http://ansinet.com/itj

ITJ

ISSN 1812-5638

# INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 8 (4): 453-464, 2009
ISSN 1812-5638


# A Tolerance Rough Set Based Semantic Clustering Method for Web Search Results

Xian-Jun Meng, Qing-Cai Chen and Xiao-Long Wang
Intelligence Computing Research Center, Harbin Institute of Technology,
Shenzhen Graduate School, 518055, People's Republic of China

**Abstract:** The objective of this study is to present a new web search results clustering algorithm which uses the tolerance rough set based approach to find the different meanings of the query in web search results and then organizes these results into different clusters according to their related meanings about query. Each meaning of the query can be represented by its contexts in each result and if there is a significant correlation between two context words, it is more likely that these two words represent the same meaning of query and also suitable as good indication of the meaning of query. In this study, the search results are organized in groups that each group of results relates to context words with high correlations and then these groups are merged into the final clusters representation using both cluster contents similarity and cluster documents overlap. The correlated context words with high documents coverage are selected as the labels of each cluster. Some experiments were conducted on different search results sets based on various queries. The results and comparisons of the proposed algorithm with that of the popular search results clustering algorithms through an empirical evaluation establish the viability of this proposed approach.

**Key words:** Search results clustering, rough set theory, web mining, part-of-speech tagging, semantic similarity

## INTRODUCTION

As an unsupervised learning method, clustering is a fundamental technique in research areas of data mining. With the aid of clustering, people can neglect the finer differences of individual objects and organize them into fewer groups based on the similarity in their common traits, which can help the user to overview the data set in a straightforward and understandable way. This property of clustering is very valuable in web information acquisition. Nowadays, popular commercial web search engines like Google (www.google.com) and Yahoo (www.yahoo.com) are considered as indispensable service for people surfing on the internet. The traditional keyword-based, Boolean search style adopted by these search engines takes great impact on users' information acquisition behavior (Mecca *et al*., 2007). But there are two obstacles that could hamper the efficient using of such kind of search engines. The first problem is that most popular search engines often return a long list of search results. Even the results were ranked by their relevance to the query, the users still need to go through the list and examine the titles and snippets one by one to find whether the results are what they want. Moreover, the queries are often short and ambiguous, if there are multiple subtopics related to the user's query and what the user's want is just minor part of them or not stand for the mainstream at that time, the query process would be a boring and time consuming task. The results related to different subtopics mixed together and the users either examine the result list one by one, or, most usually, refine their queries in a more specific way. The second problem is that the disparity of the experiences of web users is extraordinary and it is hard for most part of users to formulate their queries in a way that are both sufficiently descriptive and discriminating to find just the web pages that are relevant to their search goal (Crabtree *et al*., 2006).

These problems exist in search engines are just similar with the general problems in all systems that process natural language, where the real cause for them is the ambiguity: words have many fuzzily defined meanings and meanings shift with the contexts (Francis, 2001). Search Results Clustering (SRC) is one possible solution to handle the ambiguities in search results. It can assist users to either comprehend the semantic structure in results or refine their queries by cluster search results into different groups. Unlike classical text clustering, SRC requires not only correct clusters but also clear labels to explain each cluster. Moreover, the performance is very important for usability (Hearst and Pedersen, 1996; Zamir and Etzioni, 1999).

Many approaches of search results clustering were conducted to improve the process of search results

**Corresponding Author:** Xian-Jun Meng, Intelligence Computing Research Center, Harbin Institute of Technology, Shenzhen Graduate School, 518055, People's Republic of China

overview and help users to easily find the pages which they interest in. In the inchoate stage of search results clustering, researchers just employed the classical text clustering methods to SRC. Such kind of methods (Hearst and Pedersen, 1996; Leuski, 2001) cluster the documents into topic related groups according to content similarity and then generate summaries for clusters by selecting representative terms in each cluster. This kind of methods is difficult to meet the special requirements in SRC, for the parameters like the number of clusters is hard to establish in advance and descriptions for clusters are often unreadable (Zamir and Etzioni, 1998, 1999). Moreover, the short snippets instead of full documents are used to conduct the clustering process, which can lead to the poor results as most similarity measurement used in classical text clustering algorithm often yield many zero values (Ho and Nguyen, 2002). Later, the label based methods (Kummamuru *et al*., 2004; Ohta *et al*., 2004; Osinski *et al*., 2004; Zeng *et al*., 2004) were proposed to solve the especial problems existed in SRC and one related open source framework Carrot (project.carrot2.org) was created to facilitate researches on clustering search results. This kind of approaches reverse the process of classical text clustering and extracts descriptive terms (words or phrases) from the search results based on various methods as candidate labels first and then organize the search results into groups according to these labels. There are also several commercial clustering search engines, like Vivisimo (vivisimo.com) and Grokker (www.grokker.com) which share some common features with the research systems like Carrot. The algorithm proposed in this study is also a label based approach, but it adopt a different labels generation process that can fully utilize the words in results to produce more descriptive and discriminating labels and meanwhile use the cluster contents as criteria to merge clusters which can generate high quality final clusters representation.

Rough set theory was introduced by Pawlak (1982) to deal with the uncertainty and vagueness for data analysis and classification (Pawlak, 1991). It has been successful applied in many applications. Equivalence Rough Set Model (ERSM) (Funakoshi and Ho, 1998) applied the original rough set model into information retrieval. By means of partitioning the vocabulary into equivalence classes, ERSM can calculate the semantic relationship of terms. However, in field of information retrieval equivalence classes of terms are not always suitable because of term ambiguities and the transitive property required in equivalence relations is too strict for the meaning of words, so it is hard to automatically calculate equivalence classes of terms (Ho and Funakoshi, 1998). The Tolerance Rough Set Model (TRSM) proposed by Skowron and Stepaniuk (1996) relaxed the restrict of transitive property and allow the overlapping classes. This made the rough set based researches in information retrieval feasible to be conduct. Recently, some methods were developed which applied the TRSM in information retrieval and text mining. Kawasaki *et al.* (2000) and Ho and Nguyen (2002) separately proposed hierarchical and nonhierarchical text clustering algorithm based on a tolerance rough set model. Lingras (2002) proposed an unsupervised classification method using properties of TRSM along with genetic algorithm to represent clusters as interval sets. An *et al.* (2004) presented a feature reduction method based on the TRSM and show its effectiveness in feature selection on web page classification. Inspired by the study of Ho and Nguyen (2002) and Ngo and Nguyen (2005) also proposed a method for web search results clustering by enrich the representation of snippets returned from search engine based on TRSM.

