Optimal Meta Search Results Clustering

Claudio Carpineto
Fondazione Ugo Bordoni
Rome, Italy
carpinet@fub.it

Giovanni Romano
Fondazione Ugo Bordoni
Rome, Italy
romano@fub.it

ABSTRACT

By analogy with merging documents rankings, the outputs from multiple search results clustering algorithms can be combined into a single output. In this paper we study the feasibility of meta search results clustering, which has unique features compared to the general meta clustering problem. After showing that the combination of multiple search results clusterings is empirically justified, we cast meta clustering as an optimization problem of an objective function measuring the probabilistic concordance between the clustering combination and the single clusterings. We then show, using an easily computable upper bound on such a function, that a simple stochastic optimization algorithm delivers reasonable approximations of the optimal value very efficiently, and we also provide a method for labeling the generated clusters with the most agreed upon cluster labels. Optimal meta clustering with meta labeling is applied to three description-centric, state-of-the-art search results clustering algorithms. The performance improvement is demonstrated through a range of evaluation techniques (i.e., internal, classification-oriented, and information retrieval-oriented), using suitable test collections of search results with document-level relevance judgments per subtopic.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Clustering*

General Terms

Algorithms, Experimentation, Measurement, Performance

Keywords

Meta clustering, search results clustering, optimization

1. INTRODUCTION

Search results clustering (hereafter referred to as SRC) has become a popular means of approaching the problem of information disorganization and redundancy in lists of search results returned by current search engines in response to multi-topic queries. If the items that relate to the same topic have been correctly placed within the same cluster and if the user is able to choose the right cluster from the cluster labels, such items can be accessed in logarithmic rather than linear time. Because there are so many available SRC systems employing very different techniques and algorithms (see [4] for a review), it is tempting to combine their outputs, just as the results of several search engines can be merged into a meta search engine. To the best of our knowledge, this problem has not been addressed so far.

As every clustering algorithm implicitly or explicitly assumes a certain data model, the main rationale for combining multiple clusterings is to try to produce a clustering with improved accuracy and robustness when such assumptions are not satisfied by the sample data. Meta SRC is clearly related to the the general field of meta clustering, also known as consensus clustering or clustering ensembles (e.g., [5], [14], [6]) but it poses unique challenges that cannot be easily addressed by available techniques:

- the features or algorithms that determined the clusterings are not accessible (characterized as the hard ensemble clustering in [13];
- the need to label the generated clusters with linguistic descriptions of high quality is very important for SRC applications, whereas this aspect is usually ignored in the meta clustering field;
- the clusterings may have been formed from non-identical sets of objects;
- high computational efficiency of the meta clustering algorithm is required to support real-time applications.

In this paper we first show that the characteristics of the outputs returned by multiple SRC algorithms suggest the adoption of a meta clustering approach. Based on this observation, we introduce a novel criterion for measuring the concordance of two partitions of n objects into m disjoint clusters, using the information content associated with the series of decisions made by the partitions on single pairs of objects.[1] We then cast meta clustering as an optimization problem of the concordance between the clustering combination and the given set of clusterings. The optimization framework is the first main contribution of the paper.

[1]The terms 'partition' and 'clustering' are often used interchangeably throughout this paper.

SIGIR'10, July 19–23, 2010, Geneva, Switzerland.

To solve this problem, we test several metaheuristic methods: through an easily computable upper bound on the objective function, we show that a simple stochastic optimization method delivers fast approximations of the optimal value. The clusters of the meta clustering are then labeled with the most agreed upon cluster labels of the input clusterings. The procedure for building the meta clustering and labeling its clusters is our second main contribution.

The superior performance of meta SRC over individual clusterings is demonstrated through a range of evaluation techniques; i.e., using internal, classification-oriented, and information retrieval-oriented measures. The set of experiments are carried out on two test collections explicitly designed to evaluate the performance of clustering algorithms that post-process search results, including a newly created test collection made available for reuse. The extensive evaluation part is our third contribution. Overall, the proposed approach is empirically motivated, theoretically well-founded, computationally efficient, and highly effective.

The remainder of the paper has the following organization. We first compare the results produced by some well known SRC algorithms, making use of methods and concepts on which we build in the following sections. Then we introduce the theoretical framework of optimal probabilistic meta clustering and study approximate solutions to the problem. After describing the procedure for meta labeling the generated clusters, we present the evaluation experiments. We finally discuss related work on meta clustering and offer some conclusions.

2. COMPARING SEARCH RESULTS CLUSTERINGS

Method combination usually works well when the results of the individual methods are different and of good quality. In this section we experimentally analyze whether SRC complies with these requirements. We use four state-of-the-art SRC algorithms: Clusty, KeySRC, Lingo, and Lingo3G. KeySRC [1] and Lingo [10] are research systems[2], while Clusty and Lingo3G are commercial web clustering engines. These systems are characterized by highly descriptive phrases as cluster labels, and are known to perform well on browsing retrieval tasks [4].

