Improving the presentation of search results by multipartite graph clustering of multiple reformulated queries and a novel document representation*

N. Lytkin S. Streltsov L. Perlovsky I. Muchnik S. Petrov
LongShortWay, Inc.
muchnik@longshortway.com

Abstract

The goal of clustering web search results is to reveal the semantics of the retrieved documents. The main challenge is to make clustering partition relevant to a user's query. In this paper, we describe a method of clustering search results using a similarity measure between documents retrieved by multiple reformulated queries. The method produces clusters of documents that are most relevant to the original query and, at the same time, represent a more diverse set of semantically related queries. In order to cluster thousands of documents in real time, we designed a novel multipartite graph clustering algorithm that has low polynomial complexity and no manually adjusted hyper–parameters.

The loss of semantics resulting from the stem–based document representation is a common problem in information retrieval. To address this problem, we propose an alternative novel document representation, under which words are represented by their synonymy groups.

1 Introduction

Yahoo!'s [18] response to query "clustering search engine" demonstrates an exponential growth in the number of publications and tools related to text clustering [1, 2, 3, 4, 5, 7]. Documents, their headers, queries and emails, are just a few types of data clustered routinely [1, 2, 3, 4, 7]. Clustering is also used to compare and combine documents retrieved by multiple search engines in response to a single query [14].

There are two main reasons for high interest in clustering in information retrieval, especially related to the web:

1. Clustering provides direct means of speeding up the search
2. Clustering methods are simple to implement.

*This work was supported by Yandex grant 110104.

A separate branch of clustering methods in information retrieval is dedicated to the identification of text semantics. These methods are intended to improve the presentation of search results by displaying to a user the most relevant documents first, and to reveal the *semantic diversity* of the documents by presenting, in a single response, documents covering different aspects of the subject of interest.

Today's search engines order retrieved documents and present the documents' headers in a paged arrangement to the user. It is usually assumed that the order of presentation reflects documents' relevance to the query terms — more relevant documents receive smaller numerical ranks. Headers of the most relevant documents are presented on the first page.

Often, choosing the proper keywords to express the search intention is a challenging task. The user is often unaware of the complete semantic scope induced by the query terms. Pure relevance–based order of presentation of search responses however, completely ignores their semantic diversity. The first page of search results fails to convey to the user the entire semantic variety of the retrieved documents.

Clusty [14] was a pioneer system aimed at providing a compact representation of the diverse body of documents retrieved in response to a query. Several similar systems were proposed later [9, 11]. All these systems were designed based on the assumption that a query adequately reflects the user's intentions and the search scope. Under this assumption, the problem is reduced to a reordering of the search responses, such that the first documents presented to the user reflect the diversity of the search results rather than their relevance.

Our work is based on a different assumption. The user has in mind a large set of terms related to the topic of interest. While formulating a query, the term selection criteria may include, for example, simplicity of the query formulation rather than precision in the expression of the search intention. The resulting query, therefore, may not precisely define the search scope *initially intended* by the user.

We developed a method of clustering short document headers returned by a search engine (Yahoo!) in response to a query. Since document headers serve as the input data, we refer to them simply as documents. Our method has three novel features:

1. It simultaneously clusters documents retrieved by the original query and documents retrieved by other queries belonging to the *semantic neighborhood* of the original. The neighborhood queries broaden the semantic scope of the set of documents retrieved, which would most probably contain all items the user expected to see

2. Our clustering procedure considers similarities only between the documents retrieved by different queries all of which belong to the semantic neighborhood of the original query. Relationships between documents retrieved by the same query are not used explicitly by our method. Note that the neighborhood queries may not share any keywords with each other nor with the original query

3. Alternatively to the standard bag–of–words and, also common, bag–of–stems document representations, our method uses the synonymy between words in retrieved documents for their representation. The resulting representation remains understandable by a human, and im-

proves the method's *sensitivity*, which is especially important when dealing with short documents such as document headers.

Words synonymy information is obtained from a dictionary. Words present in the documents, but absent in the dictionary are discarded prior to clustering (see Table 1 for words retention rates). As the number of dictionary terms matching document words *increases*, the loss of information in our document representation *decreases* since fewer words are discarded from the analysis. Thus, the document representation becomes more *complete*.

The loss of information could also be reduced by introducing a more flexible dictionary matching mechanism. The flexibility of matching has to be controlled in order to discourage the matching of words with very different meanings.

We propose a method for addressing the challenge of maintaining our dictionary–based document representation while minimizing the information loss due to dictionary restrictiveness. Based on Dynamic Logic [8], we describe a probabilistic extension to the dictionary matching mechanism. In essence, we propose to associate each dictionary group of terms (e.g. a group of synonymous terms) with a probabilistic source (or *generator*) characterized by a vector of parameters. The parameters are estimated based on the terms in the dictionary group. The role of the generator is to determine the likelihood of affiliation of an arbitrary input string with the dictionary group. It is important that the likelihood scores are high for those input strings which, from a human perspective, are relevant to the group.

The paper is organized as follows. In Section 2 we describe an example of the relationship between an original query and its semantic neighborhood. The example also demonstrates a potential ability of a user to produce such semantic neighborhood. This neighborhood is subjective, because it reflects the user's *semantic expectations*. All queries in the example are constructed based on the term "influenza" and focus on its different aspects. The original query concerns socio–economic aspects of influenza epidemiology.

Sections 3 and 4 describe our document representation used for clustering. Section 5 presents our novel multipartite graph clustering algorithm. Also in Section 5, using cluster–query relationships combined with Yahoo!'s rankings of the retrieved documents, we define the notion of representative documents characterizing queries and clusters.

The results of our experiments are described in Section 6. Also discussed in that section are the advantages of the proposed clustering method using semantic neighborhoods over the standard approach of analyzing documents retrieved based on the original query only. We conclude our analysis in Section 7 and propose a probabilistic extension of the dictionary matching method based on Dynamic Logic in Section 8.

2 Semantic Neighborhood

In this section we consider one query with a manually generated set of semantic neighbors — queries that reflect other aspects of the domain specified by the original query (query (17) highlighted in the example below).

The work presented in this paper is focused on document clustering, assuming that the query's

semantic neighborhood is given. Automatic generation of semantic neighborhoods is a subject of future work discussed in Section 7.

Example. We consider a user requesting information on the following aspects of *influenza*:

1. Medical aspects of the disease
2. Epidemiological aspects of the disease
3. Information sources related to molecular level behavior
4. Fundamental mechanisms of the decease behavior, for instance its molecular level mechanisms
5. Specific information on the disease behavior in populations of patients with cancer and heart diseases.