The motivation of using tolerance rough set model in this study is inspired by Ho and Nguyen (2002). In their research they proposed text clustering algorithm that use TRSM to determine the latent relations between words in documents and then applied k-means like algorithm to perform the documents clustering. In this study, the TRSM is applied in a similar way, but through a perspective of label based Web search results clustering approach.

In this study, SRC is treated as a process of query discrimination which is used to discriminate the meanings of query and then cluster the search results into clusters that each resulting cluster represents a particular meaning of query. However, each search result is represented by only a few words and general words or documents similarity measurement often produce poor results because of the sparse features space. So, in this study a tolerance rough set based semantic clustering method is adopted to solve this problem, which referred to as TRSSC. Given a query and the ranked list of search results, this algorithm try to mine all the latent different meanings of the query from result contents and then organizes these results into different groups according to their related meanings. In this approach, the results are looked as the contexts of the query, because each result is consecutive in content and short enough in length. Meanwhile each search result can also represent a clear meaning of the query in the corresponding page. In this research, a novel method is proposed to find the different meanings of the query. Firstly, the context words are partitioned into groups and each group includes the words with high correlation. In the context of query, even just one more word can make the query more specific and

the correlated context words can supply more descriptions about the meaning of query meanwhile with a coherent mode. These groups of words can be looked as cluster label candidates. Then, the initial clusters based on these grouped context words are generated and an improved merging process is adopted to merge these clusters using both cluster contents similarity and cluster documents overlap, which can produce high quality final cluster representation.

Different with the full text documents, the words are little in the result search contents, so the correlations between words are not so obvious from literal analysis. This study extends the original results' feature space into their upper approximation space and in the new space the latent semantic correlations between context terms are measured. Experimental results show that this method can cluster the search results in a way that maximizes the coverage of each cluster and generates labels that descriptive enough to take a good representation of the meaning of cluster and meanwhile discriminating to separate it from other meanings.

## TOLERANCE ROUGH SET THEORY

In origin rough set theory, a set in a universe is expressed by a pair of crisp sets called lower and upper approximations regarding some equivalence relation in the universe. Let U be the universe and $R \subseteq U \times U$ be an equivalence relation defined on U, which also be called an indiscernibility relation. The pair A = (U, R) is an approximation space. Two objects $x, y \in U$ are indiscernible regarding R if xRy. The lower and upper approximation of any set $x \subseteq u$ with respect to R can be defined as (L (R, X), u (R, X)), where:

$$L(R, X) = \{x \in U: [x]_R \subseteq X\} \quad (1)$$

$$u(R, X) = \{x \in U: [x]_R \cap X \neq 0\} \quad (2)$$

The lower and upper approximation also be referred to as $(\underline{R}(X), \overline{R}(X))$, $[x]_R = \{y \in U \mid xRy\}$ denotes the equivalence class of objects indiscernible with x regarding the equivalence relation R. If $[x]_R \subseteq X$, then it is certain that $x \in X$. If $[x]_R \subseteq U - \overline{R}(X)$ then it is clear that $x \notin X$. $[x]_R \subseteq X$ is called rough with respect to $R \Leftrightarrow \underline{R}(X) \neq \overline{R}(X)$. Otherwise X is R-discernible. The set $BNR(X) = \overline{R}(X) - \underline{R}(X)$ is called rough boundary of X. A crisp boundary is defined as the union of some elementary objects. The inability to group elementary objects into distinct partitions leads to rough boundaries.

There are three properties of an equivalence relation R, which are reflexive, symmetric and transitive. The transitive property does not always satisfied in applications like natural language processing and information retrieval (Ho and Nguyen, 2002), because the concepts in these domains are imprecise and can be overlapped (Ngo and Nguyen, 2005). Skowron and Stepaniuk (1996) proposed a general approximation model to deal with the overlapping classes using tolerance relations in which only reflexive and symmetric properties were needed. The generalized approximation spaces are called tolerance spaces and can be formally defined as a quadruple R = (U, I, ν, P), where, U is a universe of objects; I: $U \rightarrow P(U)$ is an uncertainty function which satisfying conditions: (1) $x \in I(x)$ for $x \in U$ and (2) $y \in I(x) \Leftrightarrow x \in I(y)$ for any $x, y \in U$, I is a tolerance relation; ν: $P(U) \times P(U) \rightarrow [0,1]$ is a vague inclusion for measuring the degree of inclusion between two sets, which must be monotone with respect to the second argument of ν and P: $I(U) \rightarrow \{0, 1\}$ is a structurality function where, $P(I(x)) = 1$ for any $x \in U$, $\{I(x): x \in U\}$. The lower and upper approximation in R for any $X \subseteq U$ can be defined as:

$$L(R, X) = \{x \in U: P(I(x)) = 1 \& \nu(I(x), X) = 1\} \quad (3)$$

$$u(R, X) = \{x \in U: P(I(x)) = 1 \& \nu(I(x), X) > 0\} \quad (4)$$

The key of using TRSM in any applications is how to determine suitable I, ν and P (Ho and Nguyen, 2002).

## DETAILS OF TRSSC ALGORITHM

In text document, the meaning of a word can be inferred from its context. The intuition is that words that occur in similar contexts will tend to have similar meanings. This is known as the Distributional hypothesis (Miller and Charles, 1991). The algorithm proposed in this study clusters web search results according to the query's distributional similarity in results that each resulting cluster represents a particular meaning of query. To the best of our knowledge, there has been no earlier study treating SRC like this way.

This algorithm consists of five phases. In phase I, the context features are selected based on various approaches which including the standard feature selection and POS based feature selection. Some evaluations will be conducted to measure the impacts of these methods on the quality of final clusters. In phase II, the TRSM based method is applied to enrich the original feature space of search results to their semantic upper approximation space based on the tolerance relation of words co-occurrence. In phase III, the words similarity measurement is adopted to measure the correlations between words in the extended feature space and then

organize words into groups based on their correlations. In phase IV, an improved cluster merging method is proposed to merge clusters into the final clusters representation. In the last phase, the labels for each cluster are selected and the final results are presented to the users.