The clusterings to be compared were generated in the following way. We used the 100 most popular web queries provided by Google Trends as of February 2009. The queries were submitted to Clusty (with the clustering being restricted to the first 100 documents retrieved for each query), and the resulting set of snippets was collected and given as input to other SRC algorithms. In this way we have been able to include Clusty in the evaluation while ensuring that all the algorithms operate on the same set of snippets, for full comparability of the results.

In SRC, although the algorithms may produce a variable number of clusters, only those with the highest coverage are usually displayed on the first results page. Considering the first ten clusters is a typical choice, and is what we do in this paper. We group all the documents that are not covered by the first ten clusters in a dummy cluster 'other topics'.

We first measured coverage and overlap of truly classified documents; i.e., documents not grouped under the 'other topics' cluster. We found that the first ten clusters of each method covered, on average, 58% of the 100 input documents. The overlap of truly classified documents across pairs of methods was 45%, the documents uniquely classified by either method were 31%, and the remaining 24% of documents were unclassified in both methods.

The next step was to measure the similarity of the clusterings produced by the different methods. In the following set of experiments we considered all 100 search results, thus including those that were grouped under the 'other topics' cluster. While this cluster may be not so useful for information retrieval, it is relevant from the point of view of comparing the two partitions because it contains documents that cannot be grouped with similar documents. In the limit, if all the documents were placed into the 'other topics' cluster by both systems, the two systems would be useless but very similar. To evaluate clustering similarity, we used the Rand index [12], explained below.

Given a set O of n objects and two partitions of O to compare, Π_1 and Π_2, the Rand index (R) is defined as:

$$R = \frac{a+d}{a+b+c+d} = \frac{a+d}{\binom{n}{2}} \quad (1)$$

where:

- a: number of pairs of objects in O that are in the same cluster in Π_1 and in the same cluster in Π_2.
- b: number of pairs of objects in O that are in the same cluster in Π_1 and in different clusters in Π_2.
- c: number of pairs of objects in O that are in different clusters in Π_1 and in the same cluster in Π_2.
- d: number of pairs of objects in O that are in different clusters in Π_1 and in different clusters in Π_2.

Intuitively, one can think of $a+d$ as the number of agreements between Π_1 and Π_2, and $b+c$ as the number of disagreements between Π_1 and Π_2.[3] The Rand index has a value between 0 (i.e., no agreement on any pair of objects) and 1 (i.e., when the two partitions coincide).[4]

In Table 1, in the triangle above the main diagonal of the matrix, we report the mean Rand index value for each pair of methods, averaged over the set of queries. From these figures, the similarity among clusterings seems consistently high across method pairs. However, we must consider that a great contribution to the value of R comes from the pairs of objects that belong to different clusters both in Π_1 and Π_2; i.e., term d in Equation 1. As term a and term d do not have equal information in the SRC domain, a more reliable

[2]KeySRC and Lingo and can be tested, respectively, at `http://keysrc.fub.it/Keysrc` and `http://search.carrot2.org/stable/search`

[3]The Rand index assumes that clusters do not overlap. The clusters generated by the SRC systems were not strictly disjoint, but their overlap was very small: we found that on average less than one document in ten was assigned to more than one cluster. When multiple clusters contained the same document, we simulated a true partition by considering only the most highly ranked one in the list displayed by the system.

[4]As the expected value of the Rand index for random partitions does not take a constant value, the *adjusted Rand index* [7] is sometimes preferred, but we did not use it because its assumptions (e.g., a fixed number of objects in each cluster) do not fit the SRC data.

	Clusty	KeySRC	Lingo	Lingo3G
Clusty	1	R = 0.67	R = 0.66	R = 0.63
KeySRC	*J = 0.28*	1	R = 0.63	R = 0.64
Lingo	*J = 0.26*	*J = 0.27*	1	R = 0.60
Lingo3G	*J = 0.25*	*J = 0.30*	*J = 0.26*	1

Table 1: Pairwise similarity of four SRC methods on 100 popular web queries. We show the Rand index values in the upper triangle of the matrix and the Jaccard coefficient values in the lower triangle.

measure than the Rand index should probably underweigh the contribution of term d to the similarity.

A drastic solution [2] is to argue that pairs of type d are not clearly indicative either of similarity or of dissimilarity, as opposed to counts of "good pairs" (term a) and "bad pairs" (terms b and c), thus ending up with a formula conceptually similar to Jaccard's coefficient:

$$J = \frac{a}{a+b+c} \tag{2}$$

In the lower triangle of Table 1 we report pairwise Jaccard's coefficient values. The results clearly show that the inter-clustering similarity becomes dramatically lower than that measured with the Rand index, according to such a strong interpretation. In the next section we will define a more balanced way of assessing the importance of each term in Equation 1, depending on the number of clusters.

As the primary objective of meta SRC is to improve retrieval performance, it is useful to evaluate the effectiveness of the individual methods and analyse whether there is scope for improvement using a method combination. To this aim, we used AMBIENT, a test collection introduced in [3] and downloadable from `http://credo.fub.it/ambient`. AMBIENT is explicitly designed for evaluating the subtopic retrieval effectiveness of systems that post-process search results. It consists of 44 topics extracted from the *ambiguous* Wikipedia entries, each with a set of subtopics and a list of 100 ranked search results collected from a plain web search engine and manually annotated with subtopic relevance judgments.

