_r(t_k)} \tag{1}$$

Where t_k denotes k'th terms, D_j denotes j'th documents, $\#(t_k, D_j)$ denotes the number of times t_k occurs in D_j, and $\#T_r(t_k)$ denotes the number of documents T_r in which t_k occurs at least once (also known as the document frequency of term t_k).

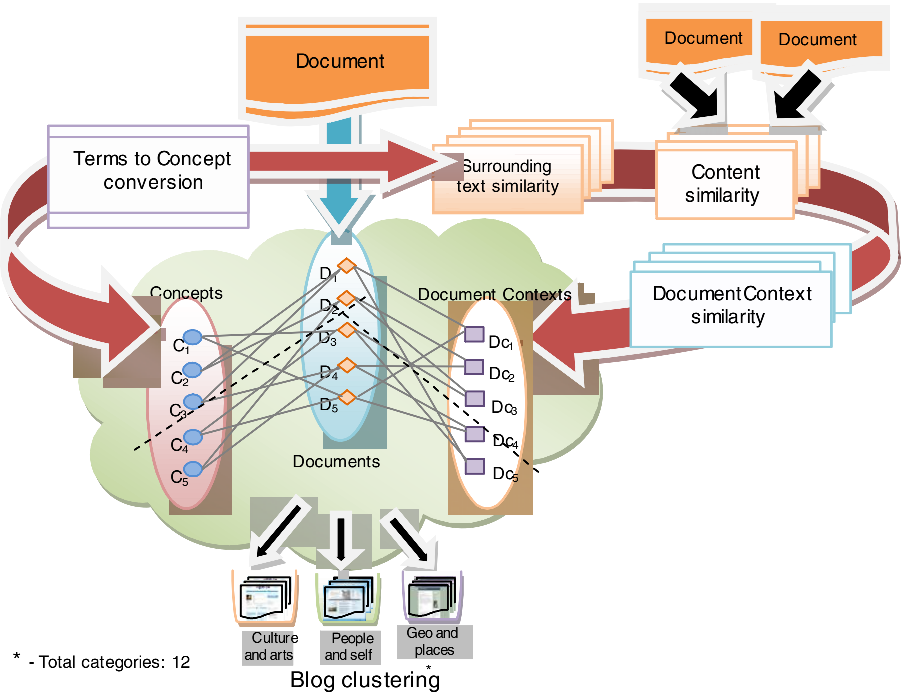


Fig. 1. Our proposed CSSGP Framework

Given this representation, the cosine similarity measure of the document pairs is represented as:

$$sim_{content}^{cosine}(D_p, D_k) = \frac{D_p.D_k}{|D_p||D_k|} \tag{2}$$

Where d_p and d_k are p'th and k'th document respectively.

Surrounding Text Similarity Measure (*Terms* to *Concepts* Conversion)

Our proposed method is a relatedness measurement method that computes the relatedness (i.e. among words) based on co-occurrence analysis. As a simple case, we map each word to a *concept*; the combination of these concepts/words can create a new *concept* (compound concept). Let us assume A is a word mapped to a *concept* C_1, B is a word mapped to a *concept* C_2, and ω is the set of extracted terms from blogpost.

Then there is a chance that the combination of A & B can produce a new concept (see Figure 2). Consider the following example,