Here, the details of the tolerance rough set based semantic clustering algorithm for web search results will be described.

**Context feature selection:** The search result returned from search engine typically contains the title, URL of the actual page, link to live and cached versions of the page and sometimes an indication of file size and type. In addition, one or more snippets are usually presented, which can help the searcher to quick preview of the related page. The snippets are usually query biased and are short fragments of text extracted from the page's content, so it can be treated as the contexts of query in result page. In this approach, only the title and snippets in each result are used because only these two kinds of information are content related.

In this study, two different methods are adopted to preprocess the search results. The first method is the standard approach which removes the punctuation, some non-informative texts and stop words. Then the Porter stemming algorithm (Porter, 2006) is used to reduce each term to their root form. The second method is Part-of-Speech (POS) based feature selection. In natural language, the noun related words are good indicator to the meaning of the content. So, in this method only the noun-related words are selected as the features for further processing. Although, POS tagging is a time consuming task, the length of snippet is very short and the performance of machine in nowadays is powerful enough, so the time is not so insufferable in this approach.

After the step of context feature selection, the snippets were just represented by a sequence of context words. Based on these words, the initial base clusters are constructed and each cluster includes list of results that containing that word. At this stage, the words that occurred in more than 50% of the results (which include the query word(s), because the meaning of query can be inferred from its context words.) were eliminated. Words with higher coverage may be more general and can cause the problem of topic drifting and finally impact the quality of clusters.

**Enriching document representation:** Generally, there are little words existed in snippets. After feature selection, the number of words in each result becomes fewer. Based on the results of experiments in this study, each result usually only contains 3 to 20 words.

The absence of shared words in search results is insufficient for mining the semantic relationship literally between words because of the sparse data problem. So, it is essential to make the most use of the features in results and mine more latent semantic associations between features. In natural language, some conceptual relations between words cannot be found directly. For example, synonyms and polysemy, that multi words have the same meaning and the same word have multi meanings, respectively. Latent Semantic Analysis (LSA) (Landauer and Dumais, 1997) is a choice for dealing with this kind of problem, but its computation complexity is heavy and not suitable for online processing. In this approach, a TRSM based method is developed to mine the latent correlations between words which can extend the original feature space of results to an upper approximation feature space and in extended feature space, two words can have high similarity even if they do not co-occur in any document.

Same with Ho and Nguyen (2002), the tolerance relation R is defined as the co-occurrence of terms in all document from dataset D. Terms are similar to the extent that they co-occur in the same context (related to the query). Let $D = \{d_1, d_2,...,d_N\}$ be a set of documents and $T = \{t_1, t_2,...,t_M\}$ be set of context terms in D. $R = (T, I_\theta, \nu, P)$ is the approximation space defined over T. The uncertainty function is defined as:

$$I_\theta(t_i) = \{t_j | f_D(t_i, t_j) \geq \theta\} \cup \{t_i\} \quad (5)$$

where, $\theta$ is a user defined co-occurrence threshold $f_D(t_i, t_j)$, is the number of documents in D that contain both terms $t_i$ and $t_j$. The set $I_\theta(t_i)$ is the tolerance class of term $t_i$. The vague inclusion function $\nu$ is defined as:

$$\nu(X, Y) = \frac{|X \cap Y|}{|X|} \quad (6)$$

The membership function $\mu$ for $t_i \in T$, $X \subseteq T$ is then defined as:

$$\mu(t_i, X) = \nu(I_\theta, X) = \frac{|I_\theta(t_i) \cap X|}{|I_\theta(t_i)|} \quad (7)$$

All tolerance classes of terms can be seen as structural subsets: $P(I_\theta(t_i)) = 1$ for all $t_i \in T$.

The lower and upper approximations of any subset $X \subseteq T$ can be determined with the tolerance space $\tau = (T, I, \nu, P)$ as:

$$L(R, X) = \{t_i \in T: \nu(I_\theta(t_i), X) = 1\} \quad (8)$$

$$U(R, X) = \{t_i \in T: \nu(I_\theta(t_i), X) > 0\} \quad (9)$$

Table 1: Approximations of results related to apple

| Doc. No. | Context words | $\overline{R}(X)$ |
|---|---|---|
| 1 | 'onlin', 'mac', 'ipod', 'store', 'comput', 'macintosh', 'laptop' | 'app', 'compani', 'comput', 'includ', 'iphon', 'ipod', 'laptop', 'latest','mac', 'macintosh', 'onlin', 'os', 'product', 'retail', 'review', 'softwar', 'steve', 'store' |
| 2 | 'ipod', 'iphon', 'user' | 'app', 'iphon', 'ipod', 'mac', 'product', 'review', 'softwar', 'store', 'user' |
| 3 | 'product' | 'ipod', 'mac', 'product', 'softwar' |
| 4 | 'comput' | 'compani', 'comput', 'includ', 'latest', 'mac', 'macintosh', 'softwar', 'steve' |
| 5 | 'iphon' | 'app', 'iphon', 'ipod', 'mac', 'softwar', 'store' |

If is treated as an concept described vaguely by terms it contains, then $\overline{R}(X)$ is the set of concepts that share some semantic meanings with X, $\underline{R}(X)$ while is a core concept of X (Ngo and Nguyen, 2005). The co-occurrence threshold $\theta$ can be used to control the preciseness of the concept represented by a tolerance class of terms.

The basic idea can be illustrated using an example. The data set was collected from the search engine using query apple which contains 200 results. The co-occurrence threshold $\theta$ can be set as 6 and the original feature space and its upper feature approximation space are shown as Table 1.

From Table 1, it can be seen that more latent semantic correlations between features could be found based on TRSM which absence in original feature space.

The quality of the clustering algorithm heavily depends on co-occurrence threshold $\theta$ in constructing the upper approximation space using tolerance rough set model. The higher the value $\theta$, the smaller the upper approximation space. Ho and Nguyen (2002) suggested that when the average number of terms in documents is high and/or the size of document collection is large, high values of $\theta$ are often appropriate and vice versa. In the experiment environments of this study (200 snippets and each snippet have 3-20 words), $\theta$ is set as 4.