As an evaluation measure, we used the *Subtopic Search Length under k document sufficiency* ($kSSL$), introduced in [1]. It is defined as the average number of items (cluster labels or search results) that must be examined before finding a sufficient number (k) of documents relevant to any of the query's subtopics, assuming that both cluster labels and search results are read sequentially from top to bottom, and that only cluster with labels relevant to the subtopic at hand are opened. The main features of $kSSL$ are that: (i) it allows evaluation of *full-subtopic* retrieval (i.e., retrieval of multiple documents relevant to any subtopic) rather than focusing on subtopic coverage (i.e., retrieving at least one relevant document for some subtopics, as e.g. with *subtopic recall at n*); (ii) the modelization of the user search behavior is realistic because the role played by cluster labels is taken into account, whereas most earlier clustering evaluation studies assume that the user can choose the best cluster regardless of its label. The $kSSL$ measure can be applied not only to clustered results but also to ranked lists, thus allowing clustering performance to be compared to the performance of the original ranked list of search results given as input to the clustering algorithms (used as a baseline).

Method	kSSL (k=1)	kSSL (k=2)	kSSL (k=3)	kSSL (k=4)
KeySRC	24.07	32.39	38.19	42.13
Lingo	24.40	30.64	36.57	40,69
Lingo3G	24.00	32.37	39.55	42.97
Best combination	21.65	29.28	33.15	37.29
Search engine	21.60	35.47	41.96	47.55

Table 2: Retrieval performance on the Ambient test collection measured as mean kSSL over the set of queries, for several values of k.

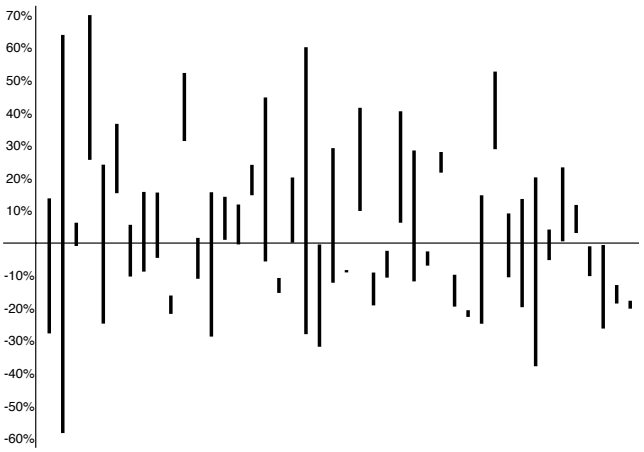

Figure 1: Retrieval performance variation of SRC methods on individual Ambient queries using kSSL with k=2 (search engine list is the baseline).

The systems being tested were ran on the 100 search results associated with each AMBIENT query and the performance of the corresponding output was evaluated using $kSSL$, with $k = 1, 2, 3, 4$. Again we considered the systems introduced above, except for Clusty, whose data were not available to us. The results, averaged over the set of queries, are reported in Table 2. Note that, for $k = 1$, SRC did not improve over using the plain list of ranked results, whereas its superiority becomes clear as the value of k increases, with a comparable mean performance improvement across different methods.

As we were interested in testing the hypothesis that individual methods behave differently on single queries, we also performed a query-by-query analysis. In Figure 1, we show the range of performance variations exhibited by the three clustering methods over the baseline on each query, using $kSSL$ with $k = 2$ as evaluation measure. The differences were ample, with a lot of scope for performance improvement: if we were able to select the best method for each query, we would get the $kSSL$ values reported in the penultimate row in Figure 1.

To summarize the results reported in this section, the individual clusterings were different and presented considerable variations of retrieval performance on single queries, while demonstrating comparable mean retrieval performance. Taken together, these findings indicate the use of a method combination strategy with the goal of maximizing the agreement with the individual clusterings, much in the same spirit

as multiple classifiers are combined through a majority vote, in the hope that the single classifiers make uncorrelated errors. This issue is dealt with in the next section.

3. OPTIMAL PROBABILISTIC META CLUSTERING: PROBLEM DEFINITION

The Rand index and the Jaccard coefficient can be used even when considering partitions with a different number of clusters. However, when the partitions to be compared have a fixed number m of clusters, it may be more convenient to weigh the different types of agreements/disagreements on the basis of their probability of occurring by chance, rather than just counting them. As m grows, the chance for a pair of objects to be placed in the same cluster decreases, while at the same time the chance to be placed in different clusters increases. We can estimate the relative importance of terms a, b, c, d as a function of m.

