# Enhancing Cluster Labeling Using Wikipedia

David Carmel, Haggai Roitman, Naama Zwerdling
IBM Research Lab
Haifa 31905, Israel
{carmel,haggai,naamaz}@il.ibm.com

## ABSTRACT

This work investigates cluster labeling enhancement by utilizing *Wikipedia*, the free on-line encyclopedia. We describe a general framework for cluster labeling that extracts candidate labels from Wikipedia in addition to important terms that are extracted directly from the text. The "labeling quality" of each candidate is then evaluated by several independent judges and the top evaluated candidates are recommended for labeling.

Our experimental results reveal that the Wikipedia labels agree with manual labels associated by humans to a cluster, much more than with significant terms that are extracted directly from the text. We show that in most cases even when human's associated label appears in the text, pure statistical methods have difficulty in identifying them as good descriptors. Furthermore, our experiments show that for more than 85% of the clusters in our test collection, the manual label (or an inflection, or a synonym of it) appears in the top five labels recommended by our system.

## Categories and Subject Descriptors

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

## General Terms

Algorithms

## Keywords

Cluster labeling, Wikipedia

## 1. INTRODUCTION

The volume of available electronic information is rapidly increasing with the advancements in digital processing. Moreover, massive amounts of textual data have brought about the need for efficient techniques that can organize the data in manageable forms. One of the popular approaches for organizing textual data is the use of clustering algorithms, which group a set of documents into coherent clusters. The algorithms' goal is to create clusters, where documents within a cluster should be as similar as possible and documents in one cluster should be dissimilar from documents in other clusters.

In many applications of clustering, particularly in user-interface based applications, human users interact directly with the created clusters. In such settings we must label the clusters so that users can understand what the cluster is about. A lot of research is being done on clustering algorithms and their applications in information retrieval and data mining. However, comparatively little work has been done on cluster labeling.

A popular approach for cluster labeling is to apply statistical techniques for feature selection. This is done by identifying "important" terms in the text that best represent the cluster topic. However, a list of significant keywords, or even phrases, will many times fail to provide a meaningful readable label for a set of documents. In many cases, the suggested terms, even when related to each other, tend to represent different aspects of the topic underlying the cluster. In other cases, a good label may not occur directly in the text. Hence user intervention is required to infer a proper label from the suggested terms to successfully describe the cluster's topic.

As an illustrating example, Table 1 shows the top five important terms extracted for six *Open Directory Project* (ODP) [14] topics using the JSD selection method (further described in Section 3). Each topic is represented by a cluster of 100 web documents randomly sampled from the corresponding ODP category. We observe that while the important terms extracted for all clusters seem to fairly represent the category's topic, only the terms of the first two topics seem to provide sufficient labels for those clusters (such terms are underlined in Table 1). This is not true for the rest of the topics. For example, none of the important terms for the *Electronics* topic seem to provide a good label, even though all terms are strongly related to *Electronics*.

By analyzing a sample of 100 ODP categories, we discovered that the original label associated with the category, as given by a human assessor, indeed appears in the category's text in 85% of the categories. However, these human labels are rarely identified as "significant" by feature selection methods. For example, the JSD method for feature selection identifies human labels as "significant" (appearing in the top five most important terms) for only 15% of the categories. This result implies that human labels are not necessarily significant from a statistical perspective. This further moti-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
*SIGIR'09,* July 19–23, 2009, Boston, Massachusetts, USA.
Copyright 2009 ACM 978-1-60558-483-6/09/07 ...$5.00.

vated us to seek out a cluster labeling method using external resources.

## 1.1 Our Approach

In this work we investigate the contribution of external knowledge-bases for cluster labeling. This approach is close in spirit to the work described by Syed et al. [19], who identified topics associated with a set of documents using Wikipedia. Our method first finds Wikipedia pages relevant to the cluster to be labeled. It then used the meta-data of these pages, such as categories and titles, for labeling the cluster. The last column of Table 1 shows the Wikipedia labels extracted by the labeling system for a set of ODP topics, which agree much more with the given human annotated labels.

Given a cluster of documents, we first extract the most important terms (keywords and phrases) from the text. We then identify a list of related Wikipedia pages by searching Wikipedia using a query that is based on those terms. The Wikipedia categories and titles of related pages serve as potential candidates for cluster labeling. In addition, all important terms extracted from the text of the cluster are also considered as candidates. Each candidate is evaluated by several independent judges. Section 3 describes all judges used by our system in full detail. Finally, all judgments are aggregated to select the top candidates for use as cluster labels.

We evaluated our work using a sample of the ODP collection and the 20 News-Group (20NG) collection. Following the evaluation framework described in [21], we extracted a uniform sample of 100 categories from ODP, each associated with a manual label. Similarly, the 20NG benchmark that is frequently used for clustering analysis contains 20 clusters of news-groups each associated with a manual label. We evaluated our labeling method by measuring its ability to predict the manual label of each of the clusters in the two benchmarks. Our experiments show that for both benchmarks, our labeling framework is able to provide *Match@5* $\geq$ *0.85*. This means that for more than 85% of the catgories, the manual label (or an inflection, or a synonym of it) appears in the top five recommended labels. To the best of our knowledge, this is the highest evaluation score reported so far for the cluster labeling task.