**Semantic correlation measurement in context:** The semantic correlations between context words are very useful indications to the meaning of query. This algorithm try to determine which of the meanings of a query is invoked in a particular result and this can be done by looking at the context of the query. Every word in context provides clues to the meaning of query. Compared to the single or multiple unrelated context words, the words with significant correlations in context of query can take a better representation of the meaning of query. For example, in results of query apple, the context word price can make a more specific restriction to the meaning of query and so does the IPod. But the query apple along only with each of them cannot provide sufficient evidences that can discriminate the meaning of query because there are lots of ambiguities related to price of apple or lots of information related to apple IPod. If both of them are used as the refinement words, the meaning of query is specific which has little ambiguities. The latent semantic correlations between context words are mined in the extended upper approximation feature space generated from previous step.

There are lots of methods used to measure the strength of words similarity or words association. One kind of approaches use a manually constructed lexicons, like WordNet (Miller *et al*., 1990) which associated with several disadvantages. First, manually created lexicons often contain rare senses. The second problem with these kinds of lexicons is that they miss many domain specific senses (Pantel and Lin, 2002). The others use the statistical analysis based methods, like Latent Semantic Analysis, Pointwise Mutual Information (PMI) (Church and Hanks, 1990), Log-likelihood ratio (Dunning, 1993) and Normalized Google Distance (NGD) (Cilibrasi and Vitanyi, 2007). In these methods, a key assumption is that similarity between words is a consequence of word co-occurrence. If there are two or more words co-occurred frequently, they may have syntactic or semantic relations to each other.

Based on the experiments in this study and the researches proposed by Pedersen and Kulkarni (2007) and Lindsey *et al.* (2007), the NGD is chosen in this study to measure the semantic correlations between words. The NGD use the counts of pages that words occurred in returned from the Google search engine (or any other data sets) as the evidences to compute the dependence between words. Words with high correlation in sense of natural language tend to have low Google distance, while words with dissimilar meanings tend to have high Google distance.

Specifically, the normalized Google distance between two terms x and y can be defined as:

$$NGD(x,y) = \frac{\max\{\log f(x), \log f(y)\} - \log f(x,y)}{\log N - \min\{\log f(x), \log f(y)\}} \quad (10)$$

where, x and y are terms, f(t) is the approximate number of documents containing t, N is the approximate total number of documents indexed by Google (or such kind of data set) and f(x, y) is the approximate number of documents containing both x and y.

If two terms never occurred together on the same document, the NGD value between them is infinite, or if they always occurred together, their NGD value is zero.

The NGD between words can be obtained online from Google search engine, but this process is time consuming and has service restriction (500 queries/IP/day), so it is not suitable in expandability. In this study, NGD is applied in a small scale data set sampled from Google, that is, the search results set which have similar distribution with the whole results set returned from Google. So, this kind of downscaling is feasible and practically sufficient for the clustering purpose in this approach. f(t) is denoted as the document frequency of term in search results collection D and N is the total number of results in collection D. The value of approximate NGD in this study is between 0 and 1 because this algorithm prevents the calculation of the correlations of words that never occurred together. Terms with low distance tend to be specific and are often unambiguous in the context of the query, while terms with a high query distance tend to be quite broad and are often ambiguous. In experiments of this study, setting the threshold value of NGD to 0.35 can achieve relative good performance of clustering and high coverage of results.

Just like PMI, NGD suffers a bias towards two words only occur with each other and give them lowest score if they only occur 1 time. The setting of minimums documents coverage and threshold of co-occurrence in this study can prevent this problem.

The co-occurrence threshold in previous phase can also affect the threshold of NGD value that the higher the value $\theta$, the higher the value threshold of NGD.

**Clusters merging:** Based on the correlation measurement using NGD, the context words are clustered into groups that each group including words with high correlations using single-link clustering method (Jain *et al*., 1999). In this approach, the context words with low correlation to any other words are discarded. So in each group, there are at least two words. Then according to these groups, the initial base clusters are merged into several interim clusters that the format looks like:

$$[c^*_1, c^*_2,...]: [c_1, c_2, c_3,...]: [d_1, d_2, d_3,...]$$

The first tuple of interim cluster is the group of the correlated context words for that cluster. The second tuple includes the words selected from all context words contained by results in this cluster which have high correlation with any word in the first tuple and in this study the top 20 correlated words are selected. These context words can be seen as the content of the interim cluster. The third tuple contains the results list in this cluster.

The number of interim clusters is far less than the number of initial base clusters, but still too many to present them to the users. Despite the correlations computed using NGD, there still have some similarity exist in interim clusters, which cannot mine by co-occurrence based words similarity measurement. Then it is essential to further emerge these interim clusters based on not only their results overlaps but also their related contents, which make sure that the interim clusters which have similar meaning should be merged and meanwhile keep the dissimilar clusters apart.

The similarity between the contents of clusters is measured by their group average NGD similarity between all pairs of their context words and similarity between documents of clusters by their overlaps. The similarity measurement used in this phase can be defined as:

$$\mathrm{Sim}(C_i, C_j) = p \times \frac{|C_i^D \cap C_j^D|}{\min(|C_i^D|, |C_j^D|)} + q \times (1 - \mathrm{AvgSim}(C_i^C - C_j^C)) \quad (11)$$

where, $C_i \neq C_j$ and $p+q = 1$, $p, q \geq 0$. p and q determine the relative weights to be given for both kind of similarity. In the experiments, p and q are both set to 0.5 and the threshold of Sim ($C_i$, $C_j$) is set to 0.6.

**Label selection:** The label selection method in this approach is simple and straightforward. The goal is to find the descriptive and discriminate labels for each cluster. By using the semantically related context words, the label can have good ability to specify the meaning the cluster and using the content based clustering merging and the clusters can be merged together with less semantic overlapping. There are three factors that can impact on the quality of labels: the first is the label's semantic correlation; the second is the label's coverage, even the semantic relation is very tight, if the coverage of the context words is very low, it is still not suitable to be selected as the labels; the third is the position of context words, two or more context words are an ordered sequence in representation, It is necessary to reorder the sequence of context words in the way they appeared in snippets. For example, association rule and rule of present association, same context words with different meaning in different order.

