Keyqueries for Clustering and Labeling

Tim Gollub, Matthias Busse, Benno Stein, and Matthias Hagen

Bauhaus-Universität Weimar, Germany
<first name>.<last name>@uni-weimar.de

Abstract In this paper we revisit the document clustering problem from an information retrieval perspective. The idea is to use queries as features in the clustering process that finally also serve as descriptive cluster labels "for free." Our novel perspective includes query constraints for clustering and cluster labeling that ensure consistency with a keyword-based reference search engine.
Our approach combines different methods in a three-step pipeline. Overall, a query-constrained variant of k-means using noun phrase queries against an ESA-based search engine performs best. In the evaluation, we introduce a soft clustering measure as well as a freely available extended version of the Ambient dataset. We compare our approach to two often-used baselines, descriptive k-means and k-means plus χ^2. While the derived clusters are of comparable high quality, the evaluation of the corresponding cluster labels reveals a great diversity in the explanatory power. In a user study with 49 participants, the labels generated by our approach are of significantly higher discriminative power, leading to an increased human separability of the computed clusters.

1 Introduction

Document clustering is a popular approach to enable the exploration of large collections such as digital libraries, encyclopedias, or web search results. The objective of clustering is to automatically organize a document collection into a small number of coherent classes or *clusters* such that documents in one cluster are more similar to each other than to those in other clusters. Along with short meaningful labels for the clusters (summarizing the cluster content) a user can get a general overview of a collection, start a systematic exploration, or narrow the focus to just a particular subset of the documents meeting an information need.

The document clustering task falls into two steps: (1) unveil the topical structure in a document collection and (2) provide meaningful descriptions that communicate this structure to a user. For the first step, referred to as *clustering*, many effective algorithms are known. However, clustering algorithms such as the popular k-means are usually not capable of producing meaningful cluster labels. This is usually treated in a subsequent second step—the *cluster labeling*. One major drawback of common keyword-based labeling techniques is their limitation to only selecting "statistical" features from the documents; for example, by concatenating the most prominent keywords occurring in a cluster. However, a list of keywords tends to represent different and unrelated aspects of the documents and will often fail to provide a readable label.

To account for the crucial aspect of meaningful labels for document clustering, we take an information retrieval perspective. Note that a user's perceived suitability of a label for a document set can be seen as similar to a search engine's decision of whether a

document matches a query. Thus, we view queries as good candidates for cluster labels—and as good features for the clustering itself. This way, we establish an explicit connection between clustering and search technology. Furthermore, the interplay between information retrieval systems and cluster analysis brings forth an intuitive approach to hierarchical search result clustering: Once the relevant aspects in a document collection are unveiled in form of search queries, each of the corresponding result sets can then serve as input for another iteration of the clustering process, which in turn leads to a new set of now more detailed aspects, i.e. search queries.

Our main contributions are threefold: (1) a flexible three-step processing pipeline for document clustering using search queries as features and labels, (2) an extended and freely available version of the Ambient data set with 4680 manually annotated documents, and (3) a user study with 49 participants comparing the explanatory power of the cluster labels generated by our approach and two often-used baselines.

2 Related Work

One of the first applications of document clustering was to improve retrieval based on the cluster hypothesis stating that "closely associated documents tend to be relevant to the same requests" [16]. Later, clusters were also used as a browsing interface to explore and organize document collections like in the famous Scatter/Gather algorithm [4]. The numerous document clustering algorithms can be classified into three classes [3]: data-centric, description-aware, and description-centric algorithms.

Data-centric algorithms typically are not limited to the text domain. The to-be-clustered data is represented in models that allow to compute similarities between the data objects; one of the most popular such algorithms being k-means (cf. Section 3.3). A popular data representation in the text domain is the Vector Space Model with $tf \cdot idf$ weights and cosine similarity [18]. However, the generation of a label that can be presented to a user is not part of data-centric algorithms but tackled as an independent, subsequent step. Examples are labels formed from keywords frequently occurring in a cluster's documents [21] or applying Pearson's χ^2 test taking into account other clusters [12] that forms our first baseline. Still, such labels often are a sequence of rather unrelated keywords rendering even the best clustering less useful to users that rely on the labels as readable descriptions—an issue that inspired a second class of cluster algorithms.