To satisfy these requirements, the user creates 5 query groups (one group per aspect) containing the total of 21 queries:

1. (1) Influenza prevention treatment symptoms
   (2) Influenza vaccination immune system
   (6) Infection vaccine immune system
2. (4) Infection prevention sex ethnicity
   (5) Flu genetics mortality rate
   (8) Infection network mortality rate
   (11) Children vaccination Infection prevention
   (12) Influenza ethnicity marital status
   (14) Flu socioeconomic survival time
   (17) Insurance doctors flu economics
   (20) Flu pandemic epidemic modeling
3. (9) Flu database protein interaction
   (10) Flu database genetic evolution
   (21) Flu virus genome database
   (18) Infection virus evolution database
4. (5) Flu genetics mortality rate
   (7) Influenza virus evolution adaptation
   (19) Comparative genomics infection diseases
5. (3) Lung cancer influenza vaccination
   (13) Flu heart cancer mortality
   (15) Influenza mortality blood pressure
   (16) Dental health flu resistance

Within the framework of this paper, we consider a query as a bag–of–words interpreted as a conjunction of all its terms. We ignore additional query syntax commonly supported by search engines, and consider the following three methods of query modification:

1. Query reduction
2. Synonym substitution
3. Usage of dictionary–based semantic relations.

The first and second modifications are actually used by the majority of search engines. Query reduction is mainly used for association of additional (often commercial) information. Synonym substitution is used in search, but usually in a very limited and doubtless way — allowing a search for "notebook" together with a search for "laptop" but not for "motherland" together with "home country." Synonym substitution is subject to combinatorial explosion, which makes its unrestricted application impractical.

3 Document Representation and Data Preprocessing

Queries are sent one by one to Yahoo! [18] search engine. For each query, the first 200 documents headers are retrieved and parsed removing HTML, URLs, blank lines, etc. The resulting file consists of 200 lines of text, one document header per line. Each line is prefixed by a document identification number. All preprocessing was implemented in PERL.

During the next stage of preprocessing, we use a dictionary — a precompiled list of WordNet nouns grouped by synonymy. A short description of WordNet, dictionary compilation, and synonymy group construction, related statistics and examples are given in Section 4.

All substrings longer than three characters of each word in every document header are matched against the dictionary. The first left longest match is chosen to be the word's representative in the dictionary. Thus, the word becomes associated with a particular synonymy group. One term from every synonymy group is declared as its *main term* (see Section 4).

Every word in a document is replaced by the main term of its synonymy group. Words that produced no matches are removed from the document. Currently, we use WordNet nouns only. Thus, a word that is not a noun will be considered only if it lexically coincides with a noun (e.g., "(a) group", "(to) group"). In our experiments, the number of terms retained in the resulting document representation was usually half of the number of words in the original document. Statistics for six arbitrary queries are presented in Table 1. The last column contains the number of distinct synonym groups present in all 200 documents retrieved in response to the corresponding query.

Table 2 illustrates that currently, nouns constitute 75% of WordNet terms. It is possible that the share of verbs in WordNet will grow to their statistical share in English.

Our system is capable of handling parts of speech other than the noun without requiring any significant modifications. We plan to incorporate other parts of speech into our analysis, but we do not expect drastic changes in the result.

The result of preprocessing one query is the representation of 200 documents containing together about 8–9 thousand words by a sparse Boolean matrix with 200 rows and approximately 1000 columns. Rows of the matrix correspond to documents, columns — to indices of the synonymy groups' main terms.

Words in original documents	Words found in dictionary	Distinct dictionary groups
8807	4480	750
8738	4482	844
8321	4789	920
8859	4238	777
8706	4380	1141
8446	4269	1015

Table 1: Statistics for six arbitrary queries.

POS	Unique strings	Synsets	Total Word–Sense Pairs
Noun	114648	79689	141690
Verb	11306	13508	24632
Adjective	21436	18563	31015
Adverb	4669	3664	5808
Totals	152059	115424	203145

Table 2: WordNet statistics for March 2005 release.

After preprocessing all 21 queries, we are left with a 4200–row sparse Boolean matrix representing the retrieved documents in a unified space of the synonymy groups. This matrix is used to compute a 4200×4200 *affinity* matrix serving as input to our multipartite clustering procedure. The similarity measure used and the clustering algorithm are discussed in Section 5.

4 WordNet, Synonymy Groups, and Dictionary Compilation

WordNet [15, 16] is a lexical reference system whose design was inspired by the current psycholinguistic theories of human lexical memory. English nouns, verbs, adjectives and adverbs are organized into synonym sets (synsets), each representing a single lexical concept. Various relations link the synonym sets. WordNet was developed by the Cognitive Science Laboratory at Princeton University, and can be used online or locally, free of charge. Installation packages exist for Windows and Unix/Solaris/Linux platforms. There is also a Prolog version. WordNet statistics for March 2005 release are in Table 2.

We used only nouns in our clustering experiment. WordNet's set of nouns can be considered as a synonymy graph containing one vertex for each noun. An edge between a pair of vertices signifies the synonymy relation reflected in WordNet between the two corresponding nouns.

The synonymy graph induced by the set of WordNet's nouns consists of 65219 connected components — *synonymy groups*. Each component contains between 1 and 36 vertices. The distribution of the number of connected components is presented in Table 3 and Figure 1.

Starting with a single dictionary term, connected components are discovered by recursively

Power	1	2	3	4	5	6	7	8	9	10	11	12	13
Number of groups	32440	19798	6167	2539	1044	499	273	135	85	48	42	18	9

Power	14	15	16	17	18	19	20	22	26	33	36
Number of groups	10	11	6	3	4	1	2	1	1	1	1

Table 3: The distribution of the number of synonymy groups over the number of terms in each group.

adding to the current set of terms synonyms of all its members. The discovery of all connected components induced by 114648 WordNet nouns took approximately 10 hours on a computer equipped with a 1.5GHz Pentium processor and 512MB of RAM.

Lexicographically shortest word in a synonymy group was selected as its main term (in examples below, it is the first word in the group, and it is highlighted).

As was expected, transitive closure of the synonymy relationship sometimes placed terms with quite different meanings into the same group (connected component). Below are a few examples of the synonymy groups:

1. The unique group of 36 terms: ***peck***, *mint, raft, slew, heap, pile, mass, flock, sheaf, spate, sight, spile, batch, stack, plenty, bundle, piling, muckle, mickle, hatful, tidy_sum, big_bucks, megabucks, whole_lot, good_deal, plenitude, big_money, whole_slew, great_deal, mint_candy, plentitude, visual_sense, plenteousness, plentifulness, quite_a_little, visual_modality.*

2. 15 variations of F.M. Dostoevsky name spelling: ***Dostoevsky***, *Dostoevski, Dostoyevsky, Fyodor_Dostoevski, Feodor_Dostoevski, Feodor_Dostoevsky, Fyodor_Dostoevsky, Fyodor_Dostoyevsky, Feodor_Dostoyevsky, Feodor_Mikhailovich_Dostoevski, Fyodor_Mikhailovich_Dostoevski, Fyodor_Mikhailovich_Dostoevsky, Feodor_Mikhailovich_Dostoevsky, Feodor_Mikhailovich_Dostoyevsky, Fyodor_Mikhailovich_Dostoyevsky.*

3. 15 variations of a popular concept: ***crap***, *dump, turd, dirt, poop, shit, grime, shite, grunge, dumpsite, wasteyard, waste-yard, trash_dump, garbage_dump, rubbish_dump.*

4. A group with a slight drift of meaning: ***keep***, *donjon, upkeep, backing, dungeon, support, funding, livelihood, sustenance, supporting, sustainment, sustentation, bread_and_butter, financial_backing, financial_support.*

Main terms serve as identifiers of the synonymy groups, and are used for document representation during clustering. Original form of the retrieved document headers is more suitable for manual (human) examination.

