Topical Clustering of Search Results

Ugo Scaiella
Dipartimento di Informatica
University of Pisa, Italy
scaiella@di.unipi.it

Paolo Ferragina
Dipartimento di Informatica
University of Pisa, Italy
ferragina@di.unipi.it

Andrea Marino
Dip. di Sistemi e Informatica
University of Florence, Italy
andrea.marino@unifi.it

Massimiliano Ciaramita
Google Research
Zürich, Switzerland
massi@google.com

ABSTRACT

Search results clustering (SRC) is a challenging algorithmic problem that requires grouping together the results returned by one or more search engines in topically coherent clusters, and labeling the clusters with meaningful phrases describing the topics of the results included in them.

In this paper we propose to solve SRC via an innovative approach that consists of modeling the problem as the *labeled clustering* of the nodes of a newly introduced *graph of topics*. The topics are Wikipedia-pages identified by means of recently proposed topic annotators [9, 11, 16, 20] applied to the search results, and the edges denote the relatedness among these topics computed by taking into account the linkage of the Wikipedia-graph.

We tackle this problem by designing a novel algorithm that exploits the spectral properties and the labels of that graph of topics. We show the superiority of our approach with respect to academic state-of-the-art work [6] and well-known commercial systems (CLUSTY and LINGO3G) by performing an extensive set of experiments on standard datasets and user studies via Amazon Mechanical Turk. We test several standard measures for evaluating the performance of all systems and show a relative improvement of up to 20%.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Clustering*; I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Text analysis*

General Terms

Algorithms, Experimentation.

1. INTRODUCTION

Search Results Clustering (referred to as SRC) is a well-known approach to help users search the web [5]. It consists of clustering the short text fragments (aka snippets), returned by search engines to summarize the context of the searched keywords within the result pages, into a list of folders. Each folder is labeled with a variable-length phrase that should capture the "topic" of the clustered result pages. This labeled clustering offers a complementary view to the flat-ranked list of results commonly returned by search engines, and users can exploit this new view to acquire new knowledge about the issued query, or to refine their search results by navigating through the labeled folders, driven by their search needs. See Fig. 1 for an example.

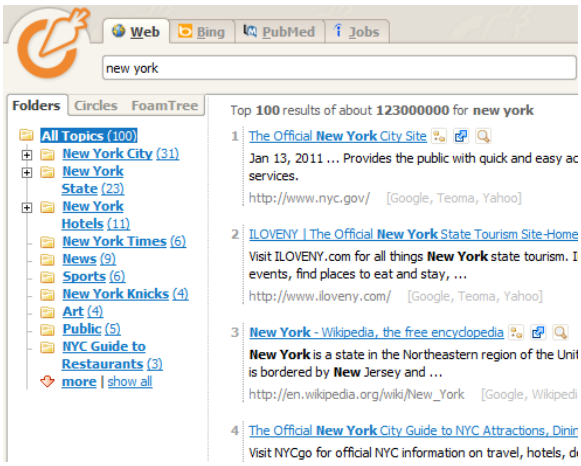

Figure 1: The web interface of LINGO3G, the commercial SRC system by CARROTSEARCH.

This technique can be particularly useful for polysemous queries, but it is hard to implement efficiently and effectively [5]. This is due to many reasons. Efficiency imposes that the clustering must use only the short text of each snippet –otherwise the download of the result pages would take too long. Efficacy requires that the size of the clusters should be reasonable –otherwise too large or too small clusters would be useless for users–, the number of clusters should be limited, e.g., to 10 –to allow a fast and simple glance of the topics of the underlying search results–, the composition of the clusters should be diversified and ensure the coverage of the topics expressed by the search results, and the labels of the clusters should be meaningful and intelligible –to allow the users an efficient and effective browsing of the search results via the folder labels.

WSDM'12, February 8–12, 2012, Seattle, Washington, USA.

These specific requirements cannot be addressed by traditional clustering algorithms. Numerous approaches have been proposed in the recent past to solve this problem both as commercial systems, CLUSTY and LINGO3G are the most representative examples, and as academic prototypes (see [5] for a survey). All of them rely on the (syntactic) *bag of words* paradigm applied to the short texts of the search-result snippets. This inevitably leads to two main limitations:

(a) the shortness and the fragmentation of the textual snippets makes it particularly difficult, if not impossible, to select meaningful and intelligible cluster labels. This problem is made today more significant by the diversification techniques applied by modern search-engines to their top-ranked results, which further reduces the applicability of statistically significant indicators;

(b) the polysemy or synonymy of terms often defeats the classical clustering approaches when they are applied onto the short snippets, because they are based on similarity measures that deploy just syntactic matches and/or *tf-idf* schemes.

1.1 Topic annotators

A recent line of research [9, 11, 16, 20] has started to successfully address the problem of detecting short and meaningful sequences of terms which are linked to relevant Wikipedia pages. These hyper-links constitute a sort of *topic annotation* for the input text and often solve synonymy and polysemy issues, because the identified Wikipedia pages can be seen as representation of specific and unambiguous topics. As an example, let us consider the following text fragment:

(1) US president issues Libya ultimatum

These topic-annotators are able to detect "US president", "Libya" and "ultimatum" as meaningful phrases to be hyperlinked with the topics represented by the Wikipedia pages dealing with the President of the United States, the nation of Libya and the threat to declare war, respectively.

We argue in the present paper that this contextualization of the input text might be very powerful in helping to detect the semantic similarity of syntactically different phrases, which is actually one of the limitations of the classical similarity measures. Indeed, consider the following text fragment:

(2) Barack Obama says Gaddafi may wait out military assault

It would be difficult to detect the tight relationship between phrases (1) and (2) by using classical similarity measures based on word matches, *tf-idf* or co-occurrences. On the contrary, the topics attached to the input texts by topic-annotators might allow one to discover easily this connection by taking into account the Wikipedia link-structure.

In addition, the disambiguation task performed by these annotators could allow to prevent correlation errors due to ambiguous words. As an example consider the following two fragments which are syntactically very similar:

(3) the paparazzi photographed the star

(4) the astronomer photographed the star

By considering just their one-word difference it would be hard to figure out the wide *topic distance* between the two fragments. On the contrary, the topic annotators would link the word "star" in the first fragment to the Wikipedia page entitled "Celebrity" and, in the second fragment, to the page that deals with the astronomical object. And since these two pages (topics) are far in the Wikipedia graph, an algorithm could easily spot the semantic distance between the two phrases.

1.2 Topical clustering of snippets

A first application of topic-annotators was presented in [14], where the authors used the annotated topics to extend the classical cosine-similarity measure in order to cluster long and well-formed texts. Apart from this result, to the best of our knowledge, no result is known in the literature that relies uniquely onto this novel annotation process in terms of both text representation and similarity measures.