Description-aware algorithms try to circumvent the labeling issue of data-centric approaches by ensuring that the construction of cluster labels produces results that are comprehensive and meaningful to a user. One way to achieve this goal is to use algorithms that assign documents to clusters based on a single feature—so-called monothetic clustering—and to then use this feature as a label. One example is the Suffix Tree Clustering [25] that exploits frequently recurring phrases as similarity features. First, base clusters are discovered by shared single frequent phrases utilizing suffix trees. Second, the base clusters are merged by their phrase overlap. However, since the merging step is based on the single-linkage criterion, the combined phrases of merged clusters forming the cluster labels often still tend to be unrelated and therefore misleading for a user.

SnakeT [5] tries to enrich the similarly obtained labels by using phrases from a predefined ontology but still, the cluster analysis precedes and dominates the labeling task—a problem the next class of algorithms tries to circumvent.

Description-centric algorithms consider the cluster labels as the crucial elements of a clustering. They assume that if a cluster cannot be described by a meaningful label, it is probably of no value to a user. The description precedes the assignment of a document to a cluster. Description-centric algorithms mainly tackle the use case of clustering web search results, with Lingo being one of the pioneering examples [14]—now part of Carrot2, an open source framework for search result clustering.[1] A singular value decomposition of a frequent term-search result snippet matrix is used to extract orthogonal vectors assumed to represent distinct topics in the snippets. The documents are then assigned to the extracted topic clusters using the Vector Space Model.

With a similar goal, Weiss revisits the data-centric k-means algorithm and adjusts it to a description-centric version: descriptive k-means [24]—our second baseline. First, k-means with $tf \cdot idf$ features is run. Then frequent phrases are extracted from the cluster's centroids as potential cluster labels. As for document assignment, the algorithm searches for cluster documents that are relevant to a phrase utilizing the Vector Space Model.

Description-centric algorithms focus on label quality but still do not use the full potential. Documents not containing a topic label but being just as relevant from an information retrieval perspective are not considered to belong to a topic's cluster. We believe that queries against suited search engines are able to overcome this drawback, exploiting the extensive information retrieval research of the last decades. Some of the respective ideas that inspired our approach are discussed in the following section.

3 Our Approach

Our approach leverages queries in the clustering process and as labels. This way, we exploit the fact that search queries linking keywords to document sets are a concept well-known to users from their daily web search experience. Both Lingo [14] and descriptive k-means [24] can be interpreted to utilize search queries in their algorithms. However, queries are only used for validating a clustering. Instead, our approach considers search queries as the driving force while deriving the clustering; inspired by Fuhr et al.'s more theoretical optimum clustering framework (OCF) that suggests search relevance scores or retrieval ranks as clustering features [6]. Still, the OCF does not address the problem of labeling the resulting clusters.

Our new approach combines the general idea of OCF with Gollub et al.'s concept of keyqueries as document descriptors [8,9] that recently has been used for recommending related documents [10]. We will use keyqueries as clustering features in an OCF-style but then will as well suggest suited keyqueries as labels. Following Stein and Meyer zu Eißen [21], meaningful cluster labels should be comprehensive (appropriate syntax), descriptive (reflect each document in a cluster), and discriminative (minimal semantic overlap between two cluster's labels). Most existing cluster labeling techniques do not sufficiently address the descriptiveness aspect but queries do as our experiments will show.

[1] http://project.carrot2.org

3.1 Queries as Label Candidates

To model the descriptiveness of cluster labels, we view a user's perception of a label as follows: The presentation of a cluster label activates a concept in the user's mind, and each document that is relevant to this concept should be classified under that label. This process is conceptually very closely related to the standard task of information retrieval—query-based search. This analogy leads us to propose the use of search queries as cluster labels that have to retrieve the documents of the associated cluster. The task of document clustering then can be formulated as the reverse of query-based search as follows: Given a set of documents, find a set of diverse search queries that together retrieve the document set when submitted to a reference search engine. Along with their retrieved documents as cluster contents, the queries then form a labeled clustering. This implies that the potential clusters of a document are given by the queries for which it is retrieved and leads to a first new constraint within the constrained clustering terminology [2]: the *common-query constraint CQ* stating that two documents cannot be in the same cluster if they are not retrievable by a common query.

