A Tolerance Rough Set Approach to Clustering Web Search Results*

Chi Lang Ngo and Hung Son Nguyen

Institute of Mathematics, Warsaw University
Banacha 2, 02-097 Warsaw, Poland
chilang@chilang.com, son@mimuw.edu.pl

Extended Abstract

Two most popular approaches to facilitate searching for information on the web are represented by web search engine and web directories. Although the performance of search engines is improving every day, searching on the web can be a tedious and time-consuming task due to the huge size and highly dynamic nature of the web. Moreover, the user's "intention behind the search" is not clearly expressed which results in too general, short queries. Results returned by search engine can count from hundreds to hundreds of thousands of documents.

One approach to manage the large number of results is clustering. Search results clustering can be defined as *a process of automatical grouping search results into to thematic groups*. However, in contrast to traditional document clustering, clustering of search results are done on-the-fly (per user query request) and locally on a limited set of results return from the search engine. Clustering of search results can help user navigate through large set of documents more efficiently. By providing concise, accurate description of clusters, it lets user localizes interesting document faster.

In this paper, we proposed an approach to search results clustering based on Tolerance Rough Set following the work on document clustering [4, 3]. Tolerance classes are used to approximate concepts existed in documents. The application of Tolerance Rough Set model in document clustering was proposed as a way to enrich document and cluster representation with the hope of increasing clustering performance.

Tolerance Rough Set Model: (TRSM) was developed in [3] as basis to model documents and terms in information retrieval, text mining, etc. With its ability to deal with vagueness and fuzziness, TRSM seems to be promising tool to model relations between terms and documents. In many information retrieval problems, defining the similarity relation between document-document, term-term or term-document is essential.

Let $D = \{d_1, \ldots, d_N\}$ be a set of documents and $T = \{t_1, \ldots, t_M\}$ set of *index terms* for D. TRSM is an approximation space (see [5]) $\mathcal{R} = (T, I_\theta, \nu, P)$ determined over the set of terms T (universe of $\mathcal{R}$) as follows:

* The research has been partially supported by the grant 3T11C00226 from Ministry of Scientific Research and Information Technology of the Republic of Poland.

J.-F. Boulicaut et al. (Eds.): PKDD 2004, LNAI 3202, pp. 515–517, 2004.

1. Let $f_D(t_i, t_j)$ denotes the number of documents in D in which both terms t_i and t_j occurs. The parameterize uncertainty function I_θ is defined as $I_\theta(t_i) = \{t_j \mid f_D(t_i, t_j) \geq \theta\} \cup \{t_i\}$. The set $I_\theta(t_i)$ is called the *tolerance class* of index term t_i.
2. The vague inclusion function is defined as $\nu(X, Y) = \frac{|X \cap Y|}{|X|}$,
3. All tolerance classes of terms are considered as structural subsets: $P(I_\theta(t_i)) = 1$ for all $t_i \in T$.

In TRSM, the lower and upper approximations of any subset $X \subseteq T$ can be determined by

$$\mathbf{L}_{\mathcal{R}}(X) = \{t_i \in T \mid \nu(I_\theta(t_i), X) = 1\}; \quad \mathbf{U}_{\mathcal{R}}(X) = \{t_i \in T \mid \nu(I_\theta(t_i), X) > 0\}$$

By varying the threshold θ (e.g. relatively to the size of document collection), one can control the degree of relatedness of words in tolerance classes. The use of upper approximation in similarity calculation to reduce the number of zero-valued similarities is the main advantage main advantage TRSM-based algorithms claimed to have over traditional approaches. This makes the situation, in which two document are similar (i.e. have non-zero similarity) although they do not share any terms, possible. Let us mention two basic applications of TRSM in text mining area:

1. Enriching document representation: In TRSM, the document $d_i \in D$ is represented by its upper approximation:

$$\mathbf{U}_{\mathcal{R}}(d_i) = \{t_i \in T \mid \nu(I_\theta(t_i), d_i) > 0\}$$

