Clustering Document with Active Learning using Wikipedia

Abstract

Wikipedia has been applied as a background knowledge base to various text mining problems, including document categorization, topic indexing and information extraction. However, very few attempts have been made to utilize it for document clustering. In this paper we propose to exploit Wikipedia and the semantic knowledge therein to facilitate clustering, enabling the automatic grouping of documents with similar themes. Although clustering is intrinsically unsupervised, recent research has shown that incorporating supervision improves clustering performance, even when limited supervision is provided. The approach presented in this paper applies supervision using active learning. We first utilize Wikipedia to create a concept-based representation of a text document, with each concept associated to a Wikipedia article. We then exploit the semantic relatedness between Wikipedia concepts to find pair-wise instance-level constraints for supervising clustering, guiding clustering towards the direction indicated by the constraints. We test our approach on three standard text document datasets. Empirical results show that our basic document representation strategy yields comparable performance to previous attempts. Adding constraints improves clustering performance further by up to 20%.

1. Introduction

Clustering is an indispensable data mining technique, especially for handling large scale data. Text document clustering automatically groups documents with similar themes together while keeping documents with different topics in separate clusters. Conventionally, a document is represented using the *bag of words* (*BOW*) document model, consisting of terms that appear in the document and associated weights. By "terms" we mean words or phrases, but in most cases they are single-word terms. With the *BOW* model, similarity between documents is measured based on co-occurrence statistics involving their constituent terms. Hence the clustering algorithm can only relate documents that use identical terminology, and important sematic relations between terms such as acronyms, synonyms, hypernyms, spelling variations and related terms are all ignored. Furthermore, the *BOW* model assumes that terms appear independently and the order is immaterial. However, this contradicts reality. For example, "launch vehicle", "launch vehicles", "carrier rocket" and "satellite launch vehicle" all refer to a rocket carrying a payload into outer space; if these phrases are decomposed into single words, such sematic information will be lost.

We therefore propose to represent text documents by concepts rather than words, so that the semantic relations between concepts mentioned in the documents can be captured and retained. External semantic knowledge bases such as the *Open Directory Project*[1], *Wikipedia*[2] and expert-defined semantic ontologies such as *WordNet*[3] can be used to identify the concepts appearing within a document. For example, Hotho et. al. [10] and Recupero [20] map document terms to WordNet entries that each represent a different concept, whereas in [8] and [25] Wikipedia is used instead of WordNet and document terms are mapped to concepts represented by Wikipedia articles. By leveraging the external knowledge base, problems such as synonyms and hypernyms, spelling variations, and related terms can potentially be dealt with in an effective manner.

Wikipedia surpasses other structural knowledge bases in its coverage of concepts and up-to-date content. In contrast to extensive research on using Wikipedia for text categorization [8, 25], little work can be found on exploiting Wikipedia for clustering. We propose to use it to solve two substantially different problems in text document clustering. First, Wikipedia is used to create a semantic representation of text documents, by mapping phrases in the document text to a set of concepts that are each represented

[1] http://www.dmoz.org/
[2] http://www.wikipedia.org/
[3] http://wordnet.princeton.edu/

by a Wikipedia article. We use "Wikipedia concept" and "Wikipedia article" interchangeably in this paper. Secondly, in the clustering process, we cluster concepts according to their pair-wised semantic relatedness as computed from Wikipedia, so as to identify the major concept groups in a document cluster to facilitate active learning. If a document cluster is coherent and the documents within the cluster have similar themes, its major concept groups will be cohesive; otherwise, if the cluster consists of documents with different topics, its concept groups are expected to be diverse and loosely connected. Upon identifying the major concept groups, we select a set of representative documents for each topic group and find constraints on these documents. These pair-wise document-level constraints are actively learned, by querying a noiseless *oracle*. Two types of constraints are formed based on the oracle's answer: if the two documents concerned must appear together in the same group, a *must-link* constraint is formed; otherwise, if they must appear in different groups, a *cannot-link* is formed. These constraints are employed during the subsequent clustering process so that clustering is guided to the direction suggested by the constraints.

The remainder of this paper is organized as follows. In the next section we describe our approach of using Wikipedia to create a concept-based document representation. In Section 3 we propose the active learning algorithm that finds pair-wise constraints by utilizing the semantic relatedness measure between concepts as computed from Wikipedia. Section 4 briefly reviews the underlying clustering algorithm we used in our experiments. Experiments and results are reported and discussed in Sections 5 and 6. Related work is reviewed in Section 7. Section 8 concludes the paper and discusses future work.

2. Enriching Text Documents with Wikipedia Concepts

When using Wikipedia for text mining, it is common to map document terms to concepts in Wikipedia [13, 8, 25]. Different approaches have been proposed to accomplish this. Gabrilovich and Markovitch [8] map a document to a Wikipedia article based on features computed from the Wikipedia article content, anchor texts of the article's incoming links, and features describing the link structure associated with the article. In [2], the entire document is treated as a query to Wikipedia and is associated with the top articles in the returned result list. In [25], titles of Wikipedia articles are searched for within fixed-length subsequences of a document; this matching method is rather brittle though: the article titles must be matched literally and completely, every word of the title must appear within the sequence.

We investigate a more efficient and flexible method for mapping a normal text document to appropriate Wikipedia concepts, by leveraging an informative and compact vocabulary—the collection of anchor texts in Wikipedia. Each link in Wikipedia is associated with an *anchor text*. The anchor text can be regarded as a descriptor of its target article. In many cases the anchor text differs from the categorical name of the target article. We observe that anchor texts have great semantic value: they provide alternative names, morphological variations and related phrases for the target articles. Furthermore, the anchor text itself is plain text which comes from and naturally fits into normal text documents. It is a natural option for building the connections between normal document terms and Wikipedia articles. However, how to implement this is not straightforward. For example, different anchor texts are used in different contexts to refer to the same article, and the same anchor text can actually link to different articles in different situations. In other words, anchor texts are ambiguous just as other human created text.