In order to find the labeling queries, the possible vocabulary has to be defined. The vocabulary generation is an important step in our pipeline since the choice of vocabulary terms determines the comprehensive power of the cluster labels. In case the terms are ambiguous, not comprehensive, or too specific, the cluster labels will inevitably also exhibit such problems and will fail to reflect the content of a cluster. Also the size of the vocabulary has an impact on the overall performance. With respect to the syntax of cluster labels, category names in classification systems or Wikipedia are considered to be ideal [21,22,24]. Category names typically are noun phrases or conjunctions of these; therefore, we consider noun phrases as suitable to serve as cluster labels. For readability reasons, we suggest to restrict the number of conjunctions to one, like in "Digital Libraries and Archives." This forms our second constraint, the *query-syntax constraint QS* stating that a cluster label consists of noun phrases or a conjunction of these.

But not all noun phrases form good candidates for cluster labels. Even though determiners are often viewed as part of a noun phrase, they are not necessary in our scenario. The same holds for post-modifiers, etc. We consider noun phrases to be a concatenation of pre-modifiers and a head noun. Still, pre-modifiers are not yet restricted in length such that arbitrarily long cluster labels could be generated. Following the distribution in the Wikipedia where a category name on average consists of 3.87 terms, we formulate our third constraint, the *query-length constraint QL* stating that a cluster label consists of maximum four terms per at most two noun phrases (i.e., maximum length is eight plus the conjunction). To find suitable phrases, we use Barker and Cornacchia's head noun extractor [1] that provides a phrase ranking from which we choose the top-6 per document (determined in pilot studies) that are then lemmatized using the Apache OpenNLP library to avoid different flections. Other keyphrase extractors can of course also be integrated.

To avoid meaningless phrases like "big issue" or "common example," we also consider a second form of vocabulary generation allowing only noun phrases from a predefined vocabulary. As the source of a controlled and predefined vocabulary consisting of well-formed and suitable phrases we choose the titles of Wikipedia articles following Mihalcea and Csomai's suggestion [13]. Applying the three constraints from above, we

select only those titles with a maximum length of four terms. In addition, we discard Wikipedia article titles that solely consist of stopwords, dates, and special or non-latin characters, because they usually do not serve as meaningful cluster labels. Our resulting vocabulary consists of 2,869,974 titles that are also lemmatized. As for ranking possible Wikipedia phrase candidates, we use the keyphraseness score [13] as the ratio of the number of articles that contain the phrase as a link and the total number of articles that contain the phrase.

3.2 Examined Search Engines / Retrieval Models

In the document indexing step of our clustering pipeline, we exploit the research effort on retrieval models of the last decades by using queries as a good means to derive clusters and labels. Of course, different retrieval models may yield different clusterings and labels. In our pipeline, we experiment with the classic Boolean model (queries based on Boolean logic but no ranking possible), the Vector Space Model with $tf \cdot idf$ weighting [18] (documents and queries modeled as vectors), BM25 [17] ("$tf \cdot idf$+ document length"), and ESA [7] with Wikipedia articles as the background collection (topic modeling approach taking semantic similarities into account). Our evaluation will show that the ESA retrieval model is best suited for our task.

For the retrieval models that rank the results, we include two further relevance constraints for setting a cut-off such that lower ranked documents are not considered part of the result set for the purpose of clustering. These relevance constraints reflect the keyquery idea of Gollub et al. [8]: a keyquery for a document is a query that returns the document in its top ranks. Our *top-k constraint* states that only the k topmost results of a query count as the result set—we set $k = 10$ following the original keyquery idea. Since a document at rank $k + 1$ could be as relevant as the one at rank k, such a static cut-off might be problematic and also limits the size of the possible clusters in our scenario—difficult if the size of the clusters is not known in advance. Hence, we propose an alternative *score constraint* stating that to be part of the result set, a document must have a retrieval score above some relevance threshold t. In our pilot experiments with different techniques of "averaging" retrieval scores, $t = \sum {s_i}^2 / \sum s_i$, where s_i denotes the retrieval score of a document, turned out to be a good choice. Compared to the standard mean $t = \sum s_i / N$, the formula emphasizes the highest scores and reduces the influence of low scores.

3.3 Query-constrained Clustering Algorithms

For every document in the to-be-clustered collection, we store all the queries for which the document is retrieved according to our above relevance constraints in a reverted index [15]. The postlists of the documents in the reverted index contain the respective keyqueries and serve as the document features for the clustering. In the following, we describe three different cluster algorithms that satisfy the common-query constraint.