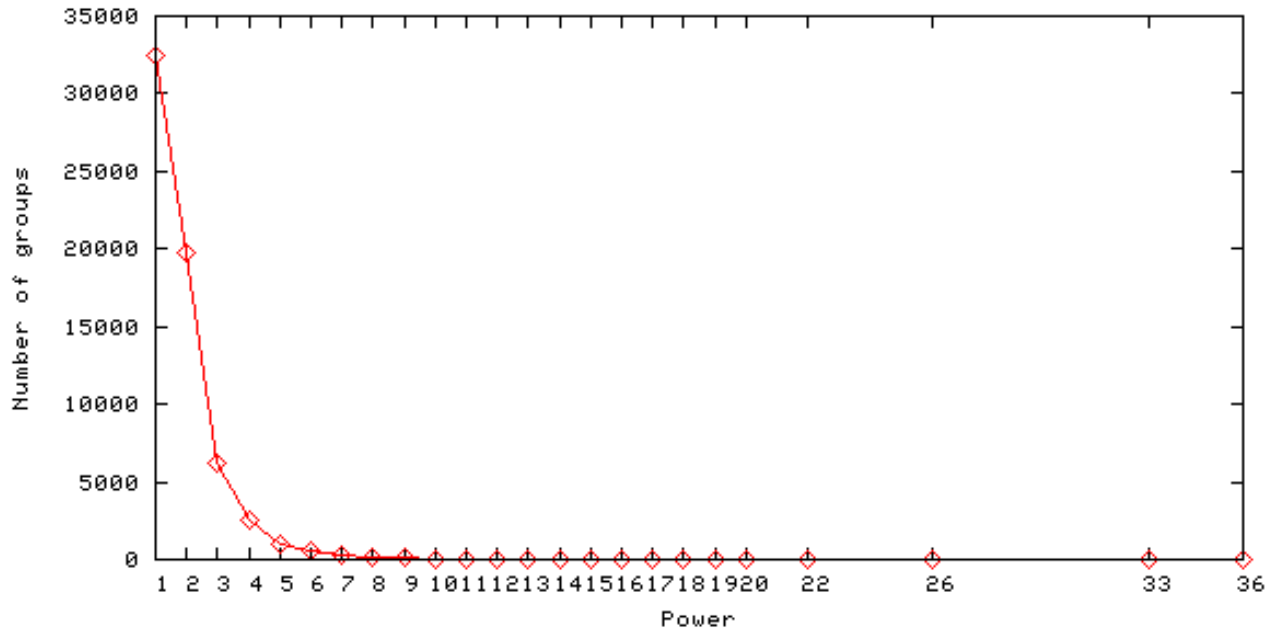

Figure 1: Plot of the data in Table 3. The distribution of the number of groups over their powers.

5 Multipartite Graph Clustering

Our multipartite graph clustering procedure uses a weighted multipartite graph as input and partitions the graph's vertex set into disjoint subsets (clusters). Each vertex corresponds to a retrieved document. A *partite set* is a set of vertices corresponding to the documents retrieved by a single query. If a document is retrieved by more than one query, it corresponds to several vertices — one in every appropriate partite set.

The vertex set V of a multipartite weighted graph $G = (V, E)$ is a union of pairwise disjoint partite sets

$$V = \bigcup_{s=1}^{r} V_s,\ V_s \cap V_t = \emptyset,\ s \neq t.$$

The weight assigned to an edge $(v, u) \in E,\ v \in V_s,\ u \in V_t,\ s \neq t$ denotes the degree of similarity between the two corresponding documents. All edges connecting vertices of the same partite set $(v, u) \in E,\ v \in V_s,\ u \in V_s,\ s = 1, 2, \ldots, r$ have zero weight.

5.1 Similarity Measure

Let L be the set of all synonymy groups, introduced in Section 3, whose terms are present in the retrieved documents. A document (document header) is represented by an $|L|$–dimensional Boolean vector with components corresponding to the synonymy groups. Non–zero components indicate the presence of the synonymy groups' term(s) in the document.

We define a similarity measure $\alpha(\mathbf{x}, \mathbf{y})$ for a pair of documents represented by Boolean vectors $\mathbf{x} = (x_1, x_2, \ldots, x_{|L|})$ and $\mathbf{y} = (y_1, y_2, \ldots, y_{|L|})$ as

$$\alpha(\mathbf{x}, \mathbf{y}) = \frac{\sum_{i=1}^{n} x_i y_i}{max(\sum_{i=1}^{n} x_i, \sum_{i=1}^{n} y_i)}, \; x_i, y_i \in \{0, 1\}. \tag{1}$$

5.2 Clustering Procedure

Our clustering procedure is based on an optimization of a quasiconcave function

$$F(H) = \min_{x \in H} \pi(\mathbf{x}, H), \; H \subseteq V, \tag{2}$$

where set H contains vertices from at least two partite sets of the complete vertex set V. The *linkage function* $\pi(\mathbf{x}, H)$ is the sum of similarities $\alpha(\mathbf{x}, \mathbf{y})$ between document $\mathbf{x} \in V_s \subset H$ and every other document $\mathbf{y} \in (H - V_s)$ that belongs to a partite set different from V_s

$$\pi(\mathbf{x}, H) = \sum_{y \in V_t \subset H} \alpha(\mathbf{x}, \mathbf{y}), \; \mathbf{x} \in V_s \subset H, \; s \neq t. \tag{3}$$

Evidently, the more similar $\mathbf{x}$ is to all elements of $(H - V_s)$, the larger $\pi(\mathbf{x}, H)$ is.