Our approach works in three steps: identifying candidate phrases within the document text, mapping them to Wikipedia articles and then selecting salient concepts. Given a plain text document as input, we first break it into sentences. In each sentence we scan for and locate a set of phrases that match anchor texts in Wikipedia. We also create a collection of n-grams from the anchor texts so that partial match of an anchor text is allowed. Each candidate phrase is mapped to an anchor text in Wikipedia and overlaps between them are handled.

For each anchor text, we then retrieve its target articles in Wikipedia and choose the most relevant article by applying context-based sense disambiguation. The result is a list of Wikipedia articles that representing the concepts and topics mentioned in the input document. However, not all of the candidate concepts are desirable for describing the document topic. On the contrary, marginally related concepts add noise to the representation, which adversely impacts the document similarity calculation and harms clustering. We thus perform attribute selection: only select salient concepts that are closely related to other concepts in the document; concepts that are loosely related to others are discarded. The outcome is a set of concepts representing the topics mentioned in the input document and each concept is weighted by its number of occurrences within the document.

Handling Overlaps. When mapping document terms to anchor texts, we search for n-grams with a maximum length of 10 words. When this is done, it is possible to have overlaps between two matches, in spite of the fact that the search is already confined to be within a sentence. For example, the text sequence "south central pennsylvania" will result in six matches: "south", "central", "pennsylvania", "south central", "central pennsylvania" and "south central

pennsylvania". We want to preserve the one that is more specific and more likely to be linked as a concept. Therefore, for each candidate phrase we calculate its *likelihood* of being a linked concept. Given a candidate phrase p, its total number of appearances within the Wikipedia anchor texts $f_a(p)$ and occurrences as plain text in Wikipedia articles $f_t(p)$, the likelihood is $\frac{f_a(p)}{f_a(p)+f_t(p)}$. The n-gram with the highest likelihood is selected. In the above example, the phrase "south central pennsylvania" appears 15 times in total in the Wikipedia snapshot we used and in 9 cases within an anchor text, which gives it the highest likelihood.

Sense Disambiguation. Article that are targets of the anchor text in Wikipedia become the possible senses of that anchor text. For example, "pluto" is used to refer to 26 different articles in Wikipedia, including the planet Pluto, Pluto in the Greek mythology, a World War II operation named Pluto, and Pluto the pup. We use a machine-learning algorithm to disambiguate the senses of an anchor [16]. Given a possible sense, we consider two factors: its prior probability, i.e. the probability of the sense being the target sense of the anchor, and its closeness to the context, which is represented by a set of unambiguous anchors. For example, out of the 870 times that "pluto" appears as an anchor text, 594 times it links to the article about the Pluto planet, therefore this article has higher likelihood to be the target article than the others. We build a classifier based on these two features and train it with a set of Wikipedia articles. The input of the classifier is a set of possible senses for a given anchor text and the context, the classifier predicts for each sense a probability of it being the intended sense. The one with the highest probability is selected. More details about the algorithm can be found in [16].

Attribute Selection. After the first two steps we have a list of candidate concepts. It is a long list, because the document phrases are matched against a huge vocabulary—the anchor texts in Wikipedia. In the snapshot we used for our experiments, there are 713,539 distinct anchor texts after all have been changed to lower case. Pruning irrelevant concepts from the list is important, for both efficiency and accuracy considerations. We want to preserve concepts that are better descriptors of document theme and can represent the major topics mentioned within the document. Therefore we measure the salience of each candidate concept by its cohesiveness with other concepts in the document: salient concepts have higher relatedness to more concepts. Concepts that are loosely related to others are discarded. This requires the calculation of sematic relatedness between concepts. The measure we use is Milne and Witten's similarity measure [15]. Given two concepts x, y and the sets of hyperlinks X and Y that appear in the text of the associated Wikipedia article, the similarity of x and y is calculated by:

$$SIM(x,y) = 1 - \frac{max(\log|X|, \log|Y|) - \log|X \cap Y|}{N - min(\log|X|, \log|Y|)} \quad (1)$$

Here $X \cap Y$ is the intersection of the two sets X and Y, N is the total number of articles in Wikipedia and $|\cdot|$ denotes the size of the sets.

We define two concepts x and y to be neighbors if the semantic relatedness between them is no less than a prespecified threshold ϵ. We denote the neighborhood of a concept c as $N_\epsilon(c)$. The more neighbors a concept has, i.e. the larger the size of $N_\epsilon(c)$, the more salient the concept is. This is similar to the density based clustering algorithm DBScan [7], where data points are connected into clusters based on the cohesiveness of their neighborhood. Upon computing $N_\epsilon(c)$ for each concept in the list, we eliminate those whose size of $N_\epsilon(c)$ falls below a threshold, say n.

Instead of finding an appropriate threshold through trial and error, we use a more flexible approach that adapts the threshold automatically to suite different situations. If no concept's $N_\epsilon(c)$ contains more than n concepts, we incrementally decrease the neighborhood size threshold n, until some of the concepts are preserved or n is zero. If n is zero, it indicates that the topics mentioned within the document are diverse. In this case, all the candidate concepts will be selected. The semantic relatedness threshold is always set to 0.6. Note that relatedness is bounded in [0,1], with 0 meaning the concepts are not related and 1 meaning they are exactly the same.