In this paper we propose to move away from the classic *bag-of-words* paradigm towards a more ambitious *graph-of-topics* paradigm derived by using the above topic-annotators, and develop a novel labeled-clustering algorithm based on the spectral properties of that graph. Our solution to the SRC problem then consists of four main steps:

1. We deploy TAGME[1] [9], a state-of-the-art topic annotator for short texts, to process on-the-fly and with high accuracy the snippets returned by a search engine.
2. We represent each snippet as a richly structured *graph of topics*, in which the nodes are the topics annotated by TAGME, and the edges between topics are weighted via the relatedness measure introduced in [19].
3. Then we model SRC as a labeled clustering problem over a graph consisting of two types of nodes: topics and snippets. Edges in this graph are weighted to denote either topic-to-topic similarities or topic-to-snippet memberships. The former are computed via the Wikipedia linked-structure, the latter are discovered by TAGME and weighted via proper statistics.
4. Finally, we design a novel algorithm that exploits the spectral properties of the above graph to construct a *good* labeled clustering in terms of diversification and coverage of the snippet topics, coherence of clusters content, meaningfulness of the cluster labels, and small number of balanced clusters.

The final result will be a *topical decomposition* of the search results returned for a user query by one or more search engines. We have tested our approach on publicly available datasets using some standard measures plus a specific measure recently introduced in [6] that estimates the search-length time for a user query. Our experiments show that our approach achieves a relative improvement of up to 20% with respect to current state-of-the-art work [6]. We also complemented these experiments with a *user study* based on Mechanical Turk[2] aimed at comparing the quality of our cluster labels against two well-known commercial SRC systems: CLUSTY and LINGO3G. In this case our system is the best in producing semantically diversified labels over a public dataset of TREC queries, and it is the second best in terms of topics coverage compared to the gold standard sub-topics provided with the queries.

[1] `http://tagme.di.unipi.it`

[2] The crowd-sourcing service hosted by Amazon. `http://mturk.com`

Summarizing, the main contributions of this work are:

- the new *graph of topics* representation for short texts, based on the annotation by TAGME, that replaces the traditional *bag of words* paradigm (see Section 3);
- a new modeling of the SRC problem as the labeled clustering of a weighted graph consisting of topics and snippets (see Section 4);
- a novel algorithm for the labeled clustering of the above graph that exploits its spectral properties and its labeling (see Section 4);
- a wide set of experiments aimed at validating our algorithmic choices, optimizing the parameter settings and comparing our approach against several state-of-the-art systems over standard datasets [6]. The result is a relative improvement up to 20% for several standard measures (see Section 5);
- a large user study conducted on Amazon Mechanical Turk, which is aimed at ascertain the quality of the cluster labels produced by our approach against two commercial systems, namely CLUSTY and LINGO3G, based on 100 queries drawn from the TREC Web Track. This provides evidence that our system is the best in producing diversified labels, and it is competitive in terms of topics coverage (see Section 5.5).

We argue that the breakthrough performance of our approach over this "difficult problem and hard datasets" [6] is due to the successful resolution of the synonymy and polysemy issues which inevitably arise when dealing with the short and sparse snippets, and constitute the main limitation of known systems [5] which rely on syntactically-based techniques.

2. RELATED WORK

An in-depth survey of SRC algorithms is available in [5]. It is worth noting that most previous works exploit just simple syntactic features extracted from the input texts. They differ from each other by the way these features are extracted and by the way the clustering algorithms exploit them. Many approaches derive single words as features, which however are not always useful in discriminating topics and not always effective in describing clusters. Other approaches extract phrases [8], or build a different representation of the input texts through a decomposition of the vector space [21], or by mining a query-log [25]. Liu et al. [17] presented an approach that exploits spectral geometry for clustering search results: the nodes of the graph they considered are the documents returned by the underlying search engine and they use cosine similarity over traditional *tf-idf* representation of texts as weights for the edges. Even this technique relies on syntactic features and it has been evaluated over datasets composed of a few thousands long documents, i.e., not just snippets, which is obviously very different from our setting where we wish to cluster on-the-fly a few hundreds of short text fragments.

Recently Carpineto et al. [6] presented a (meta-)SRC system that clusters snippets by merging partitions from three state-of-the-art text clustering algorithms such as singular value decomposition, non-negative matrix factorization and generalized suffix trees. They also introduced a new, more realistic, measure for evaluating SRC algorithms that properly models the user behavior (called SSL_k) and takes into account the time a user spends to satisfy his/her search needs. Several experiments showed that this meta-SRC system yields considerable improvements with respect to previous work on datasets specifically built for this task. Again, these approaches rely on syntactic matches and *tf-idf* features (the term-document matrix), so they suffer from the sparsity of the short texts and the polysemy/synonymy of their terms, as argued above.

The clustering of our *graph of topics* might recall to the reader the Topical Query Decomposition problem introduced in [4]. However that setting is different from ours because it deals with query-logs and tries to decompose a query into other queries by exploiting valuable, but not easily available, information about past queries and users behavior. Conversely we try to cluster search results according to their topics detected by TAGME and deploying the "semantics" underlying the link-structure of Wikipedia.

Finally we mention that from a certain point of view our work could be considered somewhat related to similarity measures for short texts such as those proposed in [10, 22], or to approaches in which the document representation is enriched with features extracted from external knowledge-bases such as [2, 12, 13, 14]. However, all these approaches are either not designed for short texts or cannot be executed on-the-fly, which are two key requirements of our scenario.

3. THE GRAPH OF TOPICS

The traditional approach to IR tasks is to represent a text as a bag of words in which purely statistical and syntactic measures of similarity are applied. In this work we propose to move away from the classic bag-of-words paradigm towards a more ambitious *graph-of-topics* paradigm derived by using the TAGME annotator [9].

The idea is to deploy TAGME to process on-the-fly and with high accuracy the snippets returned by search engines. Every snippet is thus annotated with a few topics, which are represented by means of Wikipedia pages. We then build a graph consisting of two types of nodes: the topics annotated by TAGME, and the snippets returned by the queried search engines. Edges in this graph are weighted to denote either topic-to-topic similarities, computed via the Wikipedia linked-structure, or topic-to-snippet memberships, weighted by using proper statistics derived by TAGME.

This new representation provides a stunning contextualization for the input snippets because it helps to relate them even though they are short and fragmented. Figure 2 provides an illustrative example over the query `jaguar`: on the left-hand side are shown some snippets returned by a search engine for that query, on the right-hand side are shown some of the topics identified in those snippets by TAGME. The dashed edges represent the topic-to-snippet annotations, weighted with a score (called ρ-score in [9]) that denotes the reliability/importance of that annotation for the input text. The solid edges represent the topic-to-topic similarities, weighted by the relatedness measure introduced in [19] and recalled in the following Section 4 (here, the thickness of these edges is proportional to that measure).

