What was the Query? Generating Queries for Document Sets with Applications in Cluster Labeling

Matthias Hagen(✉), Maximilian Michel, and Benno Stein

Bauhaus-Universität Weimar, Weimar, Germany
{matthias.hagen,maximilian.michel,benno.stein}@uni-weimar.de

Abstract. We deal with the task of generating a query that retrieves a given set of documents. In its abstract form, this can be seen as a "compression" of the document set to a short query. But the task also has a real-world application: cluster labeling (e.g., for faceted search). Our solution to cluster labeling is the usage of queries that approximately retrieve a cluster's documents. To be generalizable, our approach does not require access to a search index but only a public interface like an API. This way, our approach can also be implemented at client side.

In an experimental evaluation, a basic version of our approach using a simple retrieval model is on par with standard cluster labeling techniques. A further user study reveals that queries as labels are often preferred when they are not too long.

1 Introduction

In this paper, we study the problem of generating a query that would retrieve a given set of documents from some search interface. At first glance, the problem itself seems rather abstract and only of theoretical interest. However, we suggest it as a means to identify good human-understandable labels for document clusters. The labels should tell the users something about the contained documents. In our opinion, many users nowadays conceptually connect search queries with document sets—the returned results. We thus exploit this connection by using as cluster labels such queries that approximately retrieve the documents from one cluster but not from the others.

Our approach does not require full access to some search index; a public interface like an API is sufficient. This way, our approach is applicable even at client side. However, the full potential can be utilized at search engine side when for instance generating search result facets that provide some clues on what the results are about. As facets are only useful with good labels, we propose to cluster the original query's result set and to provide other search queries as labels for the different clusters/facets. In this way, facets could work similar to query suggestions. By clicking on a facet, the user implicitly submits the label as a search query and is provided with a set of results—as accepted and expected by many users.

C. Biemann et al. (Eds.): NLDB 2015, LNCS 9103, pp. 124–133, 2015.
DOI: 10.1007/978-3-319-19581-0_10

In a user side scenario, the constructed queries can also be seen as a way of "compressing" a document set using the search engine as the "compression" algorithm. Instead of the whole document set, just the query could be stored. Against some retrieval system that does not change too frequently (e.g., some research search engines but probably not the big commercial search engines), the query in some sense contains all the information necessary to retrieve the document set again. However, the main use case of our approach at user side is that of labeling small to medium sized document clusterings. Our algorithm can derive queries for each cluster that approximately retrieve the documents from the respective cluster. To this end, it is not even necessary to build a fully-fledged search engine for the whole clustering but some on-the-fly computations of retrieval scores would suffice.

We envision the usage of queries as cluster labels as particularly promising due to the nature of queries. Traditional cluster labeling approaches often purely rely on text statistics. However, many users accept queries as the dominant way of retrieving a set of documents from a larger collection. Using queries as labels, we are able to go beyond the simple text statistics model of traditional cluster labeling such that we can exploit all the tools developed for effective document retrieval and make them applicable to cluster labeling itself.

In an empirical evaluation, we show our query-based labels to be on par with standard approaches. We examine the label quality with classic measures of similarity to human-generated labels (i.e., Jaccard index or F-measure) and we also develop a new semantics-aware quality measure based on ESA. Additionally, we conduct a user study to manually assess the usefulness of the generated query labels. In all experiments it turns out that queries are a good means of labeling when they are not too long.

2 Related Work

Query Formulation. Fuhr et al. suggest an optimum clustering framework based on vectors of document-query similarities [7] that inspired our idea. One way of storing such important queries for a document is the reverted index [17] that we will also employ. For deriving queries for a *single* document, several strategies from the literature [3,5,24] were shown not to perform as well as the approach by Hagen and Stein [11] that also inspired our idea. However, contrary to the above single-document query formulation approaches, our scenario requires queries that retrieve complete document *sets*. This problem was first examined by Jordan at al. [13] who used language models based on full access to corpus statistics. Instead, we are focusing on a black-box scenario where we just apply the public search engine interface. Bonchi et al. [4] deal with a scenario very similar to ours. For a given result set of a query, they want to find queries in a query log that "cover" the result set in a set-cover manner. We generalize their setting by not requiring any log information but simply relying on public interfaces as in the maximum query setting [10].