Our strategy is efficient because no full text analysis of Wikipedia articles is involved. Although we count the total number of occurrences of a given phrase in articles, it is still a very light computation compared to the full text analysis because it can be performed based on a pre-computed index. We will show in Section 5 that the effectiveness of our strategy is comparable with other more complicated ones.

Besides using the concepts alone to represent a document, we can also combine our strategy with the *Bag of Words* model. In general, there are two strategies for merging the two representation schemes [10]. One strategy is to *add* concepts into the bag of words, thereby creating a combined scheme that is basically the concatenation of words and concepts. Alternatively, the *replace* strategy replaces document terms that have been mapped to concepts in the concept-based representation with the concepts they have been mapped to, resulting in a mixture of words and concepts. We will investigate the effectiveness of each strategy for clustering in Section 6. From now on we will use ***BOC*** and ***BOW*** to denote respectively the concept-based and word-based document representation. The two hybrid schemes will be denoted ***Combined*** (for the *add* strategy) and ***Replaced*** (for the *replace* strategy).

3. Constraining Clustering using Wikipedia

Although clustering is intrinsically unsupervised learning, recent research found that providing clustering algorithms with a certain amount of supervision significantly improves performance in terms of accuracy [23]. Particularly when it comes to real world applications, some prior knowledge is usually known beforehand or easy to obtain, and can be used to guide the clustering process. Pair-wise instance-level constraints are one such type of knowledge and have been used widely in different clustering applications [23, 3, 5, 4].

Labeling is expensive. Therefore these pair-wise constraints should be actively learned rather than randomly selected, so that clustering benefits to the greatest extend possible. Normally a pair is selected according to certain criteria and posed to a noiseless *oracle* that determines which type of relation the given query pair exhibits. Here, we propose to use Wikipedia for identifying informative document pairs to pose as queries to the oracle, by analyzing the major concept groups in a collection of documents and finding documents that are more likely to be different. Our approach combines instance level constraints in semi-supervised clustering with active learning by selective sampling [18].

3.1. Active Learning of Constraints from Concept Clusters

With documents represented by concepts, we are able to analyze the major concepts for a collection of documents. In the attribute selection step described above, we select concepts based on their relatedness to the context. Similarly, given a set of documents in a cluster, we can cluster the concepts that appear frequently in these documents based on the density of their neighborhoods, resulting in a number of disconnected concept clusters. If the documents within the cluster have different themes, the concept clusters will be loosely connected and scattered in the semantic space. On the other hand, if the documents are homogeneous with similar topics, more closely related concept groups are expected. This reflects the "clustering hypothesis" from information retrieval: thematically similar documents tend to appear in the same group, whereas it is more unlikely for documents from different groups to have similar themes [1].

Given a cluster of documents C_D as input, our active learning algorithm works in three steps as follows.

Clustering Concepts. First, we cluster the attributes, i.e. the concepts, according to their pair-wise semantic relatedness and the density of their neighborhood in the semantic space. The definition of neighbor concept and neighborhood is the same as in the attribute selection step (cf. Section 2). The specific clustering algorithm used here is the DBScan algorithm [7]. We start by randomly selecting a concept c_i with $N_\epsilon(c_i) \geq n$, a concept cluster C_{c_i} is then created to hold c_i. Then we do a depth-first search for concepts that should be added to cluster C_{c_i}, until all the concepts have been iterated. For each concept $c_k \in N_\epsilon(c_i)$, if $|N_\epsilon(c_k)| \geq n$, then c_k is added to cluster C_{c_i}, and the search continues to explore the concepts within $N_\epsilon(c_k)$. Otherwise, if $|N_\epsilon(c_k)| < n$, c_k is still added to cluster C_{c_i}, but then we return to c_i and iterate to the next neighbor of c_i. The outcome is a number of disconnected *concept clusters*, which represent the dominant concepts and therefore topics mentioned in the input document cluster. For example, Table 1 lists example concept clusters found from a document cluster containing documents from three classes of the *20Newsgroup* data set: *alt.atheism*, *sci.space* and *rec.sport.baseball*. In order to ensure the concepts clustered are representative of the themes that appear within the document group, we only cluster the top m concepts that have the highest weight in the input document cluster.

Finding Candidate Documents. For each concept cluster, we then retrieve a small number of documents that have the highest weight for the current concept cluster, and create a ranked list of these documents. Documents from different lists are more likely to be thematically different. Given a concept cluster C_c and a document $d \in C_D$, we compute the weight of C_c for d as $w(d, C_k) = \sum_c w(d, c)$, where $c \in C_c$ and $w(d, c)$ is the weight of concept c in document d, which is the number of occurrences of concept c in document d. Then all the documents in C_D are ranked in descending order of their $w(d, C_k)$ value. The top documents are the ones we are looking for.

Obtaining Pair-wise Constrains. Two documents, each from a different list, are selected as the next query to the oracle. A *must-link* and *cannot-link* constraint is created respectively if the two documents are determined to be in the same cluster or different clusters. The oracle can also return "unknown" as the answer, in which case the answer is simply discarded and the pair will not be proposed again. It is possible that the same document appears within the top document list for different concept clusters and is selected to form the next query; in this case we skip to the next candidate document in the list. Moreover, documents that have been labeled before will not be used as a query again. In our experiments, we simulate the oracle by revealing the known class labels for the two documents concerned as in [5]. If the two documents share the same label, i.e. they belong to the same category originally, they form a must-link constraint, otherwise a cannot-link constraint is created between them.

