BIOINFORMATICS ORIGINAL PAPER

Vol. 30 no. 22 2014, pages 3240–3248
doi:10.1093/bioinformatics/btu514

Data and text mining

Advance Access publication July 29, 2014

Retro: concept-based clustering of biomedical topical sets

Lana Yeganova*, Won Kim, Sun Kim and W. John Wilbur

National Center for Biotechnology Information, National Library of Medicine, NIH, 8600 Rockville Pike, Bethesda, MD 20894, USA

Associate Editor: Igor Jurisica

ABSTRACT

Motivation: Clustering methods can be useful for automatically grouping documents into meaningful clusters, improving human comprehension of a document collection. Although there are clustering algorithms that can achieve the goal for relatively large document collections, they do not always work well for small and homogenous datasets.

Methods: In this article, we present Retro—a novel clustering algorithm that extracts meaningful clusters along with concise and descriptive titles from small and homogenous document collections. Unlike common clustering approaches, our algorithm predicts cluster titles before clustering. It relies on the hypergeometric distribution model to discover key phrases, and generates candidate clusters by assigning documents to these phrases. Further, the statistical significance of candidate clusters is tested using supervised learning methods, and a multiple testing correction technique is used to control the overall quality of clustering.

Results: We test our system on five disease datasets from OMIM® and evaluate the results based on MeSH® term assignments. We further compare our method with several baseline and state-of-the-art methods, including K-means, expectation maximization, latent Dirichlet allocation-based clustering, Lingo, OPTIMSRC and adapted GK-means. The experimental results on the 20-Newsgroup and ODP-239 collections demonstrate that our method is successful at extracting significant clusters and is superior to existing methods in terms of quality of clusters. Finally, we apply our system to a collection of 6248 topical sets from the HomoloGene® database, a resource in PubMed®. Empirical evaluation confirms the method is useful for small homogenous datasets in producing meaningful clusters with descriptive titles.

Availability and implementation: A web-based demonstration of the algorithm applied to a collection of sets from the HomoloGene database is available at http://www.ncbi.nlm.nih.gov/CBBresearch/Wilbur/IRET/CLUSTERING_HOMOLOGENE/index.html.

Contact: lana.yeganova@nih.gov

Supplementary Information: Supplementary data are available at *Bioinformatics* online.

Received on February 4, 2014; revised on July 3, 2014; accepted on July 21, 2014

1 INTRODUCTION

Text-clustering algorithms group a set of documents into meaningful clusters and can be useful in many settings. Although a vast collection of clustering methods exists, most of them work best for relatively large sets of documents. However, we frequently encounter topics that are discussed in a limited number of documents. One such example is NCBI's HomoloGene resource available through PubMed (http://www.ncbi.nlm.nih.gov/homologene/), which is a system for automated detection of homologs among the annotated genes of several completely sequenced eukaryotic genomes. The resource provides the list of articles associated with every entry, and most of these sets are small and homogenous. When applied to such document collections, traditional clustering approaches often fail to produce useful results, suggesting that more sensitive methods are needed for the task.

There are two ways to fail at clustering small document collections. First, the algorithm could place all documents in a single cluster. A second way in which an algorithm may fail is to produce weak clusters that represent noise. For example, many clustering algorithms produce different results depending on the cluster initialization, which suggests some of the clusters are unstable or weak. The more homogenous the data, the more severe both of these problems are.

Another issue that motivated this research is that most existing algorithms produce clusters that are not self-descriptive, and one has to further sort through the documents to decipher the subject of the cluster. However, presenting visual cues, such as cluster titles, can significantly improve the user perception of clustering results. We emphasize the importance of cluster titles and provide clusters along with meaningful titles.

In this article, we develop a robust clustering algorithm that addresses these issues and produces meaningful clusters along with concise and descriptive titles for small document collections. Our algorithm, which we named Retro, predicts cluster titles prior to clustering. It first identifies central phrases using *P*-values based on a hypergeometric test. Each such phrase is then evaluated as a potential cluster title using supervised learning techniques. The evaluation hinges on the hypothesis that a phrase represents a useful cluster title if it has discriminative power. Thus, a cluster is formed around the phrase by collecting the documents that contain that phrase in the title and is enriched with documents that are closely related to those. Then, a machine learning method is used to measure how strongly that cluster stands out from the rest of the documents in the set. Clusters are then sorted based on that measure, and a multiple testing correction technique is used to control the number of top-scoring clusters to retain. Document titles in the resultant clusters are then used to extend the initial phrase into a more descriptive cluster title. Our approach can be characterized as a soft clustering algorithm, which allows a document to be

*To whom correspondence should be addressed.

assigned into more than one cluster and does not force every document into a cluster. It produces a flat structure, works on homogenous datasets and does not require any initialization or control parameters except for a desired confidence level of the output for multiple testing correction.

In Section 2, we review related literature. Section 3 describes in detail our clustering framework. Section 4 presents the datasets used for experiments and clustering results. In Section 5, we present an application of our system and also discuss our clustering approach and draw conclusions.

2 RELATED WORK

The problem of partitioning a set of objects into a number of groups is encountered in many research areas, and a plethora of different clustering methods exists. Clustering methods differ depending on how an object is represented, how the similarity between a pair of objects is measured and the clustering procedure used. They further vary in the structure of the final solution, which can be flat or hierarchical, and in cluster membership requirement, which can be described as hard, soft or fuzzy. Thorough surveys on clustering methods are presented by Aggarwal and Zhai (2012), Anastasiu *et al.* (2013), Jain *et al.* (1999) and Xu and Wunsch (2005).