Cluster Labeling. We suggest to use queries as a new approach to cluster labeling. In general, there are two different strategies applied to cluster labeling: differential cluster labeling and cluster-internal labeling [14]. Differential cluster labeling compares term distributions within a cluster to the distributions of other clusters. A very effective such approach is based on the χ^2-test yielding labels of k terms that have a high weight according to their presence within the cluster and their "absence" outside of the cluster [6]. The cluster-internal labeling methods instead simply construct labels from the terms appearing within a cluster's *centroid* document—a prominent example being the weighted centroid approach (WCC) [21] identified as a simple yet very effective technique based on $tf \cdot idf$ weights in a recent cluster labeling comparison [15]. Our own approach will be a mixture of both general strategies: we also exploit the centroid document to identify candidate terms as a form of cluster-internal labeling but then derive queries by paying attention to the result set in comparison to the whole clustering as a form of differential cluster labeling. A drawback for both approaches (WCC and χ^2-test) is that the size k of the label (number of desired terms) has to be pre-determined whereas in our scenario it is automatically derived. Whenever the query is not descriptive enough, another term is added. We compare our query-based labels to WCC and the χ^2-test on the Ambient dataset that has been applied in different clustering studies [16,22,23].

3 Approach

We first describe our basic approach of generating a query for a given document set against a search engine interface. In the second part, we apply this approach to cluster labeling.

3.1 Generating Queries for Document Sets

The goal of generating a query for a given document set is to find a keyword (or keyphrase) combination that approximately returns the given document set from a search engine interface but not too many other documents. In a web search scenario this setting may seem rather artificial. It becomes more applicable and tractable when in the use case of cluster labeling the search engine is set up only for the documents in the clustering (typically much smaller than the web). Still, also against some web search engine, our approach is able to "compress" a given document set to a short query. In both settings, we treat the retrieval system as a black box. Thus, no real information about the employed retrieval model or about the index structure can be used. Similar to other approaches [2,12], only the public black-box search interface needs to be available.

Reverted Index. To store some information about the to-be-retrieved document set, we employ a reverted index [17]. Instead of mapping document IDs to index terms as in the traditional inverted index, the reverted index stores for each document the queries that return that document. Pickens et al. [17] originally

```
Input: document set $D$, RevertedIndex
Output: query term candidates $W_{cand}$
$Map \leftarrow \emptyset$
for all $d \in D$ do
  $W_d \leftarrow$ RevertedIndex($d$)
  for all $w \in W_d$ do
    $Map(w) \leftarrow Map(w) + 1$
$W_{cand} \leftarrow \emptyset$
for all $w \in Map$ do
  $\#d \leftarrow Map(w)$
  $weight \leftarrow \#d/|D|$
  $W_{cand} \leftarrow W_{cand} \cup \{\langle w, weight \rangle\}$
Sort $W_{cand}$ by decreasing weight
return $W_{cand}$
```

```
Input: $D$, $W_{cand}$, threshold $k$
Output: query $q$ with $D$ in top-$k$ results
$v \leftarrow 0$
$q \leftarrow \emptyset$
for all $w \in W_{cand}$ do
  $q \leftarrow q \cup \{w\}$
  $D_{top\text{-}n} \leftarrow$ top-$k$ results of $q$
  $v' \leftarrow |D_{top\text{-}n} \cap D|/|D|$
  if $v' \geq v$ then
    $v \leftarrow v'$
  else
    $q \leftarrow q \setminus \{w\}$
    break
return $q$
```

Fig. 1. Left: Identifying candidate terms. Right: Greedy combination of candidate terms.

suggest to use query logs or frequent terms as the basis queries to automatically populate the reverted index. Each returned document in the top-k results of some basis query (e.g., the top-1000 results) becomes a key for some postlist in the reverted index. The postlist contains the queries that return the document weighted by the rank at which the document appears (i.e., the first queries rank the document higher than later queries in a postlist). Note however that query logs are not always available and that using frequent terms may result in problems of retrievability [1].