This graph enhances the traditional term-document matrix deployed in most previous works [5]. In fact, that matrix would be very sparse for our input snippets which are very short and fragmented, and thus difficult to be related by means of statistics or terms co-occurrences.

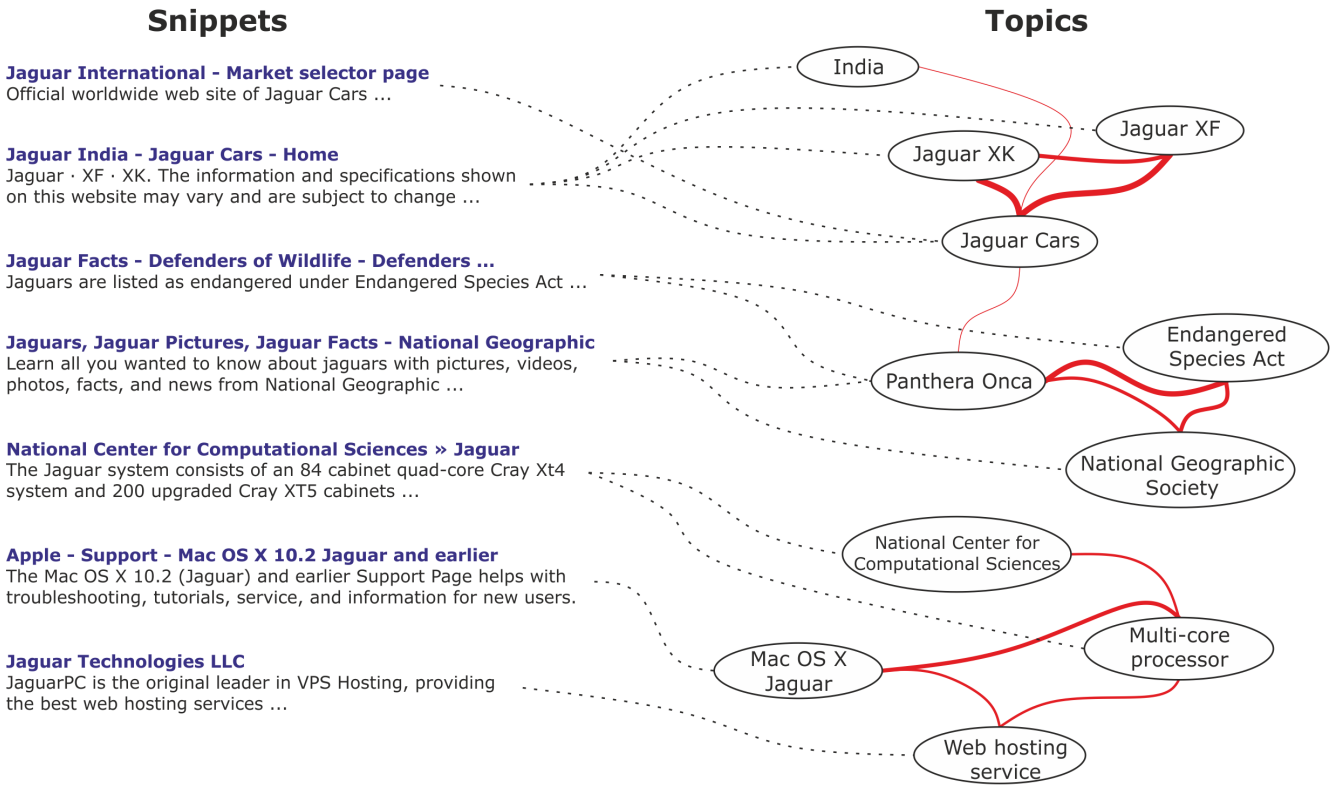

Figure 2: The *graph of topics* representation for texts.

The snapshot of the graph of topics in Fig. 2 shows the potential of this novel text representation. By looking at the graph structure one can quickly identify three main themes for the input snippets: automobiles, animals and IT. Note that the last theme is easily identifiable even though the last three snippets do not share any significant term; the second theme is identifiable even if the third and the forth snippets share just the term "jaguar", which is the query and thus obviously occurs everywhere. We finally point out that the snippet-to-topic edges could be deployed to discard some un-meaningful topics, e.g. the topic `India`, that are unrelated to the main themes of the query and are clearly "disconnected" from the rest of the graph.

It goes without saying that this graph depends strongly on the annotation of TAGME and on the content and linked-structure of Wikipedia. Moreover it does not represent a perfect ontology for the input snippets; e.g., the topic `Panthera Onca` is slightly related to `Jaguar Cars`, and more in general some relations could be missing. Nevertheless, as our wide set of experiments will show in Section 5, the coverage and the quality of our labeled-clustering algorithm proved superior to all known SRC-systems.

4. THE ALGORITHM

According to the previous section an instance of our problem consists of a graph whose nodes are n snippets $\mathcal{S} = \{s_1, ..., s_n\}$ and r topics $\mathcal{T} = \{t_1, ..., t_r\}$ that are identified by TAGME in $\mathcal{S}$.

Given a snippet s and a topic t, we denote by $\rho(s,t)$ the score assigned by TAGME to the annotation of s with topic t. This score is computed by TAGME taking into account the coherence of the disambiguated topic with respect to the other topics in the text plus some other statistical features drawn from the text corpus of Wikipedia. Indeed $\rho(s,t)$ represents the reliability/importance of the topic t with respect to the text s (for details about this score see [9]) and it is used to weight the edge (s,t) in the graph.

Given two topics t_a and t_b (i.e. Wikipedia pages), we can measure their relatedness $rel(t_a, t_b)$ by using the scoring function of [19],[3] which is mainly based on the number of citations and co-citations of the corresponding pages of Wikipedia:

$$rel(t_a, t_b) = \frac{\log(|in(t_a)|) - \log(|in(t_a) \cap in(t_b)|)}{\log(W) - \log(|in(t_b)|)} \qquad (5)$$

where $in(t)$ is the set of in-links of the page t, and W is the number of all pages in Wikipedia. We make the assumption that $|in(t_a)| \geq |in(t_b)|$ and thus this measure is symmetric.

For the sake of presentation, we denote by G_t the weighted graph restricted to the topics $\mathcal{T}$ detected in the input snippets, and appearing on the right-hand side of Fig. 2. Moreover we denote by $S(t) \subseteq \mathcal{S}$ the subset of snippets which are annotated with topic t, so the snippets s such that $\rho(s,t) > 0$; and for any set of topics T, we use $S(T) = \cup_{t \in T} S(t)$ as the set of snippets annotated with at least one topic of T.

