A New Approach to Search Result Clustering and Labeling

Anil Turel and Fazli Can

Bilkent Information Retrieval Group
Computer Engineering Department, Bilkent University
Bilkent, Ankara 06800, Turkey
aturel@cs.bilkent.edu.tr
canf@cs.bilkent.edu.tr

Abstract. Search engines present query results as a long ordered list of web snippets divided into several pages. Post-processing of retrieval results for easier access of desired information is an important research problem. In this paper, we present a novel search result clustering approach to split the long list of documents returned by search engines into meaningfully grouped and labeled clusters. Our method emphasizes clustering quality by using cover coefficient-based and sequential k-means clustering algorithms. A cluster labeling method based on term weighting is also introduced for reflecting cluster contents. In addition, we present a new metric that employs precision and recall to assess the success of cluster labeling. We adopt a comparative strategy to derive the relative performance of the proposed method with respect to two prominent search result clustering methods: Suffix Tree Clustering and Lingo. Experimental results in the publicly available AMBIENT and ODP-239 datasets show that our method can successfully achieve both clustering and labeling tasks.

Keywords: Cluster labeling, search result clustering, web information retrieval

1 Introduction

The utility of search result clustering (SRC) and associated cluster labeling algorithms for easy access to the query results has been widely investigated [6]. Without a proper arrangement of search results, finding the desired query result among ranked list of document snippets is usually difficult for most users. This problem is further aggravated when the query belongs to a general topic which contains documents from a variety of subtopics. At this point, the burden of solving inter-relations among documents and extracting the relevant ones are left to the user. More recently; however, there are continuous research and commercial efforts for developing online search result clustering and labeling methods [6].

Even though there exists some search result clustering algorithms, embedding these methods in search engines is not a common practice. There are three main reasons behind this problem: (1) existing algorithms are not able to capture the relationships among documents since the snippets are too short to convey enough information about query subtopics; (2) finding descriptive and meaningful labels for clusters is a difficult problem; (3) the evaluation of SRC task is not well-defined. Motivated by these observations, we present a new search result clustering method based on cover coefficient (C^3M) [3] and sequential k-means clustering algorithms [14].

Early works on the SRC problem include the Scatter-Gather system [13], Suffix Tree Clustering (STC) [22], and Lingo [16]. Apart from those, MSEEC [12] and SHOC [9] also contribute to the use of words proximity in the input documents. Clustering web results is also essential for mobile devices since it decreases the amount of information transmitted, provides a more effective and informative user interface that require less interactions in terms of page scroll or query reformulation [5] [6]. Search result diversification is another approach to post-processing of search results. Related studies re-rank search results for presenting documents from different subtopics at the beginning of search results list [4] which is similar to but different from the SRC problem. Although SRC seems as a subset of document clustering, it has distinguishing constraints coming from efficiency, effectiveness and labeling quality requirements [6]. While both keyword extraction and labeling task of SRC are based on frequent phrases, labeling differentiates from keyword extraction with efficiency requirement it possesses.

Note that, among all SRC methods, for comparison we study two prominent algorithms; Lingo [16] and STC [22]. Lingo uses singular value decomposition to generate cluster descriptions that are crucial for user-friendly search engines. The Lingo method is currently being used in Carrot^2 open source search result clustering engine [21]. Besides, STC introduced in [22] is based on suffix tree data structure that enables the usage of phrases instead of single words as cluster labels. In this method, clustering and labeling steps are accomplished using suffix tree.

Our search result clustering method, C^3M+K-means is based on C^3M and sequential k-means algorithms. The adaptation of these two methods to the search result clustering problem is one of the contributions of this paper. Additionally, a new labeling approach "labeling via term weighting" is introduced. The key contribution of this paper is the labeling evaluation strategy. To assess the effectiveness of cluster labeling, we introduce a new metric called $\text{sim}_{\text{F-measure}}$, by employing precision and recall. We provide experimental results by systematically evaluating the performance of our method in the AMBIENT [7] and ODP-239 [8] test collections. We show that our method can successfully achieve both clustering and labeling tasks [19].