Our study was motivated by an effort to cluster small homogenous sets of documents of the HomoloGene resource and to display these clusters along with descriptive titles. Each set represents a collection of documents on a particular genetic disease or disorder. We found that traditional clustering algorithms, such as K-means and expectation maximization, frequently fail on this clustering task. Furthermore, even when clusters can be identified, finding titles for small clusters represents another challenge.

These circumstances led us to consider the problem of clustering small document collections from a different perspective. Unlike traditional clustering algorithms, which first group the documents and then identify cluster titles, we first identify central topics (defined by key phrases) and then assign documents to these topics. The idea is also highlighted by Osinski *et al.* (2004), who, similar to our approach, suggested reversing the process to ensure they create a humanly understandable cluster title and only then assign documents to it. The authors argued that document similarity measures used to assign documents into clusters frequently do not yield coherent and meaningful titles. A similar approach is supported by Wilbur (2002), who discussed that the problems of clustering terms and documents are closely related—good term groups provide means to discover good document clusters and vice versa. This is likewise the principle behind co-clustering methods (also known as bi-clustering) (Busygin *et al.*, 2008), which define a co-cluster as a pair of cluster features and associated cluster documents, and partition a dataset into such co-clusters. Adaptive subspace iteration (Li *et al.*, 2004) is another co-clustering approach, which integrates K-means and eigenvector analysis in an alternating procedure that iteratively assigns data points into clusters and identifies cluster-dependent keywords.

When searching for central concepts in a document collection, we restrict our attention to well-formed biomedical phrases to improve the readability of the final clustering results. To that end, we represent documents in the collection by multiword phrases. Several studies have identified the importance of representing a document in terms of phrases rather than single words (Wang *et al.*, 2009). Such representation preserves word-order information and can improve the quality of the clusters by better leveraging information presented in the document. This is consistent with an observation (Yeganova *et al.*, 2009) that many biomedical concepts are expressed as multiword phrases. Zamir and Etzioni (1998) introduced a suffix tree clustering (STC), which relies on suffix arrays (Gusfield, 1997) to efficiently extract phrases and constructs clusters by identifying documents that share common phrases. Suffix tree structure is also used in the Lingo algorithm (Osinski *et al.*, 2004) to extract the key phrases. Different from these methods, we use *P*-values obtained from the hypergeometric test to predict key phrases for a set of documents. We believe this measure is more sensitive for predicting central concepts than mere phrase frequency.

As mentioned earlier, our study was motivated by an effort to cluster small and homogenous sets of documents. In that sense, our problem is closely related to the clustering of Web search results, which is often referred to as a post-retrieval clustering. Search results clustering is a problem of grouping together short snippets of text returned by a search engine in topically coherent clusters and labeling them with meaningful and descriptive phrases. Similar to our task, this cannot be addressed by traditional clustering algorithms. STC and Lingo mentioned above are some of the first systems attempting to address the problem. A commercial version of Lingo is available at http://search.carrot2.org/stable/search. Subsequently, numerous post-retrieval clustering approaches have been proposed, including most recent studies by Carpineto and Romano (2010) and Moreno *et al.* (2013). We compare our method with both of these methods on Web data, and present our findings in the Section 4.

Another approach related to clustering is topic modeling, such as latent Dirichlet allocation (LDA), which aims to find term clusters (Blei *et al.*, 2003; Papadimitriou *et al.*, 1998; Steyvers and Griffiths, 2007). Topic modeling is a probabilistic method based on the idea that documents are mixtures of topics (where a topic is a probability distribution over words) that produces document-topic distribution. Although a topic does not correspond to a cluster of documents, several studies (Lu *et al.*, 2011; Xie and Xing, 2013) have extended LDA to obtain document clusters. Following Lu *et al.* (2011), the document-topic distribution can be deemed as a mixture proportion vector over topics, and a document is assigned to its maximum probability topic. This produces a clustering that corresponds to the topics detected by LDA.

Other work similar to our approach is Simultaneous Keyword Identification and Clustering (SKWIC; Frigui and Nasraoui, 2004). It performs document clustering and cluster-dependent keyword identification simultaneously. However, SKWIC can only produce a hard clustering. Hofmann (1999) presented the cluster-abstraction model for text data. Although this model integrates clustering and keyword selection, it rather focuses on learning topic hierarchies.

3 METHODS

Let *S* denote a document collection on some topic that we aim to cluster. Retro starts by identifying phrases central to the set of documents *S*. We

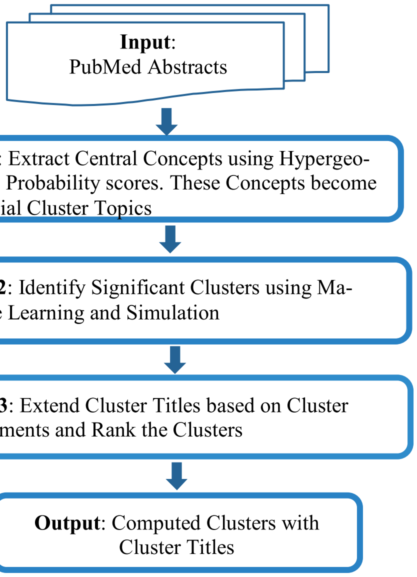
Input:
PubMed Abstracts
Step1: Extract Central Concepts using Hypergeometric Probability scores. These Concepts become Potential Cluster Topics
Step2: Identify Significant Clusters using Machine Learning and Simulation
Step3: Extend Cluster Titles based on Cluster documents and Rank the Clusters
Output: Computed Clusters with Cluster Titles

Fig. 1. The Retro processing pipeline

model the centrality of a phrase using the *P*-value based on the hypergeometric test as described in Section 3.1. These central phrases are then evaluated as potential cluster topics. Not every key phrase qualifies as a good cluster topic. Intuitively, a phrase that is general and appears in most of the documents is not appealing as a cluster topic. Yet, a phrase that is specific and is expressed in a subset of documents may define a useful cluster topic. We use machine learning to evaluate a key phrase as a candidate cluster topic based on a group of documents that relate to that phrase. The algorithm is described in Section 3.2. Once the clusters are identified, we apply a title-generating heuristic to induce a cluster title. The heuristic attempts to find a well-formed title, an extension of the existing original cluster title, and is described in Section 3.3. Final cluster titles are scored, and these title scores are used to rank the clusters for display. Figure 1 below presents a schematic of the algorithm.