Given an integer m, we solve the SRC problem by addressing three main tasks:

(a) create a *topical decomposition* for G_t consisting of a set $\mathcal{C} = \{T_1, ..., T_m\}$ of disjoint subsets of $\mathcal{T}$;

(b) identify a labeling function $h(T_i)$ that associates to each

[3]Other measures could be considered, this is however beyond the scope of the current paper.

set of topics $T_i \in \mathcal{C}$ the one that defines its general theme.

(c) derive from the topical decomposition $\mathcal{C}$ and from the labeling function $h(\cdot)$, a labeled clustering of the snippets into m groups. For each set of topics T_i, we create a cluster consisting of the snippets $S(T_i)$, and then label it with $h(T_i)$.

Such a topical decomposition has to exhibit some suitable properties, which will be experimentally evaluated:

- High *snippet coverage*, i.e. maximize the number of snippets belonging to one of the m clusters.
- High *topic relevance*, i.e. maximize the ρ scores of the topics selected in $\mathcal{C}$, namely $\sum_{s\in\mathcal{S}} \max_{t\in T_1\cup\ldots\cup T_m} \rho(s,t)$.
- High *coherence* among the topics contained in each cluster $T_i \in \mathcal{C}$, that is maximize $\sum_{t_j,t_z\in T_i} rel(t_j,t_z)$.
- Enforce *diversity* between topics contained in different clusters, namely, for each pair $T_1, T_2 \in \mathcal{C}$, minimize the value of $\sum_{t_i\in T_1, t_j\in T_2} rel(t_i,t_j)$.
- Enforce *balancing* over the sizes of the clusters induced by the topical decomposition, namely maximize $\min_{T_i\in\mathcal{C}} |S(T_i)|$ and minimize $\max_{T_i\in\mathcal{C}} |S(T_i)|$.

As in most previous work, we will aim at forming 10 clusters in order to ease the reading of their labels. Our experiments will show that it is preferable to set $m > 10$ and then merge the smallest $m-10$ clusters into a new one that represents a sort of *container* for rare or not so trasparently meaningful topics of the query (typically labeled with `"Other topics"`). Section 5.1 will evaluate the impact of the value of m onto the clustering quality.

Finally, we note that there could be snippets in which TAGME is not able to identify any topic so that they are not represented in our graph. We limit the set $\mathcal{S}$ to the snippets that obtained at least one annotation from TAGME. Section 5.2 will experimentally evaluate the coverage of $\mathcal{S}$ with respect to the total set of snippets returned by the queried search engine, showing that $\mathcal{S}$ covers the 98% of them on average.

4.1 Pre-processing

First we remove from $\mathcal{T}$ the topics that cover more than 50% of the snippets, because we argue that very generic topics are not useful for clustering. Then we select from the remaining topics the most significant ones by *greedily* solving a set-cover problem in which the universe U to be covered is formed by the input snippets $\mathcal{S}$, and the collection B of covering-sets is given by the topics of $\mathcal{T}$.

We recall that the goal of the set-covering problem is to find a minimum-cardinality set cover $C \subseteq B$ whose union gives U. The particularity of our set-covering problem is that the membership of each element s (snippet) in a set t (topic) is weighted by the value $\rho(s,t)$ computed by TAGME.[4] Hence we design a special greedy algorithm that selects the next set (topic) t not based on the number of yet-uncovered elements (snippets) it contains, as in the classic greedy-approach to set covering [7], but based on the *volume* of the edges incident to t and measured as the sum of their ρ-values. The (relevant) topics selected via this greedy approach will be the nodes eventually constituting the graph G_t whose edges are weighted according to the relatedness formula in (5).

[4]A score $\rho(s,t) = 0$ indicates that s is not annotated with t.

4.2 Topical decomposition

Given the weighted graph G_t, we aim at constructing a *good* labeled clustering via spectral geometry. The goal is to find a partition of the nodes of G_t in groups such that the edges between groups are few and have low total weight, whereas edges within a group are many and have high total weight. The interpretation in terms of *topic similarity* is straightforward since the edge weights in G_t measure the relatedness between its nodes (topics), therefore the clusters produced by the spectral approach should show high intra-cluster relatedness and low inter-cluster relatedness.

In our problem, however, we cannot rely on G_t only because of the strict interplay that exists between topics and snippets, and because of the properties we wish to guarantee with our final clustering (see Section 4). Thus we propose to operate on the entire graph of topics-and-snippets of Section 3, and design a clustering algorithm that selects the next cluster of topics to be split according to the number of contained snippets and to its spectral properties over the linked structure of G_t. This selected cluster is then split into two parts which aim at minimizing intra-similarity and maximizing inter-similarity among their topics in G_t.

This is different from traditional spectral clustering techniques which deploy the spectral properties of the input graph to map its nodes in a reduced space and then apply simple clustering algorithms, such as k-means [24].

Technically speaking, our clustering algorithm deploys the normalized Laplacian matrix L_{rw}, as defined in [24]. This way the spectral decomposition induced by L_{rw} solves a relaxed version of the *normalized cut* (Ncut) objective function introduced in [18] and defined as:

$$\sum_{i=1}^{k} \frac{cut(T_i, \mathcal{T}\setminus T_i)}{vol(T_i)} \tag{6}$$

where

$$cut(T_i,T_j) = \sum_{t_a\in T_i, t_b\in T_j} rel(t_a,t_b), \tag{7}$$

$$vol(T_i) = \sum_{t_c,t_d\in T_i} rel(t_c,t_d). \tag{8}$$

L_{rw} is tightly related to the transition matrix of the weighted random walk in G_t: it is shown that minimizing Ncut means finding a cut through the graph such that a random walk seldom transitions from a group to the other one [24].

Our clustering algorithm proceeds iteratively, starting with a single large cluster (the whole G_t), and then bi-sectioning one cluster at each iteration. We concentrate our attention over the *big* clusters, namely the ones that cover more than $\delta_{\max}$ snippets, where $\delta_{\max}$ is a parameter whose value has been evaluated in our experiments and that represents the desirable maximum number of elements contained in a cluster. Among these *big* clusters, we bi-section the one that has the lowest second eigenvalue λ_2 of its L_{rw}, i.e. the normalized Laplacian matrix computed upon the sub-graph induced by that cluster. Recall that λ_2 encodes the *sparseness* of that sub-graph: so we argue that the sparser cluster is the more appropriate to be cut in order to diversify its sub-topics. The nodes of this cluster are then sorted according to their projection onto the second eigenvector of L_{rw}, and the cut point is finally found by scanning that sorted sequence and searching for the minimum of the Ncut function defined above.

As commented above, Section 4.1, the algorithm stops when it creates approximately 10 clusters or there is no more clusters to be cut. Section 5.1 will evaluate the impact of this number onto quality of the final clustering.