2 An Approach to Search Result Clustering and Labeling

The methodology we use in this study is to extract the relationships among documents with C^3M method and to construct the final clusters through feeding the results of C^{3}M to the sequential k-means algorithm. We then use our term weighting-based approach to label the generated clusters.

2.1 Preprocessing

The first step is to clean the document text from non-letter characters and to convert all characters to lower case. Afterwards, stopwords are eliminated and stemming is applied by the Porter Stemmer [18]. Finally, the terms appearing in the 3-30% of the snippets constitute the term list (that is used for document description).

In order to generate meaningful cluster labels, phrase discovery is a crucial phase in SRC problem. Most of the time, a combination of words, namely, phrases are needed to reflect the cluster content. In this study, we use suffix tree structure [22] to extract phrases from the document snippets. Suffix tree indexes sequence of words in the nodes and stores number of occurrences. Then, the inner nodes with sufficient occurrences are considered as a phrase (in our experiments nodes that occur in more than %2 of the documents are selected as phrases) and they are added to the term list.

Before passing to the clustering phase, we index each document using its terms that appear in the term list. The term weights are computed by using the log entropy formula [10] [19]. Entropy based term weighting considers the distribution of term over documents. Finally, we reduce the weights of single-word terms by multiplying them with a constant value, in our experiments 0.3, and to increase the importance of phrases, they are multiplied with 0.7. Then, we normalize the term weights of documents and the collection becomes ready for clustering [19].

2.2 Clustering

Cover coefficient-based clustering. It is a seed oriented, partitioning, single-pass, linear-time clustering algorithm introduced in [3]. The main goal of C^{3}M is to convey the relationships among documents using a two-stage probability experiment. The efficiency and effectiveness of C^{3}M for information retrieval in texts has been experimentally demonstrated in [1]. To accomplish clustering task, briefly, ten documents are selected as seed documents and for each non-seed document we check the coverage of the document with the seed documents and select the seed that has the highest coverage over the non-seed. If none of the seeds covers the non-seed document, then, it is directly added to the *Others* cluster. Detailed information about C^{3}M can be found in [3].

Modified sequential k-means algorithm. K-means is a linear-time and widely used clustering algorithm which groups given documents after the initial centroids are provided.The success rate of the k-means algorithm highly depends on the initial cluster centroids. Therefore, we use the results of C^{3}M clustering to derive the centroids as accurately as possible. The input centroids are the vectorial averages of the documents in each C^{3}M cluster.

Sequential k-means algorithm [14] updates the cluster centroid after each document assignment to the cluster instead of after all documents distributed in original k-means. We use a modified version of the sequential k-means algorithm where we assign documents to the centroids as in k-means in the first pass. Then, the centroids are re-calculated according to the new distribution of documents. At the beginning of each following pass, we empty the cluster contents. Then, we assign each document to the nearest cluster and update that cluster's centroid again as:

$$centroid_i = \frac{\sum_{j \epsilon cluster_i} doc_j + centroid_i}{|cluster_i| + 1} \quad (1)$$

where $|cluster_i|$ is the number of documents in the cluster and 1 is added to the denominator for the centroid vector in numerator.

2.3 Labeling via Term Weighting

The final step of our method is the labeling phase. We aim to assign descriptive labels to clusters that reflect their contents. This step is very important because meaningless or confusing labels may mislead users to check the wrong clusters for the query and lose extra time. We present a novel labeling strategy called *labeling via term weighting* that assesses significance of terms for clusters. Firstly, the terms of documents in a cluster are merged, then term weighting is applied to the clusters (by assuming them as documents). We use the same term weighting formula as in Section 2.1 [10] [19]. A single-word label generally lacks expressiveness, so we give more weight to phrases than single-word terms during cluster labeling as in Section 2.1. For each cluster, we select the highest weighted terms into the candidate labels list. In our experiments, we add topmost five terms to the list. While we are assigning the final labels of the clusters from these lists, we follow the criteria below:

- Clusters are labeled in descending order of cluster size,
- Label should not be one of the previously given labels to another cluster,
- Phrase label candidate with less than five words is preferred (if exists),
- Term with a higher weight is preferred.