$\$s = "President\ of\ the\ United\ States"$

$X = "President"$, $Y = "United"$, $Z = "States"$, $\omega = \{X, Y, Z\}$

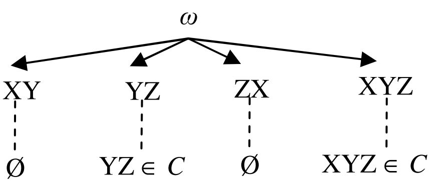


Fig. 2. Related measurement method

Where *C* is the set of concepts such that YZ is mapped to *Country* concept and XYZ is mapped to *President* Concept. Since the order of X, Y and Z can give different concept or no concept at all, we neglect the reordering. For effective clustering, we mine existing knowledge bases, which can provide a conceptual corpus. Wikipedia is one of the best online Knowledge Base to provide an individual English article with more than 3 million concepts. With that said, an important feature of Wikipedia is that on each wiki page, there is a URI which unambiguously represent the concept. Two words co-occur if they appear in an article within some distance of each other. Typically, the distance is a window of *k* words. The window limits the co-occurrence analysis area of an arbitrary range. The reason of setting a window is due to the number of co-occurring combinations that becomes too huge when the number of words in an article is large. Qualitatively, the fact that two words often occur, close to each other is more likely to be significant, compared to the fact that they occur in the same article, even more so when the number of words in an article is huge. The *n-gram based concept extraction* algorithm was briefly discussed in Ayyasamy et al.'s work [14]. We transform the surrounding text from *terms* into a weighted vector of *concepts* to represent in a vector space. The concept weights are computed with concept weighting scheme *conf.idf* [19].

$$conf \cdot idf(con,T) = conf(con,T).idf(con), \tag{3}$$

$$idf(con) = \log \frac{N_t}{pf(con)} \tag{4}$$

Where $conf(con,T)$ denotes the number of times a concept *C* occurs in the surrounding text *T*, *pf(con)* denotes the number of *documents* contains the concept *con,* and N_t is the number of documents in the collection. Our vector space model consists, the set of all vectors representing distinct documents. We need a similarity measure that compares two vectors and returns a numeric value that shows the level of similarity between them. There exists several similarity measures and *cosine metric* is the one among commonly employed similarity metric for text mining.

Document Context Similarity Measure

Document context consists of documents co-occurrence information and are constructed against the set of documents. The *document context* is a feature vector with each element representing the content similarity of the document and each document in the collection. In a given set of *n* documents $\chi = \{D_1, D_2, ..., D_n\}, n \in \mathbb{Z}^+$, the context vector of a document D_p is an *n-dimensional* vector $f(D_p) = < f_1(D_p), f_2(D_p).....f_n(D_p) >$, where each element is equal to the corresponding content similarity as follows:

$$f_k(D_p) = \alpha.Sim^{\cos ine}_{content}(D_p, D_k) + \beta.Sim^{\cos ine}_{concept}(D_p, D_k), 1 \le k \le n \quad \{\alpha, \beta \text{ are weighting factors}\} \tag{5}$$

We obtain the context similarity between two documents D_p and D_q by computing the similarity of the corresponding vectors $f(D_p)$ and $f(D_q)$. There exist various measures such as *Dice coefficient, Overlap Coefficient, Cosine Similarity, and Jaccard Similarity Coefficient* for computing the similarity in the vector space model [16]. We use *Cosine similarity*, a popular measure for text documents to measure our external cluster validation. For example, the context similarity using *Cosine Similarity* between two documents D_p and D_q defined as follows (6):

$$sim^{\cos ine}_{context(D_p, D_q)} = \frac{\sum_k f_k(D_p) \times f_k(D_q)}{\left(\sum_k f_k(D_p)^2 \times \sum_k f_k(D_q)^2 \right)^{\frac{1}{2}}} \tag{6}$$

3.2 Spectral Graph Clustering

Clustering is an unsupervised learning method, which can be formulated by partitioning the graph such that the edges between different groups have very lower weights, and the edges within the same group have higher weights. The importance of spectral graph clustering can be found on Luxburg's work [12]. The algorithm for bipartite spectral graph clustering (Figure 3) was briefly discussed in Dhillon's work [17].