The process of choosing labels of clusters following these steps:

- Choose context words with the strong semantic correlations
- Choose context words selected from earlier step with high results coverage
- Reorder the sequence of labels

The context words related to each meaning of query can provide additional information to make the query more specific.

## EXPERIMENTS AND RESULTS

The experiments in this study conducted several objective measures to validate the performance of the proposed algorithm. Evaluating the quality of search results clustering is a non-trivial task and the most suitable evaluation is judged by user's experiences. So we also made some empirical comparisons for the proposed algorithm with that of the popular search results clustering algorithms.

The algorithm in this study was implemented in Python on a CentOS 5 Linux Workstation with a Xeon 3.0 GHz processor and 2 GB memory.

**Description of datasets:** The evaluation datasets used in this study were constructed based on the top 200 search results retrieved from Google search engine according to different kind of queries. Two hundred is the number of results that can coverage most of the meanings about the query. To make a good simulation of the user's query process, three kinds of queries were used in this experiment: queries that related to ambiguous topics which represent different concepts (apple, jaguar), queries that relate to general topics which have broad various subtopics (Linux, data mining) and queries that related to specific topics (rough sets, VoIP). Table 2 shows some statistics information about the data sets used in the experiments.

**Data preprocessing:** Data sets need to be preprocessed before they were used in the experiments. In the standard approach, punctuation and some non-informative texts firstly were removed. The noise information usually has bad effect on the quality of final clusters. Then, the stop words were eliminated from the results. The stop words are the words bear no content or relevant semantics. In this approach, the English stop word list was adopted from SMART system, which consist 571 stop words. Then, the words were stemmed by using the Porter's suffix-stripping algorithm. It is important to do the stemming since it can eliminate the minor difference between words with the identical meaning. In POS based feature selection approach, the POS tagger adopted in this approach is CRFTagger (crftagger.sourceforge.net), which use the brown tag-set and has very good performance (accuracy of 97.00% and tagging speed is 500 sentences $sec^{-1}$). In this study, the words with tags NN, NNS, NNP, NNPS were extracted which means singular noun, plural noun, singular proper noun and plural proper noun, respectively. In this approach, the Document Frequency (DF) was also adopted to filter the context words with document coverage more than 50%. In the experiment, the time of POS tagging for each data set is no more than 0.4 sec.

Table 2: Statistic information of datasets

| Dataset | Terms | Uni terms | Specificity |
|---|---|---|---|
| Apple | 6196 | 2113 | Low |
| Jaguar | 6645 | 1770 | Low |
| Linux | 5423 | 1786 | Medium |
| Data mining | 6128 | 1708 | Medium |
| Rough sets | 6795 | 2028 | High |
| Voip | 5984 | 1737 | High |

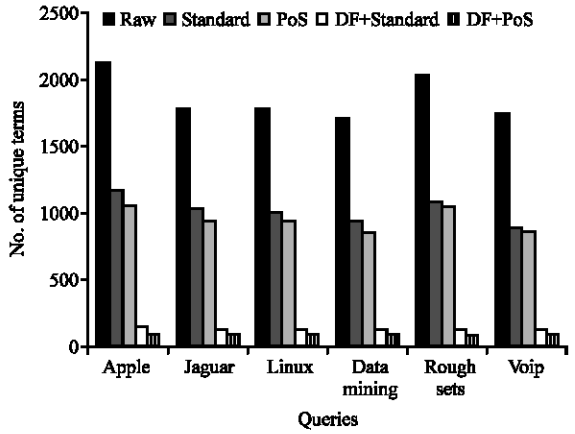


Fig. 1: Context words selection with different

From Fig. 1, it can be seen that, the number of features fell sharply after the preprocessing stage. In all datasets, POS with DF based feature selection achieved the smallest features set and compared with other feature selection methods especially the standard approach with DF, the features selected are more descriptive, because all features remained are noun related words. In this approach, only the Standard (+DF) and POS (+DF) pre-processing methods were considered.

**Initial base clusters selection:** Using a lower threshold of minimum documents coverage will increase the number of initial clusters. Clustering can be seen as an information compression process and documents grouped in the same cluster share the common traits and lost their special characters. So, more base clusters can promote the clustering performance because more information will be reserved for further merging process. But too many base clusters can slow the speed of clustering and many base clusters are useless which even may contaminate the final clusters. Paradoxically, if the threshold is too high, many minor but important clusters were omitted and the clusters coverage is low too. So, it needs balance between clustering performance and documents coverage. Using standard pre-processing method, the minimum documents coverage threshold was set as 1, 2, 3, 4, 5 and 6%. From Fig. 2, it is obviously that 3% is a suitable threshold that can preserve small number of initial base clusters while keep high documents coverage. It is same in POS based