Set Cover Clustering The first algorithm tackles clustering as a set covering problem (SCP) on the result lists of the query vocabulary. In our scenario, we apply a variant of the greedy set cover algorithm [23]. For up to k iterations, the query q is selected

whose result set size is within a certain range, covers the maximum number of documents not yet covered by previous queries, and where the not-yet-covered documents in the result set have a high *positive rate* in a graph that connects documents by an edge when they share a keyquery (i.e., multiple edges between two documents are possible). The positive rate of a new result set is the ratio of actual edges between not-yet-covered documents in the result set and the minimum number of edges if each of these documents would be retrieved by only this one query. Note that this way, documents in the clustering may be part of several result sets.

Agglomerative Clustering Our second algorithm variant follows the agglomerative strategy of hierarchical clustering. It starts with each document in its own cluster, and then merges pairs of clusters in a "bottom-up" approach. As for the merging, measures for the distance between pairs of documents and a linkage criterion specifying the distance of clusters are needed. We choose the number of shared keyqueries for both distances. As for cluster similarity, we follow a complete-linkage clustering approach (taking into account all document pair similarities between two clusters) since this avoids the chaining phenomenon of single-linkage clustering, where clusters might be merged due to a single pair of documents being close to each other, even though all other documents are very dissimilar. Our algorithm merges those two clusters, whose document pairs share the most keyqueries. In case that the maximum number is shared by more than two clusters, the algorithm decides upon the ratio of shared to non-shared queries of the document pairs. Since the documents of the two merged clusters are not necessarily the only clusters that are retrieved by the shared keyqueries, we additionally include all other remaining clusters that the shared keyqueries retrieve.

When the merging finally leads to the desired number of clusters, the algorithm stops. But simply concatenating the set of queries as the corresponding cluster label would in many cases violate our query-length constraint (e.g., when more than two queries are left in a node). We therefore strive for the query or pair of queries that "best" cover the cluster documents. Since all queries find at least the cluster documents, we choose the query (pair) that retrieves the fewest additional documents from other clusters.

Constrained *k*-means Clustering The query-constrained clustering algorithm in this section adopts the popular data-centric k-means algorithm with keyquery features. Given a collection of data points, k-means operates in three steps. (1) In the initialization, k random values within the data domain are chosen as initial cluster representatives (the centroids). In our scenario, each document is represented by a vector with a 1 at position i if the document is retrieved by that query in the reverted index or 0 otherwise. For the initialization, we randomly generate k such vectors. (2) In the assignment phase, each data point is assigned to its nearest centroid and therefore, clusters of data points are formed. In our scenario, the algorithm calculates for each document vector the dot-product to all centroid vectors and assigns the document to the centroid with the highest value. (3) In the update phase, the k centroids of the new clusters are computed and input to the assignment phase until convergence or some threshold of iterations is reached. In our scenario, for each cluster the query is selected whose result set best covers the assigned documents in terms of the F-Measure. The new centroid is computed as the mean vector of the result documents of that best query.

4 Evaluation

We compare the different variants of our three-step query-based clustering pipeline on an extended version of the Ambient dataset against two often-used approaches; among others, we conduct a user study with 49 participants on the explanatory power of the cluster labels.

4.1 AMBIENT++ Dataset

The original Ambient dataset was published by Carpineto and Romano in 2008,[2] and has become popular for document clustering and labeling evaluation [20,19]. It comprises 44 ambiguous topics with about 18 subtopics each, obtained from Wikipedia disambiguation pages. Some of the subtopics are associated with a set of documents (URL, title, snippet) that were collected by submitting every topic as a query to a web search engine, and by manually assigning each URL of the top 100 results to a subtopic. However, the documents were not stored and the subtopics are very uneven in size. Hence, we reconstruct the Ambient dataset as our extended corpus AMBIENT++ as follows.

The documents of the original Ambient URLs form the basis of our corpus extension and are crawled in a first step. The authors of the original data set assigned a total of 2257 URLs to some subtopic; in fact, most of the subtopics did not get any document assigned while others got up to 76 URLs. In early 2016, only 1697 documents of the original dataset could be crawled. After a manual inspection, 611 documents had to be discarded since they did not discuss the originally assigned subtopic anymore—only 1086 documents remain. We thus enrich the data to have at least ten documents in each of the original subtopics. To this end, the descriptions from the Wikipedia disambiguation pages for the subtopics that do not have ten documents were submitted to a web search engine and the result lists manually assessed until ten documents for the subtopic are available (excluding pages that only contain an image, video, table, etc.). In some cases, the subtopic descriptions are no successful queries (e.g., too long and specific). In such cases, our annotators manually formulated a better suited query. But a few topics still did not get ten "valid" documents although we assigned 4506 additional documents to subtopics—a total amount of 5592 documents.