The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 describes the labeling system and its components. Section 4 provides the results of some experiments we conducted with the labeling system. Section 5 summarizes and discusses future directions.

## 2. RELATED WORK

The most common approach for cluster labeling works by identifying "important" terms in the cluster content that characterize the cluster in contrast to other clusters [6]. Important terms can be identified by naively selecting the most frequent terms in the cluster, by extracting the top weighted terms in the cluster centroid, or using any other statistical feature selection techniques [13]. For example, Geraci et al. [10] used a modified version of the information gain measure to identify words that most represent the cluster's contents and are least representative of the contents of other clusters. Several works also considered frequent phrases in the text for cluster labeling [15, 21]. Toda and Kataoka [20] also used named entities extracted from the text for labeling. However, in many cases, a labeling approach that is solely based on the cluster content may have difficulties in providing discriminative labels, as illustrated in Table 1.

Several labeling solutions look for alternative resources. One of the first systems that dealt with cluster labeling is the Scatter/Gather application [6]. In this system, in addition to the cluster's important terms, the titles of documents that are mostly close to the cluster centroid are also considered for labeling, since usually titles are much more readable than a list of terms. On the web, the anchor text associated with the in-links to the cluster web pages can be considered as candidate labels since often anchors successfully describe the pages to which they link. Glover et al. [11] demonstrated that labels extracted from the anchor text provide a much better description than those extracted from the page's content.

There is a lot of research on linguistic-based summarization techniques for multiple documents which are also related to the labeling task. For example, Radev et al. [16] generate a summary of multiple documents using cluster centroids produced by topic detection and tracking. However, multi-document summaries are usually too long to be utilized as short comprehensive labels.

Several labeling approaches attempt to enrich content-only terms by exploiting external resources for labeling, for example, the WordNet lexical database [4] was used to extract root meanings of important terms and to determine semantic relationships among these terms. Gabrilovich and Markovich [9] utilized Wikipedia to represent the meaning of a text fragment as a weighted vector of Wikipedia concepts. Semantic relatedness between two fragments is then measured through the comparison of concept vectors using the cosine similarity. Similarly, Syed et al. [19] find concepts common to a document, or a set of documents, using Wikipedia articles and spreading activation on the Wikipedia's category links graph. Both works demonstrated that categories of Wikipedia articles can successfully describe the document's common concepts. Wikipedia has recently become one of the major knowledge resource for many information retrieval tasks, including text categorization and clustering [8, 17, 12], computing semantic relatedness between concepts [18, 9], and predicting document topics [19].

Our cluster labeling solution also leverages the labeling task by Wikipedia. However, our solution has several main distinctions from previous works. First, in our approach, candidate labels are extracted from Wikipedia pages in addition to the important terms that are extracted directly from the cluster content. Thus, selected Wikipedia categories "compete" with inner terms for serving as the cluster labels. In general, Wikipedia is a very successful resource for labeling; however, inner terms should be considered for the cases when Wikipedia fails to cover the cluster content.

Second, in contrast to previous works which focus on identifying document concepts [18, 9, 19] using Wikipedia, we look for a few focused labels that will be judged by a (human) user as descriptive labels for a given documents' cluster. This distinction is reflected by the novel additional candidate judgment process applied by our system, and by the different evaluation paradigm applied in our work. While Gabrilovich and Markovitch evaluated their system's ability to identify related text fragments, and Syed et al. evaluated their system's ability to predict the concepts of a

| ODP Category | Top-5 JSD important terms | Top-5 Labels Using Wikipedia Enhancement |
|---|---|---|
| Bowling | *bowl*, bowler, lane, bowl center, league | Bowls, *Bowling*, Bowling (cricket), Bowling organisations, Bowling competitions |
| Buddhism | buddhist, *buddhism*, buddha, zen, dharma | *Buddhism*, History of Buddhism, Buddhism by country, Tibetan Buddhism, Buddhists |
| Ice Hockey | hockey, nhl, hockey league, coach, head coach | *Ice hockey*, Ice hockey leagues, Hockey prospects, Canadian ice hockey coaches, National Hockey League |
| Electronics | voltage, high voltage, circuit, laser, power supply | *Electronics*, Power electronics, Diodes, Power supplies, Electronics terms |
| Tennis Players | wimbledon, tennis, defeat, match today, wta | *Tennis Players*, Tennis terminology, Tennis tournaments, 2002 in tennis, 2000 in tennis |
| Christianity | church, catholic, ministry, christ, grace | *Christianity*, Christian denominations, Non-denominational Christianity, Christian theology, Christianity in Singapore |

**Table 1: Lists of top-5 important terms extracted using the JSD selection method and top-5 labels extracted using Wikipedia for several ODP categories. While the list of important terms fairly represents the content of the categories, these terms can serve as appropriate labels for only a few categories. On the other hand, Wikipedia's labels agree with human annotated labels much more.**

unique Wikipedia article, we evaluate our system's ability to agree with a predefined human label associated with a multi-documents set[1]. By a systematic evaluation procedure which is an integral part of our labeling framework, system parameters can be optimally tuned for each collection to be clustered and labeled.