The value of the function $F(H)$ is determined by the π–value of a least connected element (an *outlier*) of H, and can be interpreted as a measure of *compactness* of set H. We define a *cluster* as a subset $H^* \subseteq V$, such that

$$H^* = \arg\max_{H \subseteq V} F(H). \tag{4}$$

Set H^* is a *maximizer* of $F(H)$ and can be found in polynomial time by an algorithm published in [13] (see also [10] for a discussion of combinatorial optimization in clustering). Low computational complexity (less than $O(|V|^2)$) and two other properties of the algorithm follow from the fact that $\pi(\mathbf{x}, H)$ is a monotonically increasing function of H:

1. $F(H)$ is a quasiconcave function

$$F(H_i \cup H_j) \geq \min\left(F(H_i), F(H_j)\right), \; \forall H_i, H_j \subseteq V,$$

   iff $\pi(\mathbf{x}, H)$ is monotone

2. Our efficient procedure finds the largest maximizer that includes all maximizers of $F(H)$.

5.2.1 Outline of The Procedure

The procedure is an iterative *shelling process* that finds sequential maximums of the quasiconcave function $F(H)$ defined by Formula 2. On every iteration, a new cluster H_i^* — a subset of the current vertex set H^i — is found, and the algorithm proceeds to the next iteration with the new

set of vertices $H^{i+1} = H^i - H_i^*$. Table 4 shows pseudocode for the cluster discovery algorithm used on every iteration.

Our procedure terminates when either $F(H^i) = 0$, or the current set of vertices H^i is empty. In the latter case, set H_{i-1}^* is the last nonempty cluster. In a case when $F(H^i) = 0$, if H^i is not empty, all its elements are considered as clusters–singletons.

In contrast with the majority of clustering techniques [6, 17] where the number of clusters is a parameter predefined by an expert, the number of clusters discovered by our method is determined simultaneously with clusters and without manual interference.

5.3 Cluster Characterization by Representatives

Multipartite graph clustering results in the following representation of the initial vertex set V

$$V = \bigcup_{s=1}^{r}\bigcup_{t=1}^{m} H_s^t,\ H_s^t = V_s \cap H_t^*,$$

where r is the number of initial partite sets, m is the number of clusters discovered by the algorithm, V_s is the document header set (the partite set) retrieved by query Q_s, and $H_s^t \subseteq V_s$ is the set of document headers that belong to cluster H_t^*. Of course, some sets H_s^t may be empty.

We take into consideration that all documents in each partite set V_s are ranked by Yahoo! search engine. The ranking is based on Yahoo!'s internal measure of relevance of a document to the query. More relevant documents receive smaller ranks.

Documents' relevance ranks combined with clusters define a rich structure on the set of retrieved document headers, and allow numerous content presentation schemes to be employed. This structure makes it possible to vary semantic diversity versus relevance when presenting search results to the user. Below, we give three definitions of representatives that will be used for cluster characterization during discussion of the results in Section 6.

A document header $\mathbf{x}_s^t \in V_s,\ \mathbf{x}_s^t \in H_t^*$ is *the representative of a query Q_s in a cluster H_t^**, if the rank of $\mathbf{x}_s^t$ is the smallest of the relevance ranks of all other documents retrieved by query Q_s and assigned to cluster H_t^*

$$\mathbf{x}_s^t = \arg\min_{\mathbf{x}\in H_s^t} \text{rank}(\mathbf{x}),\ H_s^t = V_s \cap H_t^*. \tag{5}$$

Consider a set of queries $\{Q_1, Q_2, \ldots, Q_m\}$ all of which have representatives in a cluster H_t^*. Query representatives $\{\mathbf{x}_1^t, \mathbf{x}_2^t, \ldots, \mathbf{x}_m^t\}$ determined by Formula 5 form *per query representation of the cluster H_t^**.

All documents with the smallest rank in a cluster H_t^* constitute the set of *cluster representatives*

$$\{\mathbf{x}_s^t\} = \arg\min_{\mathbf{x}\in H_t^*} \text{rank}(\mathbf{x}). \tag{6}$$

In a case when this set contains more than one document, we refrain from selecting a single document as the cluster representative due to two factors:

1. A pair of documents each from a different partite set are considered equally relevant to the two corresponding queries if the documents' ranks are equal

2. In our experiments, all partite document sets were of equal size $|V_s| = |V_t|,\ V_s \subset V,\ V_t \subset V$.

function Cluster(H^i)

input: Current vertex set H^i

output: Cluster H_i^* or a set of clusters–singletons

1 **if** all vertices in H^i are from the same partite set ($\forall v \in H^i,\ v \in V_s$)

Declare every vertex $v \in H^i$ as a cluster–singleton

2 **return** H^i as a set of clusters–singletons

3 **else** {

4 $j := 0$

5 $H_j := H^i$

6 $S := \emptyset$

7 **do** {

Find a vertex x_{H_j} that determines the value of $F(H_j) = \min\limits_{x \in H_j} \pi(x, H_j)$:

8 $x_{H_j} := \arg\min\limits_{x \in H_j} \pi(x, H_j)$

Add x_{H_j} into the ordered set $S = \{x_{H_0}, x_{H_1}, \ldots, x_{H_{j-1}}\}$:

9 $S := S \cup \{x_{H_j}\}$

10 $H_{j+1} := H_j - \{x_{H_j}\}$

11 $j := j + 1$

12 } **until** $F(H_j) \neq 0\ \wedge\ H_j \neq \emptyset$

13 $n := j - 1$

14 **if** $F(H_j) = 0\ \wedge\ H_j \neq \emptyset$ {

Add all elements of H_j in any order to set S:

15 $S := S \cup H_j$

16 $n := j$

17 }

18 $F_{max} := max(F(H_0), F(H_1), \ldots, F(H_n))$

Find the smallest index k, such that $x_{H_k} \in S$ and $F(H_k) = F_{max}$:

19 $k := \arg\min\limits_{l=1,2,\ldots,|S|} \{x_{H_l} : x_{H_l} \in S,\ F(H_l) = F_{max}\}$

Place all elements of S starting from x_{H_k} into cluster H_i^*:

20 $H_i^* := \{x_{H_l} :\ k \leq l \leq |S|,\ x_{H_l} \in S\}$

21 **return** cluster H_i^*

22 }

Table 4: The pseudocode for computing the cluster H_i^* on the i–th iteration of our multipartite graph clustering procedure. If all elements of the input vertex set H^i belong to the same partite set, then each vertex $v \in H^i$ is declared as a cluster–singleton. Consequently, H^i is returned as a set of clusters–singletons. Otherwise, a new cluster H_i^* is computed.

6 Results

Using the original query and its variations described in Section 2, we retrieved and clustered 4200 document headers. In Section 5.3, we described how to represent the clusters by a small number of documents. Below, we present three ways of characterizing queries while varying semantic diversity versus relevance of the search response:

1. The set of all clusters' representatives serves as the most diversified representation of the original query and all its semantic neighbors. At this level of diversification all queries are represented by the same set of documents.

2. Documents retrieved by a query and selected as representatives of some cluster form *per cluster representation of the query*. For example, row (*17) in Table 8 demonstrates that per cluster representation of the original query (17) is formed by documents $\{1, 37, 81, 148, 159, 200\}$ whose ranks are highlighted.

3. A query representation emphasizing relevance is achieved by the set of documents each of which is the representative of the query in a cluster (see Formula 5). In this case, the original query (17) is represented by documents $\{1, 2, 15, 37, 58, 81, 89, 133, 148, 159, 200\}$, as shown in Table 8.

