Concept Based Query Expansion and Cluster Based Feature Selection for Information Retrieval

Chellatamilan T [1], Dr. Suresh R. M [2]

[1.] Department of CSE, Arunai Engineering College, Tiruvannamalai, India
[2.] Principal, Sri Muthukumaran Institute of Technology, Chennai 600069, India
chellatamilan_t@yahoo.com, rmsuresh@hotmail.com

Abstract: With the advent of internet technology as a ubiquitous platform for sharing the educational contents and experiences, many of the institutions across the globe offer the federated search to the courses, lesson plans, contents, assignments, seminars and experiments. These learning resources are stored in the repositories of the learning content management system. Sophisticated search and information retrieval solutions are essential for efficient use of these repositories. The structure of many existing information retrieval system considers ontology for retrieval. This ontology based solution increases the accuracy of information retrieval through high precision and recall. This paper addresses the requirement for pre-processing and classification of documents in order to achieve more efficient Information Retrieval system. Tools and techniques employed for autonomous classification or clustering of documents are investigated and a new method based on concept expansion is proposed. The proposed methods are evaluated using Reuters 21578 dataset.
[Chellatamilan T., Suresh R. M. **Concept Based Query Expansion and Cluster Based Feature Selection for Information Retrieval.** *Life Sci J* 2013; 10(7s): 661-667] (ISSN:1097-8135). http://www.lifesciencesite.com. 104

Keywords: Information Retrieval, Query Expansion, language modeling, query likelihood retrieval method, Reuters, Inverse Document Frequency

1. Introduction

Information retrieval models are a process of producing ranking functions [1] that assigns score to documents with regard to given query. Every such process consists of two main tasks.

1) The task which represents the documents and query.

2) The task which computes a rank of each documents.

IR system initially builds the index terms to index and to retrieve the documents [2]. In general most of the index terms are simply any keyword that appears in the text document collections. Most of the users have no opinion or training in forming the query to retrieve the results. In order to retrieve the answers to a query, the IR system predicts the documents the users will find relevant and irrelevant. The predicting function is called as ranking algorithm to establish simple ordering of the documents that were retrieved.

The language modeling approach to retrieval has been shown to perform well empirically. One advantage of this new approach is its statistical foundations. The language modeling approach to text retrieval was first introduced by Ponte and Croft [3] and later explored in [4, 5, 6]. The relative simplicity and effectiveness of the language modeling approach, together with the fact that it leverages statistical methods that have been developed in speech recognition and other areas, make it an attractive framework in which to develop new text retrieval methodology.

Simple language models have been shown to incorporate document and collection statistics in a more systematic way than earlier tf.idf based techniques [7, 8, 9]. Language models work as well as the classical models using tf.idf, but further improvements are likely to require a broad range of techniques in addition to language modelling [10, 11, 12]. The essence of the language modeling approach, which is shared with more classical probabilistic approaches to information retrieval, is that probabilistic modeling is taken to be the primary scientific tool. At present, this appears to be the most promising framework for advancing information retrieval to meet future challenges presented by more diverse data sources and advanced retrieval tasks.

The query likelihood retrieval method [3] has enjoyed much success for many different retrieval tasks [13, 14]. The query likelihood retrieval method [3] assumes that a query is a sample drawn from a language model: given a query Q and a document D, we compute the likelihood of "generating" query Q with a model estimated based on document D. We can then rank documents based on the likelihood of generating the query.

This paper addresses the requirement for pre-processing and classification of documents based on ontology in order to achieve more efficient Information Retrieval system. Tools and techniques employed for autonomous classification or clustering of documents are investigated and a new method based on concept expansion is proposed. The

proposed methods are evaluated using Reuters 21578 dataset.