Given several candidate labels identified by the system, there is still a need to evaluate which candidate to nominate for labeling. Candidate terms are usually evaluated using term statistics gathered from a given corpus. For example, measuring the *pointwise mutual information* ($PMI$) of a term with the rest of the cluster terms is a common evaluation approach [13]. $PMI$ between two terms is usually measured by statistical co-occurrence analysis in the collection to be clustered. Recently, de-Winter and de-Rijke [7], while comparing several labeling methods for a set of blog posts, measured PMI by gathering term co-occurrence statistics from the web. Similarly, we measure $PMI$ independently over the web and over the Wikipedia corpus.

Unfortunately, there is still no standard evaluation methodology for cluster labeling and there are no standard benchmarks to compare alternative labeling methods. In this work, we follow the evaluation framework suggested by Treeratpituk and Callan [21]. We utilize the 20NG collection, and a random sample of categories from ODP as benchmarks. The labeling system is evaluated by its ability to produce a label for each category that is "equivalent" to the category's original associated label. Two labels are considered equivalent when one is identical, an inflection, or a Wordnet's synonym of its counterpart. Our experimental results will show that our labeling system performs very well compared to previously reported results on similar benchmarks.

## 3. GENERAL FRAMEWORK FOR CLUSTER LABELING

We propose a general framework for cluster labeling using external resources. Figure 1 provides an illustration of this framework, which includes five main components: indexing, clustering, important terms extraction, candidate labels extraction, and candidate evaluation.

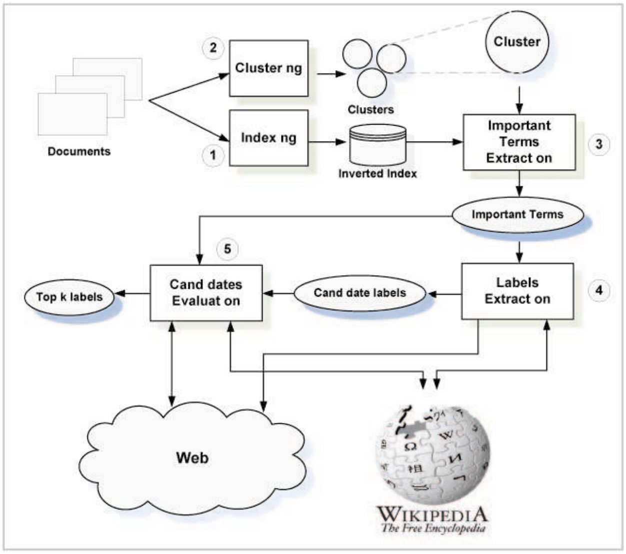


**Figure 1: A general framework for cluster labeling**

The general flow of the system can be summarized as follows. The system receives a set of textual documents as input. Initially, the documents are parsed and indexed and an inverted index is generated. This index is primarily used by other components for gathering term statistics. The documents are then clustered using the clustering component. For each generated cluster, the system extracts a set of important terms that are estimated to best represent the content of the documents of the cluster. The cluster important terms are then used to identify a list of candidate labels for the cluster. Candidate labels can be selected from the set of important terms or from external resources (e.g., Wikipedia, or the general web). Finally, the set of candidate labels is evaluated and a list of top recommended labels is returned by the system. In the rest of this section, we describe each of the system components in more detail.

[1] In addition to experimenting with individual documents, Syed et al. [19] also demonstrated their system's capabilities for a few sets of documents that are related to the same concept. However, as the authors mentioned, "results were encouraging", but their work lacks a systematic analysis for the multi-documents case.

## 3.1 Indexing

Documents are first parsed and tokenized, and then represented as term vectors in the vector space over the system's vocabulary. Term weights are determined by the well-known *tf-idf* weighting scheme of the vector-space model. The documents are indexed by generating a search index. For this purpose we use the *Lucene* open source search system[2]. The inverted index lets us obtain for each term $t$ its term frequency ($tf(t, d)$) in each document $d$, and its inverse document frequency ($idf(t)$) in the entire collection. These statistics are later used by all components of the framework.

## 3.2 Clustering

The clustering algorithms' goal is to create coherent clusters for which documents within a cluster share the same topics, and the labels provided by the system are expected to express the mutual topic of documents within the cluster.

The clustering component receives as an input the collection of documents $\mathcal{D}$ and returns a set of document clusters $\mathcal{C} = \{C_1, C_2, \ldots, C_n\}$ that cover the whole collection. Each cluster $C_i$ is a subset of documents of $\mathcal{D}$, and a document may belong to more than one cluster.

A cluster is represented by the centroid of the cluster's documents; however, the weights of terms in the cluster's centroid are slightly modified to bias terms that are distributed over many cluster documents. Thus, the weight of a term $t$ in the centroid of a cluster $C$, is set to be:

$$w(t, C) = ctf(t, C) \cdot cdf(t, C) \cdot idf(t)$$

where the cluster term frequency $ctf(t, C) = \frac{1}{|C|} \sum_{d \in C} tf(t, d)$ and the cluster document frequency $cdf(t, C) = \log(n(t, C) + 1)$; $n(t, C)$ is the document frequency of $t$ in $C$.