6.1 Clusters

Our procedure split 4200 document headers retrieved by 21 queries into 20 clusters, 9 of which were singletons. The distribution of the number of documents over clusters is given in Table 5. The first row and column contain cluster and query identification numbers, respectively. The last row indicates the number of documents in each cluster. The last column shows the number of clusters containing the documents retrieved by the corresponding query. Every other cell (s, t) represents the number of documents retrieved by query Q_s, and placed into cluster H_t^*. Results for the original query are in the highlighted row (*17).

Table 5 shows 2 large clusters. Cluster 1 contains documents retrieved in response to all 21 queries. For 12 queries, more than 190 of 200 documents belong to cluster 1. This indicates that all our queries are similar in the sense that they retrieve similar documents.

Let us consider the two large clusters (clusters 1 and 2). To interpret cluster 1 we joined all terms from 13 queries, $\{1, 2, 3, 4, 6, 7, 8, 11, 12, 13, 15, 18, 19\}$: *adaptation, blood, cancer, children, comparative, database, diseases, ethnicity, evolution, flu, genomics, heart, immune, infection, influenza, lung, marital, mortality, network, pressure, prevention, rate, sex, status, symptoms, system, treatment, vaccination, vaccine, virus*.

For each of these 13 queries, at least 87% of documents retrieved belong to cluster 1. The above list of terms covers all queries in groups 1, 4, and 5 (with the exception of query 16 in group 5) described in Section 2. These groups of queries cover all fundamental medical and biological questions concerning influenza.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
1	198	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3
2	198	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3
3	192	4	3	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	4
4	194	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3
5	165	25	7	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	6
6	197	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	4
7	191	7	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	4
8	195	4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3
9	98	94	3	0	1	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	8
10	124	75	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	3
11	197	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	4
12	188	7	4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4
13	194	4	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3
14	103	83	11	1	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	5
15	198	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3
16	154	43	2	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	4
***17**	**70**	**109**	**10**	**0**	**0**	**2**	**1**	**1**	**0**	**0**	**3**	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**1**	**11**
18	197	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3
19	194	1	2	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	6
20	123	71	4	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	5
21	170	28	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	4
All	3540	565	57	6	2	3	2	2	4	2	1	1	1	1	1	1	1	1	1	1	

Table 5: Document distribution by clusters. Rows correspond to queries, columns — to clusters. The last row indicates the number of documents in each cluster. The last column shows the number of clusters containing the documents retrieved by the corresponding query. Results for the original query are in row (*17).

Cluster 2 is related mostly to queries from groups 2 and 3. In other words, this cluster contains documents related to the socio–economic aspects of influenza epidemics, its data, data analysis and support.

The data demonstrates that the original query (17) retrieved the most diverse set of documents distributed over 11 clusters. Most of the documents retrieved in response to query (17) are contained in cluster 2 (socio–economic aspects of influenza epidemic). However, a subset of the documents of size 70 makes a strong presence in cluster 1.

We conclude that almost all generated semantic neighbors are relevant to the original query (17), and reflect its various aspects. The generated queries retrieved additional documents helping to construct a more complete representation of the original query. Moreover, multipartite clustering allows for highly sensitive selection of additional documents semantically related to the original query. Indeed, our results demonstrate that query (17) has a specific semantic relation with the most diverse queries $\{5, 7, 9, 10, 14, 15, 20, 21\}$ (i.e. those queries that retrieved the most diverse sets of documents). The query with the next highest diversity after the original query is query (9). This query relates to database support for influenza analysis and is distributed over 8 clusters.

Cluster	4 most frequent main terms
1	grippe, infection, disease, virus
2	database, doctor, indemnity, protein
3	stance, doctor, economics, indemnity
4	following, query, department, force
5	using, Dante, bovine, charge
6	blog, Aire, Dana, November
7	prof, Aussie, Canberra, Princeton
8	event, Bari, Ribes, bite
9	analysis, literature, technique, Wilkins
10	vitamin, addendum, compare, diet
11	economics, indemnity, April, decrease
12	
13	Prep
14	bacteria, causing, cleaner, feel
15	pharma, seam
16	egis, page, search
17	George, admiralty, banking, contract
18	doctor, economics, indemnity, post
19	Canada, Canuck, Florida, Laws
20	Davis, ally, berg, bill

Table 6: Clusters characterized by four most frequent main terms. Singleton clusters $\{13, 15, 16\}$ each contained less than four main terms.

We characterized each cluster by four most frequent main terms (the synonymy groups representatives) appearing in its documents. The characterizing terms are listed in Table 6. These terms reflect the clusters' semantics, although sometimes ambiguously. Note that cluster 12 has no terms, because it contained a single document in French with no matching dictionary terms. Since our initial data consisted of short document headers, it is of no surprise that three singleton clusters $\{13, 15, 16\}$ each contained less than four main terms.

Table 7 relates queries, retrieved documents and clusters. Queries are presented by lists of keywords. For every set of document headers, we collected term frequency statistics for every main term appearing in the headers. The last column of each row contains a list of clusters with their four most frequent main terms. Each cluster number is followed by the number of documents that were retrieved by the corresponding query, and placed into the cluster.

Table 7: Characterization of queries, documents and clusters.

Query # and keywords	20 most frequent main terms in the retrieved document headers (with term frequency)	4 most frequent main terms from each cluster containing the query documents
1. Influenza, prevention, treatment, symptoms	grippe 553, handling 302, prevention 253, symptom 250, cause 57, data 55, disease 53, health 50, cold 48, virus 45, control 44, diagnosis 41, infection 41, vaccine 38, park 32, sign 24, complication 22, inhalator 22, bout 21, medicine 21	Cluster 1 (198): grippe, infection, disease, virus Cluster 2 (1): database, doctor, indemnity, protein Cluster 3 (1): stance, doctor, economics, indemnity
2. Influenza, vaccination, immune, system	grippe 376, immune 289, vaccination 276, system 273, vaccine 98, disease 59, response 51, virus 51, health 41, shot 35, infection 34, cause 29, babe 24, bout 23, data 23, pack 23, heir 21, body 20, HTML 19, immunisation 19	Cluster 1 (198): grippe, infection, disease, virus Cluster 2 (1): database, doctor, indemnity, protein Cluster 3 (1): stance, doctor, economics, indemnity
3. Lung cancer, influenza, vaccination	grippe 325, cancer 300, lung 259, vaccination 239, disease 140, health 82, vaccine 58, pump 43, data 33, patient 31, handling 30, asthma 29, diabetes 29, medical 27, virus 26, high 25, kidney 24, news 24, association 23, bout 22	Cluster 1 (192): grippe, infection, disease, virus Cluster 2 (4): database, doctor, indemnity, protein Cluster 3 (3): stance, doctor, economics, indemnity Cluster 13 (1): pharma, seam
4. Infection, prevention, sex, ethnicity	prevention 281, infection 227, ethnicity 139, AIDS 103, race 86, risk 62, disease 53, have 50, health 48, Amen 47, report 38, control 37, data 34, centre 32, state 31, HTML 28, program 28, spouse 27, rates 26, view 25	Cluster 1 (194): grippe, infection, disease, virus Cluster 2 (3): database, doctor, indemnity, protein Cluster 3 (3): stance, doctor, economics, indemnity

