IMPROVED SEARCH ENGINE USING CLUSTER ONTOLOGY

Gauri Suresh Bhagat, Mrunal S. Bewoor, Suhas Patil
Computer Department, Bharati Vidyapeeth Demeed University College of Engineering, Pune
Maharashtra, India

ABSTRACT

Search engine such as Google and yahoo returns a list of web pages that match the user query. It is very difficult for the user to find relevant web pages. Cluster based search engine can provide significantly more powerful models for searching a user query. Clustering is a process of forming groups (clusters) of similar objects from a given set of inputs. When applied to web search results, clustering can be perceived as a way of organising the results into a number of easily brows able thematic groups. In this paper, we propose a new approach for applying background knowledge during pre-processing in order to improve clustering results and allow for selection between results. We preprocess our input data applying an ontology-based heuristics for feature selection and feature aggregation. The inexperienced users, who may have difficulties in formulating a precise query, can be helped in identifying the actual information of interest. Clustering are readable and unambiguous descriptions (labels) of the thematic groups. They provide the users with an overview of the topics covered in the results and help them identify the specific group of documents they were looking for.

KEYWORDS: *Cluster, stemming, stop words, Cluster label induction, Frequent Phrase Extraction, cluster content discovery.*

I. INTRODUCTION

With an enormous growth of the Internet it has become very difficult for the users to find relevant documents. In response to the user's query, currently available search engines return a ranked list of documents along with their partial content. If the query is general, it is extremely difficult to identify the specific document which the user is interested in. The users are forced to sift through a long list of off-topic documents. For example When "java Map" query submitted to Cluster based search engine The result set spans two categories, namely the Java map collection classes and maps for the Indonesian island Java. Generally speaking, the computer science student would be most likely interested in the Java map collection classes, whereas the geography student would be interested in locating maps for the Indonesian island Java. The solution is that for each such web page, the search-engine could determine which real entity the page refers to. This information can be used to provide a capability of clustered search, where instead of a list of web pages of (possibly) multiple entities with the same name, the results are clustered by associating each cluster to a real entity. The clusters can be returned in a ranked order determined by aggregating the rank of the web pages that constitute the cluster.

II. RELATED WORK

The Kalashnikov et al. Have developed a disambiguation algorithm & then studied its impact on people search [1]. The Author has proposed algorithm that use Extraction techniques to extracts entities such as names, organizations locations on each web page. The algorithm analyses several types of information like attributes, interconnections that exist among entities in the Entity-Relationship Graph.If the multiple people name web pages merged into same cluster it is difficult for

user to find relevant web pages. For the disambiguating people that have same name a novel algorithm is developed.

The Kalashnikov et al. have, discuss a Web People Search approach which is based on collecting co-occurrence information from web to make clustering decisions [2]. To classify the collected co-occurrence information a sky-line based classification technique is used.

Bekkerman and Zilberstein have proposed framework makes the heuristic search viable in the vast domain of the WWW and applicable to clustering of Web search results and to Web appearance disambiguation [3].

Chen and Kalashnikov have, presented graphical approach for entity resolution. The overall idea behind this is to use relationships & to look at the direct and indirect (long) relationships that exist between specific pairs of entity representations in order to make a disambiguation decision. In terms of the entity-relationship graph that means analyzing paths that exist between various pairs of nodes [4].

III. DESIGN OF PREPROCESSING OF WEB PAGES

The preprocessing of the web pages which include the two processing named as stemming and stops word removal. 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. For example, stemming the term may produce the term interest. Though the stem of a word might not be its root, we want all words that have the same stem to have the same root. The effect of stemming on searches of English document collections has been tested extensively. Several algorithms exist with different techniques. The most widely used is the Porter Stemming algorithm. In some contexts, stemmers such as the Porter stemmer improve precision/recall scores .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. Stop-words account for about 20% of all words in a typical document. These techniques greatly reduce the size of the search engine's index. Stemming alone can reduce the size of an index by nearly 40%. To compare a webpage with another webpage, all unnecessary content must be removed and the text put into an array.