The labeling framework is not limited to a specific clustering algorithm; however, the coherence of the clusters identified by the system is expected to significantly affect the quality of labeling. In the worst case, when there is no semantic relatedness between the cluster documents, no meaningful labels can be expected. In Section 4, we explore the effect of cluster coherency on the quality of cluster labels provided by the system.

## 3.3 Important Terms Extraction

Given a cluster $C \in \mathcal{C}$ as input, we now wish to find a list of terms $\mathcal{T}(C) = (t_1, t_2, \ldots, t_k)$, ordered by their estimated importance, to represent the content of the cluster's documents. Such terms consist of single keywords, and N-grams of different length [21].

A naive approach for term extraction is to select the top-$k$ terms with maximal weights from the cluster's centroid. Several clustering systems apply this selection approach for cluster labeling [6]. In our experiments we use this term extraction approach as one of the baseline methods for comparison.

Important term extraction is strongly related to *feature selection* which is the process of selecting a subset of the terms for text representation, and is frequently applied by text categorization and text clustering methods [13]. Common approaches for feature selection evaluate terms according to their ability to distinguish the given text from the whole text. In our case our aim is to find a set of terms $\mathcal{T}(C)$ that best separates the cluster's documents from the entire collection.

In this work, we extract important terms using the method described by Carmel et al. [3], which was originally proposed in the context of the query difficulty model. We look for a set of terms that maximizes the *Jensen-Shannon Divergence* (JSD) distance between the cluster $C$ and the entire collection. Each term is scored according to its contribution to the JSD distance between the cluster and the collection. The top scored terms are then selected as the cluster important terms. We will experimentally show the superiority of this set of terms for cluster labeling over the top weighted terms in the cluster centroid, and over sets of terms extracted by alternative standard feature selection methods.

## 3.4 Label Extraction

Given the list of important terms $\mathcal{T}(C)$, we now wish to extract candidate labels for cluster $C$. We identify two different types of sources from which it is possible to extract such candidate labels. The first type involves labels that are extracted directly from the cluster's documents content. In this case, we follow the spirit of previous works [6, 11, 21] and consider the set of important terms themselves as potential labels for the cluster. Nevertheless, there are many cases in which important terms do not provide suitable labels or are not meaningful enough for end-users, as Table 1 shows. Therefore, we turn to external sources as complimentary sources for this task.

Similar to [19], we focus on Wikipedia as an external source from which candidate cluster labels can be extracted. We note that there may be other external sources that can be utilized for this task, such as domain-specific knowledge-bases, ontologies, or even more general sources such as the web. The main reason for focusing on Wikipedia is its attractive ability to provide high quality controlled content. Moreover, Wikipedia content has also been annotated by Wikipedia's users. These manual annotations can provide high-quality meaningful labels.

The process of candidate label extraction from Wikipedia can be summarized as follows. We first generate a search index from the latest available Wikipedia dump[3] using the *Lucene* search system. Given the list of important terms $\mathcal{T}(C)$, we execute a query $q$ against the Wikipedia index which consists of the disjunction of the important terms, where the query terms are further boosted according to their relative importance in $\mathcal{T}(C)$. The result of this query is a list of documents $\mathcal{D}(q)$ sorted by their similarity score to $q$. For each document $d \in \mathcal{D}(q)$, we then consider both the document's title and the set of categories associated with the document as potential candidate cluster labels (denoted $\mathcal{L}(C)$).

## 3.5 Candidate Label Evaluation

Candidate labels are evaluated by several *judges*[4]. Each judge gets as an input the set of candidate labels $\mathcal{L}(C)$ and the set of the cluster's important terms, $\mathcal{T}(C)$. Then, each judge evaluates the candidates according to its evaluation policy. The scores of all judges are then aggregated and the labels with the highest aggregated scores are returned.

[2] http://lucene.apache.org/

[3] http://download.wikimedia.org/enwiki/20080724/

[4] Please note that the term "judge" is used throughout the paper to symbolize an *automatic heuristic* for candidate labels evaluation.

We now present the judge types in more detail and further suggest several instantiations for each type.

### 3.5.1 MI Judge

The `Mutual Information` (MI) judge scores each candidate by the average *pointwise mutual information* (PMI) of the label with the set of the cluster's important terms, with respect to a given external textual corpus. The average PMI of a given label with the set of important terms reflects the "semantic distance" of the label from the cluster content. Hence, labels that are "closer" to the cluster content are preferred. This approach is similar in nature to the evaluation process applied by [7] while evaluating labeling methods for clusters of blog-posts.

The `MI` judge gets as input the set of candidate labels $\mathcal{L}(C)$, the set of the cluster's important terms $\mathcal{T}(C)$, and a corpus that identifies an external textual source where the PMI will be measured (e.g., the web). Given a candidate label $l \in \mathcal{L}(C)$, the following score is assigned to $l$:

$$\texttt{MI}(l, \mathcal{T}(C)) = \sum_{t \in \mathcal{T}(C)} \texttt{PMI}(l, t|corpus) \times \omega(t) \tag{1}$$

where $\omega(t)$ denotes the relative importance of important term $t \in \mathcal{T}(C)$, and $\sum_{t \in \mathcal{T}(C)} \omega(t) = 1$,

The PMI between two terms is measured by:

$$\texttt{PMI}(l, t|corpus) = \log \left( \frac{Pr\,(l, t\,|corpus\,)}{Pr\,(l\,|corpus\,) \times Pr\,(t\,|corpus\,)} \right) \tag{2}$$

The probability of a term, or a pair of terms in the given corpus, is approximated by the maximum likelihood estimation

$$Pr(x|corpus) = \frac{\#(x|corpus)}{\#(corpus)}$$

where $x$ stands for a single term or a pair of terms, $\#(x|corpus)$ is the number of occurrences of $x$ in the data, and $\#(corpus)$ is an estimation of the number of terms in the corpus.