2. Material and Methods

The Reuters-21578 Text Categorization Test Collection is a standard text categorization benchmark [15]. It contains 21578 Reuters news documents from 1987. They were labeled manually by Reuters personnel. Labels belong to 5 different category classes, such as 'people', 'places' and 'topics'. The total number of categories is 672, but many of them occur only very rarely. The Reuters-21578 data set is a commonly used collection of newswire stories categorized into hand-labeled topics. Each news story has been hand-labeled with some number of topic labels such as "corn", "wheat" and "corporate acquisitions". Note that some of the topics overlap and so some articles belong to more than one category. We used the 12902 articles from the "ModApte" split of the data5 and, to stay comparable with previous studies, we considered the top ten most frequently occurring topics. The Reuters collection is distributed in 22 files. Each file begins with a document type declaration line:

<DOCTYPE lewis SYSTEM "lewis.dtd">

Each article starts with an "open tag" of the form

<REUTERS TOPICS=?? LEWISSPLIT=?? CGISPLIT=?? OLDID=?? NEWID=??>

where the ?? are filled in an appropriate fashion. Each article ends with a "close tag" of the form:

</REUTERS>

Each REUTERS tag contains explicit specifications of the values of five attributes: TOP-ICS, LEWISSPLIT, CGISPLIT, OLDID, and NEWID. These attributes are meant to identify documents and groups of documents. The values of the attributes determine how the documents are divided into a training set and a test set. In the experiments described in this work, we used the modified Apte split, which is the one that is most used in the literature.

Each document was represented as a stemmed, TFIDF-weighted word frequency vector. Each vector had unit modulus. A stop list of common words was used and words occurring in fewer than three documents were also ignored.

Inverse document frequency (IDF) is a popular measure of word's importance [16]. The IDF invariably appears in a host of heuristic measures used in information retrieval .However, so far the IDF has itself been a heuristic .It is a popular measure of a word's importance. It is defines as the logarithm of the ratio of number of documents containing the given word. This means rare words have high IDF and common function words like "the" will have low IDF. IDF is believed to measure a word's ability to discriminate between documents [17]. Text Classification involves assigning a text document to a set of pre-defined classes automatically, using a machine learning technique [18, 19]. The classification is usually done on the basis of significant words or key-features of the text document. Since the classes are pre-defined it is a supervised machine learning task.

Inverse Document Frequency **(IDF)** represents scaling factor. When a term a occurs frequently in many documents, its importance is then scaled down because of its lowered discriminative power. The $IDF(a)$ is defined as follows:

$$IDF(a) = \log \frac{1+|x|}{x_a}$$

x_a is the set of documents containing term a.

A term or phrase may have multiple meanings, while a domain specific concept is unambiguous [20]. It is more useful to use the domain specific concepts present in documents than the terms for retrieving documents belonging to a particular domain [21]. Therefore, we extract the list of concepts present in documents and annotate them with the list of concepts. For this, we need to disambiguate the meaning of a term and identify the concept it refers to. In some cases more than one term may refer to the same concept. In such cases the frequency of a concept will include the frequencies of all synonymous terms for the concept in the document.

For each term, the associated set of concepts is obtained from the ontology. A term can map to one or more number of concepts. Out of these mapped concepts, we want to find the most appropriate concept for a particular domain. To identify the correct concept, we look at the occurrences of the related concepts. We use the inter concept relationship which is captured in our ontology. A concept is more significant if more number of related concepts of that term occur in the document. The proposed algorithm takes a list of terms from the document along with their frequency as input, and returns a list of concepts along with their significance with respect to the document.

The algorithm works as follows. For each term t_i in the term list of a document D, the associated concepts c_{ij} are obtained from the ontology. Let the significance of each associated concept c_{ij} be $c_{ij}.significance$. The significance $c_{ij}.significance$ is initially taken as the normalized frequency of the term t_i i.e. $t_i.frequency$. For each associated concept c_{ij}, we look at the presence of the related concepts rc_p in the document. We then increment the significance of the associated concept c_{ij} by α* normalized term

frequency for the occurrences of the terms t_p corresponding to the concept rc_p.