4.3 Snippets clustering and labeling

The final clustering of the snippets is derived from the topical decomposition of $\mathcal{T}$: each snippet is assigned to (possibly many) clusters in accordance with the snippet-to-topic annotations discovered by TAGME. These clusters of snippets could overlap: in fact, if the snippet s has been annotated with two topics t_a and t_b, and these topics belong to distinct clusters T_1 and T_2, respectively, then we will have $s \in S(T_1)$ and $s \in S(T_2)$. This is a desirable behavior because a snippet can deal with several topics [5].

The final step is then to label these clusters of snippets. This labeling plays an important role, possibly more important than the clustering itself. In fact even a perfect clustering becomes useless if the cluster labels do not clearly identify the cluster topics. This is a very difficult task since it must be executed on-the-fly and processing only the poorly composed snippets. All previous approaches tried to address this problem by exploiting different syntactic features to extract meaningful and intelligible labels [5]. Our innovative topical decomposition allows to label easily the topical clusters thanks to the topics annotated by TAGME.

Let us define the main topic $h(T_i)$ of a topical cluster T_i as the topic $t \in T_i$ that maximizes the sum of the ρ-scores between t and its covered snippets, namely

$$h(T_i) = \arg\max_{t \in T_i} \sum\nolimits_{s \in S(t)} \rho(s,t)$$

Since each topic corresponds to a Wikipedia page, we finally derive the label for the cluster of snippets $S(T_i)$ by using the title of its main topic $h(T_i)$. It could be the case that the title of the page is the same as the query string, thus limiting its utility. In this case, we append to the title the most frequent anchor text that was used in Wikipedia to refer page $h(T_i)$.

As an example consider the term `jaguar`: it is an ambiguous term, and the Wikipedia page dealing with the animal is entitled `Jaguar` (hence identical to the query). If the user submits `jaguar` as a query, the cluster related to the animal will be labeled as `Jaguar panthera onca`, since `panthera onca` is the most frequent anchor text, different from the title, used to refer the `Jaguar` page in Wikipedia.

5. EXPERIMENTAL EVALUATION

The experimental validation of our approach to SRC relies on two publicly available datasets specifically created for this context by [6]. The former dataset is called AMBIENT (AMBIguos ENTitities) and consists of a collection of 44 ambiguous queries and a list of 100 result snippets for each of them, gathered from Yahoo!'s search engine. This dataset also offers a set of sub-topics for each query and a manual association between each snippet and (possibly many, one or none) related subtopics. The second (and larger) dataset is called ODP-239 and is built from the top levels of DMOZ directory. It includes 239 topics, each with 10 subtopics and about 100 documents (about 10 per subtopic), for a total number of 25580 documents. Each document is composed by a title and a brief description, even shorter than the typical snippet-length as returned by modern search engines.

Top-5		
m	$\delta_{\max}$	F_1 measure
12	10	0.4136
12	13	0.4134
12	12	0.4132
10	14	0.4131
10	8	0.4129

Bottom-5		
m	$\delta_{\max}$	F_1 measure
5	8	0.3961
5	7	0.3960
5	11	0.3960
5	9	0.3959
5	10	0.3957

Table 1: Top-5 and bottom-5 settings over the ODP-239 dataset according to the F_1 measure.

We complemented these two experiments with a large user study comparing the quality of the cluster labels attached by our approach or by two well-known commercial systems: CLUSTY and LINGO3G. This user study was executed through Amazon Mechanical Turk and used the queries specified as "Web Track 2009 and 2010" in the TREC competition[5]. This dataset (TREC-100) is composed by 100 queries and for each of them is given a list (possibly incomplete) of descriptions of user intents behind these queries.

5.1 Tuning of our system parameters

Recall that our algorithm relies on two parameters (see Section 4.2):

m is the maximum number of clusters created by the topical decomposition;

$\delta_{\max}$ is the lower bound to the number of snippets contained in the topic-clusters that must be cut.

We evaluated the impact of these parameters by deploying the ODP-239 dataset. In this tuning phase, we make m range within $[5, 20]$ since in our context we are aiming at display at most 10 cluster labels (see Section 4). Similarly, since the main goal of a SRC-system is to improve the retrieval performance, we aim at displaying at most 10 snippets per cluster, so we make $\delta_{\max}$ ranging from 5 to 15.

We tested all $15 \times 10 = 150$ combinations evaluating the F_1 measure. The top-5 and bottom-5 combinations are reported in Table 1: the maximum gap is less than 0.02 (2%), that shows the robustness of our algorithm to these parameter settings. The best setting is $m = 12$ and $\delta_{\max} = 10$, which validates the necessity to set $m > 10$ as argued in Section 4. This setting will be used in all the following experiments.

5.2 Coverage analysis

We experimentally assessed the suitability of using TAGME in the SRC context by measuring the number of its annotations per input snippet. Results confirmed our choice: more than 98% of snippets are covered by at least one TAGME's annotation, and 5 is the average number of annotations per snippet attached by TAGME for both datasets, AMBIENT and ODP-239, as shown in Figure 3.

We also evaluated the impact of the pruning executed by the pre-processing phase over the total set of topics extracted by TAGME (Section 4.1). It could be the case that some snippets remain orphan of topics, because their annotated topics have been pruned, and thus they are not assigned to any topical cluster. Our experiments show that less than 4% orphan snippets are generated (namely, less than 2.4% and 3.9% on average for the AMBIENT and ODP-239 dataset, respectively).

[5] `http://trec.nist.gov/`

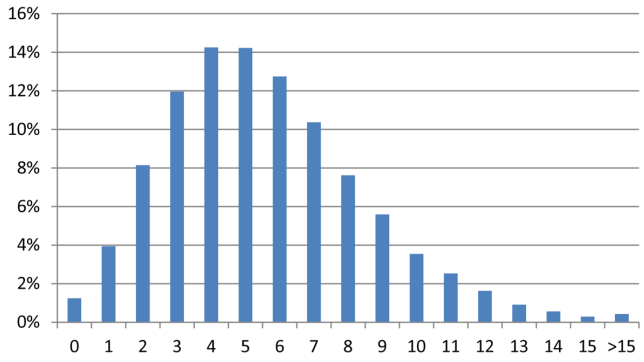

Figure 3: The distribution of the number of topic annotations per snippet, over AMBIENT and ODP-239. Column 0 corresponds to the percentage of snippets that did not get annotated by TAGME.

5.3 Subtopic retrieval

The main goal of an SRC system is to improve the retrieval performance when the user is interested in finding multiple documents of any subtopic of the query he/she issued. To this aim, [6] defined a new evaluation measure which was called the *Subtopic Search Length under k document sufficiency* (SSL_k). Basically it computes the "average number of items (cluster labels or snippets) 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 clusters with labels relevant to the subtopic at hand are opened". Moreover, if it is not possible to find the sufficient number of documents (k) via the clustering –e.g. because the clusters with an appropriate label are not enough or do not exist– then the user has to switch to the full ranked result-list and thus the search length is further increased by the number of results that the user must read in that list to retrieve the missing relevant documents.