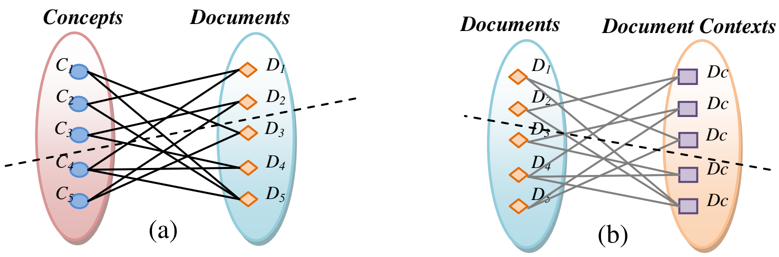


Fig. 3. The Bipartite Graph of (a) *C-D* , (b) *D-Dc* Clustering

Tripartite Spectral Graph Partitioning

In order to benefit from both the *concepts* and the *document contexts* for text clustering, we use tripartite graph as shown in Figure 4 to model the relations among the *concepts*, *documents*, and *document contexts*. In this tripartite graph (Figure 4), circle symbol represents the *concepts* $\mathrm{C} = \{C_1, C_2, C_3, ...C_i\}, i \in \mathbb{Z}^+$, diamond symbol represents the *documents* $D = \{D_1, D_2, D_3, ...D_j\}, j \in \mathbb{Z}^+$ and the rectangle symbol represents the *document contexts* $Dc = \{Dc_1, Dc_2, Dc_3, ...Dc_k\}, k \in \mathbb{Z}^+$ respectively. An undirected tripartite graph can be represented as quadruplet $G = \{C, D, Dc, E\}$, where E is the set of edges connecting vertices $\{\{C_i, D_j, Dc_k\} \mid C_i \in C, D_j \in D, Dc_k \in Dc\}$. An edge $\{C_i, D_j\}$ exists if *concepts* C_i occurs in *document* D_j, where the edges are undirected which denotes there are no edges between *concepts* or between *documents*.

The edge-weight between C_i and D_j equals the value of C_i in D_j, while the edge-weight between D_j and Dc_k equals the frequency of Dc_k in the surrounding text of document k. Consider i x j x k as *concept*-by-*document*-by-*document context* matrix,

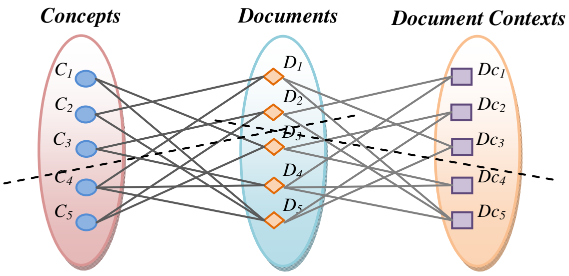


Fig. 4. Tripartite Graph of Concepts, Documents, and Document Contexts(C-D-Dc)

where A and B represent feature weight matrices for *Concepts-Documents* and *Documents-Document Contexts*, in which the A_{ij} equals the edge-weight E_{ij} and B_{jk} equals the edge-weight E_{jk} respectively. The adjacency matrix M of the tripartite graph could be written as:

$$M = \begin{array}{c|ccc} & C & D & Dc \\ C & 0 & A & 0 \\ D & A^T & 0 & B \\ Dc & 0 & B^T & 0 \end{array} \tag{7}$$