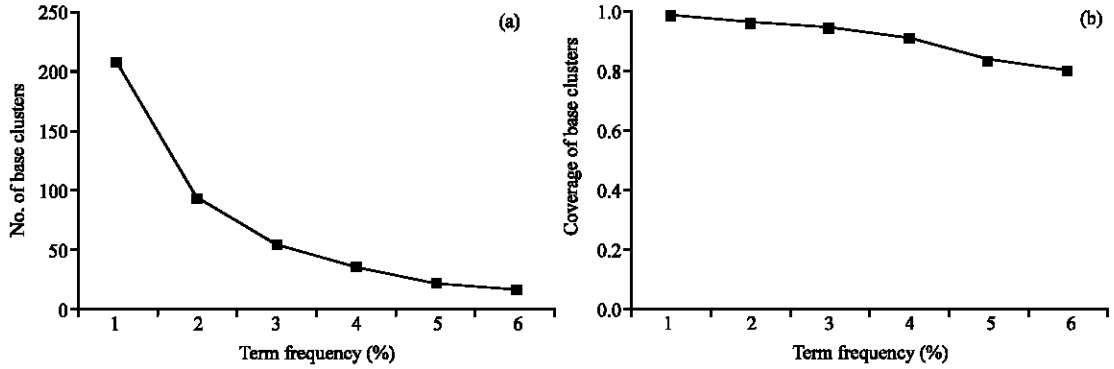


Fig. 2: No. of initial clusters and their documents

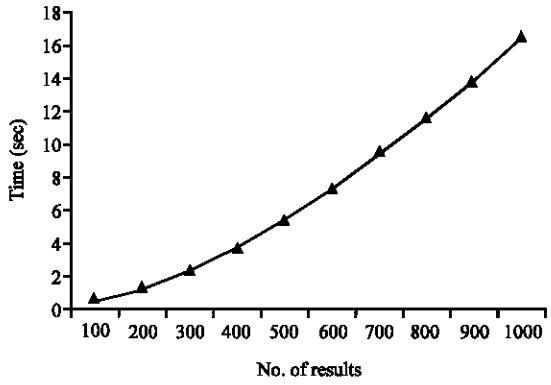


Fig. 3: Time complexity analysis

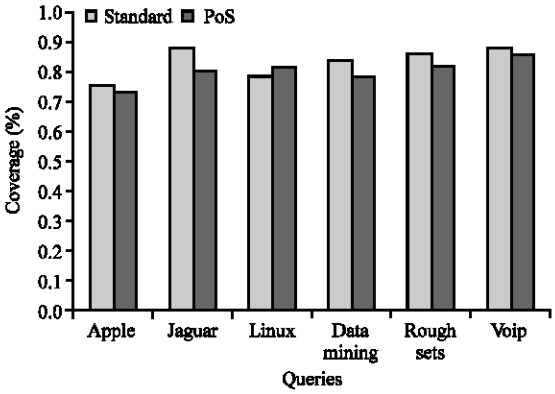


Fig. 4: Coverage of final clusters

pre-processing method. So, in this study, 3% was selected as the threshold of minimum documents coverage in both kinds of pre-processing methods.

**Time complexity:** The algorithm proposed in this study is very efficient. For example, the matrix was used as the query word and 1000 snippets were retrieved from Google search engine used to analyze the time complexity of this algorithm. The time is clustering consumed using POS based pre-processing. This algorithm was implemented using python and didn't optimize the program coded. From Fig. 3, it can be seen that the time complexity in this approach is approximately linear. So, it is fast and extensible in large scale applications.

**Coverage and overlap evaluation:** Because the document frequency based feature selection was used in this clustering algorithm and base clusters were merged according to their labels' correlation, which lead to the situation that the final clusters can hardly cover all the results in collection. This is a common problem in almost all label based clustering algorithms. It isn't a drawback in this approach because even the results from search engine have outliers that irrelevant with the query. What the user wants in search result clustering is overview of the semantic structure in results set. They may not concern the results that share little semantic correlations with others. But it doesn't work well if the coverage is too low. Figure 4 shows the coverage of clusters generated by the algorithm in this study for 6 queries. It can be seen that the algorithm with standard pre-processing usually get high coverage than that using POS based process. This mainly because the features selected using POS are less than the features selected using standard preprocess. In this approach, all the results that do not belong to any clusters were organized into a group names Miscs.

In this algorithm, one result can belong to more than one cluster. This can reflect the essence of natural language: using little words to represent more meanings. But too much overlap may cause the ambiguous situation too. The overlap measurement used in the experiment is the ratio of total number of results in final clusters divided by the number of the unique results in final clusters. From Fig. 5, it can be shown that, even the coverage using standard based clustering algorithm is bigger than POS based method, but the overlaps in POS based approach is

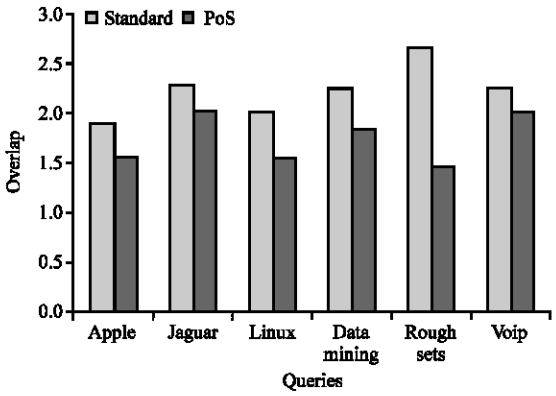


Fig. 5: Overlap of final clusters

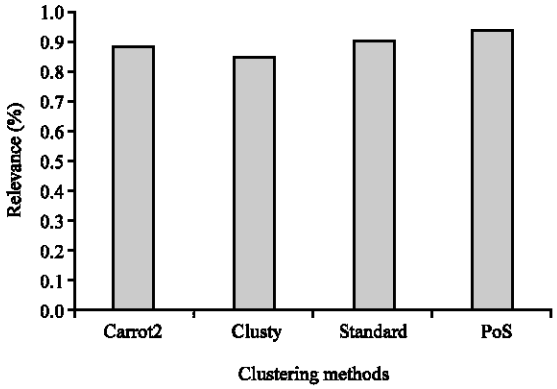


Fig. 6: Label relevance in different approach

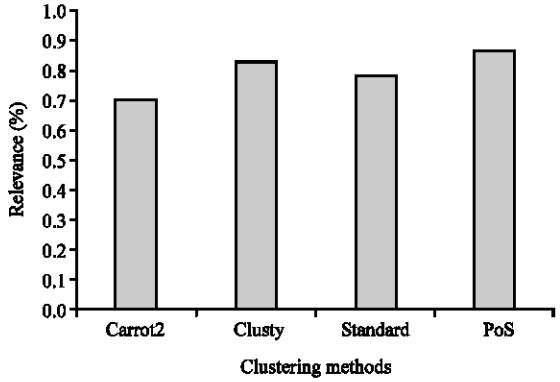


Fig. 7: Documents relevance with labels

obviously less than the standard based method. Nouns related words have a good ability of distinguishing the different meanings of the query.