This measure models in a realistic manner the time users need to satisfy their search needs by deploying the labeled clusters. In addition, this measure integrates the evaluation of clusters accuracy with the relevance of labels because, in order to minimize SSL_k, a system must create few but accurate clusters and their labels must be related to the topic which the contained snippets deal with. Also, the order of clusters affects the SSL_k measure: in this experiment we order our clusters according to their size (i.e. the biggest clusters are ranked first), as most of our competitors do[6].

However, the computation of this measure requires an expensive and intensive human work because for each query three kinds of assessments are needed: (a) it needs to create a list of sub-topics of the query; (b) it needs to relate each snippet with any (none, one or more) of the sub-topics of the query; (c) for each label produced by an SRC system to be evaluated, it needs to assess which sub-topic(s) the label is related with (if any). Thus we use the AMBIENT dataset that offers this manual annotation[7] and has been the testbed for the state-of-the-art algorithms evaluated in [6].

[6]Other rankings could be considered and this issue will be addressed in future works.

[7]We complemented the AMBIENT dataset with the assessment (c) for our system.

System	SSL_1	SSL_2	SSL_3	SSL_4
Baseline	22.47	34.66	41.96	47.55
LINGO	24.40	30.64	36.57	40.69
LINGO3G	24.00	32.37	39.55	42.97
OPTIMSRC	20.56	28.93	34.05	38.94
TOPICAL	**17.10**	**24.02**	**27.41**	**30.79**
Improv.	16.8%	17.0%	19.5%	20.9%

Table 2: Evaluation of SRC systems over the AMBIENT dataset using the SSL_k measure. The lowest the values of SSL_k, the more effective a system is. Our system is called TOPICAL.

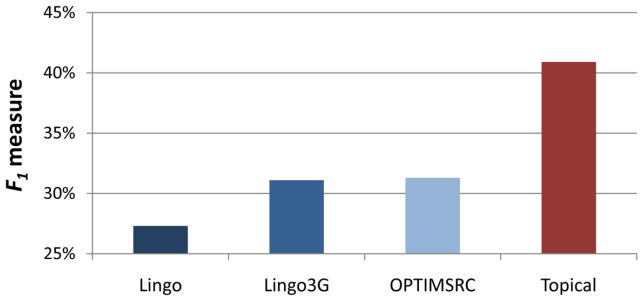

Figure 4: Evaluation of different SRC systems over the ODP-239 dataset.

Table 2 summarizes the results of our algorithm (called TOPICAL) and the main competitors on the AMBIENT dataset. LINGO [21] and OPTIMSRC [6] are two of the most recent systems appeared in the literature, and LINGO3G is a commercial system by CarrotSearch[8]. Unfortunately, to our knowledge, there is no publicly available evaluation of CLUSTY neither over this dataset nor over ODP-239.

Since the principle of the search-length can be applied also to ranked lists, we evaluated SSL_k for the flat ranked list provided by the search engine from which search results were gathered. This is used as the baseline. The experiment clearly shows that our algorithm TOPICAL outperforms the other approaches improving the SSL_k measure of about 20% on average for different values of k. The last line of Table 2 shows the relative improvement of TOPICAL over the best known approach, i.e. OPTIMSRC.

5.4 Clustering evaluation

This experiment aims at evaluating the cluster accuracy, disregarding the quality of the cluster labels. This way we can use bigger datasets because a manual assessment for each label of each algorithm is not needed. Following [6], we use the ODP-239 dataset and we use common precision and recall measures considering the subtopic memberships of DMOZ as class assignments of the ground-truth. Namely, precision P and recall R are defined as $P = \frac{TP}{TP+FP}$ $R = \frac{TP}{TP+FN}$ where True-Positives (TP) are the couples of documents of the same class assigned to the same cluster, False-Positives (FP) are the couples of documents of different classes assigned to the same cluster and False-Negatives (FN) are the couples of documents of the same class assigned to different clusters.

[8]`http://carrotsearch.com/lingo3g`

Figure 4 reports the micro-average F_1 of precision and recall over the set of queries of the ODP-239 dataset. Our approach, TOPICAL, yields an F_1 measure of 0.413 and it outperforms the previous best algorithm (OPTIMSRC) of more than 20%. This result, together with the one reported for the SSL_k in the previous section, is particularly interesting because OPTIMSRC is taking the best from three state-of-the-art clustering algorithms such as singular value decomposition, non-negative matrix factorization and generalized suffix trees.

On the other hand, the value of 0.413 for the F_1 measure could appear low in an absolute scale. However such relatively small F_1 values have been already discussed in [6], where the authors observed that the clustering task over ODP-239 is particularly hard because sub-topics are very similar to each other and textual fragments are very short.

5.5 User study

To evaluate the quality and usefulness of the labels generated by our algorithm, we devised a *user study* based on a set of 100 queries drawn from the TREC-100 dataset. To the best of our knowledge, there is no publicly available prototype for OPTIMSRC, thus we performed a comparison of our system against CLUSTY and LINGO3G.

For each system and for each query of the TREC-100 dataset, we gathered the cluster labels computed over the top-100 results returned by Yahoo!. Since CLUSTY and LINGO3G produce a hierarchical clustering, we take as cluster labels the ones assigned to the first level of the hierarchy.

To generate human ratings we used Mechanical Turk (AMT)[9]. AMT is increasingly popular as a source of human feedback for scientific evaluations (e.g., see [15]), or *artificial artificial intelligence* [3]. The evaluation task proposed to the raters needs to be designed properly; i.e., it should resemble as much as possible a natural task and it should be as simple as possible, in order to avoid unpredictable biasing and distractor effects.

We set up two evaluation tasks. The first concerns the diversification of the cluster labels. We created a survey (the *unit* of the task) for each query of the TREC-100 dataset and for each tested clustering system, by considering pairs of labels generated by each individual system. For practical reasons we limited this evaluation to the top-5 labels of each system, thus creating $\binom{5}{2} = 10$ pairs of labels per query and per system. Overall, we created about 3K units for this evaluation task. In each unit, the evaluator is given a pair of labels and we ask him/her how related they are in terms of meaning. The evaluator has to pick his/her answer from a set of four pre-defined choices: (1) unrelated; (2) slightly related; (3) very related; (4) same meaning. We required at least five different evaluators to answer each unit and we provided answers for several units (about 5%, and, obviously, they were hidden to the evaluators) as a sort of *gold standard* used to automatically discard answers from not reliable evaluators. Overall, the raters obtained about 67% total agreement and an average distance from the answer returned by AMT equals to 0.38[10].

The task breaks down the evaluation of redundancy into smaller atomic tasks where raters answer a simple question with respect to pairs of phrases. The basic assumption is that the more redundant the labels of a system the more similar they will look to the raters.