Since not every subtopic could be sufficiently enriched and some subtopics have way more than ten documents, we balance the dataset to subtopics with exactly ten documents. We discard the subtopics with less than ten documents and from the ones with too many documents we keep the ten best-ranked query results only—resulting in 481 subtopics with ten results compared to only 25 subtopics in the original Ambient dataset. During the manual filtering, we also identified a few subtopics with identical meaning (e.g., subtopic 12.11 (globe, a Japanese trance/pop-rock group) and subtopic 12.17 (globe (band), a Japanese pop music group) that are too difficult to separate in a clustering such that we only keep one of these—13 subtopics were removed. In our enriched dataset, each of the 44 topics has at least three subtopics (468 in total) each having ten documents. As for extracting the main content of the 4680 corpus documents, we use the Default Extractor from the Boilerpipe library [11] which performed best in our pilot experiments.

[2] Claudio Carpineto, Giovanni Romano: Ambient Data set (2008), `http://credo.fub.it/ambient/`

4.2 Soft F-Measure as a new Evaluation Measure

In our experimental framework, we consider each topic of the AMBIENT++ dataset as one to-be-clustered collection where the "optimal" clustering would form clusters identical to the respective subtopics. However, our query-based clusterings can result in clusters that are difficult to evaluate with the traditional F-Measure against the ground truth. For instance, a query `animal` for the topic "Camel" could retrieve documents about the humped ungulates but also about arachnids (the camel spider, both subtopics of the topic camel) such that the resulting cluster cannot really be evaluated against just one of the two ground truth subtopics/clusters. As for comparing the quality of clusterings with such ambiguous or overlapping clusters, we propose the Soft F-Measure (name inspired by soft clustering algorithms, where a document may be contained in several clusters). The measure computes true/false positives/negatives on the level of document pairs and not document-cluster pairs like the conventional F-Measure does. For each document pair in the clustering, we calculate the association strength s by the ratio of shared clusters to all clusters they are assigned to (maximum association strength is 1). If the two documents are in the same subtopic/cluster in the ground truth, s is added to the true positive score and $1 - s$ to the false negative score; if not, s is added to the false positive score and $1 - s$ to the true negative score. The scores are finally used in the "traditional" F-Measure formula. Note that the Soft F-Measure is not "symmetric" (e.g., only retrieving six of ten documents in one cluster is worse than retrieving all ten documents and four additional false positives).

4.3 Setting up our Pipeline

For each of the three pipeline steps (vocabulary generation, document indexing, constraint clustering), we compare the performance of the different variants on a training set of ten topics to a "best" clustering possible and an index clustering. The best clustering is obtained by a brute-force analysis that finds the queries from the index that best identify the subtopics with respect to the traditional F-Measure against the ground truth. The index clustering uses every postlist in the index as a cluster. Rationale for this approach is the assumption that the entries of the inverted index can be seen as support for the query-based clustering: the more queries retrieve similar result sets, the more likely these documents are grouped together.

In our pilot experiments, noun phrase vocabulary achieves slightly better best clustering performance than the Wikipedia vocabulary (F-Measure of 0.93 vs. 0.91) and also a slightly higher Soft F-Measure for index clustering (0.26 vs. 0.25) such that we choose noun phrases as the vocabulary. To decide how many phrases to extract per document, we test 1 to 20 extracted phrases per document. Interestingly, the F-Measure of the best clustering saturates at six extracted noun phrases. Hence, we decide to extract six phrases from each document.