When designing a Cluster Based Web Search, special attention must be paid to ensuring that both content and description (labels) of the resulting groups are meaningful to humans. As stated, "a good cluster—or document grouping—is one, which possesses a good, readable description". There are various algorithms such as K means, K-medoid but this algorithm require as input the number of clusters. A Correlation Clustering (CC) algorithm is employed which utilizes supervised learning. The key feature of Correlation Clustering (CC) algorithm is that it generates the number of clusters based on the labeling itself & not necessary to give it as input but it is best suitable when query is person names[9]. For general query, the algorithms are Query Directed Web Page Clustering (QDC), Suffix Tree Clustering (STC), Lingo, and Semantic Online Hierarchical Clustering (SHOC)[5].The focus is made on Lingo because the QDC considers only the single words. The STC tends to remove longer high quality phrases, leaving only less informative & shorter ones. So, if a document does not include any of the extracted phrases it will not be included in results although it may still be relevant. To overcome the STC's low quality phrases problem, in SHOC introduce two novel concepts: complete phrases and a continuous cluster definition. The drawback of SHOC is that it provides vague threshold value which is used to describe the resulting cluster. Also in many cases, it produces unintuitive continuous clusters. The majority of open text clustering algorithms follows a scheme where cluster content discovery is performed first, and then, based on the content, the labels are determined. But very often intricate measures of similarity among documents do not correspond well with plain human understanding of what a cluster's "glue" element has been. To avoid such problems Lingo reverses this process—first attempt to ensure that we can create a human-perceivable cluster label and only then assign documents to it. Specifically, extract frequent phrases from the input documents, hoping they are the most informative source of human-readable topic descriptions. Next, by performing reduction of the original term-document matrix using Singular Value Decomposition (SVD), try to discover any existing latent structure of diverse topics in the search result. Finally,

match group descriptions with the extracted topics and assign relevant documents to them. The detail description of Lingo algorithm is in [4].

IV. 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.

A complete phrase is a complete substring of the collated text of the input documents, defined in the following way: Let T be a sequence of elements $(t_1, t_2, t_3 \ldots t_n)$. S is a complete substring of T when S occurs in k distinct positions $p_1, p_2, p_3 \ldots p_k$ in T and $\exists i, j \in 1 \ldots k : t_{pi-1} \neq t_{pj-1}$ (left completeness) and $\exists i, j \in 1 \ldots k : t_{pi+|S|} \neq t_{pj+|S|}$ (right-completeness). In other words, a complete phrase cannot be "extended" by adding preceding or trailing elements, because at least one of these elements is different from the rest. An efficient algorithm for discovering complete phrases was proposed in [11].

V. 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 (tfidf) formula [12], 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. As discussed earlier, 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 A_k. 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.

VI. CLUSTER CONTENT DISCOVERY

In the cluster content discovery phase, the classic Vector Space Model is used to assign the input documents to the cluster labels induced in the previous phase. In a way, we 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.

VII. FINAL CLUSTER FORMATION

Clusters are sorted for display based on their score, calculated using the following simple formula: S_{core} = 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.

VIII. ONTOLOGY

Let tf(d, t) be the absolute frequency of term t ϵ T in document d ϵ D, where D is the set of documents and $T = \{t_1,..., t_m\}$ is the set all different terms occurring in D. We denote the term vectors $\overrightarrow{td} =$ ((tf(d, t1),....., tf(d,t_m)). Later on, we will need the notion of the centroid of a set X of term vectors. It

is defined as the mean value . As initial approach we have produced this standard representation of the texts by term vectors. The initial term vectors are further modified as follows.

Stopwords are words which are considered as non–descriptive within a bag–of–words approach. Following common practice, we removed stopwords from T.