where the vertices been ordered in such a way that the first i vertices index the *concepts* while the next j vertices index the *documents* and the last k vertices index the *document contexts*. Every entry in A and B represents the importance of particular *concepts* to *document* and *document contexts* respectively. To co-cluster C, D, and Dc simultaneously, it seems common to partition the graph in Figure 3(b) by solving the generalized eigenvalue problem corresponding to the adjacency matrix. In parallel, we found that this solution does not work, as it seems. Originally, if we move the vertices of *concepts* in Figure 3(b) to the side of the vertices of *document contexts*, then the drawn tripartite graph would turn to be a partite graph. As our work is related on (*documents*, *concepts*, and *document contexts*) bipartite graph and the loss incurred on cutting an edge between a *document* and a *concept*, which contributes the loss function similar to the loss of cutting an edge between a *document* and *document context*. In addition, these two kinds of edges are heterogeneous and incomparable. To deal with such problems, Gao et al. [9] have treated tripartite graph as two-bipartite graphs, where each graph share the central nodes. Therefore, we convert the actual problem by combining the pair-wise problems over these two bipartite graphs. During our implementation of bipartite graph (as in Figure 3), will be generated from the tripartite. In the mean time, if we transfer bipartite spectral graph partitioning [17] on Figure 3(a) and Figure 3(b) individually, that provides us with a maximum probability that the partitioning schemes for documents are different in the two solutions (i.e., *not locally optimal*). To solve this optimization problem, this work [9] has suggested a *consistent bipartite graph co-partitioning* (CBGC), which provides an optimization algorithm using positive semi-definite programming. Our work only focuses on bi-partitioning the *concepts*, *documents* and the *document contexts* into two groups simultaneously. To achieve this clustering, we allow {*C, D, Dc*} to perform as the column vectors of *i, j, k* dimensions for *concept, document* and *document context* respectively. Let $p = (C, D)^T$ and $q = (D, Dc)^T$ as the indicating vectors for the two bipartite graphs and $D^{(c)}$, $D^{(Dc)}$, $L^{(c)}$ and $L^{(Dc)}$ as the diagonal matrices and Laplacian matrices for the adjacency matrices *A* and *B*. We equate the consistent co-partitioning problem with multi-objective optimization, which is defined as:

$$\underset{p\neq 0}{\text{minimize}} \frac{p^T L^{Dc} p}{p^T D^{Dc} p} \mid p^T D^{Dc} e = 0, \tag{8}$$

and

$$\underset{q\neq 0}{\text{minimize}} \frac{q^T L^{C} q}{q^T D^{C} q} \mid q^T D^{C} e = 0, \tag{9}$$

where *e* is the column vector, which is equal to 1. To solve the multi-objective optimization problem, this work [17] provides a complete detail on the ways of using the positive semi-definite programming as an efficient solution to the optimization problem. We utilize their algorithm to solve the co-clustering of *concepts*, *document* and *context* (*C-D-Dc*) and compared with *terms*, *documents* and *category* (*T-D-Cat*). Partitioning this graph lets us achieve blog clustering by integrating information from *concepts* and *document context* synchronously.

4 Experimental Evaluation

4.1 TREC Dataset

In this section, to evaluate our CSSGP framework, we used part of TREC BLOGs08[1] dataset. TREC dataset is a well-known dataset in blog mining research area. This dataset consists of the crawl of Feeds, associated Permalink, blog homepage, and blogposts and blog comments. The BLOGs08 corpus is a sample of blogosphere crawled over from 14 January 2008 to 10 February 2009. The crawled feeds are mostly from *BlogSpot, Live-Journal, Xanga,* and *MSN Spaces*. The blog permalinks and homepages are encoded using HTML. This dataset is a combination of both English and Non-English blogs.

4.2 Experiment Dataset Preparation

As our research focus is only on blog clustering and not on language identifier, we filtered the blogs by English language. In order to extract the blogposts from blog

[1] http://ir.dcs.gla.ac.uk/test_collections/blogs08info.html

documents, the HTML source code of blog pages should be parsed (which includes removal of HTML tags, scripts, etc.), and to be converted into plain text. We used blog-post extraction program named *BlogTEX*[2] to extracts blog posts from TREC Blog dataset. We utilized the list of stop words in our program to identify the English blogs**.** During preprocessing, we removed HTML tags and used only blogposts for our experiment. Furthermore, as the dataset is huge and our motive is to test our tripartite clustering approach, we collected around 41,178 blogposts and assigned based on each category. Table 1 below shows the *CategoryTree* structure of Wikipedia and the cluster sizes. Our expected clustering result should be the same as Table 1. We built, *Document context* (*Dc)* by constructing a vector with each dimension representing the content similarity between the blogpost, and any blogpost in the dataset. We consider leftover words as textual representation of blogposts. The entire document collections represented as *word* × *document matrix*, where rows corresponds to *words* and columns to *documents*. We carried out concept indexing, using *conf.idf*, 1,55,980 *concepts* and term indexing by using *tf.idf* [15], 2,83,126 *terms* is reserved for our experiment.