Significance $c_{ij} = t_i.frequency + \alpha * t_p.frequency$

Where α is the weight given to the related concepts. In our experiment, we have taken α = ½.

For a particular term, we choose a concept with maximum significance value.

The WordNet is a lingual database for the link language English. WordNet is termed as the abounding lexical database for English that constitutes the group of nouns, verbs, adjectives and adverbs called synsets. Synsets are contrived on conceptual semantic and lingual relations. Corpus with proposed concept expansion using wordnet is formed.

A similarity thesaurus is a matrix that consists of term-term similarities. In contrast to a co-occurrence matrix, a similarity thesaurus is based on how the terms of the collection "are indexed" by the documents. A similarity thesaurus can be constructed automatically by using an arbitrary retrieval method with the roles of documents and terms interchanged. In other words, the terms play the role of the retrievable items and the documents constitute the "indexing features" of the terms.

With this arrangement a term t_i is represented by a vector $t_i = (d_{i1}, d_{i2}, ..., d_{in})^T$ in the document vector space (DVS) defined by all the documents of the collection. The d_{ik}'s signify feature weights of the indexing features (documents) d_k with respect to the item (term) t_i and n is the number of features (documents) in the collection. Normalized tf . idf weighting scheme is adopted [22] and define the feature weights d_{ik} by the feature frequency (ff), the inverse item frequency (iif), and the maximum feature frequency (maxff) as follows.

$$d_{ik} = \frac{(0.5+0.5\,\frac{ff(d_k,t_i)}{maxff(t_i)})\cdot iif(d_k)}{\sqrt{\sum_{j=1}^{n}((0.5+0.5\,\frac{ff(d_j,t_i)}{maxff(t_i)})\cdot iif(d_j))^2}}$$

where $ff(d_k,t_i)$ is the within-item frequency of feature d_k in item t_i .

iif(dk) = log(m/|dk|) is the inverse item frequency of feature d_k

m is the number of items in the collection and |dk| is the number of different items indexed by the feature d_k.

In other words, |dk| is the number of terms appearing in document dk. $maxff(t_i)$ is the maximum within-item frequency of all features in item t_i.

The feature frequency $ff(d_k,t_i)$ specifies the number of occurrences of the indexing feature d_k in item t_i . It is analogous to the term frequency $tf(t_i ,d_k)$ when the documents are indexed by terms. The definition of the inverse item frequency shows that a short document plays a more important role than a long document. If two terms co-occur in a long document, the probability that the two terms are similar is smaller than if they would co-occur in a short document as follows:

$$|\vec{t}_i| = \sqrt{\sum_{k=1}^{n} d_{ik}^2} = 1$$

This means that t_i is a unit vector representing the term in the document vector space DVS. With these definitions, we define the similarity between two terms t_i and t_j by using a similarity measure such as the simple scalar vector product:

$$SIM(t_i,t_j) = \vec{t}_i^{\,T}\cdot\vec{t}_j = \sum_{k=1}^{n} d_{ik}\cdot d_{jk}$$

The similarity thesaurus is constructed by determining the similarities of all the term pairs (t_i ,t_j). The result is a symmetric matrix whose values are in the following range:

$$0 \le SIM(t_i, t_j) \le 1$$

A query q is represented by a vector q = (q1, q2, ..., qm)^T in the term vector space (TVS) defined by all the terms of the collection. Here, the q_i's are the weights of the search terms t_i contained in the query q; m is the total number of terms in the collection.