Constructing the Basis Queries. Since we do not have up-to-date query logs at our disposal, we can only employ Pickens et al.'s suggestion of using frequent terms as the basis queries [17] but will adapt it to the use case of cluster labeling. Given a document set, we first automatically construct its centroid document. To this end, the documents are represented as *tf*-vectors (stopwords removed) and the centroid document is the arithmetic midpoint of the resulting vector space. One can think of the terms in the centroid document as the ones that on average appear at least once in each document. One crucial point is that in an online scenario of generating a good query for a given document set, each of the basis queries needs processing time when automatically submitted to a search engine. For a faster response time, we propose to have a cut-off value of using at most n terms for the basis queries. In a pilot study on the Ambient dataset (also used in our evaluation), the centroid document on average contained about 90 terms which we choose as the cut-off value for n.

Query Generation with the Reverted Index. The query generation using the reverted index runs in three phases: (1) constructing the reverted index on the fly for the given document set, (2) identifying candidate terms, and (3) composition of a good query from the candidates.

To construct the reverted index, we submit the centroid document's terms as basis queries. Having the reverted index at hand, we assign weights to the terms in the index according to the number of documents they retrieve from the document set and return the terms by decreasing weight. The respective algorithm is given in the left part of Fig. 1.

We can then combine the candidate terms to a final query in a third phase. The goal is to find a query that returns as many of the documents from the given document set as possible. To this end, we propose a greedy strategy (cf. the right part of Fig. 1). The algorithm adds terms from the candidate list to a query q. Whenever the returned result list does not get worse (i.e., does not return less of the documents from the given document set), the term is added to the query. Otherwise, it is dropped and the combination process stops since we expect the remaining terms to be of even worse quality given their smaller weight. If time is not an issue, the combination could also proceed in a backtracking manner and test several queries from which the shortest or otherwise best might be chosen.

3.2 Application to Cluster Labeling

The described query formulation approach can be easily transferred to the task of cluster labeling. The research question then is whether queries can serve as promising cluster labels.

Query formulation in the context of cluster labeling can be seen as a mixture of cluster-internal and differential labeling. The first phase of term selection is completely internal based on the cluster's centroid document. However, when weighting the terms and combining them to a query, the information of how many documents from different clusters are retrieved, is exploited. The constructed query for one cluster should return as many documents of that cluster but as few documents as possible from other clusters.

We view each of the candidate terms as a classifier that selects documents from the desired cluster and documents from the other clusters. As a weighting scheme, we propose the F-Measure derived from the recall of documents from the desired cluster and the precision in form of the retrieval of only few documents from other clusters. Note that these values can also be computed on the reverted index when constructed for the whole clustering. The set of documents that ideally should not be contained in the retrieved results forms a slight difference to the general query formulation from above. But apart from that slight difference (adding F-Measure weighting), the greedy combination works as described before.

4 Evaluation

We compare our new query-based cluster labeling approach to standard approaches from differential and cluster-internal labeling: the χ^2-test labeling [14] and weighted centroid covering [21]. Both performed very well in a recent cluster labeling study [15].

Our evaluation is divided into two parts. First, we compare the labels with traditional measures: Jaccard index and cosine similarity to reference labels. As a new measure taking also semantic similarity into account, we also propose an ESA-based similarity [8] of a generated and a reference label. This newly proposed measure is also a contribution in itself to cluster labeling evaluation.