We utilize two instantiations of this judge using two different external corpora for calculating the `PMI` values. The first judge uses the Wikipedia collection as a data source and the second evaluates PMI over the web, using the PMI evaluation scheme described in [5]. For gathering web statistics, we use the *Google n-grams* collection [2], which provides the frequency counts of English word n-grams generated from approximately 1-trillion word tokens gathered from the web, to estimate the term frequency in a large web collection.

It is important to note that the important terms are evaluated by the MI judge in exactly the same way as Wikipedia candidate labels. Each important term is scored by the average PMI with all other important terms and is compared with all other candidates according to that score.

### 3.5.2 SP Judge

The second type of judges, termed `Score Propagation` (SP) judge, scores each candidate label with respect to the scores of the documents in the result set associated with that label. This judge propagates documents' scores to candidates that are not directly associated with those documents, but share common keywords with other related labels.

Given a candidate label $l \in \mathcal{L}(C)$, by summing over the set of all documents in $\mathcal{D}(q)$ associated with label $l$, we obtain an aggregated weight for $l$, which represents the score propagation from $\mathcal{D}(q)$ to $l$:

$$\omega(l) = \sum_{d \in \mathcal{D}(q): l \in d} \frac{score(d)}{n(d)} \tag{3}$$

where $n(d)$ denotes the number of candidate labels extracted from document $d$.

After scoring the labels, we score the label keywords as follows:

$$\omega(kw) = \sum_{l \in \mathcal{L}(C): kw \in l} \omega(l) \tag{4}$$

Finally, the score assigned to each candidate label $l$ is set by the average score propagated back from its keywords. Formally:

$$\texttt{SP}(l|\mathcal{D}(q)) = \frac{1}{n(l)} \sum_{kw \in l} \omega(kw) \tag{5}$$

where $n(l)$ denotes the number of $l$'s unique keywords.

Important terms are judged identically to Wikipedia labels. Each important term is associated with all the results containing it and treated as a label. Hence, it is scored using the same scoring mechanism of the `SP` judge.

Several instantiations of this judge type can be utilized. By using different scoring mechanisms we can expect different candidates (since $\mathcal{D}(q)$) may be different) as well as different judgments. In our experiments, one such judge directly applied the Lucene scores of documents in the result set, while another judge ignored the search engine scores and used document ranking instead, setting the document score to be $score(d) \leftarrow rank^{-1}(d)$.

### 3.5.3 Score Aggregation

The final stage in candidate evaluation is to aggregate the scores from the different judges for each label. Given a list of judges, $(J_1, ...J_m)$, each candidate label is scored using a linear combination of the judge scores:

$$\texttt{score}(l|C) = \sum_{i=1}^{m} \beta_i J_i(l|C)$$

where $\sum_i \beta_i = 1$. Finally, the set of top-k scored candidates are recommended for cluster labeling.

The "optimal" set of judge weights $\{\beta\}_{i=1}^m$ can be learned using standard linear regression methods, given some training data. In this work we experimented with several "reasonable" sets of weights while leaving optimization for future work.

# 4. EXPERIMENTS

## 4.1 Data Collections

We used two data collections for experimenting with the system. The first one is the 20 News Groups (20NG) data collection [1], which consists of newsgroup documents that were manually classified into 20 different categories. Each category includes 1,000 documents, for a total collection size of about 20,000 documents. The second collection was gathered by downloading pages from the Open Directory Project (ODP) [14]. For this purpose, we randomly selected 100 different categories from the ODP hierarchy. Example categories include, among others, sub-categories of the top level ODP categories such as *Ceramic Art and Pottery.* From each category we then randomly selected up to 100 documents, resulting in a collection size of about 10,000 documents.

In both collections, the categories were manually labeled. These ground-truth "correct" labels were later used to evaluate our labeling system.

## 4.2 Evaluation and Experimental Setup

We followed the evaluation framework proposed by [21]. In this framework, a proposed label for a given cluster is considered correct if it is identical, an inflection, or a Wordnet synonym of the cluster's correct label[5]. This is a conservative evaluation approach that severely evaluates the labeling system while ignoring good labels that do not comply with these restrictive rules. Therefore, the evaluation scores reported in this work can be considered as lower bounds on the system's real performance.

Given a collection of clusters, and the parameter $k$ that indicates the number of required cluster labels, the system proposes up to $k$ labels for each cluster. The system parameters we experimented with are the feature selection method, the number of important terms for querying Wikipedia, the number of Wikipedia results to be used for label extraction, and the judges used for candidate evaluation.