We have processed our text documents using the Porter stemmer. We used the *stemmed terms* to construct a vector representation $\overrightarrow{td}$ for each text document. Then, we have investigated how pruning rare terms affects results. Depending on a pre-defined threshold δ, a term t is discarded from the representation (i. e., from the set T), if $\sum_{d \in D}$ tf (d,t) ≤ δ. We have used the values 0, 5 and 30 for δ. The rationale behind pruning is that infrequent terms do not help for identifying appropriate clusters.

Tfidf weighs the frequency of a term in a document with a factor that discounts its importance when it appears in almost all documents[14]. The tfidf (term frequency-inverted document frequency) of term t in document d is defined by:

$$\text{tfidf}(d,t) = \log(\text{tf}(d,t)+1) * \log\left(\frac{|D|}{\text{df}(t)}\right)$$

where df(*t*) is the document frequency of term *t* that counts in how many documents term *t* appears If tfidf weighting is applied then we replace the term vectors $\overrightarrow{td}$ = ((tf(d, t1),....., tf(d,t_m)) by $\overrightarrow{td}$ = ((tfidf(d, t1),....., tfidf(d,t_m)) [13]. A core ontology is a tuple O := (C, $\leq_C$) consisting of a set C whose elements are called concept identifiers, and a partial order $\leq_C$ on C, called concept hierarchy or taxonomy . This definition allows for a very generic approach towards using ontologies for clustering.

IX. RESULTS AND DISCUSSION

The system was implemented using Net bean 6.5.1 as development tool & Jdk 1.6 development Platform .Also it was tested for variety of queries under following four categories and the results obtained where satisfactory.

9.1 Web pages retrieval for the query

This module gives the facilities for specifying the various queries to the middleware. The front end developed so far is as follows. The Figure 1 shows user interface, by using that the user enters the query to the middleware. Along with the query, user can also select the number of results (50/100/150/200) to be fetched from source. In Figure.1, query entered is “mouse” & result selected is 100.The user issues a query to the system (middleware) sends a query to a search engine, such as Google, and retrieves the top-K returned web pages. This is a standard step performed by most of the current systems. The Figure1 shows that the 200 results were fetched from the source Google for query “mouse” Input: Query “mouse” & k=50/100/150/200 page. Output: Web pages of Query “mouse”.

The system was assessed for a number of real-world queries; also analyzed the results obtained from our system with respect to certain characteristics of the input data. The queries are mainly categorized in four types such as Ambiguous Query, General Query, Compound Query, People Name, The system was tested for all these queries & the result obtained is satisfactory.