Continued on next page

Table 7 – continued from previous page

Query # and keywords	20 most frequent main terms in the retrieved document headers (with term frequency)	4 most frequent main terms from each cluster containing the query documents
5. Flu, genetics, mortality, rate	rate 209, mortality 203, genetics 160, health 78, disease 67, high 54, virus 38, vaccine 36, grippe 33, doll 32, babe 30, have 29, news 29, dying 27, query 26, reverse 22, world 22, cancer 21, infection 20, nidus 20	Cluster 1 (165): grippe, infection, disease, virus Cluster 2 (25): database, doctor, indemnity, protein Cluster 3 (7): stance, doctor, economics, indemnity Cluster 4 (1): following, query, department, force Cluster 9 (1): analysis, literature, technique, Wilkins Cluster 11 (1): economics, indemnity, April, decrease
6. Infection, vaccine, immune, system	vaccine 420, immune 398, system 375, infection 251, AIDS 54, virus 48, disease 44, body 41, event 38, response 36, cancer 35, fight 32, query 32, cell 29, babe 24, health 23, data 22, produce 22, researcher 21, homo 20	Cluster 1 (197): grippe, infection, disease, virus Cluster 2 (1): database, doctor, indemnity, protein Cluster 3 (1): stance, doctor, economics, indemnity Cluster 10 (1): vitamin, addendum, compare, diet
7. Influenza, virus, evolution, adaptation	virus 387, grippe 289, evolution 273, try-on 192, homo 64, HTML 39, host 39, view 36, gene 29, disease 27, infection 23, ring 23, species 23, swine 23, biology 22, change 22, world 22, genet 21, cell 19, nidus 19	Cluster 1 (191): grippe, infection, disease, virus Cluster 2 (7): database, doctor, indemnity, protein Cluster 12 (1): Cluster 13 (1): prep
8. Infection, network, mortality, rate	mortality 269, rate 258, infection 244, network 210, babe 81, health 54, high 54, data 40, AIDS 35, news 30, mate 27, dying 26, Natal 25, disease 25, cancer 23, hospital 22, cause 21, fear 21, report 20, patient 19	Cluster 1 (194): grippe, infection, disease, virus Cluster 2 (4): database, doctor, indemnity, protein Cluster 3 (1): stance, doctor, economics, indemnity

Continued on next page

Table 7 – continued from previous page

Query # and keywords	20 most frequent main terms in the retrieved document headers (with term frequency)	4 most frequent main terms from each cluster containing the query documents
9. Flu, database, protein, interaction	protein 240, database 190, interaction 165, data 44, health 37, search 35, virus 30, drug 27, vaccine 25, disease 24, cell 23, grippe 23, symptom 23, medical 22, sequence 22, news 21, query 21, biology 19, science 19, homo 18	Cluster 1 (98): grippe, infection, disease, virus Cluster 2 (94): database, doctor, indemnity, protein Cluster 3 (3): stance, doctor, economics, indemnity Cluster 5 (1): using, Dante, bovine, charge Cluster 6 (1): blog, Aire, Dana, November Cluster 9 (1): analysis, literature, technique, Wilkins Cluster 10 (1): vitamin, addendum, compare, diet Cluster 11 (1): economics, indemnity, April, decrease
10. Flu, database, genetic, evolution	genet 198, evolution 192, database 177, virus 62, sequence 60, science 50, biology 41, data 41, homo 38, news 37, gene 36, grippe 36, doll 32, search 29, nidus 25, technology 24, health 23, genome 22, mole 21, group 20	Cluster 1 (124): grippe, infection, disease, virus Cluster 2 (75): database, doctor, indemnity, protein Cluster 9 (1): analysis, literature, technique, Wilkins
11. Children, vaccination, Infection, prevention	babe 287, infection 275, vaccination 267, prevention 265, hepatitis 106, disease 84, control 72, vaccine 60, grippe 54, immunisation 45, health 35, risk 35, good_word 33, Amen 31, virus 29, data 26, teenager 24, adult 23, varicella 23, young 23	Cluster 1 (197): grippe, infection, disease, virus Cluster 2 (1): database, doctor, indemnity, protein Cluster 3 (1): stance, doctor, economics, indemnity Cluster 14 (1): egis, page, search

Continued on next page

Table 7 – continued from previous page

Query # and keywords	20 most frequent main terms in the retrieved document headers (with term frequency)	4 most frequent main terms from each cluster containing the query documents
12. Influenza, ethnicity, marital, status	stance 257, grippe 186, Mari 183, ethnicity 149, race 103, health 94, pneumonia 63, HTML 57, view 50, training 49, disease 48, vaccination 40, gender 35, data 32, kin 32, state 29, babe 25, birth 25, report 25, query 24	Cluster 1 (188): grippe, infection, disease, virus Cluster 2 (7): database, doctor, indemnity, protein Cluster 3 (4): stance, doctor, economics, indemnity Cluster 4 (1): following, query, department, force
13. Flu, heart, cancer, mortality	cancer 336, pump 244, mortality 229, disease 153, health 152, news 61, risk 53, titty 51, cold 48, rates 48, loser 44, high 40, dying 34, lung 34, stroke 34, attack 32, doll 30, vaccine 30, rate 29, study 24	Cluster 1 (194): grippe, infection, disease, virus Cluster 2 (4): database, doctor, indemnity, protein Cluster 3 (2): stance, doctor, economics, indemnity
14. Flu, socioeconomic, survival, time	time 213, survival 180, health 92, stance 63, cancer 44, news 41, data 29, factor 29, bout 28, patient 28, fear 27, have 25, report 25, vaccine 25, babe 24, gain 23, rates 23, disease 22, medical 21, over 21	Cluster 1 (103): grippe, infection, disease, virus Cluster 2 (83): database, doctor, indemnity, protein Cluster 3 (11): stance, doctor, economics, indemnity Cluster 4 (1): following, query, department, force Cluster 11 (2): economics, indemnity, April, decrease
15. Influenza, mortality, blood, pressure	mortality 261, roue 256, grippe 243, pressure 224, high 122, disease 84, health 51, pneumonia 51, pump 43, diabetes 42, cholesterin 35, gain 33, infection 32, risk 32, rate 29, cause 28, control 28, vaccine 25, dying 24, medical 22	Cluster 1 (198): grippe, infection, disease, virus Cluster 2 (1): Database, doctor, indemnity, protein Cluster 4 (1): following, query, department, force