Table 1. The Main *CategoryTree* Structure of Wikipedia and the Cluster Size

Categorical Index	Sub-Categories	# of documents	# of concepts	Cluster
General Reference	Almanacs • Atlases … more	878	3758	C_1
Culture and Arts	Baseball • Basketball •……	8580	18245	C_2
Geography and Places	Cities • Continents • …more	4558	23220	C_3
Health and Fitness	Health promotion • health ….	4082	21409	C_4
History and Events	Africa • Asia• ….. more	2471	12746	C_5
Mathematics and Logic	Algebra • Analysis ….. more	213	2561	C_6
Natural and Physical sciences	Astronomy • Chemistry….	1560	8530	C_7
People and Self	Children • Political people•…	6279	10125	C_8
Philosophy and Thinking	Theories • Aesthetics•….	2345	7591	C_9
Religion and Belief	Allah • Bible • Buddha•….	2345	8315	C_{10}
Society and Social sciences	Business • Family • …. More	4657	13170	C_{11}
Tech and Applied sciences	Bioengineering • Chemical …	3210	26310	C_{12}

4.3 Evaluation Results

Our CSSGP framework have been evaluated by comparing clustering output with true class labels (Wikipedia categories). There are number of comparison metrics and we use cluster *purity* and *entropy*. The clustering metric, *purity* measures the extend to which each cluster contained documents from primarily one class. The second clustering metric, *entropy* measures the various classes of documents that are distributed within each cluster. The value for the relative *entropies* and *purities* metrics are shown in Table 2. The higher purity $C_2$and lower entropy C_4 values are

[2] http://sourceforge.net/projects/blogtex

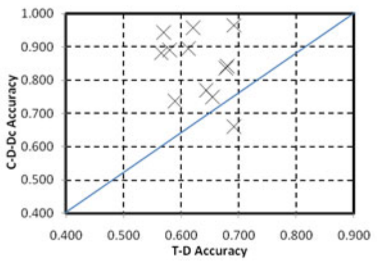
highlighted in the above Table 2. As most of the documents have discussed on *culture and arts*, *health and fitness*, we got higher purity and lower entropy during clustering.

Table 2. Results from Clustering evaluating metrics *Purity* and *Entropy*

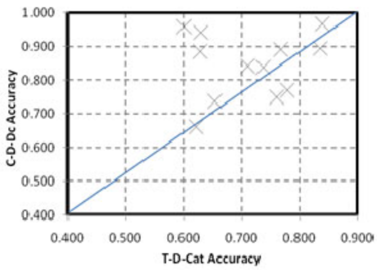
	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}
Purity	0.843	**0.881**	0.835	0.859	0.455	0.668	0.856	0.753	0.817	0.795	0.720	0.813
Entropy	0.285	0.259	0.223	**0.210**	0.675	0.419	0.265	0.299	0.238	0.329	0.368	0.233

4.4 Performance Comparison

In this section, we compared the clustering performance for all 12 possible categories with existing clustering technique. As blog document clustering research is in its nascent stage, there is no current benchmark to prove our work. To test the performance of our tripartite spectral graph clustering (*C-D-Dc*), we compared with the existing technique, *T-D* bipartite spectral graph clustering, and *T-D-Cat* tripartite spectral graph clustering. *T-D* bipartite graph clustering [18] consists of raw terms collected from the documents. *T-D-Cat* tripartite spectral graph clustering [9] consists of raw terms, documents and the Wikipedia categories. We plot two-dimensional graph to compare performance between *C-D-Dc* tripartite spectral graph clustering accuracy and *T-D* bipartite spectral graph clustering accuracy in Figure 5.