2. Extended weighting scheme for upper approximation: To assign weight values for document's vector, the TF*IDF weighting scheme is used (see [6]). In order to employ approximations for document, the weighting scheme need to be extended to handle terms that occurs in document's upper approximation but not in the document itself. The extended weighting scheme is defined as:

$$w_{ij} = \begin{cases} (1 + log(f_{d_i}(t_j)) * \log \frac{N}{f_D(t_j)} & \text{if } t_j \in d_i \\ \min_{t_k \in d_i} w_{ik} * \frac{\log \frac{N}{f_D(t_j)}}{1 + \log \frac{N}{f_D(t_j)}} & \text{if } t_j \in \mathbf{U}_{\mathcal{R}}(d_i) \backslash d_i \\ 0 & \text{if } t_j \notin \mathbf{U}_{\mathcal{R}}(d_i) \end{cases}$$

The extension ensures that each terms occurring in upper approximation of d_i but not in d_i, has a weight smaller than the weight of any terms in d_i.

The TRC Algorithm: The Tolerance Rough set Clustering algorithm is based primarily on the K-means algorithm presented in [3]. By adapting K-means clustering method, the algorithm remain relatively quick (which is essential for online results post-processing) while still maintaining good clusters quality. The usage of Tolerance Space and upper approximation to enrich inter-document and document-cluster relation allows the algorithm to discover subtle similarities not detected otherwise. As it has been mentioned, in search results clustering, the proper labelling of cluster is as important as cluster contents quality.

Since the use of phrases in cluster label has been proven [7] to be more effective than single words, TRC algorithm utilize n-gram of words (phrases) retrieved from documents inside cluster as candidates for cluster description. The TRC algorithm consists of five phases (depicted in Fig .1). It is widely known (see [2]) that preprocessing text data before feeding it into clustering algorithm is essentials and can have great impact on algorithm performance. In TRC, the following standard preprocessing steps are performed on snippets: *text cleansing, text stemming, and Stop-words elimination.* As TRC utilizes Vector Space Model for creating document-term matrix representing documents, in *document representation building* step, two main standard procedures: *index term selection and term weighting* are performed.

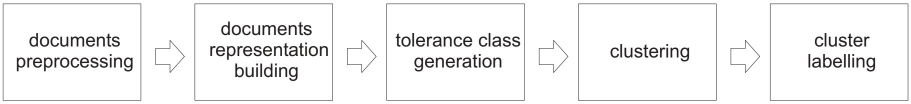

Fig. 1. Phases of TRC algorithm

We have implemented the proposed solution within an open-source framework, *Carrot*^{2}. The implementation of algorithm presented in this paper, including all source codes, will be contributed to the *Carrot*^{2} project and will be available at http://carrot2.sourceforge.net. We hopes that this will foster further experiments and enhancements to the algorithm to be made, not only by the author but also other researchers.

References

1. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. 1st edn. Addison Wesley Longman Publishing Co. Inc. (1999)
2. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. 1st edn. Morgan Kaufmann (2000)
3. Ho, T.B, Nguyen, N.B.: Nonhierarchical document clustering based on a tolerance rough set model. International Journal of Intelligent Systems **17** (2002) 199–212
4. Kawasaki, S., Nguyen, N.B., Ho, T.B.: Hierarchical document clustering based on tolerance rough set model. In Zighed, D.A., Komorowski, H.J., Zytkow, J.M., eds.: Principles of Data Mining and Knowledge Discovery, 4th European Conference, PKDD 2000, Lyon, France, September 13-16, 2000, Proceedings. Volume 1910 of Lecture Notes in Computer Science., Springer (2000)
5. Skowron, A., Stepaniuk, J.: Tolerance approximation spaces. Fundamenta Informaticae **27** (1996) 245–253
6. Salton, G.: Automatic text processing: the transformation, analysis, and retrieval of information by computer. Addison-Wesley Longman Publishing Co., Inc. (1989)
7. Zamir, O., Etzioni, O.: Grouper: a dynamic clustering interface to web search results. Computer Networks (Amsterdam, Netherlands: 1999) **31** (1999) 1361–1374