3 Performance Measures

3.1 Clustering Evaluation

To be able to quantify clustering performance, we first need to define a success measure which reflects the actual performance of clustering results as fairly as possible, regardless of the clustering method we choose. In this paper, we use weighted average F-measure ($\mathrm{w}_{\text{F-measure}}$) [20] which is the average of total weighted F-measure of each class. Intuitively, precision reflects to what extent presented cluster includes documents of ground truth class and recall reflects to what extent ground truth class is presented to the user. The necessary equations to measure the similarity between a ground truth class i and represented cluster j are given.

$$precision(i,j) = \frac{class_i \cap cluster_j}{|cluster_j|} \tag{2}$$

$$recall(i,j) = \frac{class_i \cap cluster_j}{|class_i|} \tag{3}$$

$$F\text{-}measure(i,j) = \frac{2 \times recall(i,j) \times precision(i,j)}{recall(i,j) + precision(i,j)} \tag{4}$$

For each class in the ground truth we find the best matching cluster (that has the maximum F-measure among all clusters). We are interested in weighted F-measure to better evaluate the contribution of each class to the overall performance. Clustering performance is computed as follows.

$$w_{F\text{-}measure} = \frac{1}{\sum_{i=1}^{n_{class}} |class_i|} \sum_{i=1}^{n_{class}} \left(\max_j\{F\text{-}measure(i,j)\}\ |class_i|\right) \tag{5}$$

where n_{class} represents the number of classes.

3.2 Labeling Evaluation

Although human judgment is preferred to evaluate the labeling performance of most of the SRC methods, this approach is very expensive and difficult to repeat for different parameters. It is also difficult to compare distinct labeling methods based on human judgment. Due to such drawbacks, we propose a new labeling evaluation measure called $\mathrm{sim}_{\mathrm{F\text{-}measure}}$ based on the assessment of similarity between two labels (ground truth and generated label).

3.2.1 Comparison of Ground Truth and Generated Label We use four similarity metrics to automatically find similarity between generated label and ground truth label and they are semantic similarity, exact, partial and overlap match. Each metric reflects the labeling performance of SRC methods from different aspects. While exact match is strict to the ground truth, partial match requires the ground truth structure (also human readability) is preserved partially. Overlap match considers how close suggested labels are to the ground truth. Lastly, semantic similarity finds the indirect relationship between labels. Before applying these metrics, stopwords are eliminated and stemming is applied. If the ground truth class is *Others* cluster, and algorithm cluster is not, or vice versa, the similarity score between labels is set to 0. Similarity metrics give Boolean output; 1 for similarity and 0 for dissimilarity, except semantic similarity.

Semantic similarity. It is a research field in artificial intelligence, that aims to determine the similarity between concepts by mapping them into an ontology and investigating their relationship within the ontology. In this paper, we use semantic similarity to detect the similarity between the ground truth and proposed labels. For the experiments, we use Java WordNet Similarity Library [17] that exploits WordNet [11] as the ontology source. The semantic similarity metric outputs a similarity value within the range of 0 and 1 to quantify the measure of similarity between two labels. Although there are different formulations of this metric, we are using the approach presented in [15] that uses the information content concept of information theory. For example, in our experiments, ground truth and generated label pairs “News” - “Broadcasts” and “Sound Files” - “Streaming Audio” are found to share respectively 0.90 and 0.78 similarity according to the semantic similarity metric of [15].

Exact match. It suggests similarity if the generated label is the same as the ground truth or the generated label covers the other. To exemplify, when ground truth and generated label pair is “Instruments” - “Musical Instrument,” exact match is ensured.

Partial match. It suggests similarity if the cluster label covers the ground truth label or vice versa. For instance, the ground truth - extracted label pair “USS Coral Sea, disambiguation” - “USS Coral Sea” is accepted. The partial and exact match do not cover the case when the words in ground truth change order in generated label.