3.1 Identification of central concepts

In this section, we automatically identify central multiword concepts that characterize a set of documents. We will interchangeably refer to such concepts as central concepts or key phrases. A phrase in our context is an ordered sequence of two or more words.

We start by preprocessing the entire Medline and compiling a comprehensive list of phrases. We collect from Medline all multiword text strings bounded by punctuation or stop words, and require that each text string selected has appeared at least twice. This process results in 8.3 million unique text strings, which we will refer to as *AllPhrases* and treat as our phrase vocabulary.

We further analyze these phrases to identify synonyms and string variants. We detect the synonymy between two strings by consulting the UMLS® Metathesaurus®. A detailed explanation of UMLS can be found at http://www.nlm.nih.gov/research/umls/. We define two strings to be variants of each other if one string can be derived from the other by word order permutation, or if they both stem to the same phrase. String variants and synonyms are grouped into synonymy classes producing 1.3 million classes involving 3.2 million phrases of the original 8.3 million phrases. The remaining 5.1 million phrases are singletons. An example of a synonymy class consisting of four phrases is *cardiac muscle myosin*; *cardiac muscle myosins*; *cardiac myosin*; *cardiac myosins*. We further use a method for detecting well-formed biomedical phrases (Kim *et al.*, 2012) to score the phrases in each synonymy class and select the highest scoring phrase as a representative for that class. The phrase *cardiac muscle myosin* scores highest among the four phrases and is selected to represent the class. This concludes the preprocessing step.

We now process the set of documents S to find central concepts. First we map document titles to vocabulary phrases in *AllPhrases*. Thus, document titles are represented in terms of vocabulary phrases. For example, the title '*Siblings with prune belly syndrome and associated pulmonic stenosis, mental retardation, and deafness*' yields the phrases *eagle barrett syndrome*; *intellectual disabilities; prune belly*; *pulmonic stenosis*. If a phrase belongs to a synonymy class, then that phrase is mapped to the representative phrase for that synonymy class. In the above example, the phrase *prune belly syndrome* is mapped to *eagle barrett syndrome*.

Next, we use the hypergeometric distribution to identify central phrases as follows. Let N_s be the size of the document set S, N be the size of Medline, N_t be the number of documents in Medline that contain phrase t and N_{st} be the number of documents in the set S containing t. The random variable Y representing a number of documents containing the phrase in the set S is a hypergeometric random variable with parameters N_s, N_t and N (Larson, 1982). The probability distribution of Y is shown as follows:

$$P(y) = \binom{N_t}{y}\binom{N - N_t}{N_s - y} \Bigg/ \binom{N}{N_s}.$$

From N_{st} we compute the P-value, i.e. the probability of the observed (N_{st}) or a higher frequency arising by chance as follows:

$$P\text{-value} = \sum_{y = N_{st}}^{\min\,(N_s, N_t)} P(y)$$

The P-value reflects how strongly the phrase is represented in S as compared with all of Medline. A low P-value indicates that we are observing a rare event and that the observed phrase represents a statistical discovery, indicating that the phrase is central for the set S. For example, a few top-scoring key phrases for *hypertrophic cardiomyopathy* are *beta-myosin heavy-chain gene mutations*; *familial hypertrophic cardiomyopathy*; *familial wolff parkinson white syndrome*; *alpha tropomyosin gene*, etc. Every significant phrase identified is evaluated at the next stage as a potential cluster topic.

3.2 Cluster definition and evaluation

Now we evaluate significant key phrases as potential cluster topics using our machine learning approach. First, given such a phrase, we define its base cluster as all documents that contain the phrase in their title. There are several reasons for doing so as opposed to defining base clusters to include documents that contain the phrase either in the title or the abstract. First, we believe that titles generally are carefully chosen to be good indicators of contents of scientific articles. Moreover, by analyzing user behavior (Islamaj Doğan and Lu, 2010) in response to PubMed retrieval (PubMed retrieves articles that contain the query either in the title or in the abstract), it was found that an article was more likely to be clicked on if the query terms appeared in the title. And finally, we compared the correlation of the computed clusters with Medical Subject Headings (MeSH) assignments for the two scenarios. We found the F-measure is significantly higher for the title-based clustering, due to a considerable difference in precision. These experiments are presented in Section 4.

At the same time, by requiring the presence of a key phrase in document titles, a cluster may be constrained to a focused set of documents. While some documents may not contain the phrase in the title, they may still be relevant for the cluster. Hence, we try to enrich the base cluster by adding documents to it described in a following way. Given a phrase, we label documents in the base cluster as positive. Let N_p represent the number of documents in the cluster. Remaining documents in the set

S are labeled negative. We train a naïve Bayes classifier to learn the difference between documents in positive and negative classes. Single words and two-word phrases from titles and abstracts are used as features. Then document scores for both positive and negative sets are computed as the sum of all positively weighted features in a document. The highest scoring negative document is then iteratively added to the positive class, and the classifier is re-trained. We iterate as long as the average score of documents in the expanding positive class increases and the average score of documents in the original base class increases. We do not add more than N_p documents to the positive class to avoid topic drift.

