International Research Journal of Computer Science (IRJCS) ISSN: Applied
Volume 1 Issue 1 (September 2014) www.irjcs.com

Conceptual Algorithm for Clustering Search Result

Supriya Domkundwar
PG student, SCOE Vadgaon BK,

Prof.Kirti Korabu
Asso.Prof, SCOE Vadgaon BK

ABSTRACT - ***Search Result Clustering problem means clustering of search result returned by search engine. Web contains large amount of data. So finding relevant information is a difficult task. Lingo algorithm is used for clustering search result. This algorithm finds out meaningful cluster label. Documents are grouped under that cluster label. It retrieves the document based on conceptual content rather than term matching method.***

The Lingo algorithm combines common phrase discovery and latent semantic indexing techniques to separate search results into meaningful groups. It looks for meaningful phrases to use as cluster labels and then assigns documents to the labels to form groups. Lingo algorithm is different from traditional method because traditional method groups the documents first. After this it decides the cluster label. So Lingo is description-comes-first algorithm. In this paper we have proposed Lingo Algorithm for clustering search result.

Keywords- Latent semantic indexing, Singular value decomposition, Vector space model.

I. INTRODUCTION

With the growth of internet, it has become difficult for user to find relevant document. In response to user's query, Search engines returns list of documents. So the task of finding relevant information becomes difficult for user.
Web search is difficult because it is hard for users to construct queries that are both sufficiently descriptive and sufficiently discriminating to find just the web pages that are relevant to the user's search goal. Queries are often ambiguous: words and phrases are frequently polysemantic and user search goals are often narrower in scope than the queries used to express them. This ambiguity leads to search result sets containing distinct page groups that meet different user search goals. Often users must refine their search by modifying the query to filter out the irrelevant results. Users must understand the result set to refine queries effectively; but this is time consuming, if the result set is unorganised.Web page clustering is one approach for assisting users to both comprehend the result set and to refine the query. Web page clustering identifies semantically meaningful groups of web pages and presents these to the user as clusters. The clusters provide an overview of the contents of the result set and when a cluster is selected the result set is refined to just the relevant pages in that cluster. Here, we have proposed Lingo's algorithm for clustering of documents returned from web. This clustering algorithm used for groping of similar documents in search result list returned from search engine. This algorithm finds out the cluster label. Cluster label should be meaningful. After finding cluster label documents are grouped under that label.

RELATED WORK

Originally derived from full-text clustering and classification, topic-grouping of search results has its subtleties. Contextual descriptions (snippets) of documents returned by a search engine are short, often incomplete, and highly biased toward the query, so establishing a notion of proximity between documents is a challenging task. Clustering systems initially used classic information retrieval
algorithms, which converted documents to a term-document matrix before clustering. We use the same technique but combine clustering and smart cluster label induction to provide stronger cluster descriptions. The Scatter-Gather system [1], For example, used the Buckshot-fractionation algorithm. Other researchers used agglomerative hierarchical clustering (AHC) but replaced single terms with lexical affinities (2-grams of words) as features [2].
Unfortunately, strictly numerical algorithms require more data than is available in a search result. Raw numerical outcome is also difficult to convert back to a cluster description that human users can understand. Phrase-based methods evolved to address this problem. The suffix tree clustering (STC) algorithm [3] and the Multisearch Engine with Multiple Clustering system [4] form clusters based on recurring phrases instead of numerical frequencies of isolated terms. STC, for instance, implicitly assumes correlation between a document's topic and its most frequent phrases. Clustering in STC is thus basically finding groups of documents sharing a high ratio of frequent phrases; cluster descriptions are a subset of the same phrases used to form the cluster.

Phrase-based methods, albeit simple, usually yield good results. Unfortunately, when one topic highly outnumbers others, the algorithms usually discard smaller clusters as insignificant. Recently, researchers have applied matrix decomposition methods to the term-document matrix to fully explore the underlying latent topic structure and provide a diverse cluster structure. For example, researchers have used singular value decomposition for this purpose,[5] and nonnegative matrix factorization in a more general context of document clustering.[6]

To our knowledge, no other researchers have successfully integrated numerical and phrase-based methods. Lingo bridges existing phrase-based methods with numerical cluster analysis to form readable and diverse cluster descriptions.