Overlap match. It aims to catch the slightest similarity between labels. If the intersection between the label and ground truth label is not empty, then the overlap match accepts the label. As an example; if the ground truth label is “Editorial Illustration,” the overlap match accepts the generated label “Digital Illustrations.”

3.2.2 Labeling Evaluation Measure: sim$_{\textbf{F-measure}}$

In order to obtain a robust labeling evaluation metric for the entire clustering structure, we introduce a new measure, sim$_{\text{F-measure}}$, based on precision and recall. It is inspired by [20]. In this formulation, similarity precision (sim$_{\text{precision}}$) represents to what extent labels presented to the user resemble ground truth labels and similarity recall (sim$_{\text{recall}}$) defines to what extent ground truth labels are reflected to the user. The methodology for computing the overall similarity can be summarized as follows. For each class in the ground truth, we find the matching cluster that gives the highest F-measure with the class. Then, we compute the similarity between the labels by using one of the similarity metrics (represented as *similarity* function in equation 6). After that, we sum up the similarity scores for all classes and normalize by the number of classes to find the sim$_{\text{recall}}$. We find the sim$_{\text{precision}}$ by applying the same procedure to the clusters. Finally, sim$_{\text{F-measure}}$ is computed as the harmonic mean of sim$_{\text{recall}}$ and sim$_{\text{precision}}$. The necessary formulation for this procedure can be derived as follows (note that all of them have a value between 0 and 1).

$$sim_i = similarity\,\{label(class_i),\ label(cluster_{\max F\text{-}measure(i,j)})\} \tag{6}$$

$$sim_j = similarity\,\{label(cluster_j),\ label(class_{\max F\text{-}measure(i,j)})\} \tag{7}$$

$$sim_{precision} = \frac{\sum_{j=1}^{n_c} sim_j}{n_c} \quad sim_{recall} = \frac{\sum_{i=1}^{n_{class}} sim_i}{n_{class}} \tag{8}$$

$$sim_{F\text{-}measure} = \frac{2 \times sim_{recall} \times sim_{precision}}{sim_{recall} + sim_{precision}} \tag{9}$$

where n_{class} and n_c are respectively the number of classes and clusters.

3.3 Experimental Results

In order to assess the the performance of clustering and cluster labeling algorithms, we perform experiments in two publicly available datasets specific to SRC task: the AMBIENT Dataset [7] and ODP-239 Dataset [8]. They consist of 44 and 239 queries, respectively and 100 snippets for each query. We present both the results of C^{3}M and C^{3}M+K-means methods to discuss the effect of using sequential k-means clustering. We use a comparative strategy to derive the relative performance of our algorithm with respect to the two state-of-the-art algorithms: Lingo and Suffix Tree Clustering (STC). Implementation of these methods are available in Carrot^2 API [21].

Clustering results. The first step of the clustering evaluation is to prove that the algorithm shows significant difference from random clustering according to the Monte Carlo method [14]. If the cluster sizes are preserved and documents are added to the clusters randomly, we obtain random clustering. A target cluster of a class contains at least one relevant document of the class. As a rule, the average number of target clusters of the clustering method should be significantly less than the average number of target clusters of random clustering [3]. The random clustering is performed 1000 times and as a result, on the average, the proposed method outperforms %97.3 (in AMBIENT) and %98.8 (in ODP-239) of the 1000 random clusterings. So we conclude that the proposed method performs significantly different from random.

Table 1. Clustering results in terms of $\mathrm{w_{F\text{-}measure}}$

Algorithm	AMBIENT	ODP-239
C^{3}M	0.444	0.386
C^{3}M+K-means	**0.603**	0.464
STC	0.413	**0.510**
Lingo	0.370	0.420

Afterwards, we test our algorithm in the AMBIENT and ODP-239 datasets by using $\mathrm{w_{F\text{-}measure}}$ success measure. Table 1 details the average results for all queries in both datasets including the results for STC and Lingo. As seen in this table, the proposed C^{3}M+K-means algorithm performs the best among all methods in the AMBIENT dataset when we look at the $\mathrm{w_{F\text{-}measure}}$ results. To prove that our results are statistically significantly different from those of the other algorithms, we also run a paired t-test over $w_{\text{F-measure}}$ scores of all queries in AMBIENT. With a threshold level of 0.01, we achieve statistical significance in our results. The proposed method ranks second in the ODP-239 dataset after STC, but the difference between the proposed method and STC is not statistically significant. Therefore, we conclude that the proposed method is successful at clustering search results. Notice that, the usage of sequential k-means as a secondary clustering mechanism after the C^3M method increases the clustering performance significantly.