Table 1. Average label quality (791 Ambient subtopics with Wikipedia disambiguation description as the reference label). The computed labels' quality is measured by the traditional measures F-Measure (precision and recall of the computed label terms against the reference), Jaccard index (overlap of computed and reference terms), and cosine similarity of the *tf*-weighted term vectors of the computed and the reference labels, as well as the newly proposed ESA similarity between the computed and the reference label. Bold font depicts the best approach in a row.

	Query generation	χ^2	Weighted centroid covering
F-Measure	0.103	**0.137**	0.056
Jaccard index	0.051	**0.068**	0.028
Cosine similarity	**0.367**	0.352	0.188
ESA similarity	**0.443**	0.434	0.311

Second, to complement the machine-computable measures, we also conduct a small-scale user study on the quality of the derived labels.

4.1 Evaluation Corpus

Our evaluation corpus is based on the Ambient dataset[1] often used in cluster evaluation [16,22,23]. The dataset contains 44 topics referring to ambiguous terms with a Wikipedia disambiguation page. The short descriptions of the 791 subtopics in the disambiguation pages form the reference labels. The original corpus contains documents obtained by submitting the 44 topics to a commercial search engine. However, since only the top-100 documents for each of the 44 topics were fetched and some topics contain as many as 37 subtopics, there are a lot of subtopics with only very few or no assigned documents. To enlarge the corpus, we submitted all the 791 subtopics as search queries to the Bing API and fetched the top-50 results for each query. Note that in the evaluation, we do not run a clustering algorithm but use the "correct" clustering given by the enlarged Ambient subtopics' document sets as the reference—a standard procedure in evaluating cluster labeling.

We set up a BM25F index [19,20] for the enlarged Ambient corpus. To simulate web-scale search, queries against this small index are also submitted to the BM25F-based ChatNoir search engine [18] for the ClueWeb09. The results of our local Ambient search and the accompanying ChatNoir search are always merged using the BM25F-scores.

4.2 Automatic Label Evaluation

For each of the 791 subtopics, the three cluster labeling approaches χ^2-test, weighted centroid covering, and our newly proposed query-based method are run. In a first evaluation phase, we employ the standard evaluation scheme

[1] http://credo.fub.it/ambient/, last accessed: May 20, 2014.

Table 2. User study results for the query-based labels ("Query"), the χ^2-based labels, and the weighted centroid covering ("WCC"). Shown are the absolute and relative number of votes from our 29 participants on the 100 sampled subtopics. The last two columns show for how many of the subtopics an approach received the most votes ("Winner") and the absolute majority of votes.

Approach	User votes (absolute)	User votes (relative)	Winner	Absolute majority
Query	1276	0.44	43	31
χ^2	1160	0.40	33	16
WCC	463	0.16	4	2
Total	2900	1.00	80	49

of comparing the reference labels in the AMBIENT dataset (the disambiguation descriptions) to the computed labels. Standard measures of similarity are F-Measure (precision and recall of the computed compared to the reference label terms), Jaccard index (overlap of computed and reference terms), and cosine similarity of the *tf*-weighted term vectors of the computed and the reference labels. Since these measures are only able to capture lexical similarity, we also propose to use a semantics-aware measure in form of the ESA-similarity [8]. In this case, also semantically related terms that have no or only a very low lexical similarity are counted as "correct." The background collection for the ESA-similarity is formed by a random sample of 100,000 English Wikipedia articles. Note that the usage of ESA as a cluster labeling quality measure is novel and a contribution in itself. Before, only lexical similarity was measured.

The results can be found in Table 1. For evaluation, we set the label length $k = 5$ for the approaches χ^2-test and weighted centroid covering since this is the average length of the query generation labels. Interestingly, the measures that simply evaluate the term overlap with the reference label (F-Measure and Jaccard) favor the χ^2-labels while the more advanced ESA similarity favors the query labels. Thus, depending on the used evaluation measure, our new query generated labels are somewhat on par with the standard χ^2 labeling approach and clearly improve upon the weighted centroid covering.