The probability that a term t is similar to the concept of query q is P(S|q,t). In order to estimate the probability, Bayes' theorem is applied:

$$P(S|q,t) = P(S|t)\cdot\frac{P(q|S,t)}{P(q|t)} = \frac{P(S|t)}{P(q|t)}\cdot P(q|S,t)$$

It is assumed that the distribution of terms in all the queries to which a term is similar is independent:

$$P(S|q,t) = \frac{P(S|t)}{P(q|t)}\cdot\prod_{i=1}^{m} P(q_i|S,t)$$

$$= \frac{P(S|t)}{P(q|t)}\cdot\prod_{i=1}^{m}\frac{P(S|q_i,t)}{P(S|t)}\cdot P(q_i|t)$$

$$= \frac{1}{P(q|t)\cdot P(S|t)^{m-1}}\cdot\prod_{i=1}^{m} P(S|q_i,t)\cdot P(q_i|t)$$

An additional assumption is that the similarity between a term and the concept of a query depends only on the terms contained in the query and not on other terms. Hence,

$$P(S|q,t) = \frac{1}{P(q|t)\cdot P(S|t)^{m-1}}\cdot\prod_{t_i\in q} P(S|t_i,t)\cdot P(t_i|t)$$

Here, $P(S|t_i,t)$ is the probability that the query term t_i is similar to the term t. $P(t_i|t)$ is the probability that the query term ti represents the query q. P(q|t) is the probability that the query q will be submitted to the IR system. P(S|t) is the probability that the term t is similar to an arbitrary query.

The probability of a term to be similar to a query depends on the following factors:

- The similarities between the term and all the query terms;
- The weights of the query terms.

As mentioned above, the objective of our query expansion scheme is to find suitable additional query terms. They should have the property of being similar to the entire query rather than to individual query terms. We showed that such terms can only be found when an overall similarity scheme is taken into account. Since the similarity thesaurus expresses the similarity between the terms of the collection in the DVS (defined by the documents of the collection), we map the vector q from the TVS (defined by the terms of the collection) into a vector in space DVS. This way, the overall similarity between a term and the query can be estimated. Each query term t_i is defined by the unit vector t_i which itself is defined by a number of documents. q_i is the weight of term t_i in the query. In other words, the concept expressed by the term t_i in the query has an importance of q_i .t_i for the query. We assume that the concept expressed by the entire query depends only on the terms in the query. Therefore, the vector q_c representing the query concept in space DVS is the virtual term vector:

$$\vec{q}_c = \sum_{t_i \in q} q_i \cdot \vec{t}_i$$

The similarity between a term and the query q is denoted by Simqt(q,t). The scalar vector product is used as similarity measure:

$$Simqt(q,t) = \vec{q}_c^{\,T} \cdot \vec{t} = \left(\sum_{t_i \in q} q_i \cdot \vec{t}_i \right)^T \cdot \vec{t}$$

$$= \sum_{t_i \in q} q_i \cdot (\vec{t}_i^{\,T} \cdot \vec{t})$$

where $(t_i^T.t)$ is the similarity between two terms.

Data clustering is one of the most popular data labeling techniques. In data clustering, we are given unlabeled data and we are to put similar samples in one pile, called a cluster, and the dissimilar samples should be in different clusters. Usually, neither cluster's description nor its quantification is given in advance unless a domain knowledge exists, which poses a great challenge in data clustering.

Clustering is useful in several machine learning and data mining tasks including: image segmentation, information retrieval, pattern recognition, pattern classification, network analysis, and so on. It can be seen as either an exploratory task or preprocessing step. If the goal is to explore and reveal the hidden patterns in the data, clustering becomes a standalone exploratory task by itself. There are many clustering methods in the literature. These methods can be categorized broadly into: partitioning methods, hierarchical methods, and density-based methods. The partitioning methods use a distance-based metric to cluster the points based on their similarity [6].

Clustering is an important unsupervised classification technique. When used on a set of objects, it helps identify some inherent structures present in the objects by classifying them into subsets that have some meaning in the context of a particular problem. More specifically, objects with attributes that characterize them, usually represented as vectors in a multi-dimensional space, are grouped into some clusters. When the number of clusters, K, is known a priori, clustering may be formulated as distribution of n objects in N dimensional space among K groups in such a way that objects in the same cluster are more similar in some sense than those in different clusters. This involves minimization of some extrinsic optimization criterion.