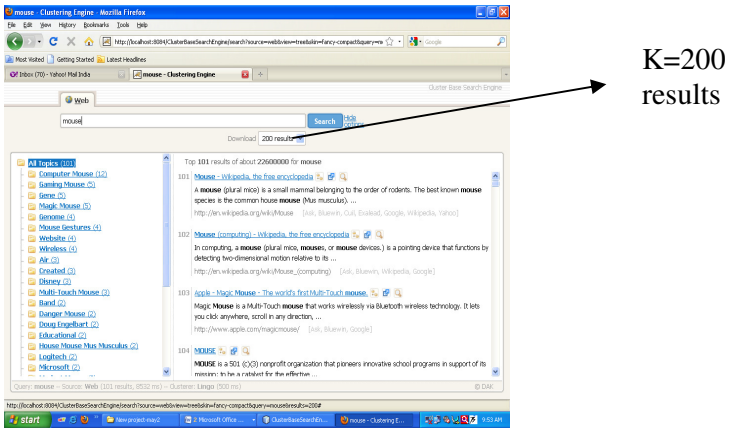

Figure 1. Clustering results for a ambiguous query "mouse" & k=200 results

X. QUALITY OF GROUP IDENTIFICATION

Figure 1 demonstrates the overall disambiguation quality results on WWW 2005 and WEPS data sets. We also compare the results with the top runners in the WEPS challenge [6]. The first runner in the challenge reports 0.78 for Fp and 0.70 for B-cubed measures. The proposed algorithm outperforms all of the WEPS challenge algorithms. The improvement is achieved since the proposed disambiguation method is simply capable of analyzing more information, hidden in the data sets, and which [8] and [7] do not analyze. That algorithm outperforms [7] by 11.8 percent of F-measure, as illustrated in Table 1 and Table 2. In this experiment, F-measure is computed the same way as in [7].The field "#W" in Table 1. is the number of the to-be found web pages related to the namesake of interest. The field "#C" is the number of web pages found correctly and the field "#I" is the number of pages found incorrectly in the resulting groups. The baseline algorithm also outperforms the algorithm proposed in [7].

Table 1. F- Measures Using WWW'05 Algo.

Name	#W	WWW'05 Algo.		
		#C	#I	F-measure
Adam cheyer	96	62	0	78.5
William cohen	6	6	4	75.0
Steve hardt	64	16	2	39.0
David Israel	20	19	4	88.4
Leslie kaelbling	88	84	1	97.1
Bill Mark	11	6	9	46.2
Mouse	54	54	2	98.2
Apple	15	14	5	82.4
David Mulford	1	1	0	100.0
Java	32	30	6	88.2
Jobs	32	21	14	62.7
Gauri	1	0	1	0.0
Overall	**455**	**313**	**47**	**80.3**

F-measure: let Si be the set of the correct web pages for cluster-i and A_i be the set of web pages assigned to cluster-i by the algorithm .Then, $\text{Precision}_i = \frac{|Ai \cap Si|}{|Ai|}$, $\text{Recall}_i = \frac{|Ai \cap Si|}{|Si|}$ and F is their harmonic mean[10]. And F_p is referred to as $F\alpha = 0.5$ [8].

Table 2. F- Measures using Baseline Algo

Name	#W	Baseline Algo		
		#C	#I	F-measure
Adam cheyer	96	75	1	87.2(+8.7)
William cohen	6	5	0	90.9(+15.9)
Steve hardt	64	40	7	72.1(+33.1)
David Israel	20	14	2	77.8(-10.6)

Leslie kaelbling	88	66	0	85.7(-11.4)
Bill Mark	11	9	17	48.6(+2.4)
Mouse	54	52	0	98.1(-0.1)
Apple	15	15	2	93.8(+11.4)
David Mulford	1	0	1	0.0(-100.0)
Java	32	27	1	90.0(+1.8)
Jobs	32	23	17	63.9(+1.2)
Gauri	1	1	0	100.0(+100.0)
Overall	**455**	**327**	**47**	**82.4(+2.1)**

Table 3. F-Measure using Cluster Based Algo

Name	**#W**	**Cluster based Algo.**		
		#C	**#I**	**F-measure**
Adam cheyer	96	94	0	98.9(+20.4)
William cohen	6	4	0	80.0(+5.0)
Steve hardt	64	51	2	87.2(+48.2)
David Israel	20	17	2	87.8(-1.2)
Leslie kaelbling	88	88	1	99.4(+2.3)
Bill Mark	11	8	1	80.0(+33.8)
Mouse	54	54	1	99.1(+0.9)
Apple	15	12	5	75.0(-7.4)
David Mulford	1	1	0	100.0(+0.0)
Java	32	25	1	86.2(-2.0)
Jobs	32	25	11	73.5(+10.8)
Gauri	1	0	0	0.0(+0.0)
Overall	**455**	**379**	**24**	**92.1(+11.8)**

XI. CONCLUSION

The number of outputs processed for a single query is likely to have impact on two major aspects of the results: the quality of groups' description and the time spent on clustering .The focus is made on the evaluation of usefulness of generated clusters. The term usefulness involves very subjective judgments of the clustering results. For each created cluster, based on its label, decided whether the cluster is useful or not. Useful groups would most likely have concise and meaningful labels, while the useless ones would have been given either ambiguous or senseless. For each cluster individually, for each snippet from this cluster, judged the extent to which the result fits its group's description. A very well matching result would contain exactly the information suggested by the cluster label.