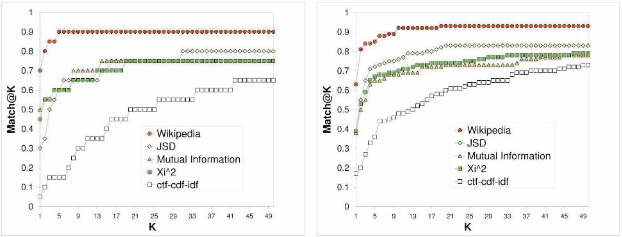
For each configuration, we evaluated the system's performance using two measures:

- **Match@K**: The relative number of clusters for which at least one of the top-k labels is correct.
- **Mean Reciprocal Rank (MRR@K)**: Given an ordered list of $k$ proposed labels for a cluster, the reciprocal rank is the inverse of the rank of the first correct label, or zero if no label in the list is correct. The mean reciprocal rank at $k$ (MRR@K) is the average of the reciprocal ranks of all clusters.

Recall that given a set of top-$k$ proposed cluster labels, a user might need to determine which labels among the top-$k$ labels are the correct ones. Therefore, it is preferable for the system to provide the shortest possible list of suggestions that contains the correct label. Both measures evaluate this system capability. The higher these measures are and the lower $k$ is, the better the system's effectiveness as perceived by the user.

## 4.3 The Effectiveness of Using Wikipedia to Enhance Cluster Labeling

We now explore the effectiveness of using candidate labels extracted from Wikipedia in addition to important terms extracted directly from the cluster data. We also compare four different feature selection methods for identifying important terms: the JSD method described in Section 3, the terms with highest *ctf-cdf-idf* values in the cluster centroid, and two standard feature selection methods, namely the *mutual information* and $\chi^2$ methods (see [13] pages 263-267). For this purpose, we evaluated the system's proposed labels with and without the candidate labels extracted from Wikipedia. In this experiment, we fixed the system's parameters to be 1) 20 important terms for querying Wikipedia, 2) 100 Wikipedia results for candidate extraction, and 3) the `SP(rank)` judge for candidate evaluation.



(a) 20NG

(b) ODP

**Figure 2: The effectiveness of Wikipedia labels enhancement for (a) 20NG data and (b) ODP data. Wikipedia labels provide an overall best performance for cluster labeling.**

Figure 2 reports on the Match@K scores of each method for increasing values of $k$. As can be observed, using the important terms extracted by the JSD method is much more effective than the highest weighted terms baseline (up to 100% improvement for the 20NG data and 60% for the ODP data). It further performs well compared to the two other standard methods, where for the ODP data it even completely dominates these methods. We can further observe that enhancing the JSD important terms with Wikipedia's labels provides the overall best performance, with Match@5>0.85 and up to 39% and 21% improvement over the JSD method for the 20NG and ODP data, respectively.

It is also interesting to note that feature selection methods on ODP data require at least 50 terms to cover 85% of the clusters with a correct label, while the same effectiveness is achieved by a list of 5 terms only using Wikipedia. A similar observation holds for the 20NG dataset – 17 terms are needed to cover 75% of the clusters while only 2 terms are needed using Wikipedia. Moreover, all the four feature selection methods fail to achieve the same performance of Wikipedia's top-5 terms, even when considering the top-50 labels.

To conclude, these results clearly show that combining labels extracted from Wikipedia together with important terms extracted from the text is a very effective cluster labeling approach.

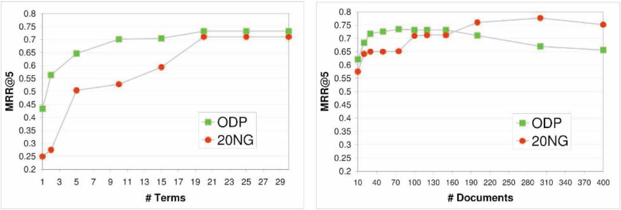
## 4.4 Candidate Labels Extraction

We identify two significant parameters that can affect the quality of Wikipedia's labels. The first parameter is the number of important terms that are used to query Wikipedia. The second is the number of top scored results from which candidate labels are extracted.

The number of the query's important terms may affect the quality of documents returned as results. As we add more terms to the query, precision is expected to drop hence the higher the chance to have irrelevant documents to the cluster's topic, therefore low quality candidates can be expected. The number of top results to consider affects the total number of unique candidates that are extracted and evaluated. Analyzing too few results implies that good candidates will be omitted. Analyzing too many results corresponds to low quality candidates.

We now analyze the effect of these two parameters on the performance of the system using both data collections. We used the `SP`(rank) judge and measured the MRR score

[5] A candidate is considered WordNet synonym of a label if both appear in the same WordNet's synset.



(a) # terms (b) # documents

**Figure 3: System's performance with respect to the number of query terms and the number of top scored documents that are considered for the candidate labels extraction.**

for different parameter values on both collections. Figure 3 shows the results of this analysis. Figure 3(a) illustrates the effect of the number of query terms. We observe that, for both data collections, with up to 20 terms, the system keeps gaining in performance and then, there is a plateau. We attribute this plateau to our choice of the query's terms boosting scheme, which boosts terms according to their JSD score. The results show that adding very low boosted terms to the query is not detrimental.