Sets are a priori partitioned into positive and negative classes depending on presence or absence of a key phrase in the document titles. Therefore, supervised learning would rely heavily on single and double terms from the key phrase to learn the difference between the two classes. We avoid that by excluding features generated by the key phrase from training. For example, given a key phrase *sarcomere protein*, we exclude the features *sarcomere*, *protein* and *sarcomere protein* from training.

At this point, we have compiled the cluster and have to determine whether the cluster is significant. Let S_p and S_n represent the average scores of documents in the resultant positive and negative sets, and let $S_{diff} = S_p – S_n$ represent the difference between these scores. Can S_{diff} tell us whether the group of documents in the positive class stands out as a cluster? Because we are working with small sets, it is important to do statistical testing to verify that we are not just detecting noise. We propose a simulation framework to evaluate whether S_{diff} is statistically significant.

To that end, we randomly draw $N_p^{'}$ (size of final cluster) documents from set *S* and label them positive. Remaining documents in the set *S* are treated as negative, and the naïve Bayes classifier is trained. To be consistent with the training above, for a key phrase consisting of *k* tokens, we exclude the top weighted (*2*k–1*) features from scoring. The average score RS_p (over positive random set), RS_n (over negative random set) and $RS_{diff} = RS_p – RS_n$ are computed as above. We repeat this random sampling and training 1000 times, obtain the distribution of $\{RS_{diffi}\}_{i\,=\,1,\ldots,1000}$ over 1000 experiments and compute the *P*-value of S_{diff} as the fraction of times that $RS_{diff} > S_{diff}$ over the 1000 experiments. A low *P*-value implies that S_{diff} is significant, indicating that documents in the positive set represent a strong cluster, and suggesting, in turn, that the phrase is discriminating. Intuitively, significant clusters tend to emerge around phrases that are essential and specific. On the contrary, frequent general phrases do not have discriminative power because documents in the positive class defined by that phrase cannot be distinguished from the rest of documents in set *S*.

At this stage, we need to decide which clusters to retain and which clusters to drop from consideration. A naive approach is to establish a desired confidence level-based threshold and retain all clusters whose *P*-value is below that threshold. However, when a family of clusters is tested simultaneously, one needs to control the confidence level for the whole family of clusters tested. The expected proportion of false-positive clusters among the family of clusters tested is referred to as a false discovery rate, and that rate increases with the number of clusters tested.

In this study, we adopt a useful statistical correction for multiple comparisons, the Benjamini–Hochberg correction (Benjamini and Hochberg, 1995), which allows one to achieve the desired confidence level for the family of clusters tested as follows. Assume *m* clusters are being tested. Let

$$P_1 \leq P_2 \leq \ldots \leq P_m$$

be the ordered list of *P*-values, and let *k* be the largest *i* for which

$$P_i \leq \alpha * i/_m$$

Then by choosing only the first *k* clusters, the Benjamini–Hochberg multiple-testing procedure controls the false discovery rate at the desired level of α. In this study, we set $\alpha = 0.01$. In other words, we expect to see no more than 1% false-positive clusters in the final clustering result.

A final postprocessing step merges the clusters that share >80% of the documents. We also merge the clusters with significant title overlap. This concludes the process of computing the clusters, and next we induce a cluster title from the titles of the documents in that cluster.

3.3 Expanding cluster titles

Here we apply a series of rules to induce a well-formed, descriptive and humanly understandable title that contains the original key phrase. To achieve this goal:

1. Extract all possible candidate *n*-grams whose tokens appear in at least half the titles, where *n* ranges from 2 to 20 tokens. Here we want the noun phrases to remain intact. Thus, noun phrases found in a title are not allowed to break into smaller pieces but instead are treated as a unit.
2. Check part of speech tags for the first and the last word in a candidate title. Conjunctions, verbs, prepositions and symbols are not allowed as the first token. Conjunctions, verbs, prepositions, symbols, articles, determiners, adjectives and certain pronouns are not allowed as the last token.
3. Retain only the candidate phrases that contain the original key phrase that was used to compute the cluster. Discard any candidates that start or end with '-' or '.'. We also found that certain characters such as '/', ';', ':' within the candidate title suggest that the title is not well formed. We discarded such candidate titles.
4. Check grammatical dependency relations. We discard candidates for which the head word of a preposition does not appear in the same candidate. For instance, candidate title *cystic fibrosis carrier screening to an HMO population* is discarded because no headword can be found for 'to'. We validate a few other patterns such as *between A and B* ensuring that they are not broken.

After candidate titles are generated for each cluster, we score each of them as a sum of negative log of hypergeometric *P*-values of single terms that appear in the candidate title. The highest scoring candidate title is selected as the cluster title. That title score is also used to rank the clusters for final presentation.

4 RESULTS AND EVALUATION

Evaluating the performance of clustering algorithms formally is a challenging task. It is challenging not only because manually created gold standards are required, but also because creating such gold standards is not well-defined. Clustering results may vary depending on the goal of the clustering task, but be equally useful for their particular tasks. Many papers in the clustering literature assemble artificial datasets from documents originating from different topics, cluster that assembled dataset and evaluate their methods based on how well their clusters resemble the initial partition. However, that type of evaluation may not be illuminating when one is trying to cluster a small set of documents that are all on the same narrow topic, as we are.

One way to evaluate our system in the biomedical domain is by using the MeSH resource. MeSH is a controlled vocabulary of terms that is used for indexing PubMed articles. MeSH terms are manually assigned to most PubMed articles and indicate the topics of an article. A detailed explanation of MeSH can be found at http://www.nlm.nih.gov/mesh/. We rely on these

MeSH term assignments to evaluate how well a set of documents is grouped by topic into clusters. This is possible because our clustering method depends exclusively on text and makes no use of MeSH terms.