The active learning approach is applicable to any document representation scheme where concept-level semantic relatedness is available. This includes *BOC* and the other two hybrid models: the *Combined* and *Replaced* mod-

Concept Clusters	
1	Qur'an Muslim Allah Hadith
2	Acceleration Mechanical work
3	Application software User (computing)
4	Earth's atmosphere Ozone
5	Breaking ball List of baseball jargon (P)

Table 1. Example of concept clusters

els. The only case where it cannot be applied is the *BOW* model because semantic knowledge at the concept level is not available.

4. Constrained K-MEANS Clustering

Two clustering algorithms are used in our experiments: the basic K-MEANS algorithm is used for clustering without constraints and its constrained version COP-KMEANS [23] for clustering with constraints. Here we briefly review the K-MEANS algorithm and describe our implementation of the COP-KMEANS algorithm.

K-MEANS is primarily used to evaluate the effect of different document representations on clustering performance. Taking a collection of instances and a pre-specified number of clusters k as input, it first select k random instances to seed k clusters, and each cluster is represented with a centroid instance. Then all the instances are assigned to their closest cluster centroid. This is followed by an update of each cluster centroid by averaging all the instances that have been assigned to the cluster. This step is repeated until there is no change in the cluster assignments of the instances or after a certain number of iterations.

When constraints are imposed in our experiments, COP-KMEANS is used as the clustering algorithm. COP-KMEANS works in a very similar way as K-MEANS. The only difference is that when predicting the cluster assignment for an instance, it will check the existing must-link and cannot-link constraints so that none of them will be violated. When an instance cannot be assigned to the nearest cluster because it violates existing constraints, COP-KMEANS will check whether the next nearest cluster can legally hold the instance, until the instance can be assigned to a cluster without violating any constraints; otherwise the instance will remain as an outlier.

COP-KMEANS only handles hard constraints—it requires that all the constraints must be satisfied. In practice, this often results in outliers that cannot be assigned to any cluster. In order to compare our empirical results with previous ones, we assign these outliers to the cluster that causes the smallest number of constraints to be violated. If two such clusters exists, the instance is assigned to the closer cluster.

The active learning step and the COP-KMEANS step are performed repeatedly and active learning is performed after the COP-KMEANS algorithm converges. In each iteration, if new constraints have been found after active learning, COP-KMEANS clustering starts again with the updated set of constraints. This process terminates when either of the following two criteria is satisfied: there is no change in the COP-KMEANS clustering process; or the number of queries asked exceeds the pre-specified number of queries that the clustering algorithm can pose to the oracle. It is worth noting that if there are no new cannot-link constraints found in an active learning iteration, the algorithm terminates. This is because the search for constraints is restricted to be within a cluster: when active learning starts, all previous constraints have already been satisfied with the minimum cost; new must-link constraints found within a cluster will not bring any changes to the current solution.

5. Experiments

We tested the proposed methodology with six test sets created from three standard text document data sets. In order to compare our results with previous ones, three evaluation measures were adopted: *Purity*, *Entropy* and the modified *Rand* Index. All implementations use a Wikipedia snapshot as of November 20, 2007. We first collected all the anchor text in the Wikipedia snapshot. Upon lower casing all the anchor texts, 713,539 distinct anchor terms were left, linking to 581,267 articles out of the entire 1,985,523 articles in the snapshot. Given a sentence, all n-grams with 10 words or less were mapped against the anchor text vocabulary.

5.1. Datasets

The following test collections were used:

20 Newsgroups (20NG) [12] contains messages from 20 different newsgroups, with 1000 messages for each newsgroup. Three test sets were created from this data: *20NG_Diff3*, *20NG_Sim3*, and *20NG_Multi10*. *20NG_Diff3* consists of the three substantially different categories *alt.atheism*, *sci.space*, and *rec.sport.baseball*, whereas the three categories in *20NG_Sim3* are significantly more similar: *comp.windows.x*, *comp.graphics*, and *comp.os.ms-windows.misc*. *20NG_Multi10* combines the original 20 categories into 10 major classes. For all the test sets, a random subsample of 100 documents is selected per category.

Reuters-21578 [21] consists of short news articles dating back to 1987. We created two test sets: the *R_Min15Max100* set was created following [10], resulting

in 43 categories with size varying from 15 documents to 100 documents, and 2484 documents in total; and the *R_Top10* set, consisting of the largest 10 categories in the original data set with 100 documents randomly selected per category.

Classic3 [6] is a much easier test set as compared to the first two. The three categories in this data set are well separated and balanced. Documents are the concatenation of title (if available) and abstract of academic papers. There are 3893 documents in total: 1400 documents in the *CRANFIELD* class, which is about aeronautical systems, 1033 *MEDLINE* documents from medical journals, and the remaining 1460 *CISI* documents are about information retrieval.

5.2. Methodology

For each dataset, we create four representations as described previously: *BOW*, *BOC*, and two hybrid ones *Combined* and *Replaced*. We also created a simple *2-gram* document model. The *BOW* and *2-gram* representations were compared to as baselines.

The preprocessing of documents differed for different document models. When creating the *BOW* representation, we selected alphabetical sequences, lowercased them, and removed stop words. To create the *Replaced* representation, we first removed all the words that had been mapped to a concept in the *BOC* representation. Then the remaining text was converted to the *BOW* with the same preprocessing as above. Finally we combined the resulting *BOW* and *BOC* representations. Rare attributes—words or concepts—that appeared just once were discarded in all models.