**Quality of the algorithm:** The evaluation of the quality of algorithm in this study can be conducted from an information retrieval view. The final cluster representation list all latent different meaning of query which implicated in the search results and each cluster accompany with its labels provides a refinement to the query which leading to more specific results set. The evaluation includes two parts. One is the relevance between the labels and the queries and the other is the relevance between results in clusters and their corresponding labels. The comparisons were conducted between the algorithm proposed in this study with two other popular clustering systems, which are Carrot[2] (www.carrot2.org, use Lingo algorithm and limit its source search engine as Google) and Clusty (www.clusty.com). In this approach, the label relevance was computed by dividing the number of relevant labels generated by different systems by their number of total labels and precision (P) at top N results (P@N) was used to measure the documents relevance to their related labels:

$$P@N = \frac{|C \cap R|}{|R|} \quad (12)$$

where, R is the set of top documents in a cluster and C is the relevant documents in this cluster that relevant to their labels. In the experiments, P@10 was used as evaluation criteria, because not so many results are examined by most users, especially in clusters.

The labels relevance comparison can be referred from Fig. 6 and the documents relevance comparison with labels can be referred from Fig. 7. The results are the mean value obtained by the blind test from five evaluators based on 6 queries and the score of each result includes 2 grades: relevant and irrelevant. From Fig. 6 and 7, it can be shown that the label's semantic relationships with queries in all platforms are good and some improvements are obtained using the algorithm of this study with both pre-processing methods. Usually, there is a general problem in the pure label based clustering algorithm, that the meanings of several labels are identical. Take example of the results from (Zeng *et al*., 2004), there are three labels with the same meaning about the query juguar, like panthera onca, cats and animal. The algorithm proposed in this study can generate labels with semantically coherence and prevent the generation of redundant labels.

In the evaluation of relevance between results in cluster with their related labels, the standard based approach TRSSC algorithm performed not well as Clusty, but the POS based algorithm performed better than Clusty. This explains that noun related labels can make a better representation for the content. Compared with Carrot and Clusty, a drawback of the system developed in this study is that only single source search engine was used which lead to the situation that some knowledge

Table 3: Clusters presentation with standard pre-processing

| Queries | Results | Queries | Results |
|---|---|---|---|
| Apple | Jobs steve | Data mining | Book concept |
| | Iphone ipod store | | Extract hidden |
| | Computer macintosh | | Customer market |
| | Aapl stock | | Machine learning |
| | Offer resource | | Model prediction |
| | Mac os | | Business technique |
| | Macbook pro | | Kdd discovery |
| Jaguar | Classified listing | | sql server |
| | Car xf xkr | Rough sets | Current trend |
| | Company ford lincon | | Application pawlak |
| | America cat onca | | Price compare |
| | Buy guid luxuri price | | Discovery technology |
| | $x_j$, $x_k$ | | Feature selection |
| | Parts accessories | | Call paper |
| | Dealer local sale | | Data mine |
| | Model specification | | Base analysis |
| | Land rover | | Deal mathematical |
| | Test drive | | International conference |
| Linux | Code tool | Voip | Host pbx |
| | Free software open source unix | | Small business |
| | Dell ubuntu | | Low cost |
| | Hardware network How to | | Feature residential |
| | Red hat | | Internet counection |
| | Suse novel serve | | Router adaptor |
| | Link portal | | Long distance |
| | Creator torvald | | Unlimit call |

Table 4: Clusters presentation with POS based pre-processing

| Queries | Results | Queries | Results |
|---|---|---|---|
| Apple | Jobs steve | Data mining | Book concept |
| | Product software | | Customer relationship |
| | Iphone ipod | | Discovery research |
| | Aapl stock | | Intelligence analytic |
| | Macbook pro | | Student note |
| Jaguar | Car motor vehicle | | Sql server |
| | Type x xf xkr s | | Decision support |
| | Aston martin | Rough sets | Granular soft |
| | Blue catalog | | Current trend |
| | America cat panthera | | Methodology theory |
| | Jag xk | | International conference |
| | Part performance | | Method approach |
| | Luxury price | | Computer science |
| | Dealer sale | | Feature selection |
| | Comparison guide | | Data knowledge mining |
| | Land rover | | Book price |
| | Drive test | | Attribute reduction |
| Linux | Dell ubuntu | Voip | Phone rate |
| | Operating software | | Forum guide |
| | Red hat | | Voice technology |
| | Free foundation | | Gateway ip |
| | Base knowledge | | Directory resource |
| | Programming guide | | Networking router |
| | Open source | | Case platform |
| | Community system | | Connection test |
| | Novell view | | Performance management |
| | Server suse | | Service provide |
| | | | Center link |

coverage is not so good in this system. Fox example, in search results return from Google using the query apple, only two snippets contain the meaning about fruit. This situation leads to the poor performance of the algorithm in this study with some queries. There are no single search engine can cover all good results about a query. So more choices, more probabilities that a good clusters representation could be found.

**Examples of search results clusters:** Table 3 and 4 present all labels generated by the algorithm proposed in this study using both standard and POS based pre-processing. From Table 3 and 4, it can be found that both methods can lead to the concise labels for each cluster and the labels from POS based method are more meaningful.

## CONCLUSIONS

In this study, a new web search results clustering algorithm named TRSSC is presented which try to find all the different meanings of the query from results' contents and then organizes these results into different groups according to their related meanings. In this study, the results are treated as contexts about the query and salient context words were selected from results based on two different pre-processing methods and then the initial base clusters were constructed. The correlation between contexts words can be measured using NGD in the upper approximation feature space of results which using co-occurrence as tolerance relation. If there is significant correlationship between two context words, it is more likely that these two words represent the same meaning of query, so the base clusters leading by those two words can be merged together for consistent meaning representation. An improved merging process is then applied to merge clusters using both cluster contents similarity and cluster overlap, which can produce high quality final cluster representation. Finally, the most salient context words are selected with high correlation as the labels of each cluster. In general, this algorithm clusters the search results in a way that maximizes the coverage of each cluster and generates labels that descriptive enough to take a good representation of the meaning of cluster and meanwhile discriminating to separate it from other meanings. The experiments based on various queries were conducted which establish the viability and efficiency of this algorithm.

There are still some improvements need to be prospected in future research. First, it is essential to use multiple source search engines instead of only one to find whether there are improvements in quality of clustering. One problem in this direction may be how to fuse all different results sets into one comprehensive collection which cover more broad topics and still avoid contents redundancy. Second, this algorithm should be applied to full text documents instead of the snippets. This can extend the algorithm to various applications like the desktop search clustering. In the full text environment, more valuable context information can be mined which may helpful to the quality of clustering.