Assuming that each object is randomly assigned to one cluster, the probability that two objects are in a same cluster in both partitions is

$$p_a = \frac{1}{m^2}, \tag{3}$$

the probability that two objects are in different clusters in both partitions is

$$p_d = \left(\frac{m-1}{m}\right)^2, \tag{4}$$

and the probability that two objects are in the same cluster in one partition and in different clusters in the other partition is

$$p_b = p_c = \frac{(m-1)}{m^2}, \tag{5}$$

with $\sum_h p_h = 1$. The smaller the probability, the larger the information content associated with the observation that the event indeed occurred. We estimate the weights associated with each of the four possible types of agreements or disagreements with the *self information*, i.e:

$$w_h = -log_2\left(p_h\right) \tag{6}$$

Such weights (taken with positive sign for agreements and negative sign for disagreements) can be used to define a measure of probabilistic *concordance* of two partitions, as detailed below.

Given a set of n objects $O = \{o_1, o_2, ...o_i, ..., o_n\}$, consider two partitions Π_1, Π_2 of O into m clusters, with corresponding object-cluster assignments $c_i^{\Pi_1}$ and $c_i^{\Pi_2}$. The concordance of the two partitions at the object-pair level, denoted OC (*object-pair concordance*), is:

$$OC_{\Pi_1,\Pi_2}(i,j) = \begin{cases} -log_2\left(\frac{1}{m^2}\right) & \text{if } (c_i^{\Pi_1} = c_j^{\Pi_1}) \cap (c_i^{\Pi_2} = c_j^{\Pi_2}) \\ +log_2\left(\frac{m-1}{m^2}\right) & \text{if } (c_i^{\Pi_1} = c_j^{\Pi_1}) \cap (c_i^{\Pi_2} \neq c_j^{\Pi_2}) \\ +log_2\left(\frac{m-1}{m^2}\right) & \text{if } (c_i^{\Pi_1} \neq c_j^{\Pi_1}) \cap (c_i^{\Pi_2} = c_j^{\Pi_2}) \\ -log_2\left(\frac{m-1}{m}\right)^2 & \text{if } (c_i^{\Pi_1} \neq c_j^{\Pi_1}) \cap (c_i^{\Pi_2} \neq c_j^{\Pi_2}) \end{cases} \tag{7}$$

where i, j are two objects, with $i \neq j$.

The concordance between the two partitions, denoted PC (*partition concordance*), is defined as the average OC value of all pairs of objects:

$$PC(\Pi_1, \Pi_2) = \frac{1}{\binom{n}{2}} \sum_{\substack{i,j \\ i<j}} OC_{\Pi_1,\Pi_2}(i,j) \tag{8}$$

Based on this pairwise measure of partition concordance, we define the *meta partition concordance* (MPC) between a single (meta) partition Π^* and a set of q partitions $\Pi_1, \Pi_2, ..., \Pi_q$ as:

$$MPC(\Pi^*, \Pi_1, \Pi_2, ..., \Pi_q) = \frac{1}{q}\sum_{r=1}^{q} PC(\Pi^*, \Pi_r) \tag{9}$$

This is our objective function. The optimal partition Π^{opt} is the one that has maximal concordance with the given partitions:

$$\Pi^{opt} = \arg\max_{\Pi^*} \sum_{r=1}^{q} MPC(\Pi^*, \Pi_r) \tag{10}$$

As the optimization of MPC is computationally expensive, it is useful to derive an upper bound that can be easily computed. An upper bound MPC^V on MPC can be defined by considering for each pair of objects a contribution to MPC equal to its maximum theoretical contribution according to the given partitions:

$$MPC^V = \frac{1}{q\binom{n}{2}} \sum_{\substack{i,j \\ i<j}} \arg\max_{\substack{c_i^{\Pi_V} = c_j^{\Pi_V} \\ c_i^{\Pi_V} \neq c_j^{\Pi_V}}} \sum_{r=1}^{q} OC_{\Pi_V,\Pi_r}(i,j) \tag{11}$$

In other words, it suffices to consider the two possible cases (either o_i and o_j in the same cluster, or o_i and o_j in different clusters) and compute the corresponding concordance values at the object-pair level with each given partition. Note that, in general, there is no guarantee that that there will be an admissible partition Π_V that fulfills the decisions made when computing MPC^V; e.g., think of $c_i^{\Pi_V} = c_j^{\Pi_V}$, $c_j^{\Pi_V} = c_k^{\Pi_V}$, and $c_i^{\Pi_V} \neq c_k^{\Pi_V}$. The upper bound corresponds to an actual value if the input partitions coincide: the optimal partition in this case is the given partition.

As an illustration, consider the following three partitions of ten objects into four clusters (each position of the vector corresponds to an object and the value is the cluster to which the object has been assigned):

Π_1 = (1, 1, 2, 2, 3, 4, 3, 3, 4, 4)
Π_2 = (1, 1, 3, 4, 2, 2, 3, 4, 3, 4)
Π_3 = (3, 4, 1, 1, 2, 2, 3, 4, 4, 3)

The partitions were chosen in such a way that the same number of object pairs are grouped together in any pair of partitions, i.e., the first two objects in Π_1 and Π_2, the third and fourth in Π_1 and Π_3, the fifth and sixth in Π_2 and Π_3.