4.3 User Study

Complementing the automatic evaluation of similarity to reference labels, we also conduct a user study in which human participants should select the best label from the three approaches according to their personal perceived similarity to the also displayed reference label.

For the user study, we sampled 100 of the 791 subtopics that had to be evaluated by each of our participants. The study was conducted online with a short introduction to the idea of cluster labeling. To ensure a meaningful word order of the generated cluster labels (remember that χ^2 and weighted centroid covering just present labels composed of 5 single words), we post-processed the labels to find frequent word n-grams in the cluster's documents and in the Google

n-grams. The label terms were re-ordered whenever a frequent n-gram like `new york` was identified and the ordering in the original label was `york new`. This improves the labels' readability for our human participants and could possibly be a useful post-processing step in any labeling approach working with single words.

In our study, 29 subjects each spent 15–30 min on their judgments. Table 2 shows the aggregated results. According to the number of votes, the users favor query- and χ^2-based labels. However, the situation changes when looking at the number of topics where one approach received the most votes (column "Winner"; for 20 subtopics there was a tie) and where one approach got an absolute majority of at least 6 out of 10 votes (no such majority for 51 topics). Here, the users clearly favor the query-based labels. However, a general critique amongst our users that could also be observed from the votes was the label length. Whenever the query labels are longer than 5 terms (the threshold for the other two approaches), the users often favored the χ^2 labels or even the weighted centroid covering.

4.4 Discussion

The traditional automatic evaluation of similarity against the reference labels results in a tie between the query-based and the χ^2 labels. But our user study indicates the promising potential of query-based labels since many users favor them and if they do not, the query labels often are almost as popular as the χ^2 labels.

5 Conclusion and Outlook

We have presented a solution to the abstract problem of automatically formulating a query that retrieves a given document set. This abstract problem has an interesting use case in the scenario of cluster labeling where the task is to generate good labels for the individual clusters that "tell" the user something about the contained documents. Our idea of using queries as the labels (derived by solving the abstract query formulation problem) has shown promising performance when compared against standard cluster labeling approaches. Using traditional and our newly proposed ESA-based evaluation measure, our query-based cluster labels are on par with the standard methods. A further user study showed a clear tendency that users prefer the idea of queries as cluster labels over the standard methods.

As for future research, the full potential of our query-based cluster labeling idea should be exploited by enhancing the currently used rather basic BM25F retrieval model. Including for instance synonyms and putting more emphasis on keyphrases as the basis queries, we envision an even better quality of queries as cluster labels. It also would be very interesting to examine the usage of queries itself to guide the whole clustering process by for instance using a document's keyqueries [9] as the clustering features. The queries used for clustering would then directly form appropriate labels at no additional costs.