**ACKNOWLEDGMENTS**

This investigation has been supported in part by the National Natural Science Foundation of China (No. 60703015 and 90612005) and the National 863 Program of China (No. 2006AA01Z197).

**REFERENCES**

An, A., Y. Huang, X. Huang and N. Cercone, 2004. Feature selection with rough sets for web page classification. Trans. Rough Sets, 2: 1-13.

Church, K. and P. Hanks, 1990. Word association norms, mutual information and lexicography. Comput. Linguist., 16: 22-29.

Cilibrasi, R.L. and P.M.B. Vitanyi, 2007. The Google similarity distance. IEEE Trans. Knowl. Data Eng., 19: 370-383.

Crabtree, D., P. Andreae and X. Gao, 2006. Query directed web page clustering. Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, Hong Kong, Dec. 18-22, pp: 202-210.

Dunning, T., 1993. Accurate methods for the statistics of surprise and coincidence. Comput. Linguist., 19: 61-74.

Francis, H., 2001. Mining associative meanings from the web: From word disambiguation to the global brain. Proceedings of Trends in Special Language and Language Technology, Mar. 29-30, Standard Publishers Brussels, pp: 15-44.

Funakoshi, K. and T. Ho, 1998. A Rough Set Approach to Information Retrieval. In: Rough Sets in Knowledge Discovery, Polkowski, L. and A. Skowron (Eds.). Physica-Verlag, USA., ISBN: 978-3790811209, pp: 166-177.

Hearst, M.A. and J.O. Pedersen, 1996. Reexamining the cluster hypothesis: Scatter/gather on retrieval results. Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1996, Zurich, Switzerland, pp: 76-84.

Ho, T.B. and K. Funakoshi, 1998. Information retrieval using rough sets. J. Japan Soc. Artif. Intell., 13: 424-433.

Ho, T.B. and N.B. Nguyen, 2002. Nonhierarchical document clustering based on a tolerance rough set model. Int. J. Intell. Syst., 17: 199-212.

Jain, A.K., M.N. Murty and P.J. Flynn, 1999. Data clustering: A review. ACM Comput. Surveys, 31: 264-323.

Kawasaki, S., N.B. Nguyen and T.B. Ho, 2000. Hierarchical document clustering based on tolerance rough set model. Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery, Sept. 13-16, Lyon, France, pp: 458-463.

Kummamuru, K., R. Lotlikar, S. Roy, K. Singal and R. Krishnapuram, 2004. A hierarchical monothetic document clustering algorithm for summarization and browsing search results. Proceedings of the 13th International Conference on World Wide Web, May 17-20, ACM New York, USA., pp: 658-665.

Landauer, T. and S. Dumais, 1997. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. Psychol. Rev., 104: 211-240.

Leuski, A., 2001. Evaluating document clustering for interactive information retrieval. Proceedings of the 10th International Conference on Information and Knowledge Management, Oct. 5-10, ACM Atlanta, Georgia, USA., pp: 33-40.

Lindsey, R., V. Veksler, A. Grintsvayg and W. Gray, 2007. Be wary of what your computer reads: the effects of corpus selection on measuring semantic relatedness. Proceedings of the 8th International Conference on Cognitive Modeling, Jul. 27-29, Erlbaum Oxford, UK., pp: 279-284.

Lingras, P., 2002. Rough set clustering for web mining Fuzzy Syst., 2: 1039-1044.

Mecca, G., S. Raunich and A. Pappalardo, 2007. A new algorithm for clustering search results. Data Knowl. Eng., 62: 504-522.

Miller, G., R. Beckwith, C. Fellbaum, D. Gross and K. Miller, 1990. Introduction to word Net: An on-line lexical database. Int. J. Lexicography, 3: 235-244.

Miller, G. and W. Charles, 1991. Contextual correlates of semantic similarity. Lang. Cogn. Proc., 6: 1-28.

Ngo, C.L. and H.S. Nguyen, 2005. A method of Web search result clustering based on rough sets. Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence, Sept. 19-22, IEEE Computer Society, Washington DC. USA., pp: 673-679.

Ohta, M., H. Narita and S. Ohno, 2004. Overlapping clustering method using local and global importance of feature terms at NTCIR-4 Web Task. Working Notes NTCIR, 4: 37-44.

Osinski, S., J. Stefanowski and D. Weiss, 2004. Lingo: Search results clustering algorithm based on singular value decomposition. Proceedings of the International Conference on Intelligent Information Systems (IIPWM), May 17-20, Zakopane, Poland, pp: 359-367.

Pantel, P. and D. Lin, 2002. Discovering word senses from text. Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Jul. 23-26, ACM Edmonton, Alberta, Canada, pp: 613-619.

Pawlak, Z., 1982. Rough sets. Int. J. Comput. Inform. Sci., 11: 341-356.

Pawlak, Z., 1991. Rough Sets: Theoretical Aspects of Reasoning About Data. 1st Edn., Kluwer Academic Publishers, UK., ISBN: 978-0792314721.

Pedersen, T. and A. Kulkarni, 2007. Discovering identities in web contexts with unsupervised clustering. Proceedings of the IJCAI-2007 Workshop on Analytics for Noisy Unstructured Text Data, Jan. 8, Springer, pp: 23-30.

Porter, M., 2006. An algorithm for suffix stripping. Program: Electron. Lib. Inf. Syst., 40: 211-218.

Skowron, A. and J. Stepaniuk, 1996. Tolerance approximation spaces. Fundam. Infor., 27: 245-253.

Zamir, O. and O. Etzioni, 1998. Web document clustering: A feasibility demonstration. Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Aug. 24-28, ACM Melbourne, Australia, pp: 46-54.

Zamir, O. and O. Etzioni, 1999. Grouper: A dynamic clustering interface to Web search results. Comput. Networks, 31: 1361-1374.

Zeng, H., Q. He, Z. Chen, W. Ma and J. Ma, 2004. Learning to cluster web search results. Proceeding of 27th Annual International ACM SIGIR Conference on Research and Development in Informing Retrieval, Jul. 25-29, Sheffield, South Yorkshire, UK., pp: 210-217.