With such small partitions, the optimal meta clustering can be found by brute force search. The number N of distinct partitions of n objects into m non-empty clusters is [**?**]:

$$N(n,m) = \frac{1}{m!}\sum_{i=0}^{m}(-1)^{m-i}\binom{m}{i} i^n \tag{12}$$

For $n = 10$, $m = 4$, we get: $N = 34,105$, $w_a = 4.0$, $w_b = w_c = 1.41$, $w_d = 0.83$. We generated all possible partitions and computed the MPC score associated with each, seen as a meta clustering of Π_1, Π_2, Π_3. The optimal partition is:

$$\Pi^{opt} = (1, 1, 2, 2, 3, 3, 4, 4, 4, 4),$$

with $MPC = 0.66$, while the value of the upper bound MPC^V is 0.77. Note that if we chose one of the original clusterings as meta clustering, we would get the same MPC value for each (i.e., $MPC = 0.59$), due to their regularities. Intuitively, the optimal meta clustering retained the three clusters shared by pairs of individual methods and placed the remaining objects in the fourth cluster. In this example, the improvement attainable by the optimal MPC value is limited, because with few clusters and few objects choosing one of the original partitions as meta clustering ensures a fair concordance with the other partitions (including itself). We will see in Section 4.6 that in practical situations the increase of the MPC value is larger.

Clearly, brute force methods cannot be applied to find Π^{opt} for values of interest in the SRC domain. For instance, with 100 objects and 10 clusters, we get $\approx 10^{39}$ distinct partitions. In the next section we study approximate solutions to this problem.

4. METAHEURISTIC OPTIMIZATION OF CLUSTERING CONCORDANCE

The impracticability of examining every possible partition naturally leads to the adoption of a *hill climbing* strategy, which essentially consists of iteratively rearranging existing partitions by moving individual objects to different clusters, and keeping the new partition only if it provides an improvement of the objective function. Metaheuristic algorithms [8] are elaborate combinations of hill climbing and random search to deal with local maxima. In this section we study the applicability of some well known metaheuristic algorithms to the clustering concordance optimization problem. Any of the following algorithms starts with a random (potentially poor) solution, because we observed that choosing more valuable starting points, such as using one of the given partitions, did not have a clear impact on the algorithm performance.

4.1 Steepest ascent hill climbing

In steepest ascent hill climbing all successors of a current partition are evaluated and the partition with the highest MPC is chosen. The computation halts when the movement of single objects no longer causes the objective function to improve.

4.2 Stochastic hill climbing

Stochastic hill climbing does not examine all successors before deciding how to move. Rather, it selects a successor at random, and moves to that successor provided that there is an improvement of MPC. The computation usually halts when we have not been able to choose a better successor after a fixed number of attempts. In our case, consistently with the termination criterion used for the steepest ascent hill climbing algorithm, we test all possible successors before halting the search.

Method	MPC	Imprv of meta over best Π_i	Upper bound	Num. of candidates
SAHC	2.08	12.3%	2.38	14,950
SHC	2.07	12.1%	2.38	4,967

Table 3: Performance of stochastic optimization methods on the Ambient collection.

Method	MPC	Imprv of meta over best Π_i	Upper bound	Num. of candidates
SAHC	1.29	21.9%	1.70	23,147
SHC	1.29	21.4%	1.70	5,891

Table 4: Performance of stochastic optimization methods on the ODP-239 collection.

4.3 Random restart hill climbing

Stochastic hill climbing with random restart iteratively does hill climbing for a random amount of time, each time with a new random partition. The best partition is kept: if a new run of hill climbing produces a better solution, it replaces the stored one.

4.4 Simulated annealing

In simulated annealing, the current state may be replaced by a successor with a lower quality. That is, the algorithm sometimes goes down hills. If the objective function value of the successor (MPC') is lower than that of the current best partition (MPC), we move to the successor with a probability equal to $e^{\frac{MPC'-MPC}{T}}$, where T is a parameter that decreases slowly, eventually to 0, at which point the algorithm is doing plain hill climbing. We used $T = \frac{T_0}{log_2(i+1)}$, with $T_0 = 10$ and i set to the iteration index.

4.5 Quantum annealing

Unlike previous algorithms, in quantum annealing the successors of the current partition are not generated by moving a single object. Ideally, the neighborhood of states explored by the method should initially extend over the whole search space, and then should shrink through the computation to the nearest states. We mimic this behavior by allowing moves involving both pairs of objects and single objects. Note that in this way the number of successors grows from nm to n^2m^2.

4.6 Testing metaheuristic optimization methods

The last three algorithms introduced above favor global optimization, at the cost of exploring a larger portion of the search space. However, preliminary tests suggested that they converged to acceptably good solutions more slowly. Based on this observation, we focused on the first two algorithms, namely steepest ascent hill climbing (SAHC) and stochastic hill climbing (SHC), and made a systematic evaluation of their performance.

For each query and each method, we computed the MPC value of the meta partition, the improvement over the initial partition with the *best* MPC score, the upper bound value on Π^{opt}, and the number of candidate partitions generated during the search. In Table 3 we show the results for Ambient, averaged over the query set. In Table 4 we show the analogous results on a different test collection, termed ODP-239, that will be discussed in Section 6.1.