Continued on next page

Table 7 – continued from previous page

Query # and keywords	20 most frequent main terms in the retrieved document headers (with term frequency)	4 most frequent main terms from each cluster containing the query documents
16. Dental, health, flu, resistance	health 419, immunity 201, slit 200, news 72, fear 68, cold 63, antibiotic 55, disease 53, diabetes 49, drug 36, data 35, babe 30, cancer 29, infection 29, vaccine 28, medical 26, More 24, insulin 24, topic 24, virus 24	Cluster 1 (154): grippe, infection, disease, virus Cluster 2 (43): database, doctor, indemnity, protein Cluster 3 (2): stance, doctor, economics, indemnity Cluster 12 (1): bacteria, causing, cleaner, feel
17. Insurance, doctors, flu, economics	indemnity 219, doctor 212, economics 180, health 139, fear 49, medical 46, vaccine 43, exam 39, news 38, shot 36, have 32, medicine 24, More 23, bout 23, heir 23, cover 22, patient 22, technology 21, need 20, costs 19	Cluster 1 (70): grippe, infection, disease, virus Cluster 2 (109): database, doctor, indemnity, protein Cluster 3 (10): stance, doctor, economics, indemnity Cluster 6 (2): blog, Aire, Dana, November Cluster 7 (1): prof, Aussie, Canberra, Princeton Cluster 8 (1): event, Bari, Ribes, bite Cluster 11 (3): economics, indemnity, April, decrease Cluster 15 (1): George, admiralty, banking, contract Cluster 16 (1): doctor, economics, indemnity, post Cluster 17 (1): Canada, Canuck, Florida, Laws Cluster 18 (1): Davis, ally, berg, bill

Continued on next page

Table 7 – continued from previous page

Query # and keywords	20 most frequent main terms in the retrieved document headers (with term frequency)	4 most frequent main terms from each cluster containing the query documents
18. Infection, virus, evolution, database	virus 395, infection 251, evolution 211, database 193, homo 70, immunodeficiency 59, hepatitis 42, data 36, sequence 31, ring 30, type 30, AIDS 29, grippe 24, query 23, drug 22, disease 21, search 21, immunity 20, site 19, figurer 18	Cluster 1 (197): grippe, infection, disease, virus Cluster 2 (2): database, doctor, indemnity, protein Cluster 3 (1): stance, doctor, economics, indemnity
19. Comparative, genomics, infection, diseases	genomics 276, disease 241, comparative 222, infection 208, function 50, query 50, genome 37, homo 33, immunity 32, genetics 30, microbiology 30, bacteria 27, centre 27, informatics 26, mole 26, animal 25, medicine 24, host 23, journal 23, institute 22	Cluster 1 (194): grippe, infection, disease, virus Cluster 2 (1): database, doctor, indemnity, protein Cluster 3 (2): stance, doctor, economics, indemnity Cluster 4 (1): following, query, department, force Cluster 5 (1): using, Dante, bovine, charge Cluster 9 (1): analysis, literature, technique, Wilkins
20. Flu, pandemic, epidemic, modeling	epidemic 217, pandemic 203, mold 163, grippe 108, disease 46, news 42, health 41, doll 34, vaccine 32, outbreak 31, world 31, data 30, virus 29, AIDS 28, have 25, baht 23, figurer 23, glob 22, Spanish 21, bout 21	Cluster 1 (123): grippe, infection, disease, virus Cluster 2 (71): database, doctor, indemnity, protein Cluster 3 (4): stance, doctor, economics, indemnity Cluster 8 (1): event, Bari, Ribes, bite Cluster 11 (1): economics, indemnity, April, decrease

Continued on next page

Table 7 – continued from previous page

Query # and keywords	20 most frequent main terms in the retrieved document headers (with term frequency)	4 most frequent main terms from each cluster containing the query documents
21. Flu, virus, genome, database	virus 319, genome 219, database 216, grippe 102, sequence 55, homo 45, data 42, doll 32, news 32, project 31, world 31, genet 30, health 24, search 24, query 23, bout 22, gene 22, SARS 21, free 21, vaccine 21	Cluster 1 (170): grippe, infection, disease, virus Cluster 2 (28): database, doctor, indemnity, protein Cluster 4 (1): following, query, department, force Cluster 7 (1): prof, Aussie, Canberra, Princeton

Table 7 confirms the consistency between query keywords and most frequent main terms in the retrieved documents. At the same time, the fact that not all query keywords correspond to highest frequency terms supports the idea of query extension based on main term frequencies. The clearest case in our data (Table 7) is the appearance of the term "grippe" in the documents retrieved in response to a query containing the term "flu".

Columns 1 and 3 in Table 7 demonstrate that frequent terms in queries and clusters are *terminologically complementary*. For instance, the term "indemnity" used in socio–economic analysis of influenza is frequent in many clusters, but is not present in all queries. At the same time, we should emphasize that this analysis was manually filtered. We have nothing to say about trivial cases like frequency of the term "doctor", or irrelevancy (in our context) of such terms as "AIDS" that appear frequently in the texts, but not in the queries. Note, however, that the term "AIDS" does not appear as one of the four most frequent main terms in any of the clusters. Concluding, we state that term frequency comparison in queries and clusters proves to be very informative, and leads to observations that are much more interesting than those made on the basis of the similarity of term frequencies in queries and the retrieved documents.

6.2 Sets of Representatives

Every document in a cluster has a relevance rank assigned by Yahoo!. Since a single cluster may contain documents retrieved by different queries, the cluster may contain several documents with equal ranks. A representative of a cluster was defined in Section 5.3 as a document with the lowest rank in the cluster. As was noted previously, a cluster may have several representatives.

A query representative in a cluster was defined by Formula 5 as the lowest–ranked document retrieved by the query, and assigned to the cluster. The distribution of query and cluster representatives is given in Table 8. Rows correspond to queries, columns — to clusters. An entry in a cell (s, t) indicates the rank of the representative document of query Q_s in cluster H_t^*.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	**1**	29	190																	
2	**1**	144	148																	
3	**1**	111	55												**56**					
4	**1**	183	2																	
5	**1**	4	42	**13**					**102**		**41**									
6	**1**	157	80							**125**										
7	**1**	91										**160**	**112**							
8	**1**	9	77																	
9	2	**1**	28		169	**50**			140	146	179									
10	**1**	5							133											
11	**1**	195	10													**191**				
12	**1**	60	56	189																
13	**1**	27	40																	
14	**1**	8	30	76							48									
15	**1**	195		198																
16	**1**	8	84											**18**						
*17	15	2	**1**			89	133	**81**			58						**148**	**159**	**37**	**200**
18	**1**	150	158																	
19	**1**	4	30	167	**162**				151											
20	**1**	5	99					139			162									
21	**1**	8		105			**61**													

Table 8: The distribution of query and cluster representatives over clusters. Rows correspond to queries, columns — to clusters. Results for the original query are in row (*17). Cluster representatives are highlighted.