Labeling results. Labeling performances of the proposed method are provided in Table 2. Success rates are shown based on the previously mentioned semantic similarity, exact, partial and overlap match similarity metrics applied on *similarity F-measure* ($\text{sim}_{\text{F-measure}}$) label evaluation measure. In contrast to the smaller exact match scores by all methods in AMBIENT relative to ODP-239, we observe higher scores in the other measures. The reason behind is that the ground truth labels, which define the meaning of ambiguous words, are too long in the AMBIENT dataset (on average 8.6, 1.63 words in AMBIENT and ODP-239 datasets, respectively). Note that scores are low by all methods because according to the labeling evaluation strategy, success of labeling depends on how good clusters are obtained.

For the AMBIENT dataset, our algorithm performs best with overlap match, while ranking second in other measures following the STC algorithm. We show the significance of these results using a t-test as described previously. In contrast, our method outperforms the other methods in all the success metrics in the ODP-239 dataset (with one exception and in that case there is a tie with STC). However, statistical significance is not observed due to the close results of the proposed method and STC. In the light of these results, it can be concluded that, the proposed method shows comparable performance on labeling clusters.

Table 2. Labeling results in terms of $\text{sim}_{\text{F-measure}}$. Similarity between labels are decided by exact (E), partial (P), overlap (O) match and semantic similarity (S) metrics.

Dataset	Algorithm	E	P	O	S
AMBIENT	C^3M	0.002	0.151	0.481	0.214
	C^3M+K-means	0.005	0.235	**0.488**	0.261
	STC	**0.086**	**0.335**	0.455	**0.331**
	Lingo	0.049	0.209	0.406	0.225
ODP-239	C^3M	0.091	0.112	0.149	0.108
	C^3M+K-means	**0.151**	**0.185**	**0.221**	**0.172**
	STC	0.119	0.176	0.195	**0.172**
	Lingo	0.112	0.144	0.168	0.137

In fact, the automatically computed similarity metrics are more strict than human judgment and they produce smaller similarity scores since they only compare with ground truth label, while human can also consider cluster content. In addition, automatic evaluation finds similarity between labels if they share words or have a relationship in the ontology, but human infer similarity intuitively, even such an association does not exists. However, the disadvantage of such an evaluation method is that the results may vary from person to person. Therefore, we can say that, using an automatic similarity metric simplifies the comparison of search result labeling methods. Inserting F-measure constraint into the computation of $\text{sim}_{\text{F-measure}}$ provides that the cluster content should match with the class content. This ensures that not only the label similarity is enough but also the documents in the cluster should be common with the ground truth subtopic.

4 Conclusion

In this paper, we propose methods for solving two key information retrieval problems; search result clustering and cluster labeling. Our study addresses the difficulty of clustering and labeling search results. Our contribution on SRC can be summarized as taking document relationships into account by using cover coefficient-based clustering method and using its results as an initial clustering structure for the sequential k-means clustering algorithm to improve the SRC performance. We experimentally show that our approach generates meaningful clustering structures.

A novel cluster labeling approach called “labeling via term weighting” is introduced. This labeling method observes both the behavior of terms within the documents of cluster and in the document collection. The key contribution of this study is the proposed labeling evaluation strategy. We introduce a new metric, similarity F-measure, by employing precision and recall, to assess the effectiveness of cluster labeling. The resemblance between the generated and ground truth labels is determined by semantic similarity, exact, partial, and overlap match metrics.

Extensive experimental results for both clustering and labeling show that the proposed method successfully cluster and label search results while maintaining a performance competitive with the two state-of-the-art methods Lingo and Suffix Tree Clustering. In our future research we plan to embed the proposed method to the information retrieval interface of Bilkent News Portal [2].