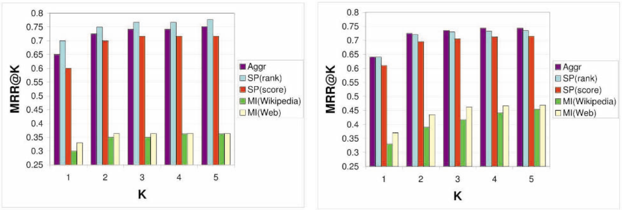
Figure 3(b) illustrates the effect of the number of results that are considered. We observe that with up to 300 results for the 20NG data and with up to 75 results for the ODP data the system keeps gaining in performance, probably since more good candidates are discovered. For more documents, the system's performance degrades. This is probably due to the fact that low scored results are irrelevant to the cluster's topic. Hence, candidate labels extracted from those results are irrelevant and introduce noise into the system's decision making scheme.

## 4.5 Evaluating Judge Effectiveness

We next compare the performance of the different judges proposed in Section 3.5 for candidate evaluation. The several judges are evaluated according their ability to identify the correct labels, or more precisely, to rank the correct labels on top of the label list. For this evaluation, we separately apply each judge in the candidate evaluation process and measured the MRR score achieved by the system. We further apply all judges together, aggregating their scores while assigning each judge $J_i$ a weight $\beta_i$ that is relative to its performance when it is was evaluated separately. We report on the MRR@5 score of each of the judges and the aggregated score of all judges for both data collections. Figure 4 shows the results of this comparison.

We first observe that for all judges, as $k$ increases (i.e., more cluster labels are proposed) the MRR score increases. Overall, among the four different judges, the SP(rank) judge performs the best. Among the two instantiations of the MI judge, the one using the web corpus (denoted MI(Web)) outperforms the one using the Wikipedia corpus (denoted MI(Wikipedia)), with up to 10% better MRR score for both datasets. This may be attributed to the fact that the web corpus is much larger then the Wikipedia corpus hence statistics may be more accurate.

The two instantiations of the SP judge completely dominate the two instantiations of the MI judge (up to almost 70% better MRR score). Finally, we observe that using the aggregated score of all judges slightly improves the system's performance for the ODP data.



(a) 20NG (b) ODP

**Figure 4: The system's MRR score for (a) 20NG data and (b) ODP data, using every judge separately and aggregating the scores of all judges.**

## 4.6 The Effect of Clusters' Coherency on Label Quality

A cluster of documents given to the labeling component is usually the corresponding result of the clustering algorithm used by the system. Given a collection of documents clustered by a specific clustering algorithm, the clusters' quality expresses how documents within any cluster are similar, and how dissimilar the pairs of clusters are. We now explore the effect of the cluster's coherency on the labeling process.

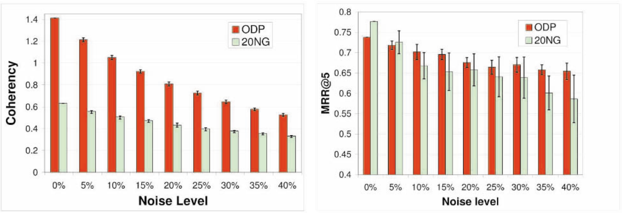
Given a set of clusters $\mathcal{C} = \{C_1, C_2, \ldots, C_n\}$ we measure the clusters' coherency as follows:

$$coherency(\mathcal{C}) = \frac{\sum_{i=1}^{n} \frac{|C_i|}{|\mathcal{D}|} sim_{in}(C_i)}{sim_{out}(\mathcal{C})} \quad (6)$$

where $sim_{in}(C_i)$ is the cluster's $C_i$ inner similarity, which is defined as the (weighted) average (cosine) similarity between cluster $C_i$'s documents and its centroid. The outer similarity $sim_{out}(\mathcal{C})$ is a normalization factor that captures the similarity between each pair of clusters, and is defined as the average (cosine) pairwise similarity between $\mathcal{C}$'s cluster centroids. Therefore, the higher the inner similarity and the lower the outer similarity, the more coherent the clusters are.

To simulate noisy clusters in our framework, we introduced different noise levels into the original clusters, which resulted in different levels of clusters' coherency. To produce noisy clusters, we adapted the method of [21]. For each cluster $C_i \in \mathcal{C}$ and a noise level $p \in [0, 1]$, each document $d \in C_i$ is swapped with a random document $d'$ from another random cluster $C_{j \neq i} \in \mathcal{C}$ with probability $p$ [21].

We now report on the effect of cluster coherency on the cluster labels proposed by the system using the two datasets. We introduced different levels of noise (up to noise level $p = 0.4$) into the each dataset's original clusters. For each noise level $p$, we repeated the experiment 10 times and measured the average MRR score of the SP(rank) judge with and without Wikipedia labels. Figure 5(a) shows the average cluster coherency level as a function of the noise level we introduced to the two datasets. We observe that, introducing more noise will result in less coherent clusters. Figure 5(b) shows the average MRR score per noise level measured for the two datasets. We observe that as expected, introducing more noise will result in less coherent clusters and therefore, the MRR score drops. Nevertheless, the drop in MRR score



(a) (b)

**Figure 5: (a) The effect of noise level on the clusters' coherency. (b) The effect of noise level on the system's performance. A higher noise level implies less coherent clusters with a *moderate* drop in the MRR score.**

per noise level is quite moderate for both datasets which implies that the proposed system is robust and has good resiliency to noise.