To overcome the influence of possibly insufficient phrases for comparing the different retrieval models (Boolean, $tf \cdot idf$, BM25, ESA) and the relevance constraint parameter settings (rank or score), we manually generated appropriate queries for each of the subtopics in our training set and compare the F-measure of the result lists with respect to the subtopic the query belongs to. Not too surprising, in our AMBIENT++ scenario, a fixed cut-off constraint at rank 10 performs much better than a score constraint that

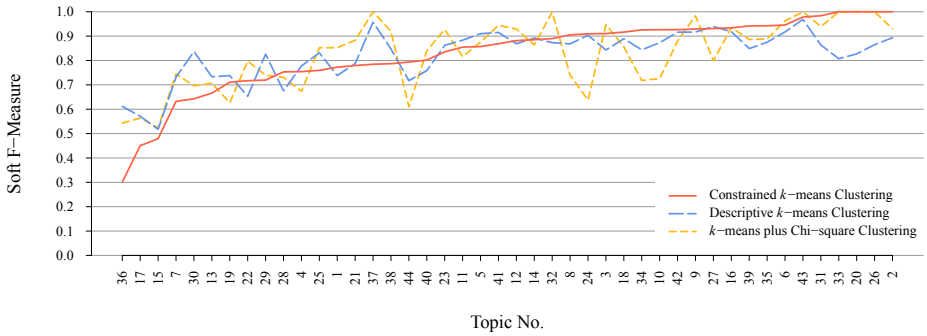


Figure 1. Comparison of our constrained k-means clustering with the baselines.

would yield clusters with 70+ documents (remember that each subtopic has ten documents). Except a few outliers, all three ranking-based retrieval models outperform the Boolean model while the ESA model outperforms the other models on 8 of 10 topics. As for ESA on our training set, the full Wikipedia articles as the background concept collection perform better than just the first paragraphs of each article.

From the three clustering methods in our pipeline (set cover, agglomerative, constrained k-means) the constrained k-means achieves the highest Soft F-Measure scores with ESA on our training set (0.83 vs. 0.77 for the other two) but is still way below the best clustering with an average Soft F-Measure of 0.94.

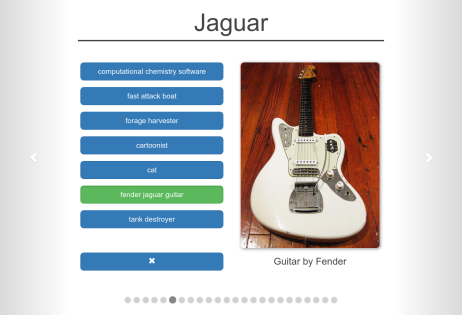
Our best pipeline set-up (constrained k-means clustering with six extracted noun phrases per document and top-10 results of ESA with the complete Wikipedia articles as background collection) is now compared to two often-used clustering+labeling approaches: k-means plus χ^2 baseline [12] representing the data-centric algorithms and descriptive k-means [24] representing the description-centric algorithms.

4.4 Clustering Quality

The clustering quality evaluation is performed on all topics of our new AMBIENT++ dataset employing the Soft F-Measure for the clusters that an algorithm derived for a topic. With their average Soft F-Measure of 0.83 our new constrained k-means and k-means plus χ^2 are slightly better than the 0.82 of descriptive k-means (k always set to the true number of subtopics for every algorithm)—hence, query integration does not harm clustering quality. Figure 1 shows the distribution on a topic level indicating quite different performance for specific topics but similar general trends (topics ordered by our algorithm's performance) as well as some rather difficult to-be-clustered topics (also our set cover and agglomerative clustering methods have similar problems on these).

4.5 Cluster Label Quality

Since our new approach is comparable to two often-used approaches from a cluster quality perspective, we also compare the label quality. As the appropriateness of a cluster label for a cluster is challenging to evaluate, we conduct a user study with 49 participants (23 female, 26 male, mean age 18.3, $SD = 6.8$) on the AMBIENT++ dataset with two experiments that evaluate (1) the discriminative power and (2) the descriptive power of the cluster labels.



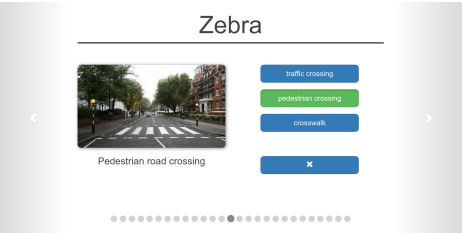
Judgment	CKM	DKM	χ^2
✓	**213**	180	152
–	**15**	25	39
×	**21**	44	58
F_1	**0.92**	0.84	0.76
p	–	0.005	0.000

Figure 2. (Left) screenshot of the first user study experiment, (right) judgment distribution (CKM = constrained k-means, DKM = descriptive k-means, χ^2 = k-means + χ^2) indicating that our approach's labels are significantly more discriminative than the baselines' labels.