Each document was represented with a vector $\overrightarrow{t_d}$. The attributes of the vector can be either words or concepts and their values are their *TFIDF* weights, which is defined as $tfidf(d,t) = tf(d,t) \times log(\frac{|D|}{df(t)})$. Here, $tf(d,t)$ is the number of occurrences of attribute t in document d, $df(t)$ is the document frequency of t, i.e. the number of documents in which t appears, and $|D|$ denotes the total number of documents in the data set. Therefore we used the vector $\overrightarrow{t_d} = tfidf(d,t_1), \ldots, tfidf(d,t_n)$.

The similarity between two documents d_1, $d_2 \in D$ was measured by the cosine of the angle between its corresponding vectors $\overrightarrow{t_1}$, $\overrightarrow{t_2}$:

$$cos(\overrightarrow{t_1}, \overrightarrow{t_2}) = \frac{\overrightarrow{t_1} \cdot \overrightarrow{t_2}}{|\overrightarrow{t_1}| \times |\overrightarrow{t_2}|} \tag{2}$$

Since all the test set have pre-specified class labels, we used stratified 10-fold cross-validation for all experiments and report results as the average of 5 runs. In each fold, the clusterer is built on 90% of the entire data and then tested on the other 10% data with the same class distribution that the clusterer has not seen before. In contrast to classification, where performance on test data is particularly emphasized, for clustering the performance on both training and test data are useful. However, we only report results on the 10% test set due to space limits, even though the performance on the 90% training data is usually better.

Both K-MEANS and COPKMEANS require a prespecified number of clusters k. For the purpose of comparing relative performance, we set k equal to the number of classes in the data. Before evaluation, each cluster was labeled with the most frequent class label appearing in the cluster.

5.3. Evaluation Measures

Purity, Entropy, and the Rand Index were used to evaluate clustering performance. *Purity* evaluates the coherence of a cluster, that is, the degree to which a cluster contains members from a single class. Given a particular cluster C_i of size n_i, the purity of C_i is formally defined as $P(C_i) = \frac{1}{n_i} \max_h(n_i^h)$. Here $max_h(n_i^h)$ is the number of members from the most frequent category in cluster C_i and n_i^h denotes the number of members in C_i belonging to the hth class. A weighted average is then formed as:

$$Purity = \sum_{i=1}^{k} \frac{n_i}{n} P(C_i) \tag{3}$$

Entropy evaluates the disorder within the clusters. The entropy of cluster C_i with size n_i on a data set with k classes is defined to be $E(C_i) = \sum_{h=1}^{k} -\frac{n_i^h}{n_i} \log(\frac{n_i^h}{n_i})$, with the same notation as above. Then the overall entropy is the weighted sum of each individual clusters' entropy:

$$Entropy = \sum_{i=1}^{k} \frac{n_i}{n} E(C_i) \tag{4}$$

Following [23], we also use the Rand index to measure the consistency between the clustering solution and the original class structure. The Rand index is defined as [19]:

$$Rand(P_1, P_2) = \frac{a+b}{n \times (n-1)/2} \tag{5}$$

Here a is the number of pair-wise *true-positive* predictions: given a pair of documents d_i and d_j, the two documents belong to the same class and cluster. In contrast, b is the number of pair-wise *true-negative* predictions: d_i and d_j belong to different classes and also appear in different clusters.

6. Results and Discussion

In this section we report and discuss the results obtained. We will investigate four issues relevant to our approach:

Dataset	Words	2-grams	Concepts
20NG_Diff3	4,487	9,148	2,409
20NG_Sim3	3,819	6,805	1,808
20NG_Multi10	9,872	22,762	6,281
R_Min15Max100	7,615	35,075	6,085
R_Top10	5,029	16,397	3,972
Classic3	11,570	56,633	8,435

Table 2. Comparison of Dimensionality

how does the concept-based representation affect the feature space? How do the different representation schemes created by using Wikipedia as knowledge base impact clustering performance? Do the constraints improve clustering performance? How effective is the active learning algorithm?

6.1. Reduction in Dimensionality

High dimensionality is an important issue in text document clustering, known as the "curse of dimensionality". Techniques such as *Principal Component Analysis* and *Latent Semantic Analysis* reduce the dimensionality of high dimensional feature spaces by finding informative attribute surrogates, which are usually obscured. Considering that the number of distinct concepts appearing in a document is usually much lower than the number of words, we are expecting a significant reduction in the feature space dimensionality by using the *BOC* model.

Table 2 lists the number of dimensions in each kind of feature space. Since we also map multi-word phrases to concepts, it is only fair to compare the dimensionality of the concept feature space to an n-gram representation of the documents. The number of 2-grams was counted after removing single-word words. As expected, there is a dramatic drop in dimensionality between the *BOC* and the original *BOW* representation, and in some cases the dimensionality is almost halved. Moreover, if we compare the *BOC* with the 2-gram representation, the reduction is even greater.

It is worth noting that although the *R_Min15Max100* set is much more complicated than *20NG_Multi10* because it has more categories and significantly more documents, the vocabulary of *R_Min15Max100* is much smaller than that of *20NG_Multi10*, both in the *BOW* and the *BOC* model. Similarly, the *R_Top10* set only has about half the number of features as the *20NG_Multi10* set; however, these two sets have equal numbers of categories and documents per category. This is because documents in the Reuters data set are often duplicated, and multiple labels can be assigned to one document. Following [10], we use the first class label of a document as its class label in the test set. Some of the categories have very similar vocabulary, such as the *wheat*, *corn*, *grain* categories, which makes it more difficult for the clustering algorithm to distinguish between documents from different categories. This indicates that in general the Reuters data set is more difficult to cluster, as has been found empirically in [17].