The K-means algorithm, starting with k arbitrary cluster centers, partitions a set of objects into k subsets and is one of the most popular and widely used clustering techniques because it is easy to implement and very efficient, with linear time complexity [23]. However, the K-means algorithm suffers from several drawbacks. The objective function of the K-means is not convex and hence it may contain many local minima. Consequently, in the process of minimizing the objective function, there exists a possibility of getting stuck at local minima, as well as at local maxima and saddle points [24]. The outcome of the K-means algorithm, therefore, heavily depends on the initial choice of the cluster centers.

Data clustering, which is an NP-complete problem of finding groups in heterogeneous data by minimizing some measure of dissimilarity, is one of the fundamental tools in data mining, machine learning and pattern classification solutions [25]. Clustering in N-dimensional Euclidean space RN is the process of partitioning a given set of n points into a number, say k, of groups (or, clusters) based on some similarity (distance) metric in clustering procedure is Euclidean distance, which derived from the Minkowski metric.

$$d(x,y) = \left(\sum_{i=1}^{m} \left| x_i - y_j \right|^r \right)^{1/r}$$

$$d(x,y) = \sqrt{\sum_{i=1}^{m} (x_i - y_j)^2}$$

A popular performance function for measuring goodness of the k clustering is the total within cluster variance or the total mean-square quantization error (MSE), [26]

$$Per\,f(X,C) = \sum_{i=1}^{N} Min\{\|X_i - C_l\|^2 \mid l = 1,..K\}$$

A genetic algorithm-based clustering technique, called GA-clustering. The searching capability of genetic algorithms is exploited in order to search for appropriate cluster centres in the feature space such that a similarity metric of the resulting clusters is optimized. The chromosomes, which are represented as strings of real numbers, encode the centres of a fixed number of clusters. The superiority of the GA-clustering algorithm over the commonly used *K*-means algorithm is extensively demonstrated for four artificial and three real-life data sets.

3. Results

The Reuters dataset is used for evaluating the proposed methods. The experiments are conducted as detailed in the previous chapter, with the inclusion of proposed concept query expansion method. Precision and recall values for various techniques for dataset are evaluated. The techniques used were tdf.idf, Language modelling using query likelihood, proposed concept expansion with feature selection method. The experimental results for Reuters 21758 dataset for precision and recall and F measure are tabulated in Table 1 and Table 1 respectively. Figure 1 and 2 show the same.

Table 1: Precision values for various techniques for Reuters 21758 dataset

Recall	TDF-IDF	Language modeling using Query likelihood	Concept expansion
0.01	0.831644106	0.875672961	0.89949308
0.1	0.769754052	0.827139846	0.851704269
0.2	0.748400666	0.782014571	0.773306475
0.3	0.748902623	0.755021124	0.764249838
0.4	0.689139901	0.711668277	0.734923375
0.5	0.6774131	0.709199407	0.724206713
0.6	0.627466661	0.700349971	0.732498213
0.7	0.592573475	0.682966143	0.69528761
0.8	0.551441906	0.651598754	0.667393946
0.9	0.543356262	0.582080004	0.620455421
1	0.310100304	0.492848736	0.530449696

Table 2: Average F measure values for various techniques for Reuters 21758 dataset

	TDF-IDF	Language modeling using Query likelihood	Concept expansion
F measure	0.563152052	0.585548757	0.592410495