The meta partition was always much better than the best initial partition across both data sets. Furthermore, the results show that the meta clusterings generated by the two methods were reasonable approximations of the optimal meta clusterings because the gap between the heuristic values and the upper bound of the optimal value was not large. This was especially true for the Ambient collection, where the heuristic MPC value was indeed very close to the upper bound value for several queries. The results also show that the two methods built meta clusterings with very similar MPC values, but SHC explored a much smaller portion of the search space than SAHC. In practice, the processing times were in the order of hundreds of milliseconds on a computer of medium power (2.8 Ghz CPU, 4GB RAM), with SHC being about four times faster than SAHC.

5. META LABELING

After finding the optimal partition, we need to label its clusters. This is a very important key to the success of meta SRC as a browsing retrieval system, because a cluster with a poor label is very likely to be entirely omitted by the user even if it points to a group of strongly related and relevant documents. We do not generate cluster labels on our own, because this is a difficult task and it would require accessing the input documents. We take advantage of the fact that the individual SRC algorithms return phrase labels of high quality, and devise a procedure to select the most agreed upon labels from those given as input.

The procedure takes into account the characteristics of the set of labels provided by the individual SRC algorithms. We observed that, on average, the number of labels that are shared by a pair of SRC algorithms is only 9% of the total labels generated. By contrast, if we consider the single distinct non-stop words contained in the set of cluster labels generated for a given query, the similarity between pairs of methods is much higher, with a mean Jaccard index value of 0.24.

The meta labeling algorithm consists of three steps:

1. We associate with each cluster of the meta partition a set of candidate labels, formed by all labels under which each document in the cluster has been classified in at least one individual method.

2. We assign a score to each candidate label based on both its *extensional* coverage of the set of objects and *intensional* coverage of the set of labels. The purpose of the intensional factor is to promote syntactically different labels that refer to the same concept. The exact formula is the following:

$$Score(l) \;=\; count(obj) \cdot \sum_{w \in l} count(w), \qquad (13)$$

where $count(obj)$ is the number of search results in the cluster that are labeled by l, w is a non-stop word contained in l, and $count(w)$ is the number of distinct labels in the cluster that contain word w.

3. We select the label with the highest score.

Hereafter, the full clustering method consisting of generation of the meta partition with stochastic hill climbing followed by meta labeling will be referred to as OPTIMSRC (OPTImal Meta Search Results Clustering). As an illustration, consider the query 'Bronx'. We show in Table 5 the set of cluster labels generated by each clustering algorithm (including OPTIMSRC) that were judged to be relevant to at least one of the subtopics of 'Bronx' defined in the Ambient collection.

	kSSL (k=1)	kSSL (k=2)	kSSL (k=3)	kSSL (k=4)
OPTIMSRC	20.56	28.93	34.05	38.94
Imprv over KeySRC	14.5%*	10.6%*	10.8%*	7.5%*
Imprv over Lingo	15.7%*	5.5%	6.8%	4.3%
Imprv over Lingo3G	14.3%*	10.6%*	13.9%*	9.4%*
Imprv over search eng	4.8%	18.4%*	18.8%*	18.1%*
Imprv over best comb	5.0%	1.1%	-2.7%	-4.4%

Table 6: Retrieval performance improvement of OPTIMSRC over individual clustering methods, baseline, and best combined method.

6. EVALUATION

In this section we describe the test collections used in the experiments and three complementary techniques to validate the results of OPTIMSRC compared to those of its input algorithms.

6.1 Test collections

There is no standard test collection for evaluating SRC algorithms. In addition to using Ambient, introduced in Section 2, we created a new, larger test collection, termed ODP-239. ODP-239 combines the features of search results data with those of classification benchmarks. It consists of 239 topics, each with about 10 subtopics and 100 documents. Each document is represented by a title and a very short snippet. The topics, subtopics, and their associated documents were selected from the top levels of the Open Directory Project (`http://www.dmoz.org`), in such a way that the distribution of documents across subtopics reflects the relative importance of subtopics. Unlike Ambient, all documents are relevant to at least one subtopic and the document-subtopic assignment comes for free. ODP-239 and Ambient have complementary aspects: the former collection deals with ambiguous queries and is suitable for information retrieval, the latter is about truly multi-topic queries and is aimed at classification. ODP-239 is available for download at `http://credo.fub.it/odp239`.

6.2 Subtopic retrieval

In this section we evaluate the subtopic retrieval effectiveness of OPTIMSRC. We use the same experimental setting as previous experiments with individual methods reported in Table 2. For each topic we found the meta partition associated with the clusterings produced by KeySRC, Lingo, and Lingo3G, and computed the corresponding $kSSL$ values. This experiment was limited to the Ambient collection.[5]. The results are shown in Table 6. Asterisks are used to denote that the difference is statistically significant, using a two-tailed paired t test with a confidence level in excess of 95%.