We also perform experiments to evaluate and compare our system with other state-of-the-art and baseline systems on benchmark datasets. We have identified two datasets that are particularly useful for evaluating our methodology. One is a subset of the 20-Newsgroups (http://people.csail.mit.edu/jrennie/20Newsgroups). The 20-Newsgroup set consists of messages collected from 20 different Usenet newsgroups. The News-Similar-3 subset of 20-Newsgroup is challenging and of a particular interest because it contains messages from three similar topics: *comp.graphics* (973 messages), *comp.os.ms-windows* (985 messages) and *comp.windows.x* (988 messages). We compare our method with K-means, and Expectation Maximization (EM) on the News-Similar-3 collection.

Another challenging benchmark dataset is ODP-239 (Carpineto and Romano, 2010). This recently created collection was put together for evaluating the Web search results clustering (SRC) algorithms. ODP-239 consists of 239 topics, each with about 10 subtopics and 100 documents. Each document consists of a title and a short snippet. The task of clustering the ODP-239 is particularly hard because subtopics are similar to each other, and text fragments are short. We compare our algorithm with three state-of-the art search result clustering methods, which are Lingo (Osinski *et al.*, 2004), OPTIMSRC (Carpineto and Romano, 2010) and adapted GK-means (Moreno *et al.*, 2013). We also compare our algorithm with LDA-based clustering (Lu *et al.*, 2011) on the ODP-239 dataset.

4.1 Example

We begin with an illustrative example and demonstrate our system with the experiments carried out on a collection of five disease datasets that we will refer to as the Disease5 collection. This collection includes cystic fibrosis (273 documents), hearing loss (731 documents), DiGeorge syndrome (110 documents), autism (136 documents) and hypertrophic cardiomyopathy (HCM) (138 documents). PubMed articles for these disease sets were collected from OMIM® (Online Mendelian Inheritance in Man, http://www.omim.org/) in 2011. The OMIM database is an online catalog of human genes and genetic disorders that provides referenced overviews for its contents. The articles included in the Disease5 collection are references provided for the human-generated overviews in the corresponding OMIM topic pages.

Complete clustering results of applying our algorithm to the Disease5 dataset are presented in Supplementary Appendix A in the Supplementary Materials. Table 1 lists the results of our clustering applied to one of the conditions, *hypertrophic cardiomyopathy*. The HCM dataset contains 138 records and yields 12 clusters. HCM is a primary disease of the myocardium, the muscle of the heart, in which a portion of the myocardium is hypertrophied (thickened) without any obvious cause. HCM is attributed to mutations in one of a number of genes that encode for one of the sarcomere proteins. A significant number of mutations occur in the beta myosin heavy chain gene on chromosome 14 q11.2-3, which is discussed in one of the clusters. The mutation may be inherited from one of the parents (referred to as *familial hypertrophic cardiomyopathy*, found as a cluster) or be a *de novo* mutation. Mutations in the *PRKAG2* gene may also cause HCM as well as familial *Wolff-Parkinson-White syndrome*. Another group of papers discusses *glycogen storage diseases*, which may present as HCM. The *severe neonatal hypertrophic cardiomyopathy* cluster discusses specific challenges when the disease is manifested in the neonatal stage. Different *missense mutations* leading to HCM are discussed in the next cluster. And the *chromosome 14q1* cluster discusses localization of genes mutated in HCM to chromosome 14q1.

Table 1. Clusters for HCM (138 documents)

Score	Cluster title
53.01	beta-myosin heavy-chain gene mutations
47.10	familial hypertrophic cardiomyopathy
41.94	familial wolff parkinson white syndrome
26.13	alpha tropomyosin gene
18.11	cardiac troponin t mutations cause familial hypertrophic cardiomyopathy
14.26	prkag2 gene causing HC
13.63	glycogen storage cardiomyopathy
13.17	severe neonatal hypertrophic cardiomyopathy
11.84	activator protein
11.63	missense mutations
10.88	apical hypertrophic cardiomyopathy
10.61	chromosome 14q1

Table 2. Evaluating the significance of clusters using MeSH terms assigned to documents in clusters

Disease name	Number of docs	Number of cluster	Average *P*-value
Digeorge syndrome	110	8	3.75E-3
Autism	136	13	6.55E-3
Hypertrophic cardiomyopathy	138	12	1.17E-3
Cystic fibrosis	273	31	7.87E-4
Hearing loss	731	82	7.03E-4

4.2 Evaluating with MeSH terms

We identify MeSH terms that appear within a cluster and compute their *P*-values using the hypergeometric distribution just as we did for the phrases. If documents were grouped randomly, MeSH terms assigned to documents within a cluster would have high *P*-values. On the contrary, low *P*-value MeSH terms indicate that documents in that cluster share the topic and are closely related. Table 2 presents the number of clusters and average *P*-values for each dataset in the Disease5 collection. *P*-values shown in the table are the average over the three most significant MeSH terms obtained from each cluster. The average *P*-values

Table 3. Top three most significant MeSH terms assigned to documents in top 8 clusters for HCM set from Disease5 collection

Score	Cluster title	Top 3 MeSH terms
53.01	beta-myosin heavy-chain gene mutations	myosin heavy chains, genetics* myosin heavy chains, genetics myosin heavy chains
47.10	familial hypertrophic cardiomyopathy	genetic linkage chromosomes, human, pair 14 base sequence
41.95	familial wolff parkinson white syndrome	multienzyme complexes protein-serine-threonine kinases, genetics protein-serine-threonine kinases
26.13	alpha tropomyosin gene	tropomyosin, genetics* tropomyosin tropomyosin, genetics
18.11	cardiac troponin t mutations cause familial hypertrophic cardiomyopathy	troponin, genetics* cardiomyopathy, dilated, genetics* troponin, genetics
14.26	prkag2 gene causing hypertrophic cardiomyopathy	amp-activated protein kinases myocardium, ultrastructure glycogen storage disease, genetics*
13.63	glycogen storage cardiomyopathy	multienzyme complexes, genetics multienzyme complexes protein-serine-threonine kinases
13.17	severe neonatal hypertrophic cardiomyopathy	carrier proteins, genetics carrier proteins carrier proteins, genetics*

"*" at the end of a MeSH term indicates that the concept expressed by a MeSH term is central to the article.