II. System Architecture

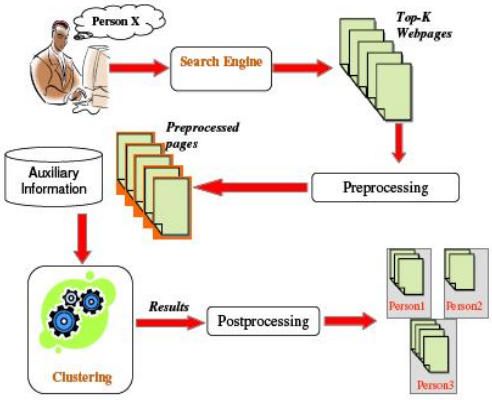

Fig 1. Overview of Processing Steps

The architecture is a pipeline that receives the input query, obtains search results from a search engine, filters the results applying a clustering algorithm and then gets the clusters. The steps of overall approach are illustrated in Fig 1. Here we have used Lingo's Algorithm for clustering of search result. Top K pages are retrieved using search engines like Google. Pages are also retrieved from document management system using term frequency inverse document frequency formula.

III. Algorithm

1. Preprocessing Phase:

After retrieving the top pages related to the query, the pages are processed by using IR techniques. There are various algorithms which are simply a set of instructions, usually mathematical, used to calculate a certain parameter and perform some type of data processing. The job is to generate a set of highly relevant documents for any search query. Fig 2 shows the preprocessing of the web pages which include the two processes named as stemming & stop word removal.

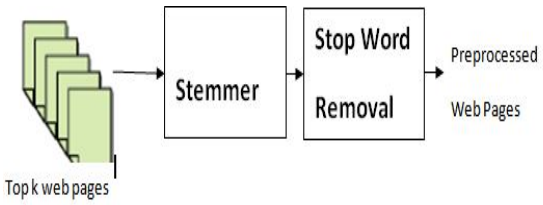

Fig 2.Preprocessing Phase

Stemming:

Stemming algorithms are used to transform the words in texts into their grammatical root form, and are mainly used to improve the Information Retrieval System's efficiency. To stem a word is to reduce it to a more general form, possibly its root. Porter Stemming algorithm is used for Stemming.

Elimination of Stop words

After stemming it is necessary to remove unwanted words. There are 400 to 500 types of stop words such as "of", "and", "the," etc., that provide no useful information about the document's topic. Stop-word removal is the process of removing these words.

2. Frequent Phrase Extraction

The frequent phrases are defined as recurring ordered sequences of terms appearing in the input documents. Intuitively, when writing about something, we usually repeat the subject-related keywords to keep a reader's attention. Obviously, in a good writing style it is common to use synonymy and pronouns and thus avoid annoying repetition. The Lingo can partially

overcome the former by using the SVD-decomposed term document matrix to identify abstract concepts—single subjects or groups of related subjects that are cognitively different from other abstract concepts.

To be a candidate for a cluster label, a frequent phrase or a single term must:

1. Appear in the input documents at least certain number of times (term frequency threshold),
2. Not cross sentence boundaries,
3. Be a complete phrase (see definition below),
4. Not begin nor end with a stop word.

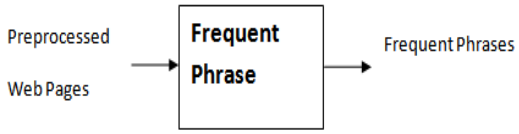

Fig.3 Phrase Extraction Phase

3. Cluster label induction

Once frequent phrases (and single frequent terms) that exceed term frequency thresholds are known, they are used for cluster label induction. There are three steps to this: term-document matrix building, abstract concept discovery, phrase matching and label pruning.

The term-document matrix is constructed out of single terms that exceed a predefined term frequency threshold. Weight of each term is calculated using the standard term frequency, inverse document frequency (tf-idf) formula, terms appearing in document titles are additionally scaled by a constant factor. In abstract concept discovery, Singular Value Decomposition method is applied to the term-document matrix to find its orthogonal basis. The vectors of this basis (SVD's U matrix) supposedly represent the abstract concepts appearing in the input documents. It should be noted, however, that only the first k vectors of matrix U are used in the further phases of the algorithm. We estimate the value of k by selecting the Frobenius norms of the term-document matrix A and its k-rank approximation Ak. Let threshold q be a percentage-expressed value that determines to what extent the k-rank approximation should retain the original information in matrix A. We hence define k as the minimum value that satisfies the following condition: $||A_k||F / ||A||F \geq q$, where $||X||F$ symbol denotes the Frobenius norm of matrix X. Clearly, the larger the value of q the more cluster candidates will be induced. The choice of the optimal value for this parameter ultimately depends on the users' preferences. Therefore make it one of Lingo's control thresholds—Candidate Label Threshold.