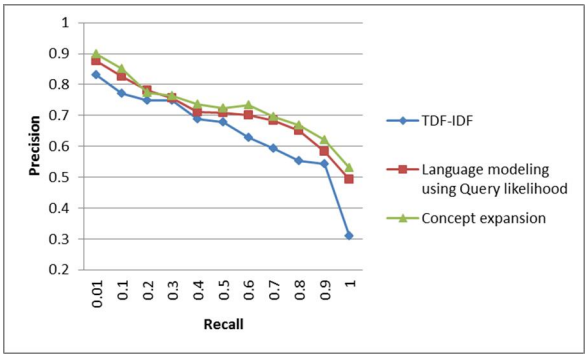

Figure 1: Precision values for various techniques for Reuters 21758 dataset

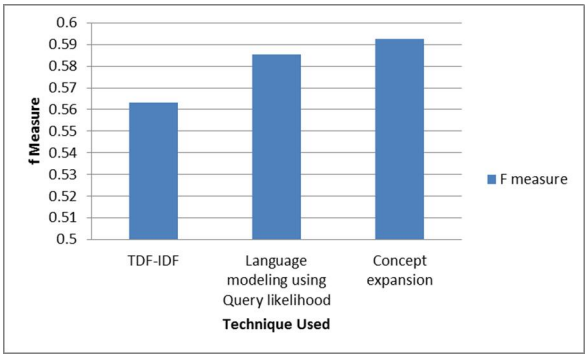

Figure 2: Average F measure values for various techniques for Reuters 21758 dataset

In the second set of experiments, additional terms with query are used. Experiments are conducted for 5, 10, 15, 20, 25 additional terms. Precision and recall values for various techniques for Reuters dataset is evaluated. The techniques used were tdf.idf, Language modelling using query likelihood, and proposed concept expansion with cluster based feature selection method. The experimental results of percentage improvement for Reuters dataset for precision and recall are tabulated in Table 3 and Table 3 respectively.

Table 3: Percentage Improvement in Precision for Reuters dataset

Additional terms	Percentage improvement		
	TDF-IDF	Language modeling using Query likelihood	Concept expansion
5	2.8	2.74	2.6
10	5.275	5.12	5.18
15	6.48	6.39	6.16
20	4.68	4.32	4.78
25	4.2	4.57	4.18

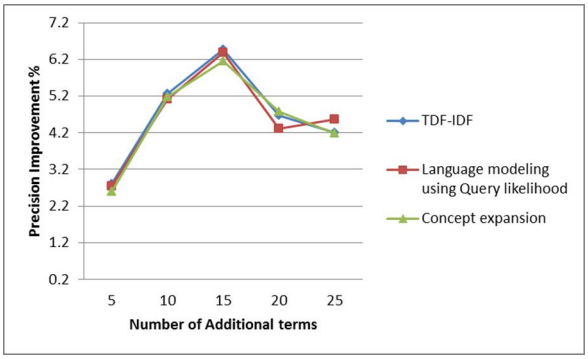

Figure 3: Percentage Improvement in Precision for Reuters dataset

4. Conclusion

In this study, a probabilistic query expansion model is presented based on a similarity thesaurus which was constructed automatically. A similarity thesaurus reflects domain knowledge about the particular collection from which it is constructed. The two important issues with query expansion are addressed: the selection and the weighting of additional search terms. In contrast to earlier methods, in the proposed method queries are expanded by adding those terms that are most similar to the concept of the query, rather than selecting terms that are similar to the query terms. Experiments are conducted for varying number of additional terms (5, 10, 15, 20, 25). Experimental results demonstrate the superiority of the proposed concept based query expansion method with respect to the precision. It is also observed that 15 additional terms achieve the maximum precision.