Experiment 1: Discriminative Power In the first experiment, we examine to what extent the cluster labels can discriminate documents from one cluster to other clusters. We conduct an empirical browser-based study in a within-subjects design meaning that each participant is asked about labels of every approach. For a given subtopic, a participant is given a manually prepared short description of up to five words and a selected identifying image (instead of the often lengthy original disambiguation text) and cluster labels of one algorithm derived for the subtopic's topic. The participant then has to choose the label that best fits the given subtopic (forced-choice). For time constraints, we only consider a subset of 22 random topics from each of which we choose at most four subtopics with the highest average clustering Soft F-Measure over all three approaches (always higher than 0.8 but some topics only have three subtopics (average at 3.77)). At most eight labels are presented to the user (some topics have fewer subtopics, from the others 7 additional random ones are chosen). Each subtopic-algorithm combination in our study was judged by three participants resulting in 747 judgments ($(22 \cdot 3.77 \cdot 3) \cdot 3)$); on average around 15 judgments per participant ensuring that no participant judged for the same subtopic twice (not even for another algorithm).

Figure 2 shows a screenshot and the result of the first experiment. In the screenshot, the name of the topic (Jaguar) is shown at the top, the to-be-judged subtopic is presented by an image and a short description at the right-hand side, and a randomly shuffled list of cluster labels for clusters in the topic at the left-hand side. If none of the labels is satisfying, the participant should click the lowermost cross-button.

In the result table, the first row denotes the number of judgments where the selected label is the label generated by the approach (i.e., true positive), the second row lists the number of judgments where the participant selected a different label than the one generated by the approach (i.e., false positive), and the third row gives the judgments where the participant selected neither of the presented labels (i.e., false negatives). A common single measure is the reported F_1-score and to statistically estimate the per-individual effect, we compare the ratio of correct label assignments (true positives) among all assignments given for a subtopic (true positives, false positives, false negatives). Each subtopic is judged by three participants, and the assigned labels split into correct (true positives) and incorrect (false positives and false negatives). In case that all three participants select the correct label, the ratio equals $\frac{3}{3} = 1$. If only one participant decided



Voting	CKM	DKM	χ^2
✓✓✓	48	45	19
✓✓	52	45	36
✓	51	49	55
–	75	87	116
total votes	**299**	274	184
p-value	–	0.3525	0.0000

Figure 3. (Left) screenshot of the second user study experiment, (right) judgment distribution (CKM = constrained k-means, DKM = descriptive k-means, χ^2 = k-means + χ^2) indicating that our approach's labels are more descriptive than the baselines' labels.

for the correct label, the ratio is $\frac{1}{3}$. According to a Shapiro-Wilk test, the individual participants' ratios are not normally distributed for either approach such that we choose the non-parametric Wilcoxon signed rank test known as a suitable significance test in our within-subjects design with ratio data and three to-be-compared approaches. For the 49 participants' ratios we get a p-value of 0.005 when comparing the distribution of our approach to descriptive k-means and a p-value below 0.001 compared to k-means plus χ^2 indicating that our approach significantly increases the discriminative power of the cluster labels over the baselines.

Experiment 2: Descriptive Power In the second experiment, we examine the descriptive power of the cluster labels. A participant is shown the different cluster labels that are generated by the approaches for one subtopic, and has to select that label which best describes the given subtopic. We ensure that the clusters of the approaches cover the same subtopic by calculating their F-measures to the subtopic. Only if the cluster of each approach exceeds the threshold of 0.8 with regard to the subtopic documents, we include that subtopic to the data set of this experiment ensuring that all three approaches derived good clusterings. We obtain judgments by three participants for 226 of the 468 subtopics similar to the setting in Experiment 1; again not showing the same subtopic to the same user twice.

The first four rows in the table in Figure 3 denote the number of judgments where either all three, two, one or no participant(s) voted for the corresponding approach. For all three approaches, the numbers accumulate to the 226 judged subtopics. Our approach is better than descriptive k-means (although not significant on the per-topic vote distribution) and both outperform k-means plus χ^2.