Phrase matching and label pruning step, where group descriptions are discovered, relies on an important observation that both abstract concepts and frequent phrases are expressed in the same vector space—the column space of the original term-document matrix A. Thus, the classic cosine distance can be used to calculate how "close" a phrase or a single term is to an abstract concept. Let us denote by P a matrix of size $t \times (p+t)$ where t is the number of frequent terms and p is the number of frequent phrases. P can be easily built by treating phrases and keywords as pseudo-documents and using one of the term weighting schemes. Having the P matrix and the i^{th} column vector of the SVD's U matrix, a vector mi of cosines of the angles between the i^{th} abstract concept vector and the phrase vectors can be calculated: $m_i = U_i^T P$. The phrase that corresponds to the maximum component of the mi vector should be selected as the human-readable description of ith abstract concept. Additionally, the value of the cosine becomes the score of the cluster label candidate. A similar process for a single abstract concept can be extended to the entire U_k matrix—a single matrix multiplication $M = U_k^T P$ yields the result for all pairs of abstract concepts and frequent phrases. On one hand we want to generalize information from separate documents, but on the other we want to make it as narrow as possible at the cluster description level. Thus, the final step of label induction is to prune overlapping label descriptions. Let V be a vector of cluster label candidates and their scores. We create another term-document matrix Z, where cluster label candidates serve as documents. After column length normalization calculates ZTZ, which yields a matrix of similarities between cluster labels. For each row we then pick columns that exceed the Label Similarity Threshold and discard all but one cluster label candidate with the maximum score.

4. Cluster content discovery

In the cluster content discovery phase, the classic Vector Space Model (VSM) is used to assign the input documents to the cluster labels induced in the previous phase. In a way, re-query the input document set with all induced cluster labels. The assignment process resembles document retrieval based on the VSM model. Let us define matrix Q, in which each cluster label is represented as a column vector. Let $C = Q^T A$, where A is the original term-document matrix for input documents. This way, element c_{ij} of the C matrix indicates the strength of membership of the j^{th} document to the i^{th} cluster. A document is added to a cluster if c_{ij} exceeds the Snippet Assignment Threshold, yet another control parameter of the algorithm. Documents not assigned to any cluster end up in an artificial cluster called others.

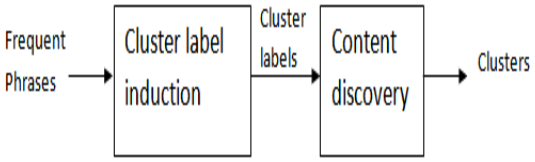

Fig.4 Cluster content discovery

5. Final cluster formation

Finally, clusters are sorted for display based on their score, calculated using the following simple formula: Cscore = label score × ||C||, where ||C|| is the number of documents assigned to cluster C. The scoring function, although simple, prefers well-described and relatively large groups over smaller, possibly noisy ones. For the time being, no cluster merging strategy or hierarchy induction is used for Lingo.

IV. Conclusion

We have presented a novel algorithm for clustering of Web search results. The inspiration for the algorithm was taken from both existing scientific work, and a commercial system—Vivisimo.Our algorithm, however, took a different path in many areas. Specifically, our contribution is in presenting a description-comes-first algorithm; to our best knowledge, no similar algorithms have been published so far. Lingo achieves impressing empirical results, but the work on the algorithm is obviously not finished. Cluster label pruning phase could be improved by adding elements of linguistic recognition of nonsensical phrases. Topic separation phase currently requires computationally expensive algebraic transformations—incremental approaches with small memory footprint would be of great importance for algorithm's scalability. It is tempting to find a method of inducing hierarchical relationships between topics. Finally, a more elaborate evaluation technique will be necessary to establish weak points in the algorithm

V. References

1. M.A. Hearst and J.O. Pedersen, "Reexamining the Cluster Hypothesis: Scatter/Gather on Retrieval Results," Proc. 19th ACM SIGIR Int'l Conf. Research and Development in Information Retrieval, ACM Press, 1996, pp. 76–84.
2. Y.S. Maarek et al., Ephemeral Document Clustering for Web Applications,tech. report RJ 10186, IBM Research, 2000.
3. O. Zamir and O. Etzioni, "Grouper: A Dynamic Clustering Interface to Web Search Results," Computer Networks, vol. 31, no. 11–16,1999, pp. 1361–1374.
4. P. Hannappel, R. Klapsing, and G. Neumann, "MSEEC: A Multisearch Engine with Multiple Clustering," Proc. 99 Information Resources Management Assoc. Int'l Conf., Idea Group Publishing,1999.
5. Z. Dong, Towards Web Information Clustering, doctoral dissertation,Southeast Univ., Nanjing, China, 2002.
6. W. Xu, X. Liu, and Y. Gong, "Document Clustering Based on NonnegativeMatrix Factorization," Proc. 26th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval, ACM Press, 2003, pp. 267–273.
7. Stanis law Osinski, Jerzy Stefanowski, and Dawid Weiss. Lingo: Search Results Clustering Algorithm Based on Singular Value Decomposition.