6.2. Different Document Representations

Table 6.2 lists the *Purity* of clustering with different document representations. These results are reported on the 10% test data only and without employing any constrains during clustering. The results are disappointing at first sight. The *BOW* model often outperforms the *BOC* model, except for one test set (the *20NG_Sim3* dataset). However, combining the two models improves clustering, with a maximum increase of 14.8% in our experiments. The results for the *Entropy* measure are highly correlated to the *Purity* results, therefore we only report *Purity* due to space limits.

On two data sets even the combined model still loses to *BOW*: *20NG_Multi10* and *Classic3*. However, the decrease is trivial compared to the increases on other data sets. A possible cause for the reduction in performance is the curse of dimensionality: these two data sets have the most dimensions in their combined document model, more than 16,000 and 20,000 dimensions respectively.

The biggest improvement comes from the *20NG_Sim3* data set, where documents are about three very similar topics. The concept-based representation yields a much better result than *BOW*, and it is the only case where using the concepts alone produces substantially better clustering performance. By mapping terms to concepts, *BOC* circumvents the limitations of merely counting co-occurrences between single words. For instance, synonyms like "true color", "16.8 million colors", "24 bit color" and spelling variations such as "truecolor", "true colour" and "24-bit color" are all mapped to one concept "Truecolor"; this strengthens the similarity between documents that have these phrases.

Little research has been done on using features generated by using Wikipedia for clustering. The only directly comparable related result is on the *R_Min15Max100* data set, where Hotho et al. [10] achieved 0.618 purity after utilizing the concept hierarchy in WordNet and adding five hypernyms to *BOW*. However, they cluster the data set into 60 clusters instead of the actual number of classes in the data. After setting the number of clusters to 60 in our experiments, we achieved a purity score of 0.623 on the *R_Min15Max100* set, which is comparable to Hotho's.

6.3. Clustering with Constraints

The active learning algorithm discussed in Section 3 is applicable to any representation where the semantic relatedness between concept is available, that is, all models except for *BOW*. We experimented with each of them. Performance is compared with the situation when no constraints are used, i.e. the results in Table 6.2. The result were again calculated on the 10% test data, so that they indicate the

Dataset	Baseline BOW	Baseline 2-grams	BOC	Combined	Replaced	Improvement
	avg ± std	avg ± std	avg ± std	avg ± std	avg ± std	
20NG_Diff3	0.757 ± 0.180	0.420 ± 0.065	0.635 ± 0.128	**0.793 ± 0.131**	0.767 ± 0.086	4.76%
20NG_Sim3	0.443 ± 0.128	0.370 ± 0.062	0.460 ± 0.074	**0.497 ± 0.086**	0.453 ± 0.106	**14.8%**
20NG_Multi10	**0.467 ± 0.050**	0.179 ± 0.027	0.427 ± 0.060	0.464 ± 0.063	0.410 ± 0.058	-0.64%
R_Min15Max100	0.560 ± 0.021	0.454 ± 0.024	0.553 ± 0.028	**0.576 ± 0.041**	0.532 ± 0.019	2.86%
R_Top10	0.538 ± 0.031	0.522 ± 0.051	0.539 ± 0.048	**0.564 ± 0.044**	0.539 ± 0.044	4.83%
Classic3	**0.965 ± 0.078**	0.904 ± 0.069	0.964 ± 0.077	0.940 ± 0.103	0.964 ± 0.078	-0.10%

Table 3. Results on the effect of different document representations for clustering in terms of *Purity*

Dataset	BOC		Combined		Replaced	
	avg ± std	improvement	avg ± std	improvement	avg ± std	improvement
20NG_Diff3	0.696 ± 0.116	6.59%	0.820 ± 0.161	3.41%	0.787 ± 0.092	2.61%
20NG_Sim3	0.513 ± 0.106	11.5%	0.557 ± 0.058	5.09%	0.508 ± 0.081	12.9%
20NG_Multi10	0.457 ± 0.063	7.03%	0.478 ± 0.067	3.01%	0.438 ± 0.049	6.83%
R_Min15Max100	0.572 ± 0.028	3.32%	0.582 ± 0.041	1.04%	0.546 ± 0.024	2.63%
R_Top10	0.547 ± 0.040	1.48%	0.565 ± 0.033	0.02%	0.539 ± 0.049	0.00%
Classic3	0.964 ± 0.077	0.20%	0.940 ± 0.103	0.00%	0.964 ± 0.078	0.00%

Table 4. Comparison of *Purity* on test data between constrained and unconstrained clustering

ability of predicting unseen instances after being informed about various constrains based on the training data, avoiding optimistic Purity estimates. Table 4 compares the Purity scores whereas Table 5 compares the Entropy scores. As Table 4 demonstrates, employing constraints improves clustering performance, but to a different extend for different data sets. Again, the largest improvement comes from the *20NG_Sim3* test set.

We can also see that constrained clustering achieves little improvement on the Reuters data set, which is not unexpected. Because of the strong resemblance in the vocabulary used for different categories, the categories are more indistinguishable. In contrast, on the *20NG_Multi10* data set, which is equivalent in scale to the *R_Top10* test set, an average 5.6% increase in prediction accuracy is achieved by using constraints.

Table 6 lists the Rand index obtained when constraints are applied with each of the three representations. We list the Rand index value obtained when using standard K-Means on *BOW* in the first column, so as to compare to the baseline where no constraints are used. In each fold, after the clustering process with active learning as described in Section 3 converges, we got a set of clusters, each represented by a centroid instance. The instances in the remaining 10% test data were clustered to their closest centroids. Neither constraints nor active learning were used in the testing process. Figure 1 plots the Rand index against the number of constraints added for the *20NG_Sim3* and *20NG_Diff3* test sets. The number of constraints varies from zero to 1000 in this figure because it is unlikely that we would be able to obtain more than 1000 constraints in practice.