References

1. Azzopardi, L., Vinay, V.: Retrievability: an evaluation measure for higher order information access tasks. In: Proceedings of the 17th ACM conference on Information and knowledge management (CIKM 2008), pp. 561–570, ACM, New York (2008)
2. Bar-Yossef, Z., Gurevich, M.: Random sampling from a search engine's index. J. ACM **55**(5), 1–74 (2008)
3. Bendersky, M., Croft, W.B.: Finding text reuse on the web. In: Proceedings of the Second ACM International Conference on Web Search and Data Mining (WSDM 2009), pp. 262–271, ACM, New York (2009)
4. Bonchi, F., Castillo, C., Donato, D., Gionis, A.: Topical query decomposition. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2008), pp. 52–60, ACM, New York (2008)
5. Dasdan, A., D'Alberto, P., Kolay, S., Drome, C.: Automatic retrieval of similar content using search engine query interface. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM 2009), pp. 701–710, ACM, New York (2009)
6. Fuglede, B., Topsøe, F.: Jensen-Shannon divergence and Hilbert space embedding. In: Proceedings of International Symposium on Information Theory (ISIT 2004), pp. 31, IEEE, Piscataway (2004)
7. Fuhr, N., Lechtenfeld, M., Stein, B., Gollub, T.: The optimum clustering framework: implementing the cluster hypothesis. Inf. Retrieval **15**(2), 93–115 (2011)
8. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using wikipedia-based explicit semantic analysis. In: Proceedings of the 20th International Joint Conference on Artifical Intelligence (IJCAI 2007), pp. 1606–1611, Morgan Kaufmann Publishers Inc, San Francisco (2007)
9. Gollub, T., Hagen, M., Völske, M., Stein, B.: From keywords to keyqueries: content descriptors for the web. In: Proceeding of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2013), pp. 981–984, ACM, New York (2013)
10. Hagen, M., Stein, B.: Search strategies for keyword-based queries. In: 7th International Workshop on Text-Based Information Retrieval (TIR 2010), pp. 37–41, Piscataway, IEEE (2010)
11. Hagen, M., Stein, B.: Candidate document retrieval for web-scale text reuse detection. In: Grossi, R., Sebastiani, F., Silvestri, F. (eds.) SPIRE 2011. LNCS, vol. 7024, pp. 356–367. Springer, Heidelberg (2011)
12. Huston, S., Croft, W.B.: Evaluating verbose query processing techniques. In: Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2010), pp. 291–298, ACM, New York (2010)
13. Jordan, C., Watters, C., Gao, Q.: Using controlled query generation to evaluate blind relevance feedback algorithms. In: Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2006), pp. 286–295, ACM, New York (2006)
14. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, New York (2008)
15. Muhr, M., Kern, R., Granitzer, M.: Analysis of structural relationships for hierarchical cluster labeling. In: Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2010), pp. 178–185, ACM, New York (2010)

16. Navigli, R., Crisafulli, G.: Inducing word senses to improve web search result clustering. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP 2010), pp. 116–126, Stroudsburg (2010) (Association for Computational Linguistics)
17. Pickens, J., Cooper, M., Golovchinsky, G.: Reverted indexing for feedback and expansion. In: Proceedings of the 19th ACM international conference on Information and knowledge management (CIKM 2010), pp. 1049–1058, ACM, New York (2010)
18. Potthast, M., Hagen, M., Stein, B., Graßegger, J., Michel, M., Tippmann, M., Welsch, C.: ChatNoir: A search engine for the ClueWeb09 corpus. In: The 35th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2012), pp. 1004, ACM, New York (2012)
19. Robertson, S.E., Zaragoza, H.: The probabilistic relevance framework: BM25 and beyond. Found. Trends Inf. Retrieval **3**(4), 333–389 (2009)
20. Robertson, S.E., Zaragoza, H., Taylor, M.J.: Simple BM25 extension to multiple weighted fields. In: Proceedings of the 13th ACM International Conference on Information and Knowledge Management (CIKM 2004), pp. 42–49, ACM, New York (2004)
21. Stein, B., zu Eißen, S.M.: Topic identification: framework and application. In: Proceedings of the 4th International Conference on Knowledge Management (I-KNOW 2004), Journal of Universal Computer Science, Know-Center, pp. 353–360, Graz (2004)
22. Stein, B., Gollub, T., Hoppe, D.: Beyond precision@10: Clustering the long tail of web search results. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM 2011), pp. 2141–2144, ACM, New York (2011)
23. Turel, A., Can, F.: A new approach to search result clustering and labeling. In: Salem, M.V.M., Shaalan, K., Oroumchian, F., Shakery, A., Khelalfa, H. (eds.) AIRS 2011. LNCS, vol. 7097, pp. 283–292. Springer, Heidelberg (2011)
24. Yang, Y., Bansal, N., Dakka, W., Ipeirotis, P., Koudas, N., Papadias, D.: Query by document. In: Proceedings of the Second ACM International Conference on Web Search and Data Mining (WSDM 2009), pp. 34–43, ACM, New York (2009)