Table 4. Recall, Precision and F-measure of title-based clusters versus abstract-based clusters for Disease5 collection

Clustering	Title-based	Abstract-based
Precision	0.6541	0.5343
Recall	0.5644	0.5830
F1	0.6059	0.5576

are low, ranging from 1.05E-2 to 2.76E-4, indicating that documents form strong clusters.

Table 3 provides the three most significant MeSH terms for the top-scoring computed clusters for the HCM set. Examining these terms reveals that cluster titles highly correlate with the MeSH terms humanly assigned to documents. This in turn suggests that the proposed clustering method effectively captures subtopics of a set of documents and extracts meaningful clusters.

Using MeSH terms we can also compute the standard recall-precision values for the clustering. To do that, we compute the average recall and precision values over the three most significant MeSH terms in each cluster and further average them over all clusters. Results are presented in Table 4. Here we also present results in association with our earlier discussion of title-based versus abstract-based clustering. We performed the same computations for abstract-based clustering and compare the recall, precision, and F1 score of title-based versus abstract-based clustering. We observe that F1 score is considerably higher for title-based clustering, owing to a large difference in precision.

4.3 Comparative evaluation

First, we compare our algorithm with two baseline methods, expectation maximization and K-means, on the News-Similar-3 benchmark dataset. Table 5 presents paired precision, recall and F1 score for each of these methods. Waikato Environment for Knowledge Analysis (Hall *et al.*, 2009) open-source tool was used to run EM and K-means clustering. Unigrams and bigrams with normalized *tf-idf* weights were used as features, and two input parameters were tested as number of clusters: $K_1 = 3$ and $K_2 = 100$. For $K_1 = 3$, EM produced three groups of documents (2623 in group 1, 177 in group 2 and 146 in group 3), whereas K-means gave three imbalanced groups of size 2943, 2 and 1, essentially failing on this dataset. Moreover, precision of both methods was $P = 0.33$, which is equal to the percentage of positive pairs among all pairs when everything is assigned to one group (i.e. when no clustering is performed). News-Similar-3

Table 5. Performance comparison of Retro with EM and K-means (KM) on News-Similar-3

Number of clusters	K = 3		K = 100		K = 1	N/A
Method	EM	KM	EM	KM	N/A	Retro
Precision	0.333	0.333	0.374	0.336	0.333	0.679
Recall	0.799	0.998	0.140	0.666	1.000	0.017
F1	0.470	0.499	0.204	0.447	0.499	0.033

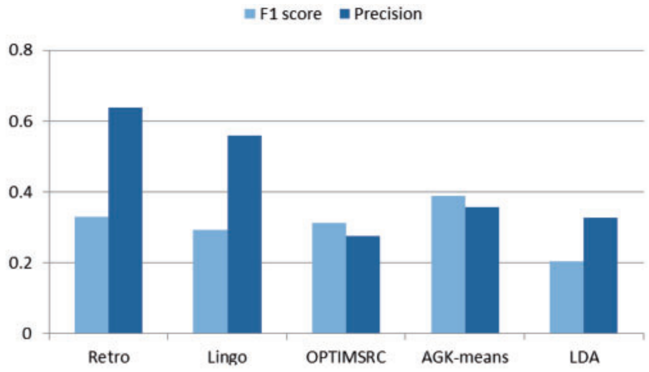

Fig. 2. Evaluation of Retro, Lingo, OPTIMSRC, AGK-Means and LDA on ODP-239 dataset

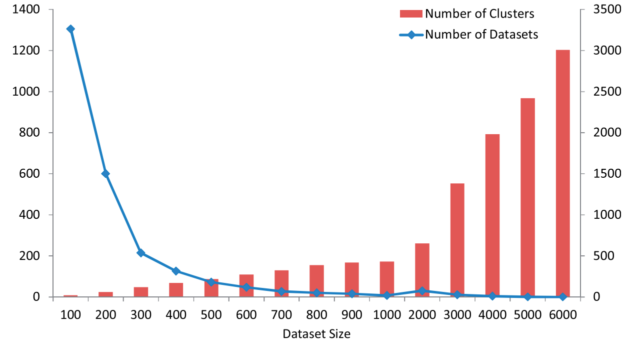

Fig. 3. Average number of clusters in datasets of varying size (bar graph), and frequency of datasets of varying size (line graph)

set is not a particularly small set; however, it is homogeneous, and that is what we believe misleads K-means.

Retro is quite different from both EM and K-means in that it does not try to find a group for every document. It only finds significant, high-precision and meaningful clusters. Hence for this particular dataset, Retro covers about 52% of the documents and detects many useful and concise clusters such as *ftp site, x server, video card, windows nt*, etc. The full list of clusters (102 clusters total) can be found in Supplementary Appendix A in the Supplementary Materials. The average precision of clusters found by Retro is 0.679, which is remarkably high for this challenging dataset.