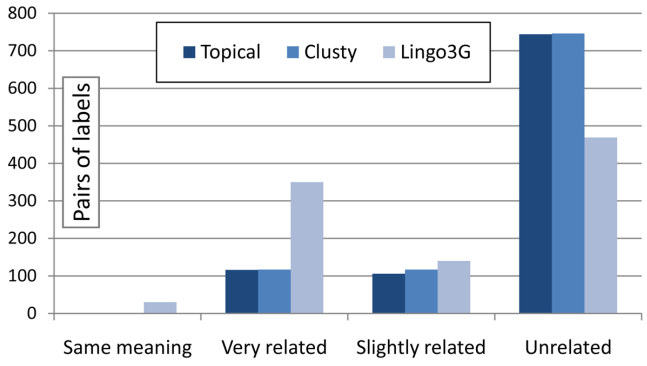

Figure 5: Evaluation of diversification of labeling produced for all queries of the TREC-100 dataset by our tested systems.

Results of this tasks are summarized in Figure 5 which clearly shows that our system TOPICAL and CLUSTY produces better diversified and less redundant labels. If we assign a rating value for each answer, starting from 0 (Unrelated) to 3 (Same meaning), we can compute a sort of "redundancy" factor for each system and it results that our system TOPICAL yields a score of 0.34, CLUSTY 0.36 and LINGO3G 0.93, where smaller means better.

The second evaluation task concerns the effectiveness of the cluster labels in *matching* the potential user intent behind the query. We created one task unit for each sub-topic, for each query of the TREC-100 dataset and for each tested system. The sub-topics are given in the TREC-100 dataset. Thus we created about 1300 units, since each query has less than 5 sub-topics on average. In each unit the evaluator is given the query, the description of the sub-topic (user intent) and the list of top-5 labels of the system to be checked, and he/she is asked to assess if the intent provided matches with at least one label provided by the system. The answer has to be taken from the list: (1) there is a label with the same meaning of the topic described by the user intent; (2) there is at least one label that is very related; (3) there is at least one label that is slightly related; (4) none of the labels are related. Thus this task intuitively aims at capturing, at least partially, the coverage guaranteed by each system with respect to a set, possibly non-exhaustive, of sub-topics which can be assumed being relevant. For this task, the rates obtained about 50% total agreement and an average distance from the answer returned by AMT equals to 0.61.

Figure 6 summarizes the results for this task[11]. In this evaluation, LINGO3G yields the best performance overall, slightly better than our approach (TOPICAL). However some comments are in order on these figures.

It is worth noticing that the top-5 labels of LINGO3G are shorter than the ones produced by our system: 1.73 versus 1.95 words per label on average. Thus they result more general and therefore might be more likely to partially match

[9]Via the `crowdflower.com` interface to AMT.

[10]Please refer to `http://crowdflower.com/self-service/faq` to read about the way answers are aggregated by `crowdflower.com`.

[11]We deployed the same kind of checks to avoid unreliable evaluators and we required at least ten different evaluators to answer to each unit because we argue that this task was more subjective with respect to the previous one.

TREC's sub-topics	LINGO3G	TOPICAL
• AVP, sponsor of professional beach volleyball events. • AVP antivirus software. • Avon Products (AVP) company. • "Alien vs. Predator" movie. • Wilkes-Barre Scranton Airport in Pennsylvania (airport code AVP).	Alternatives to Violence Project Alien Avon Products Video Volleyball Equipment Definition of AVP LCD Projectors Group Anti-Violence	Alien vs. Predator Association of Volleyball Professionals Alternatives to Violence Project Sales Avon Products Leggings NEU2 Category 5 cable LCD projector The Academical Village People

Table 3: The list of sub-topics for the query `avp` of the TREC-100 dataset and the top-ten labels produced for that query by LINGO3G algorithm and our proposed approach TOPICAL.

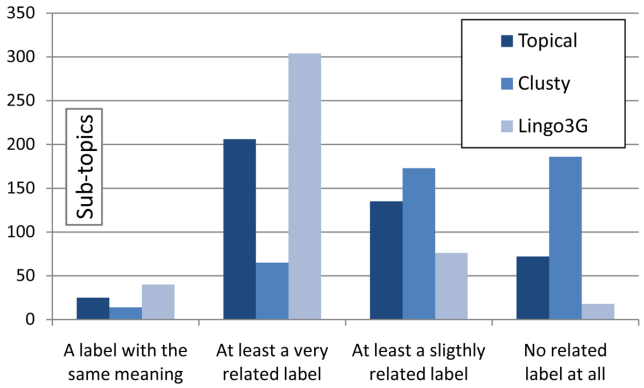

Figure 6: Evaluation of the effectiveness of the cluster labeling produced for all queries of the TREC-100 dataset by our tested systems.

one of the identified user intents. On the other hand, being more general, the labels in each set might be more likely to overlap to some extent, which seems consistent with the worse results obtained by LINGO3G in the previous redundancy evaluation.

Another potential issue to be considered is that the list of sub-topics for each query is partial and thus we are not taking into account all possible intents of the query, thus possibly giving an advantage to LINGO3G and, conversely, penalizing our system which instead offers better diversification. As an example consider the data for the query `avp` of the TREC-100 dataset showed in Table 3. The list of sub-topics is incomplete: AVP is also the name of a company that produces networking cables, the acronym of "Alternatives to Violence Project", the name of a LCD projectors manufacturer, the name of a vocal group The Academical Village People, the name of a gene that produces the NEU2 proteins. If we would have checked also these missing sub-topics, our system would have been successful whereas LINGO3G would have missed these three topics.

Even for the labels, our system is more precise and less redundant: the label *Alien* of LINGO3G corresponds to our label *Alien vs. Predator*, *Volleyball* to *Association of Volleyball Professionals*, *Group* to *The Academical Village People*. However evaluators assessed for this query `avp` that our system is the best for just one sub-topic, while for other sub-topics the outcome was a "draw".

5.6 Time efficiency

Time efficiency is another important issue in the context of SRC because the whole clustering process has to be performed on-the-fly to be useful to a user of a search engine. All figures are computed and averaged over the ODP-239 dataset and carried out on a commodity PC.

The set-covering problem, executed in the pre-processing step (Section 4.1), can be solved in $O(|\mathcal{S}| \cdot |\mathcal{T}|)$ time, where these two cardinalities are about 100 and 350, respectively, in practice. This means about 30ms.

The clustering algorithm (Section 4.2) mainly depends on the number of topics in $\mathcal{T}$, hence the number of nodes in the graph G_t. However, thanks to the pruning performed by the set-covering algorithm, this number is very small, 40 on average. Thus our spectral clustering is fast because the Laplacian matrix has a dimension of about 40. The final result is that the spectral approach takes about 350ms.