OPTIMSRC obtained better results than any individual method for all evaluation measures, with most differences being statistically significant. Unlike the individual meth-

[5]The computation of $kSSL$ requires that we know which cluster labels are relevant to each query's subtopic. Such relevance judgments were available to us for Ambient, but not for ODP-239, which is why this set of experiments could not be replicated on the former test collection.

Bronx query	OPTIMSRC	KeySRC	Lingo	Lingo3G
Subtop1	Borough of New York city / Bronx County	city's borough / Bronx NY / Bronx district	Borough of New York city / Bronx County / Bronx district	New York / Bronx County
Subtop2	Bronx River	Bronx River	Bronx River	Bronx River
Subtop3	Bronx Music		Music	Bronx Music
Subtop4	Bronx Zoo	Bronx Zoo	Bronx Zoo	Bronx Zoo

Table 5: Cluster labels relevant to the Ambient subtopics for the query 'Bronx'. The subtopic definitions are the following. Subtop1: 'The Bronx, one of the five boroughs of New York City'; Subtop2: 'Bronx River, a river that flows south through The Bronx'; Subtop3: 'The Bronx(band), an American punk rock band'; Subtop4: 'Bronx Zoo'.

Collection	OPTIMSRC	KeySRC	Lingo	Lingo3G
Ambient	0.27	0.21	0.22	0.14
ODP-239	0.25	0.20	0.19	0.15

Table 7: Mean silhouette coefficient value of SRC systems on the Ambient and ODP-239 collections.

ods, OPTIMSRC improved on the baseline not only for $k \geq 2$ but also for $k = 1$. Note that for $k = 1, k = 2$, the performance of OPTIMSRC was even better than the performance that we would obtain by selecting the best individual method for each query. Although somewhat surprising, this is perfectly consistent with the approach proposed here, because the meta strategy does not combine the performance results of individual methods but truly integrates their performance components.

6.3 Internal measures: the silhouette coefficient

Internal indices of cluster validity use only information present in the data set. Most of them (see [11] for a comprehensive summary) are based on the notions of cluster *cohesion*, i.e., how close the objects in a cluster are, and cluster *separation*, i.e., how distinct a cluster is from other clusters. The popular method of *silhouette* coefficients combines both cohesion and separation. The silhouette coefficient s_i for an individual object i is defined as:

$$s_i = \frac{b_i - a_i}{max\,(a_i, b_i)}, \tag{14}$$

where a_i is the average distance of object i from all other objects in its cluster, and b_i is the minimum average distance to objects in another cluster. The value of the silhouette coefficient can vary between -1 and 1; a negative value is undesirable because this corresponds to $a_i > b_i$, meaning that the distance of data objects to the center of their cluster is greater than the distance to the next nearest cluster. The silhouette coefficient S of a clustering is defined as the average silhouette coefficient of all objects.

We computed the S value of the individual clusterings and of OPTIMSRC for each query, representing objects as tf-idf weighted term vectors (up to text normalization) and using the formula $1 - \textit{cosine similarity}$ as distance function. In Table 7, we show the results for each collection averaged over the corresponding query set.

As OPTIMSRC achieved much better results than the individual methods, it may be conjectured that the use of meta clustering helped to remove noisy object-cluster assignments (i.e., objects that could not be clearly assigned to one cluster), thus leading to more recognizable cluster structures in the feature space.

On the other hand, the relatively low mean absolute values of S for all SRC systems, including OPTIMSRC, indicate that the separation between clusters was not always clear. This is not surprising, given that neither the individual clusterings nor OPTIMSRC were explicitly optimized for cohesion or separation. Also, perhaps more importantly, we must consider that such algorithms perform sophisticated forms of feature construction to detect inter-document similarities that go beyond the bag-of-words approach. Thus, the feature space in which we defined the distance function used to compute S is not the same as that employed to do the clustering. The former is the space of the original single terms describing the documents, the latter is a space of phrases that do not necessarily occur in exactly the same form in the documents to which they are assigned by the algorithms.

6.4 Ground-truth validation

Ground truth validation is aimed at assessing how good a clustering method is at recovering known clusters (referred to as classes) from a gold standard partition. For this experiment we use only the ODP-239 collection, because many documents in Ambient are not assigned to any category (i.e., those search results that were not relevant to any of the query's subtopics listed in Wikipedia).

Several evaluation measures are available for this task (see [9] for a detailed summary), including the well known F_β *measure* [15] that combines precision P and recall R:

$$F_\beta = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}, \tag{15}$$

with

$$P = \frac{TP}{TP + FP},\ R = \frac{TP}{TP + FN} \tag{16}$$

where TP, TN, FP, and FN are respectively, the number of true-positives (i.e., two documents of the same class assigned to the same cluster), true-negatives (i.e., two documents of different classes assigned to different clusters), false-positives (i.e., two documents of different classes assigned to the same cluster), and false-negatives (i.e., two documents of the same class assigned to different clusters). The parameter β is a weighting factor for the importance of the recall (or precision).