(a)



(b)

Fig. 5. Performance Comparison between (a) ***C-D-Dc*** vs. ***T-D***, (b) ***C-D-Dc*** vs. ***T-D-Cat*** clustering

C-D-Dc Accuracy is plotted in the X-axis and *T-D* Accuracy is plotted in the Y-axis. Each dot in the graph represents a possible category pair. We see that most of the dots are in the *C-D-Dc* clustering accuracy side, which indicates that *C-D-Dc* clustering performs better than the *T-D* clustering. This graph interprets that tripartite spectral graph clustering using *document context* (*Dc*) data gives better clustering. As similar to the above comparison, we plot two-dimensional graph to compare tripartite spectral graph clustering performance between *C-D-Dc* and *T-D-Cat* clustering accuracy in Figure 5. *C-D-Dc* Accuracy is plotted in the X-axis and *T-D-Cat* Accuracy is plotted in the Y-axis. Each dot in the graph represents a possible category pair. We see that most of the dots are in the *C-D-Dc* clustering accuracy side, which indicates that *C-D-Dc* clustering performs better than the *T-D-Cat* clustering. This graph interprets that tripartite spectral graph clustering using *document context* (*Dc*) data gives better clustering. To summarize this performance comparison, *C-D-Dc* tripartite spectral graph clustering provides better performance than the other two clustering *T-D*, and *T-D-Cat* clustering.

4.5 Average Performance

To measure the average performance, we used tripartite graph clustering results using *C-D-Dc* with the *T-D-Cat* and bipartite graph clustering using *T-D*. The average performances between each clustering and all other 12 categories are shown in Table 3. We highlighted the exceeding values. Here *Terms and Documents (T-D) clustering* follows conventional bipartite spectral graph clustering. Our three clustering results are compared with the expected result, which is mentioned in Table 3.

Table 3. The Average Performance Result among *T-D, T-D-Cat* and *C-D-Dc* clustering

Category Name	*T-D*	*T-D-Cat*	*C-D-Dc*
General Reference	0.590	0.655	0.736
Culture and Arts	0.622	0.660	0.692
Geography and Places	0.581	0.768	**0.890**
Health and Fitness	0.692	**0.840**	**0.966**
History and Events	0.655	0.761	0.749
Mathematics and Logic	0.681	0.711	**0.841**
Natural and Physical sciences	0.565	0.629	**0.883**
People and Self	0.678	0.738	**0.836**
Philosophy and Thinking	0.621	0.601	**0.957**
Religion and Belief	0.643	0.750	0.768
Society and Social sciences	0.570	0.630	**0.942**
Tech and Applied sciences	0.613	**0.837**	**0.895**
Average performance	**0.626**	**0.715**	**0.846**

Out of 12-main categories in Table 3, 8-categories-*Health and Fitness*, *Philosophy and Thinking*, *Society and Social sciences*, *Tech and Applied sciences*, *Geography and Places*, *Natural and Physical sciences*, *Mathematics and Logic* and *People and Self* shown better performance on *C-D-Dc* graph clustering. The *T-D-Cat* clustering shown good performance on categories such as *Health and Fitness,* and *Tech and Applied sciences*. This implies that our proposed method has 84% success rate and achieved better performance, compared to the other two clustering, *T-D*, and *T-D-Cat*.

5 Conclusion and Future Work

In this paper, we developed a CSSGP framework that represents concepts, documents and document contexts as a tripartite graph structure and demonstrated the effectiveness of document context as a significant feature that improves the document clustering results. The primary contributions of this paper are the following: First by using Wikipedia, we perform an explicit semantic based topic analysis by converting terms to concepts. We calculated the surrounding text similarity using *conf.idf*. Second we used Content Similarity and surrounding text similarity to calculate Document Context Similarity. We used *document context, concepts,* and *documen*ts to produce a tripartite spectral graph, which helps for efficient document clustering. As shown in Table 3, our experiment results indicate that our framework enhances cluster quality than the traditional document clustering approaches. In addition, our proposed solution blends well with real examples. In the future, we plan to expand our experiment using the full

TREC Blogs08 dataset to measure the clustering performance and its scalability. We plan to include other blog features such as comments and tags, as one of the data for tripartite graph clustering and measure its performance for document clustering.