Next, we evaluate our algorithm on the ODP-239 benchmark dataset. We compare Retro with three state-of-the-art Web search result clustering algorithms: Lingo, OPTIMSRC and adapted GK-means (AGK-means). We also compare our algorithm with LDA-based clustering. Figure 2 shows performance comparison of these methods. As mentioned earlier, our algorithm is most akin to Lingo. OPTIMSRC and AGK-means are two more recent search result clustering algorithms tested on ODP-239 collection. OPTIMSRC is a meta search result clustering algorithm; it combines the output of four different systems. AGK-means is a variation of classical K-means, which allows labeling each cluster directly from its centroid. Both OPTIMSRC and AGK-means are partitioning algorithms. To compare Retro and Lingo with OPTIMSRC, AGK-means and LDA, we group all the documents that are not covered by clusters in Retro and Lingo into a dummy cluster ‘other topics’ following a standard approach. As shown in Figure 2, the comparison yields the following average paired F1 scores: Retro (0.331), Lingo (0.294), OPTIMSRC (0.313), AGK-means (0.39) and LDA (0.205). These numbers demonstrate that our method is comparable with the state-of-the art SRC methods and superior to LDA on this challenging task. However, we believe that including a dummy cluster in the evaluation obscures the true performance of our algorithm, and we continue by comparing our method with other methods based only on clusters extracted.

Both Retro and Lingo put an emphasis on extracting high precision clusters, while OPTIMSRC and AGK-means do not achieve such high precision, as they try to include every document into a cluster. Figure 2 displays the average paired precision for Retro, Lingo, OPTIMSRC, AGK-means and LDA averaged over 239 topics. We observe that Retro has a significant advantage in precision over other methods, reaching 8% improvement in precision over Lingo, the second best scoring method.

Evaluation results for Lingo are based on a raw results file provided by the authors of Lingo. F1 scores and precision values for OPTIMSRC and AGK-means are extracted from their corresponding evaluation tables (Carpineto and Romano, 2010; Moreno *et al.*, 2013). For LDA topic modeling, we used the MALLET implementation at http://mallet.cs.umass.edu (McCallum, 2002).

4.4 Application

We applied our algorithm to HomoloGene, a reference database and a system for automated detection of homologs among the annotated genes of several completely sequenced eukaryotic genomes. It provides information on genes, proteins, conserved domains and protein alignments. The resource also provides the list of articles associated with genes and sequences for every homology group automatically retrieved from PubMed. The list of linked PubMed records is often relatively small in size compared with the datasets that are generally used for document clustering.

The HomoloGene collection consists of 6248 gene records selected from the HomoloGene resource. Of these 6248 sets, 3263 contain 40–100 documents, 2827 contain 101–1000 documents and 369 sets contain >1000 records, with the largest set containing 8282 records (tumor protein p53).

Figure 3 provides statistics on the average number of clusters produced on applying our algorithm to datasets in the HomoloGene database. The *x* axis represents the dataset size, the bar graph represents the average number of clusters in each bin, and the line graph represents the total number of datasets in the bin. Overall, we compute at least one cluster for 6210 of the 6248 HomoloGene datasets. On average, we identify 8.1 significant clusters for small sets containing 100 or fewer documents. That group constitutes over half of all datasets and, thus, is important. The average size of clusters over the whole collection is 7 documents per cluster, and for the sets containing 100 or fewer documents it is 5.6 documents. A few examples of small HomoloGene datasets along with computed clusters are presented in Supplementary Appendix B in the Supplementary Materials. Clustering results of the full collection are made publicly available at http://www.ncbi.nlm.nih.gov/IRET/HomoloGene.

5 DISUSSION AND CONCLUSIONS

In this article, we propose an algorithm for clustering biomedical topical sets that identifies clusters along with descriptive titles. While there do exist clustering algorithms that can achieve the goal for relatively large document collections, they are less effective for small document sets. Yet, in the gene-related literature, we frequently encounter small document sets that can benefit from clustering. Retro is designed for such small topical datasets, and, to our knowledge, no such clustering system exists in the biomedical domain. Our strength is at extracting accurate and meaningful clusters, a challenging task when dealing with small and homogenous datasets.

Retro is not limited to biomedical literature and, as demonstrated in this article, can be successfully applied in other domains. The advantage of applying it in the biomedical domain is the availability of the UMLS resource, which we consult for synonym detection.

The novel key phrase identification approach is different from those that have been proposed in the literature in that we model the centrality of concepts using the hypergeometric probability distribution as opposed to simple phrase frequency. While frequency can produce reasonable results, we find the hypergeometric model is more sensitive for identifying key phrases.

Another original contribution of our algorithm is the use of a supervised learning framework to evaluate the statistical significance of induced clusters. This is particularly important when dealing with small datasets for verifying that we are not just detecting noise. Combined with a multiple testing correction technique, we are able to control the overall quality of the output. A single required parameter in our system is the desired confidence level, which controls the tail of the clustering, and varying it will result in a shorter or longer list of clusters being displayed. We set that parameter to 0.01 leading to no more than 1% of clusters being erroneous.

An additional favorable characteristic of our method is the reproducibility of the clustering results, i.e. the method will consistently induce the same clustering given a dataset and the same control parameter value. Most clustering algorithms, on the other hand, will produce different results depending on the cluster initialization method and input parameters, and there is generally no good way to test the significance of the clusters produced.

We also emphasize our choice of naïve Bayes as a supervised classifier. Naïve Bayes treats each term independently, and in the learning process, terms do not obscure each other. Learned positive term weights provide an intuitive measure of how specific and important that term is to the positive set versus a negative one.

As with any unsupervised clustering method, this approach has its weaknesses. For example, the *autism* dataset in the Disease5 collection contains *genomewide screen* and *genomic screen* as separate clusters. It would be desirable to merge them; however, our method does not find enough evidence to merge them. Another example is in the *DiGeorge syndrome* dataset in the Disease5 collection, where the most significant cluster found is *familial third-fourth pharyngeal pouch syndrome*. Our knowledge base is aware that *pharyngeal pouch syndrome* and *digeorge syndrome* are synonyms; however, it does not recognize *familial third-fourth pharyngeal pouch syndrome* as a synonym of *digeorge syndrome*. Some cluster titles may be too general, like *missense mutations* in the HCM dataset or *collaborative study* in the *cystic fibrosis* dataset in the Disease5 collection. However, they appear toward the tail of the output. Moreover, we noticed that even these general clusters are frequently meaningful in light of the specific dataset.