5 Conclusion and Outlook

We have presented a novel query-based clustering pipeline that uses keyqueries as features for the clustering process and as labels for the resulting clusters. The comparison to two often-used baselines shows that our constrained k-means approach with the ESA retrieval model is competitive from a clustering quality perspective and significantly improves the label quality. Thus, our idea of revisiting the clustering problem from an information retrieval perspective combining ideas from the optimal clustering framework and keyquery research is a promising direction for supporting users engaged in exploratory

search tasks that need guidance in form of document clusterings with good labels. As part of our evaluation, we have also introduced an enriched AMBIENT++ dataset including 4680 manually annotated documents that will be made publicly available and a Soft F-Measure cluster quality evaluation measure.

Interesting directions for future research could be the inclusion of terms from predefined taxonomies from which we only evaluated Wikipedia titles as a first step. Still, we predict much potential to be explored in that direction as well as in the evaluation on other datasets and with further different retrieval models since the performance of all models still was way below an oracle best query clustering.

References

1. K. Barker and N. Cornacchia. Using noun phrase heads to extract document keyphrases. In *AI 2000*, pp. 40–52.
2. S. Basu, I. Davidson, and K. Wagstaff. *Constrained Clustering: Advances in Algorithms, Theory, and Applications*. Chapman & Hall/CRC, 2008.
3. C. Carpineto, S. Osiński, G. Romano, and D. Weiss. A survey of web clustering engines. *ACM Comp. Surv.*, 41(3):17:1–17:38, 2009.
4. D. R. Cutting, D. R. Karger, J. O. Pedersen, and J. W. Tukey. Scatter/Gather: A cluster-based approach to browsing large document collections. In *SIGIR 1992*, pp. 318–329.
5. P. Ferragina and A. Gullì. The anatomy of SnakeT: A hierarchical clustering engine for web-page snippets. In *PKDD 2004*, pp. 506–508.
6. N. Fuhr, M. Lechtenfeld, B. Stein, and T. Gollub. The optimum clustering framework: Implementing the cluster hypothesis. *Inf. Retr.* 15(2):93–115, 2012.
7. E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI 2007*, pp. 1606–1611.
8. T. Gollub, M. Hagen, M. Michel, and B. Stein. From keywords to keyqueries: Content descriptors for the web. In *SIGIR 2013*, pp. 981–984.
9. T. Gollub, M. Völske, M. Hagen, and B. Stein. Dynamic taxonomy composition via keyqueries. In *JCDL 2014*, pp. 39–48.
10. M. Hagen, A. Beyer, T. Gollub, K. Komlossy, and B. Stein. Supporting scholarly search with keyqueries. In *ECIR 2016*, pp. 507–520.
11. C. Kohlschütter, P. Fankhauser, and W. Nejdl. Boilerplate detection using shallow text features. In *WSDM 2010*, pp. 441–450. ACM.
12. C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.
13. R. Mihalcea and A. Csomai. Wikify!: Linking documents to encyclopedic knowledge. In *CIKM 2007*, pp. 233–242.
14. S. Osiński, J. Stefanowski, and D. Weiss. Lingo: Search results clustering algorithm based on singular value decomposition. In *IIPWM 2004*, pp. 359–368.
15. J. Pickens, M. Cooper, and G. Golovchinsky. Reverted indexing for feedback and expansion. In *CIKM 2010*, pp. 1049–1058.
16. C. J. v. Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, 1979.
17. S. Robertson and H. Zaragoza. The probabilistic relevance framework: BM25 and beyond. *FnTIR*, 3(4):333–389, 2009.
18. G. Salton, A. Wong, and C.-S. Yang. A vector space model for automatic indexing. *CACM*, 18(11):613–620, 1975.
19. U. Scaiella, P. Ferragina, A. Marino, and M. Ciaramita. Topical clustering of search results. In *WSDM 2012*, pp. 223–232.
20. B. Stein, T. Gollub, and D. Hoppe. Beyond precision@10: Clustering the long tail of web search results. In *CIKM 2011*, pp. 2141–2144.
21. B. Stein and S. Meyer zu Eißen. Topic identification: Framework and application. In *I-KNOW 2004*, pp. 522–531.
22. P. Treeratpituk and J. Callan. An experimental study on automatically labeling hierarchical clusters using statistical features. In *SIGIR 2006*, pp. 707–708.
23. V. V. Vazirani. *Approximation Algorithms*. Springer, 2001.
24. D. Weiss. *Descriptive Clustering as a Method for Exploring Text Collections*. PhD thesis, University of Poznan, 2006.
25. O. Zamir and O. Etzioni. Web document clustering: A feasibility demonstration. In *SIGIR 1998*, pp. 46–54.