References

1. Ruthven, I., & Lalmas, M. (2002). Using Dempster-Shafer's theory of evidence to combine aspects of information use. Journal of Intelligent Information Systems, 19(3), 267-301.
2. Yan, H., Ding, S., & Suel, T. (2009, April). Inverted index compression and query processing with optimized document ordering. In Proceedings of the 18th international conference on World wide web (pp. 401-410). ACM.
3. Ponte, J. M., & Croft, W. B. (1998, August). A language modeling approach to information retrieval. In Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval (pp. 275-281). ACM.
4. F. Song and B. Croft. A general language model for information retrieval. In 22nd ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99), pages 279-280, 1999. Allan et al 2002,
5. Balog, K., Azzopardi, L., & de Rijke, M. (2009). A language modeling framework for expert finding. Information Processing & Management, 45(1), 1-19. [Zhai et al 2001]
6. Alelyani, S., Tang, J., & Liu, H. (2013). Feature Selection for Clustering: A Review.
7. Lafferty, J., & Zhai, C. (2001, September). Document language models, query models, and risk minimization for information retrieval. In Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 111-119). ACM.
8. Kurland, O., & Lee, L. (2004, July). Corpus structure, language models, and ad hoc information retrieval. In Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 194-201). ACM.
9. Abdulmutalib, N., & Fuhr, N. (2010). Language models, smoothing, and idf weighting. Proc. of the" Information Retrieval, 169-174.
10. Xia, T., & Du, Y. (2011, August). Improve VSM text classification by title vector based document representation method. In Computer Science & Education (ICCSE), 2011 6th International Conference on (pp. 210-213). IEEE.
11. Meij, E., Trieschnigg, D., De Rijke, M., & Kraaij, W. (2010). Conceptual language models for domain-specific retrieval. Information Processing & Management, 46(4), 448-469.
12. Jimeno-Yepes, A., Berlanga-Llavori, R., & Rebholz-Schuhmann, D. (2010). Ontology refinement for improved information retrieval.

Information Processing & Management, 46(4), 426-435.
13. Zhai, C., & Lafferty, J. (2001, September). A study of smoothing methods for language models applied to ad hoc information retrieval. In Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 334-342). ACM.
14. Lv, Y., & Zhai, C. (2009, November). A comparative study of methods for estimating query language models with pseudo feedback. In Proceedings of the 18th ACM conference on Information and knowledge management (pp. 1895-1898). ACM.
15. Lewis, D. (1997). Reuters-21578 dataset. URL= http: www. research. att. com lewisreuters21578. html.
16. Papineni, K. (2001, June). Why inverse document frequency?. In Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies (pp. 1-8). Association for Computational Linguistics.
17. Metzler, D., Bernstein, Y., Croft, W. B., Moffat, A., & Zobel, J. (2005, October). Similarity measures for tracking information flow. In Proceedings of the 14th ACM international conference on Information and knowledge management (pp. 517-524). ACM.
18. Tong, S., & Koller, D. (2002). Support vector machine active learning with applications to text classification. The Journal of Machine Learning Research, 2, 45-66.
19. Baharudin, B., Lee, L. H., & Khan, K. (2010). A review of machine learning algorithms for text-documents classification. Journal of advances in information technology, 1(1), 4-20.
20. Cao, D., Ramani, K., & Li, Z. (2010, April). Guiding concept generation based on ontology for customer preference modeling. In International Symposium series on Tools and Methods of Competitive Engineering (TMEC), Ancona, Italy.
21. Dinh, D., & Tamine, L. (2012). Towards a context sensitive approach to searching information based on domain specific knowledge sources. Web Semantics: Science, Services and Agents on the World Wide Web, 12, 41-52.
22. Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. Information processing & management, 24(5), 513-523.
23. Chen, C.-Y., & Ye, F. (2004). Particle swarm optimization algorithm and its application to clustering analysis. In Proceedings of the 2004 IEEE International Conference on Networking, Sensing and Control, Taipei, Taiwan (pp. 789–794).
24. Selim, S. Z., & Ismail, M. A. (1984). K-means type algorithms: a generalized convergence theorem and characterization of local optimality. IEEE Transaction of Pattern Analysis Machine Intelligent, 6, 81–87.
25. C.S. Sung, H.W. Jin, A tabu-search-based heuristic for clustering, Pattern Recognition 33 (5) (2000) 849–858.
26. T. Niknam, J. Olamaie, B. Amiri, A hybrid evolutionary algorithm based on ACO and SA for cluster analysis, Journal of Applied Science 8 (15) (2008) 2695–2702.

1/8/2013