References

1. Ounis, I., Macdonald, C., Soboroff, I.: On the TREC BlogTrack. In: ICWSM, USA (2008)
2. Halkidi, M., Batistakis, Y., Vazirgiannis, M.: On Clustering Validation Techniques. J. of Intelligent Information System (2001)
3. Berkhin, P.: Survey of clustering data mining techniques. Accrue Software Inc., Technical report (2002)
4. Xu, R., Wunsch II, D.: Survey of clustering algorithms. IEEE Trans. Neural Netw. 16(3), 645–678 (2005)
5. Gabrilovich, E., Markovitch, S.: Overcoming the brittleness bottleneck using Wikipedia: Enhancing text categorization with encyclopedic knowledge. In: AAAI (2006)
6. Huang, A., Milne, D., Frank, E., Witten, I.: Clustering documents using a Wikipedia-based concept representation. In: PAKDD, pp. 628–636 (2009)
7. Hu, J., Fang, L., Cao, Y., Hua-Jun Zeng, H., Li, H.: Enhancing Text Clustering by Leveraging Wikipedia Semantics. In: ACM SIGIR, pp. 179–186 (2008)
8. Yoo, I., Hu, X., Song, I.Y.: Integration of semantic-based bipartite graph representation and mutual refinement strategy for biomedical literature clustering. In: KDD (2006)
9. Gao, B., Liu, T., Zheng, X., Cheng, Q., Ma, W.: Consistent Bipartite Graph Co-Partitioning for Star-Structured High-Order Heterogeneous Data Co-Clustering. In: SIGKDD (2005)
10. Xu, W., Liu, X.: Gong. Y.: Document clustering based on nonnegative matrix factorization. In: SIGIR 2003, pp. 267–273 (2003)
11. Baker, L., McCallum, A.: Distributional Clustering of Words for Text Classification. In: ACM SIGIR, pp. 96–103 (1998)
12. von Luxburg, U.: A tutorial on Spectral Clustering. In: MPI-Technical Reports No.149. Tubingen: Max Planck Institute for Biological Cybernetics
13. Dhillon, I., Guan, Y., Kulis, B.: Kernel k-Means, Spectral Clustering and Normalized Cuts. In: KDD, pp. 551–556 (2004)
14. Ayyasamy, R.K., Tahayna, B., Alhashmi, S.M., Siew, E., Egerton, S.: Mining Wikipedia knowledge to improve document indexing and classification. In: 10th International Conference on Information Science, Signal Processing and their Applications, ISSPA 2010, pp. 806–809 (2010)
15. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. J. Information Processing & Management 24, 513–523 (1988)
16. Strehl, A., Ghosh, J., Mooney, R.: Impact of similarity measures on web-page clustering. In: AAAI Workshop on AI for Web Search, pp. 58–64 (2000)
17. Dhillon, I.: Co-clustering documents and words using bipartite spectral graph partitioning. In: ACM SIGKDD, pp. 269–274 (2001)
18. Sun, A., Suryanto, M.A., Liu, Y.: Blog Classification Using Tags: An Empirical Study. In: Goh, D.H.-L., Cao, T.H., Sølvberg, I.T., Rasmussen, E. (eds.) ICADL 2007. LNCS, vol. 4822, pp. 307–316. Springer, Heidelberg (2007)
19. Tahayna, B., Ayyasamy, R.K., Alhashmi, S.M., Siew, E.: A Novel Weighting Scheme for Efficient Document Indexing and Classification. In: 4th International Symposium on Information Technology, ITSIM 2010, pp. 783–788 (2010)
20. Rui, X., Li, M., Li, Z., Ma, W.Y., Yu, N.: Bipartite graph reinforcement model for web image annotation. In: Multimedia 2007 (2007)
21. Zhang, D.Q., Lin, C.Y., Chang, S.F., Smith, J.R.: Semantic Video Clustering Across Sources Using Bipartitie Spectral Clustering. In: ICME (2004)
22. Zha, H., Ding, C., Gu, M.: Bipartite graph partitioning and data clustering. In: CIKM (2001)