The most time consuming step in our approach is the computation of the relatedness measure defined in Section 4 that is based on the Wikipedia link-structure. Nonetheless, since we keep the whole graph indexed in internal-memory[12], the above computations are affordable in the indicated time constraints.

It goes without saying that we have to add the cost of annotating the short texts with TAGME. Although not yet engineered, TAGME is the fastest in the literature being able to annotate a snippet in about 18 ms on average with a commodity PC [9]. If the snippets to be clustered are about 100 per query (as for the datasets in our experiments), we have to add less than 2 seconds to the overall processing time. Of course, we could drop this time cost by assuming that the underlying search engine, which produces the snippets, has pre-processed with TAGME the whole collection of its indexed documents. Such a pre-processing step might also improve the topic annotation quality since more context would be available for disambiguation.

As a final note, we suggest another improvement to our system that we plan to implement in a future release of the software. It regards the spectral-decomposition step which exploits just the second eigenvalue and the second eigenvector of the Laplacian matrix. This could be quickly approximated with the well-know power method, thus avoiding the computation of all eigenvalues and all eigenvectors as it is in the current prototype.

[12]The size of such a graph is about 700Mb.

6. CONCLUSION AND FUTURE WORK

We presented a new approach to the problem of Search Results Clustering that deploys a representation of texts as *graph of concepts* rather than *bag of words*. We then designed a novel clustering algorithm that exploits this innovative representation and its spectral properties. We finally showed with a large set of experiments over publicly available datasets and user studies that our algorithm yields (significant) improvements over state-of-the-art academic and commercial systems.

Because of the lack of space we could not discuss another clustering algorithm that we have designed and tested over all datasets of Section 5. This algorithm proceeds bottom-up by carefully combining a star-clustering approach [1] with some balancedness checks on the size of the snippet-clusters to be "merged". This approach is complementary to the spectral-approach adopted by TOPICAL but, surprisingly, their performance are very close over all measures deployed in all our experiments (although TOPICAL results are still better). We argue that this is a further indication of the robustness of our experimental results and of the potentiality of the labeled and weighted graph of topics we introduced in this paper.

We strongly believe that other IR applications could benefit from this representation, indeed we are currently investigating:

(a) the design of novel similarity measures between short texts, inspired by the Earth mover's distance but now applied on subset of nodes drawn from the topic-based graphs built upon the short texts to compare;

(b) concept-based approaches to classification of news stories, or short messages in general (like tweets) [23];

(c) the application of such representation of texts to the context of Web Advertising, in which the *bag of keywords* bidden by the advertiser could be replaced by our *graph of topics* in order to enhance ad-searches or ad-page matches.

Acknowledgements

This research has been partially funded by a Google Faculty Award and Italian MIUR grants "PRIN–MadAlgo" and "FIRB–Linguistica". The authors thank professors F. Romani and G. Del Corso for insightful discussions about Laplacian matrices.

7. REFERENCES

[1] J.A. Aslam, E. Pelekhov, and D. Rus. The star clustering algorithm for static and dynamic information organization. *J. Graph Algorithms Appl.*, 8:95–129, 2004.

[2] S. Banerjee, K. Ramanathan, and A. Gupta. Clustering short texts using Wikipedia. In *ACM SIGIR*, 787–788, 2007.

[3] J. Barr, and L.F. Cabrera. AI Gets a Brain. *ACM Queue*, 4(4):24–29, 2006.

[4] F. Bonchi, C. Castillo, D. Donato, and A. Gionis. Topical query decomposition. In *ACM KDD*, 52–60, 2008.

[5] C. Carpineto, S. Osiński, G. Romano, and D. Weiss. A survey of web clustering engines. *ACM Comput. Surv.*, 41(3):1–38, 2009.

[6] C. Carpineto and G. Romano. Optimal meta-search results clustering. In *ACM SIGIR*, 170–177, 2010.

[7] V. Chvátal. A greedy heuristic for the set-covering problem. *Mathematics of Operations Research*, 4(3):233–235, 1979.

[8] P. Ferragina and A. Gulli. A personalized search engine based on web-snippet hierarchical clustering. In *WWW*, 801–810, 2005.

[9] P. Ferragina and U. Scaiella. TAGME: On-the-fly annotation of short text fragments (by Wikipedia entities). In *ACM CIKM*, 2010.

[10] E. Gabrilovich and S. Markovitch. Wikipedia-based semantic interpretation for natural language processing. *J. Artif. Int. Res.*, 34(1):443–498, 2009.

[11] J. Hoffart, M.A., Yosef, I., Bordino, H., Fürstenau, M., Pinkal, M., Spaniol, B., Taneva, S., Thater and G., Weikum. Robust Disambiguation of Named Entities in Text. In *EMNLP*, 782–792, 2011.

[12] J. Hu, L. Fang, Y. Cao, H.-J. Zeng, H. Li, Q. Yang, and Z. Chen. Enhancing text clustering by leveraging Wikipedia semantics. In *ACM SIGIR*, 179–186, 2008.

[13] X. Hu, N. Sun, C. Zhang, and T.-S. Chua. Exploiting internal and external semantics for the clustering of short texts using world knowledge. In *ACM CIKM*, 919–928, 2009.

[14] A. Huang, D. Milne, E. Frank and I. H. Witten. Clustering documents using a Wikipedia-based concept representation. In *PAKDD*, 628–636, 2009.

[15] G. Kazai, J. Kamps, M. Koolen, M. Koolen, and N. Milic-Frayling. Crowdsourcing for book search evaluation: impact of hit design on comparative system ranking. In *ACM SIGIR*, 205–214, 2011.

[16] S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti. Collective annotation of Wikipedia entities in web text. In *ACM KDD*, 457–466, 2009.

[17] Y. Liu, W. Li, Y. Lin, and L. Jing. Spectral geometry for simultaneously clustering and ranking query search results. In *ACM SIGIR*, 539–546, 2008.

[18] M. Meila and J. Shi. A random walks view of spectral segmentation. *International Workshop on Artificial Intelligence and Statistics (AISTATS)*, 2001.

[19] D. Milne and I. H. Witten. An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. *AAAI Workshop on Wikipedia and Artificial Intelligence*, 2008.

[20] D. Milne and I. H. Witten. Learning to link with Wikipedia. In *ACM CIKM*, 509–518, 2008.

[21] S. Osinski and D. Weiss. A concept-driven algorithm for clustering search results. *IEEE Intelligent Systems*, 20(3):48–54, 2005.

[22] M. Sahami and T. Heilman. A web-based kernel function for measuring the similarity of short text snippets. In *WWW*, 377–386, 2006.

[23] D. Vitale, P. Ferragina, and U. Scaiella. Classification of Short Texts by Deploying Topical Annotations. To appear on *ECIR*, 2012.

[24] U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[25] X. Wang and C. Zhai. Learn from web search logs to organize search results. In *ACM SIGIR*, 87–94, 2007.