Our immediate future work intends to enhance the recognition of titles that mention the key phrase albeit using different terms. With that, given a key phrase, the algorithm would be able to include a wider group of documents and start with a stronger initial base cluster. In the future, we would also like to improve the running time of the algorithm. The current implementation of the algorithm takes slightly less than 2 min to process a set of 100 documents, and about 6 min to process a set of 300 documents. Such an approach can be applied in parallel on multiple machines to process a large number of small datasets. And this is how we use the current implementation. However, to process a single large set with multiple CPUs, the code needs parallelizing. This can be done for the second (testing cluster significance) and third (extending cluster titles) steps of the algorithm. We plan to do this in future work.

ACKNOWLEDGEMENTS

The authors thank Natalie Xie for her help in preparing the demonstration Web site. They also thank Dawid Weiss and Stanislaw Osinski for providing clustering results of Lingo on the ODP-239 collection, and Jose Moreno for providing recall and precision values of their clustering of the ODP-239 collection. The authors further thank the anonymous reviewers for valuable suggestions.

Funding: This research was supported by the Intramural Research Program of the NIH, National Library of Medicine.

Conflict of interest: none declared.

REFERENCES

Aggarwal,C. and Zhai,C. (2012) A survey of text clustering algorithms. In: Aggarwal,C. and Zhai,C. (eds) *Mining Text Data*. Springer-Verlag, New York.

Anastasiu,D. *et al*. (2013) Document clustering: the next frontier. In: Aggarwal,C. and Reddy,C. (eds) *Data Clustering: Algorithms and Applications*. CRC Press, Boca Raton, FL.

Benjamini,Y. and Hochberg,Y. (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society*, **57**, 289–300.

Blei,D. *et al*. (2003) Latent dirichlet allocation. *Journal of Machine Learning Research*, **3**, 993–1022.

Busygin,S. *et al*. (2008) Biclustering in data mining. *Computers and Operations Res*., **35**, 2964–2987.

Carpineto,C. and Romano,G. (2010) Optimal meta search results clustering. In: *Proceedings of the 33rd Annual ACM SIGIR Conference*. Geneva, Switzerland, pp. 170–177.

Frigui,H. and Nasraoui,O. (2004) Simultaneous clustering and dynamic keyword weighting for text documents. In: Berry,M. (ed.) *Survey of Text Mining*. Springer-Verlag, New York, Inc. pp. 45–70.

Gusfield,D. (1997) *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology*. Cambridge University Press, New York, NY, USA.

Hall,M. *et al*. (2009) The WEKA data mining software: an update. *SIGKDD Explor*., **11**, 10–18.

Hofmann,T. (1999) The cluster-abstraction model: unsupervised learning of topic hierarchies from text data. In: *Proceedings of the International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, pp. 682–687.

Islamaj Doğan,R. and Lu,Z. (2010) Click-words: learning to predict document keywords from a user perspective. *Bioinformatics*, **26**, 2767–2775.

Jain,A. *et al*. (1999) Data clustering: a review. *ACM Comput. Surveys*, **31**, 264–323.

Kim,W. *et al*. (2012) Identifying well-formed biomedical phrases in MEDLINE® text. *J. Biomed. Inform*., **45**, 1035–41.

Larson,H.J. (1982) *Introduction to Probability Theory and Statistical Inference*. John Wiley & Sons, New York.

Li,T. *et al*. (2004) Document clustering via adaptive subspace iteration. In: *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Sheffield, UK, pp. 218–225.

Lu,Y. *et al*. (2011) Investigating task performance of probabilistic topic models: an empirical study of PLSA and LDA. *Inform. Retrieval*, **14**, 178–203.

McCallum,A. (2002) *MALLET: A Machine Learning for Language Toolkit*. http://mallet.cs.umass.edu (14 August 2014, date last accessed).

Moreno,J. *et al*. (2013) Post-retrieval clustering using third-order similarity measures. In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, Sophia, Bulgaria, August 4-9, 2013, pp. 153–158.

Osinski,S. *et al*. (2004) Lingo: search results clustering algorithm based on singular value decomposition. *Intell. Inform. Syste. Adv. Soft Comput*., 359–368.

Papadimitriou,C. *et al*. (2000) Latent semantic indexing: a probabilistic analysis. Journal of Computer and System Sciences - Special issue on the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on principles of database systems, **61**, pp. 217–235.

Steyvers,M. and Griffiths,T. (2007) Probabilistic topic models. In: Landauer,T. *et al*. (eds), Handbook of Latent Semantic Analysis, Lawrence Erlbaum Associates, Mahwah, NJ.

Wang,A. *et al*. (2009) Text clustering based on key phrases. In: *The 1st International Conference on Information Science and Engineering, ICISE 2009*, pp. 986–989.

Wilbur,W.J. (2002) A thematic analysis of the AIDS literature. *Proc. Pac. Symp. Biocomput*., **7**, 386–397.

Xie,P. and Xing,E. (2013) Integrating document clustering and topic modeling. In: *Proceedings of Conference on Uncertainty in Artificial Intelligence*, pp. 694–703.

Xu,R. and Wunsch,D. (2005) Survey of clustering algorithms. *IEEE Trans. Neural Netw*., **16**, 645–678.

Yeganova,L. *et al*. (2009) How to interpret PubMed queries and why it matters. *J. Am. Soc. Inform. Sci*., **60**, 264–274.

Zamir,O. and Etzioni,O. (1998) Web document clustering: a feasibility demonstration. In: *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'98)*, pp. 46–54.