References

1. Can, F., Altingovde, I.S., Demir, E.: Efficiency and Effectiveness of Query Processing in Cluster-based Retrieval. Information Systems, vol. 29, n.8, pp. 697-717 (2004)
2. Can, F., Kocberber, S., Baglioglu, O., Kardas, S., Ocalan, H. C., Uyar, E.: Bilkent News Portal: A personalizable system with new event detection and tracking capabilities. In: Proceedings of the 31st International ACM SIGIR Conference on Research and Development in Information Retrieval, Singapore. ACM Press, pp. 885 (2008)

3. Can, F., Ozkarahan, E.A.: Concepts and Effectiveness of the Cover-Coefficient-based Clustering Methodology for Text Databases. ACM Transactions on Database Systems, vol. 15, n.4, pp. 483-517 (1990)
4. Carbonell, J., Goldstein, J.: The Use of MMR, Diversity-based Reranking for Reordering Documents and Producing Summaries. In: Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval, Melbourne, Australia. ACM Press, pp. 335-336 (1998)
5. Carpineto, C., Mizzaro, S., Romano, G., Snidero, M.: Mobile Information Retrieval with Search Results Clustering: Prototypes and evaluations. Journal of the American Society for Information Science and Technology, vol. 60 n. 5, pp. 877-895, (2009)
6. Carpineto, C., Osinski, S., Romano, R., Weiss, D.: A Survey of Web Clustering Engines. ACM Computing Surveys, vol. 41, issue 3, n. 17, ISSN:0360-0300 (2009)
7. Carpineto, C., Romano, G.: AMBIENT Dataset. http://credo.fub.it/ambient/ (2008)
8. Carpineto, C., Romano, G.: ODP239 Dataset. http://credo.fub.it/odp239/ (2009)
9. Dong, Z.: Towards Web Information Clustering. PhD thesis, Southeast University, Nanjing, China (2002)
10. Dumais., S.: Improving the Retrieval of Information from External Sources. Behavior Research Methods, Instruments, and Computers, vol. 23, n. 2, pp. 229-236 (1991)
11. Fellbaum, C.: WordNet: An Electronic Lexical Database. Cambridge, MA: MIT Press (1998)
12. Hannappel, P., Klapsing, R., Neumann, G.: MSEEC a Multi Search Engine with Multiple Clustering. In: Proceedings of the 99 Information Resources Management Association Conference (1999)
13. Hearst, M.A., Pedersen, J.O.: Reexamining the Cluster Hypothesis: Scatter/Gather on Retrieval Results. In: Proceedings of the 19th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 76-84 (1996)
14. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice Hall (1988)
15. Jiang, J., Conrath, D.: Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy, In: Proceedings of the International Conference Research on Computational Linguistics (ROCLING) (1997)
16. Osinski, S., Stefanowski, J., Weiss, D.: Lingo: Search Results Clustering Algorithm Based on Singular Value Decomposition. In: Proceedings of the International Conference on Intelligent Information Systems (2004)
17. Pirro, G.: A Semantic Similarity Metric Combining Features and Intrinsic Information Content. Data and Knowledge Engineering, vol. 68, n. 11, pp. 1289-1308 (2009)
18. Porter, M.F.: An Algorithm for Suffix Stripping. Program, vol. 14, n. 3, pp. 130-137 (1980)
19. Turel, A.: A New Approach to Search Result Clustering and Labeling. M.Sc. thesis, Bilkent University, Ankara, Turkey (2011)
20. Weiss, D.: Descriptive Clustering as a Method for Exploring Text Collections. PhD thesis, Poznań University of Technology, Poznań, Poland (2006)
21. Weiss, D., Osinski, S.: Carrot2 Open Source Search Results Clustering Engine. http://project.carrot2.org/ (2002)
22. Zamir, O., Etzioni, O.: Web Document Clustering: A Feasibility Demonstration. In: Proceedings of the 19th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp 46-54 (1998)