ACKNOWLEDGEMENTS

We would like to acknowledge and extend our heartfelt gratitude to the following persons who have made the completion of this paper possible: my guide Prof. M.S.Bewoor and Our H. O. D, Dr. Suhas H. Patil for his vital encouragement and support. Most especially to our family and friends and to God, who made all things possible!

REFERENCES

[1] D.V. Kalashnikov, S.Mehrotra, R.N.Turenand Z.Chen, "Web People Search via Connection Analysis" IEEE Transactions on Knowledge and data engg.Vol 20,No11,November 2008.

[2] D.V. Kalashnikov, S. Mehrotra, Z. Chen, R. Nuray-Turan, and N.Ashish, "Disambiguation Algorithm for People Search on the Web," Proc. IEEE Int'l Conf. Data Eng. (ICDE '07), Apr. 2007.

[3] R. Bekkerman, S. Zilberstein, and J. Allan, "Web Page Clustering Using Heuristic Search in the Web Graph," Proc. Int'l Joint Conf. Artificial Intelligence (IJCAI), 2007.

[4] Z. Chen, D.V. Kalashnikov, and S. Mehrotra, "Adaptive Graphical Approach to Entity Resolution," Proc. ACM IEEE Joint Conf. Digital Libraries (JCDL), 2007.

[5] Zamir, O.E.: Clustering Web Documents: A Phrase-Based Method for GroupingSearch Engine Results. PhD thesis, University of Washington (1999).

[6] J. Artiles, J. Gonzalo, and S. Sekine, "The SemEval-2007 WePSEvaluation: Establishing a Benchmark for the Web People Search Task," Proc. Int'l Workshop Semantic Evaluations (SemEval '07), June 2007.

[7] R. Bekkerman and A. McCallum, "Disambiguating Web Appearancesof People in a Social Network," Proc. Int'l World Wide Web Conf. (WWW), 2005.

[8] J. Artiles, J. Gonzalo, and F. Verdejo, "A Testbed for People Searching Strategies in the WWW," Proc. SIGIR, 2005.

[9] N. Bansal, A. Blum, and S. Chawla, "Correlation Clustering,"Foundations of Computer Science, pp. 238-247, 2002.

[10] D.V.Kalashnikov, S.Mehrotra, R.N.Turenand Z.Chen, "Web People Search via Connection Analysis" IEEE Transactions on Knowledge and data engg.Vol 20,No11,November 2008.

[11] Zhang Dong. Towards Web Information Clustering. PhD thesis, Southeast University, Nanjing, China, 2002.

[12] Gerard Salton. Automatic Text Processing — The Transformation, Analysis, and Retrieval of Information by Computer. Addison–Wesley, 1989.

[13] G. Amati, C. Carpineto, and G. Romano. Fub at trec-10 web track: A probabilistic framework for topic relevance term weighting. In *The Tenth Text Retrieval Conference (TREC 2001)*. National Institute of Standards and Technology (NIST), online publication, 2001.

[14] Hotho A., Staab S. and Stumme G, (2003) WordNet improves text document clustering, Proc. of the SIGIR 2003 Semantic Web Workshop, Pp. 541-544.

Authors

Gauri S. Bhagat is a student of M.Tech in Computer Engineering, Bharati Vidyapeeth Deemed University College of Engg, Pune-43.

M. S. Bewoor working as an Associate Professor in Computer Engineering Bharati Vidyapeeth Deemed University college of Engg, Pune-43.She is having total 10 years of teaching experience.

S. H. Patil working as a Professor and Head of Department in Computer engineering, Bharati Vidyapeeth Deemed University college of Engg,Pune-43. He is having total 22 years of teaching experience & working as HOD from last ten years.