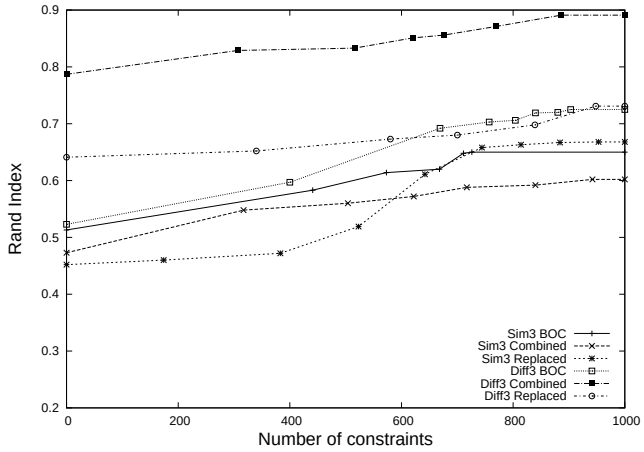

Figure 1. Results of employing constrains

7. Related Work

Related work consists of two groups: work on using Wikipedia for text document clustering and work on active learning and constrained clustering in semi-supervised learning.

Most of the previous research on using Wikipedia for automatically organizing text documents into thematically coherent groups is targeted at categorization [25, 8]. Both categorization and clustering involve a core step—generating features and producing an informative and accurate representation of a document. This is where external semantic knowledge bases can come in to help. The most closely related work to our approach on using semantic knowledge for clustering is by Hotho et. al. [10] and Recupero [20], where semantic relations formally defined in WordNet are used to enrich the *BOW* representation. Moreover, Hotho et. al. [10] also investigated the enrichment strategies we use here. Their result showed that the "add" strategy outperforms the "replace" and "only" strategies in terms of

Dataset	BOC		Combined		Replaced	
	avg ± std	improvement	avg ± std	improvement	avg ± std	improvement
20NG_Diff3	0.801 ± 0.251	18.9%	0.537 ± 0.380	16.1%	0.676 ± 0.218	2.60%
20NG_Sim3	1.296 ± 0.255	7.96%	1.383 ± 0.085	1.92%	1.424 ± 0.106	1.71%
20NG_Multi10	1.922 ± 0.182	0.77%	1.746 ± 0.228	2.68%	1.913 ± 0.176	3.43%
R_Min15Max100	1.306 ± 0.093	0.61%	1.228 ± 0.110	0.22%	1.395 ± 0.064	0.43%
R_Top10	1.268 ± 0.113	0.24%	1.179 ± 0.087	0.17%	1.249 ± 0.093	0.34%
Classic3	0.139 ± 0.164	0.20%	0.183 ± 0.240	0.15%	0.137 ± 0.173	0.21%

Table 5. Comparison of *Entropy* on test data between constrained and unconstrained clustering

Dataset	Baseline BOW	BOC	Combined	Replaced	Improvement
20NG_Diff3	0.776 ± 0.157	0.741 ± 0.077	**0.827 ± 0.135**	0.792 ± 0.071	6.19%
20NG_Sim3	0.477 ± 0.094	**0.575 ± 0.083**	0.569 ± 0.105	0.518 ± 0.061	**20.6%**
20NG_Multi10	0.830 ± 0.029	0.825 ± 0.038	**0.842 ± 0.031**	0.816 ± 0.043	1.46%
R_Min15Max100	0.967 ± 0.002	0.971 ± 0.003	0.972 ± 0.003	**0.972 ± 0.002**	0.50%
R_Top10	0.885 ± 0.006	0.887 ± 0.011	**0.892 ± 0.009**	0.890 ± 0.007	0.79%
Classic3	0.962 ± 0.077	**0.962 ± 0.068**	0.962 ± 0.1	0.961 ± 0.076	0.00%

Table 6. Comparison of *Rand index* on test data between constrained and unconstrained clustering

achieving higher purity scores. However, their approaches do not consider the relatedness between concepts when creating document representations and clustering.

Using Wikipedia to predict the semantic relatedness between concepts has recently attracted a significant amount of interest. Determining the similarity of two concepts is a highly intelligent task that can conventionally only be performed by humans. Alternatives to the measure from Milne and Witten [15] used in our experiments include WikiRelate! [22], which utilizes the Wikipedia category structure to compute similarity between articles; *explicit semantic analysis* from Gabrilovich and Markovitch [9], where sophisticated content analysis is used; and Wang et al. [25]'s work, which models the relatedness between two Wikipedia concepts as the linear combination of the similarity between three aspects–the content of the article, similarity of outlink categories, and the shortest path distance between two articles in the Wikipedia category structure. As a whole, the diverse research on this venue is attractive, particularly for such tasks like text document clustering or categorization.

Upon representing documents with concepts rather than words, it is natural to think about devising new document similarity metrics based on concepts and their similarities, such as in [24]. However, there are some issues pertaining to this direction. First, computing the pair-wise similarity between concepts will result in a $O(n^2)$ complexity just for computing of document similarity. Moreover, concept-based document similarity is not readily applicable to hybrid document models such as the *Combined* and *Replaced* models.

We are not the first to exploit the anchor text vocabulary and its associated link structure to utilize semantic knowledge residing in Wikipedia. It has been used in various other text mining problems. Mihalcea and Csomai [14] use it in a similar way—to *wikify* a given document by linking terms in the document to the appropriate Wikipedia article. Medelyan [13] use it to find candidate index terms for topic indexing. Both define a set of measures reflecting the salience of a given candidate for the task at hand. In contrast, we use an unsupervised clustering algorithm to find salient concepts for representing documents.