Highlighted entries in each column of Table 8 identify the set of cluster's representatives. Thus, cluster 1 has 19 representatives. This cluster contains documents retrieved by every query, and 19 of those documents have been assigned the highest relevance rank of one. Clusters 2–20 each have a single representative.

Comparison of Tables 8 and 5 indicates a correlation between the rank of a query's representative in a cluster and the number of documents retrieved by the query and assigned to the cluster.

Cluster 1 contains 19 of 21 rank–one documents, and therefore represents most completely all queries. Clusters 2 and 3 contain the two documents most relevant (according to Yahoo!'s ranking) to the original query (17). We conclude that query (17) is best represented by the union of documents from clusters 2 and 3. The entire set of documents retrieved in response to query (17) is spread over 11 clusters. All other queries reflect different aspects of the semantic area outlined by query (17).

7 Discussion

A novel method of data analysis described in this paper was applied to a collection of document sets retrieved in response to a set of queries — the original query and a set of semantically related queries. The three main stages of our method are:

1. Emphasizing the similarity between documents retrieved by different queries, cluster all documents using a multipartite graph clustering algorithm

2. Characterize relations between clusters and queries by term frequencies in retrieved documents and the documents' relevancy ranks, in our case, assigned by Yahoo! search engine

3. Examine the distribution over clusters of the documents retrieved by the original query. Use that information to replace the response to the original query with a new set of documents constructed on the basis of relationships discovered between clusters and queries.

In our example, stage 3 is accomplished by adding all documents from clusters 2 and 3 to the response to the original query (17). A more restrictive approach would be to add only the documents from cluster 3, because it contains the document most relevant to the original query (according to Yahoo!'s ranking).

Our approach relies on a set of neighborhood queries semantically related to the original query provided by the user. A manually generated set of neighborhood queries was used in our experiments. We intend to investigate dictionary–based methods of automatic query generation using semantic relationships between the terms.

There are cases, however, where the neighborhood queries are also provided by the user. Namely, when performing a search on a subject that a user has very limited knowledge of and that is too sophisticated to be covered by a single query. In such cases, the user produces a query set extensively covering the subject, and therefore allowing for direct application of all the methods presented in this paper. Our clustering technique would produce a manually observable set of relevant responses that reflect various aspects of the subject.

Finally, we would like to emphasize that the proposed clustering method has low polynomial computational complexity and does not require setting any initial hyper–parameters.

8 Future Work

As noted earlier, the completeness of our document representation is restricted by the dictionary used. Document words not matched against the dictionary terms are discarded from the analysis. In order to make the document representation more complete (i.e. reduce the number of words discarded), a more flexible dictionary matching mechanism is required. At the same time, the flexibility of matching has to be controlled to prevent words irrelevant from a human perspective to be associated with a dictionary group.

In this section, we discuss a probabilistic extension of the current dictionary matching method. For the purposes of our work, we consider dictionary terms grouped by synonymy. However, the matching mechanism described below is not specific to synonymy groups and could operate on terms grouped by a different criteria.

We propose to associate each dictionary group of terms with a Markov chain — a generator for the group. States of the chain correspond to letters in the alphabet. The generator is characterized by an appropriate prior probability distribution together with transition probabilities between states. The prior and transition probabilities are estimated based on terms present in the dictionary group.

Given a document word as input and a set of generators (one for each dictionary group), the final assignment of the word to a group is achieved by a Dynamic Logic (DL) method published

in [12]. Via an optimization procedure, DL allows to perform classification assignment in a very broad probabilistic framework of which our generators are a particular instance.

References

[1] Giordano Adami, Paolo Avesani, and Diego Sona, *Clustering documents in a web directory*, WIDM '03: Proceedings of the 5th ACM international workshop on Web information and data management (New York, NY, USA), ACM Press, 2003, pp. 66–73.

[2] S. Brin and L. Page, *Dynamic data mining: Exploring large rule spaces by sampling*, 1999, http://www-db.stanford.edu/∼sergey.

[3] Joaquim Ferreira da Silva, João Mexia, Carlos Agra Coelho, and José Gabriel Pereira Lopes, *Document clustering and cluster topic extraction in multilingual corpora.*, ICDM (Nick Cercone, Tsau Young Lin, and Xindong Wu, eds.), IEEE Computer Society, 2001, http://terra.di.fct.unl.pt/∼jfs/publicacoes/ICDM01.ps, pp. 513–520.

[4] Michael Dittenbach, Helmut Berger, and Dieter Merll, *Improving domain ontologies by mining semantics from text*, CRPIT '04: Proceedings of the first Asian-Pacific conference on Conceptual modelling (Darlinghurst, Australia, Australia), Australian Computer Society, Inc., 2004, pp. 91–100.

[5] Eibe Frank and Gordon W. Paynter, *Predicting library of congress classifications from library of congress subject headings*, J. Am. Soc. Inf. Sci. Technol. **55** (2004), no. 3, 214–227.

[6] Nizar Grira, Michel Crucianu, and Nozha Boujemaa, *Unsupervised and semi-supervised clustering: a brief survey*, 2005, http://www-rocq.inria.fr/∼crucianu/src/BriefSurveyClustering.pdf.

[7] Bin He and Yongzheng Zhang, *Clustering documents in large text corpora*, 2003, http://flame.cs.dal.ca/∼yongzhen/course/6505/report.pdf.

[8] L.I.Perlovsky, *Integration of language and cognition at pre-conceptual level*, International Conference on Integration of Knowledge Intensive Multi-Agent Systems, 2003, pp. 280–285.

[9] *Metacrawler web search*, 2005, http://metacrawler.com.

[10] B. Mirkin and I. Muchnik, *Combinatorial optimization in clustering*, Handbook of Combinatorial Optimization (Ding-Zhu Du and P.M. Pardalos, eds.), vol. 2, Kluwer Academic Publisher, Boston, MA, 1998, pp. 261–329.

[11] E. A. Nowick, K.M. Eskridge, D. A. Travnicek, X.Chen, and J. Li, *A model search engine based on cluster analysis of user search terms*, Library Philosophy and Practice **7** (2005), no. 2.

[12] Leonid I. Perlovsky, *Neural networks and intellect: Using model-based concepts*, Oxford University Press, 2000.

[13] A. Vashist, C. Kulikowsky, and I. Muchnik, *Automatic screening for groups of orthologous genes in comparative genomics using multipartite clustering*, Technical Report 33, DIMACS, 2004, p. 23.

[14] Vivisimo, *Clusty the clustering engine*, 2004, http://www.clusty.com.

[15] *Wordnet: A lexical database for english language*, Cognitive Science Laboratory, Princeton University, http://wordnet.princeton.edu.

[16] *Wordnet search 2.1*, http://wordnet.princeton.edu/perl/webwn.

[17] Rui Xu and D. Wunsch II, *Survey of clustering algorithms*, IEEE Transactions on Neural Networks **16** (2005), no. 3, 645–678.

[18] *Yahoo! search*, http://search.yahoo.com/search.