Because separating documents of a same class usually has a worse effect than placing pairs of documents of different classes in the same cluster, at least in the SRC domain,

Method	F_1	F_2	F_5
OPTIMSRC	0.313	0.341	0.380
KeySRC	0,295	0.318	0.341
Lingo	0.273	0.283	0.294
Lingo3G	0.311	0.292	0.285

Table 8: Classification performance on the ODP-239 collection measured as mean (micro-averaged) F_β over the set of queries, for several values of β.

we give more weight to recall. In Table 8 we show the mean (micro-averaged) F_β values for each clustering method, with $\beta = 1, 2, 5$.

OPTIMSRC clearly outperformed the other methods for all evaluation measures, with higher values of β leading to greater performance improvements, up to 11.76 % over the best individual method for $\beta = 5$. Note that although OPTIMSRC was a clear improvement over individual methods, its performance is, on an absolute scale, still relatively low. This is probably due to the intrinsic difficulty of the classification task on the ODP-239 collection, where documents are very short and subtopics do not always have very distinct meanings.

7. RELATED WORK

While there is no earlier work on meta SRC, the general problem of finding a meta (or consensus) clustering from multiple partitions has been approached from various perspectives (e.g., graph-based, statistical, and combinatorial), using, among others: hypergraph partitioning [13], co-association matrix [5], mixture model [14], and Bayesian approach [17].

Most relevant to us is the work on finding the *median* partition, that is the partition that minimizes the distance to the given partitions. Similar to our paper, the key to finding the most important commonalities and differences among partitions are the decisions made on single pairs of objects. Under the hypothesis that the distance between partitions is measured by the number of disagreements, i.e., $b + c = \binom{n}{2} - (a + d)$, the median partition problem is known to be NP-complete [16] and various heuristics can be used for approximating it [6]. In contrast to our work, the objective function is strictly modeled after the notion of binary agreements/disagreements, in a Rand index style.

8. CONCLUSIONS AND FUTURE WORK

In this paper we studied the problem of meta search results clustering. We introduced a novel probabilistic criterion for combining the results of a given set of partitions of n objects into m clusters, and showed that a simple stochastic optimization algorithm delivers fast approximations of the optimal value. Through a range of evaluation techniques, we showed that optimal meta SRC, enriched with meta labeling, is more effective than the individual SRC algorithms, and it is also efficient for real-time applications.

Two natural research directions are the extension of the proposed framework to partitions with a variable number of clusters and to partitions of different but overlapping sets of objects (e.g., web clustering engines that fetch their search results from distinct search engines). Future work will also include an experimental comparison with other meta clustering methods such as finding the median partition.

9. ACKNOWLEDGMENTS

We would like to thank Stanislaw Osiński and Dawid Weiss for providing us with the results of Lingo and Lingo3G, and four anonymous reviewers for their comments.

10. REFERENCES

[1] A. Bernardini, C. Carpineto, and M. D'Amico. Full-Subtopic Retrieval with Keyphrase-Based Search Results Clustering. In *Proceedings of Web Intelligence 2009, Milan, Italy*, pages 206–213. IEEE Computer Society, 2009.

[2] R. J. G. B. Campello. A fuzzy extension of the Rand index and other related indexes for clustering and classification assessment. *Pattern Recognition Letters*, 28(7):833–841, 2007.

[3] C. Carpineto, S. Mizzaro, G. Romano, and M. Snidero. Mobile Information Retrieval with Search Results Clustering: Prototypes and Evaluations. *JASIST*, 60(5):877–895, 2009.

[4] C. Carpineto, S. Osiński, G. Romano, and D. Weiss. A survey of Web clustering engines. *ACM Computing Survey*, 41(3), 2009.

[5] A. Fred and A. Jain. Data clustering using evidence accumulation. In *ICPR*, pages 276–280, 2002.

[6] A. Goder and V. Filkov. Consensus Clustering Algorithms: Comparison and Refinement. In *Proceedings of ALENEX 2008, San Francisco, CA, USA*, pages 109–117, 2008.

[7] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985.

[8] S. Luke. *Essentials of Metaheuristics*. 2009. available at http://cs.gmu.edu/∼sean/book/metaheuristics/.

[9] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[10] S. Osiński and D. Weiss. A Concept-Driven Algorithm for Clustering Search Results. *IEEE Intelligent Systems*, 20(3):48–54, 2005.

[11] M. S. P.-N. Tan and V. Kumar. *Introduction to Data Mining*, chapter 8: Cluster analysis: basic concepts and algorithms, pages 487–568. Addison Wesley, 2005.

[12] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of American Statistical Association*, 66:846–850, 1971.

[13] A. Strehl and J. Ghosh. Cluster Ensembles – A Knowledge Reuse Framework for Combining Multiple Partitions. *Journal of Machine Learning Research*, (3):583–617, 2002.

[14] A. Topchy, A. K. Jain, and W. Punch. Clustering ensembles: models of consensus and weak partitions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1866–1881, 2005.

[15] K. van Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, 1979.

[16] Y. Wakabayashi. The Complexity of Computing Medians of Relations. *Resenhas*, 3(3):323–349, 1998.

[17] H. Wang, H. Shan, and A. Banerjee. Bayesian Cluster Ensembles. In *SDM 2009*, pages 209–220, 2009.