Pair-wise instance level constraints have been reported as effective supervision that improves clustering performance in many different applications [3, 23, 5, 4]. There has been less work on active learning for clustering. Our active learning approach falls in the area of active learning by selective sampling—we select samples from the data at hand to form a query for the oracle. Most active learning algorithms are for supervised learning, where certain objective functions can be formulated on the effectiveness of the active learning algorithm based on the existing category structure [18]. Few active learning algorithms have been proposed for unsupervised learning where class labels are not as readily available beforehand. Basu et al. [3] proposed an active learning algorithm that searches for the two instances that are farthest from each other and pose them to the oracle as the next query. Our approach has a similar motivation—to find the documents that are most likely to be different by analyzing the concepts that appear in them.

8. Conclusions and Future Work

The semantic knowledge in Wikipedia is abundant and valuable. In this paper we utilized Wikipedia and the semantic information therein for text document clustering: to create more informative document representations and to facilitate active learning of pair-wise relations between documents by explicitly analyzing the topic distributions within document groups. Empirical results on three standard document data sets show the effectiveness of the concept-based representation and the improvement obtained by using active learning for guiding clustering.

Our method of exploring the semantic information in Wikipedia for document clustering is only a first step. There are many interesting avenues to explore. The semantic knowledge in Wikipedia can potentially be used in document clustering in four ways: to represent text documents with concepts rather than words, to reduce the feature space dimensionality, to instill the grouping of documents with semantic information about the concepts that appear in them, and to create better representations of the resulting document clusters. We have investigated the first two aspects in this paper, and leave the other two aspects for future work.

Considering the machine-learning-related aspects of the problem, devising new document similarity measures based on concept similarities is an interesting and fundamental problem for document clustering. We will further investigate this direction in future work. Moreover, the supervision employed in our approach is at the instance-level. Recent research on transforming instance-level constraints to have global impact are also of interest, such as the work presented in [11, 26].

References

[1] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley Longman Publishing Co. Inc., New York, 2004.

[2] S. Banerjee, K. Ramanathan, and A. Gupta. Clustering short texts using wikipedia. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, 2007.

[3] S. Basu, A. Banerjee, and R. J. Mooney. Active semi-supervision for pairwise constrained clustering. In *Proceedings of the SIAM International Conference on Data Mining (SDM-2004)*, pages 333–344, 2004.

[4] S. Basu, M. Bilenko, and R. J. Mooney. A probabilistic framework for semi-supervised clustering. In *Proceedings of the Tenth International Conference on Knowledge Discovery and Data Mining (KDD-2004)*, pages 59–68, 2004.

[5] D. Cohn, R. Caruana, and A. McCallum. Semi-supervised clustering with user freedback. Technical report, 2000.

[6] I. S. Dhillon and D. S. Modha. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42:143–175, 2001.

[7] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of second International Conference on Knowledge Discovery and Data Mining*, 1996.

[8] E. Gabrilovich and S. Markovitch. Overcoming the brittleness bottleneck using wikipedia: Enhancing text categorization with encyclopedic knowledge. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI)*, pages 1301–1306, 2006.

[9] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 2007.

[10] A. Hotho, S. Staab, and G. Stumme. Wordnet improves text document clustering. In *Proceeding of the SIGIR 2003 Semantic Web Workshop*, 2003.

[11] D. Klein, S. D. Kamvar, and C. D. Manning. From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In *Proceedings of the Nineteenth International Conference on Machine Learning*, 2002.

[12] K. Lang. Newsweeder: Learning to filter news. In *ICML'95*, pages 331–339, 1995.

[13] O. Medelyan, I. H. Witten, and D. Milne. Topic indexing with wikipedia. *To appear in the AAAI 2008 Workshop on Wikipedia and Artificial Intelligence (WIKIAI 2008)*, 2008.

[14] R. Mihalcea and A. Csomai. Wikify! linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, 2007.

[15] D. Milne and I. H. Witten. An effective, low-cost measure of semantic relatedness obtained from wikipedia links. *To appear in the AAAI 2008 Workshop on Wikipedia and Artificial Intelligence (WIKIAI 2008)*, 2008.

[16] D. Milne and I. H. Witten. Learning to link with wikipedia. In *To appear in CIKM 2008*, 2008.

[17] Z. Minier, Z. Bodó, and L. Csató. Wikipedia-based kernels for text categorization. In *Proceedings of the Ninth International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, pages 157–164, 2007.

[18] I. A. Muslea. *Active Learning with Multiple Views*. Phd thesis, 2002.

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

[20] D. R. Recupero. A new unsupervised method for document clustering by using wordnet lexical and conceptual relations. *Information Retrieval*, (10):563–479, 2007.

[21] Reuters. *Reuters-21578 text categorization test collection, Distribution 1.0.* 1997.

[22] M. Strube and S. P. Ponzetto. Wikirelate! computing semantic relatedness using wikipedia. In *Proceedings of AAAI*, pages 1419–1424, 2006.

[23] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. Constrained k-means clustering with background knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 557–584, 2001.

[24] J. Z. Wang and W. Taylor. Concept forest: A new ontology-assisted text document similarity measurement method. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, pages 395–401, 2007.

[25] P. Wang, J. Hu, H.-J. Zeng, L. Chen, and Z. Chen. Improving text classification by using encyclopedia knowledge. In *Proceedings of the Seventh IEEE International Conference on Data Mining*, pages 332–341, 2007.

[26] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. pages 521–528, 2003.