## 5. SUMMARY

In this work we investigated how cluster labeling can be enhanced by utilizing the Wikipedia knowledge-base. We described a general framework for cluster labeling that extracts candidate labels from the text and from Wikipedia and then retrieves the top scored candidates according to the evaluation of several independent judges. Our experimental results reveal that meta-data associated with Wikipedia articles, which are similar to the cluster's textual content, can provide very good labels for clusters of textual documents.

Our candidate extraction approach is based on identifying Wikipedia articles that are similar to the cluster's content and then extracting titles and categories from those pages. This process can be enhanced in several ways. First, other types of meta-data from Wikipedia pages can be considered for labeling. For example, anchors of Wikipedia pages (the fragments of text associated with the page's in-links and out-links) might provide good labels since an anchor-text of a hyper-link to a Wikipedia page often successfully describes the topic of that page. Another enhancement direction is to consider the hierarchical structure of Wikipedia categories. By analyzing the categories graph, candidates can be evaluated according to their specificity versus generality, i.e their relation with their ancestors and descendants in the hierarchy.

Cluster labeling with Wikipedia is extremely successful, as shown by our results, especially in collections of documents whose topics are covered well by Wikipedia concepts. For domain specific collections, with topics that are not completely covered by Wikipedia, the proposed candidates may hurt the system's performance due to their irrelevance to the documents' topics. For such collections, an intelligent decision should be made regarding the use of Wikipedia or another external resource; alternatively, a choice could be made to focus only on inner terms for labeling. The decision should be made by analyzing the given collection with respect to Wikipedia. Developing such a collection specific decision making as part of the labeling framework is left for further research.

## Acknowledgment

This work was done as a project for the IBM Classification Module (ICM). We would like to thank our colleagues: Michal Rosen-Zvi, Yaara Goldschmidt, and Josemina Magdalen, for useful discussions and feedback.

## 6. REFERENCES

[1] 20 News Group (20NG) data. http://people.csail.mit.edu/jrennie/20newsgroups.

[2] T. Brants and A. Franz. Web 1T 5-gram Version 1. 2006.

[3] D. Carmel, E. Yom-Tov, A. Darlow, and D. Pelleg. What makes a query difficult? In *SIGIR '06*, pages 390–397. ACM Press, 2006.

[4] O. S. Chin, N. Kulathuramaiyer, and A. W. Yeo. Automatic discovery of concepts from text. In *WI '06*, pages 1046–1049, Washington, DC, USA, 2006. IEEE Computer Society.

[5] R. Cilibrasi and P. M. B. Vitányi. The google similarity distance. *IEEE Transactions on Knowledge and Data Engineering*, 19(3):370–383, 2007.

[6] D. R. Cutting, D. R. Karger, J. O. Pedersen, and J. W. Tukey. Scatter/gather: a cluster-based approach to browsing large document collections. In *SIGIR '92*, pages 318–329, New York, NY, USA, 1992. ACM.

[7] W. de Winter and M. de Rijke. Identifying facets in query-biased sets of blog posts. In *ICWSM'07*, pages 251–254, 2007.

[8] E. Gabrilovich and S. Markovitch. Overcoming the brittleness bottleneck using wikipedia: Enhancing text categorization with encyclopedic knowledge. In *AAAI '06*, pages 1301–1306, Boston, MA, 2006.

[9] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI '07*, pages 1606–1611, Hyderabad, India, 2007.

[10] F. Geraci, M. Maggini, M. Pellegrini, and F. Sebastiani. Cluster generation and cluster labelling for web snippets:a fast and accurate hierarchical solution. *Internet Mathematics*, 2007.

[11] E. Glover, D. M. Pennock, S. Lawrence, and R. Krovetz. Inferring hierarchical descriptions. In *CIKM '02*, pages 507–514, New York, NY, USA, 2002. ACM.

[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 *SIGIR '08*, pages 179–186, New York, NY, USA, 2008. ACM.

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

[14] Open Directory Project (ODP). http://www.dmoz.org/.

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

[16] D. R. Radev, H. Jing, M. Styś, and D. Tam. Centroid-based summarization of multiple documents. *Information Processing Management*, 40(6):919–938, 2004.

[17] P. Schönhofen. Identifying document topics using the wikipedia category network. In *WI '06*, pages 456–462, 2006.

[18] M. Strube and S. P. Ponzetto. Wikirelate! computing semantic relatedness using wikipedia. July 2006.

[19] Z. S. Syed, T. Finin, and A. Joshi. Wikipedia as an ontology for describing documents. In *ICWSM '08*, 2008.

[20] H. Toda and R. Kataoka. A clustering method for news articles retrieval system. In *WWW '05*, pages 988–989, New York, NY, USA, 2005. ACM.

[21] P. Treeratpituk and J. Callan. Automatically labeling hierarchical clusters. In *DG.O '06*, pages 167–176, New York, NY, USA, 2006. ACM.