Similarity Measures for Text Document Clustering

Adaikalam S

Similarity Measures for Text Document Clustering

Anna Huang
Department of Computer Science
The University of Waikato, Hamilton, New Zealand
lh92@waikato.ac.nz

ABSTRACT

Clustering is a useful technique that organizes a large quantity of unordered text documents into a small number of meaningful and coherent clusters, thereby providing a basis for intuitive and informative navigation and browsing mechanisms. Partitional clustering algorithms have been recognized to be more suitable as opposed to the hierarchical clustering schemes for processing large datasets. A wide variety of distance functions and similarity measures have been used for clustering, such as squared Euclidean distance, cosine similarity, and relative entropy.

In this paper, we compare and analyze the effectiveness of these measures in partitional clustering for text document datasets. Our experiments utilize the standard K-means algorithm and we report results on seven text document datasets and five distance/similarity measures that have been most commonly used in text clustering.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Clustering; I.5.3 [**Clustering**]: Similarity measures

General Terms

Performance

Keywords

Similarity measures, partitional clustering, text clustering

1. INTRODUCTION

We are facing an ever increasing volume of text documents. The abundant texts flowing over the Internet, huge collections of documents in digital libraries and repositories, and digitized personal information such as blog articles and emails are piling up quickly everyday. These have brought challenges for the effective and efficient organization of text documents.

Clustering in general is an important and useful technique that automatically organizes a collection with a substantial number of data objects into a much smaller number of coherent groups [8, 20]. In the particular scenario of text documents, clustering has proven to be an effective approach for quite some time—and an interesting research problem as well. It is becoming even more interesting and demanding with the development of the World Wide Web and the evolution of Web 2.0. For example, results returned by search engines are clustered to help users quickly identify and focus on the relevant set of results. Customer comments are clustered in many online stores, such as Amazon.com, to provide collaborative recommendations. In collaborative bookmarking or tagging, clusters of users that share certain traits are identified by their annotations.

Text document clustering groups similar documents that to form a coherent cluster, while documents that are different have separated apart into different clusters. However, the definition of a pair of documents being similar or different is not always clear and normally varies with the actual problem setting. For example, when clustering research papers, two documents are regarded as similar if they share similar thematic topics. When clustering is employed on web sites, we are usually more interested in clustering the component pages according to the type of information that is presented in the page. For instance, when dealing with universities' web sites, we may want to separate professors' home pages from students' home pages, and pages for courses from pages for research projects. This kind of clustering can benefit further analysis and utilize of the dataset such as information retrieval and information extraction, by grouping similar types of information sources together.

Accurate clustering requires a precise definition of the closeness between a pair of objects, in terms of either the pair-wised *similarity* or *distance*. A variety of similarity or distance measures have been proposed and widely applied, such as cosine similarity and the Jaccard correlation coefficient. Meanwhile, similarity is often conceived in terms of *dissimilarity* or *distance* as well [15]. Measures such as Euclidean distance and relative entropy have been applied in clustering to calculate the pair-wise distances.

Given the diversity of similarity and distance measures available, their effectiveness in text document clustering is still not clear. Although Strehl et al. compared the effectiveness of a number of measures [17], our experiments extended their work by including more measures and experimental datasets, such as the averaged Kullback-Leibler divergence, which has

This paper was published in the proceedings of the New Zealand Computer Science Research Student Conference 2008. Copyright is held by the author/owner(s).

NZCSRSC 2008, April 2008, Christchurch, New Zealand.

shown its effectiveness in clustering text and attracted considerable research interest recently. More specifically, we evaluated five measures with empirical experiments: Euclidean distance, cosine similarity, Jaccard coefficient, Pearson correlation coefficient and averaged Kullback-Leibler divergence. Each of the measures are further discussed in Section 3.

In order to come up with a sound conclusion we have performed an empirical evaluation with seven data sets that each have different characteristics. They contain such things as newspaper articles, newsgroup posts, research papers, and web pages (see Table 1). They all come with a set of categorizing labels, with one category attached to each document. These pre-assigned labels are very useful for cluster validation; we use them to measure the consistency between the resulting clusters and the categories created by human experts. We use two measures to evaluate the overall quality of clustering solutions—*purity* and *entropy*, which are commonly used in clustering [23, 22]. Section 5.2 further explains the evaluation approaches. However, manually assigned labels are normally not available in clustering, and in these cases other measure such as within-cluster distances and between-cluster distances [13] can be used for evaluation. These are not used in this paper because all the datasets already have labels.

This paper is organized as follows. The next section describes the document representation used in the experiments. Section 3 discusses the similarity measures and their semantics. Section 4 presents the K-means clustering algorithm and Section 5 explains experiment settings, evaluation approaches, results and analysis. We point to some related work in Section 6 and Section 7 concludes and discusses future work.

2. DOCUMENT REPRESENTATION

There are several ways to model a text document. For example, it can be represented as a bag of words, where words are assumed to appear independently and the order is immaterial. The bag of word model is widely used in information retrieval and text mining [21]. Words are counted in the bag, which differs from the mathematical definition of *set*. Each word corresponds to a dimension in the resulting data space and each document then becomes a vector consisting of non-negative values on each dimension. Here we use the frequency of each term as its weight, which means terms that appear more frequently are more important and descriptive for the document.

Let $D = \{d_1, \dots, d_n\}$ be a set of documents and $T = \{t_1, \dots, t_m\}$ the set of distinct terms occurring in D. We discuss more precisely what we mean by "terms" below: for the moment just assume they are words. A document is then represented as a m-dimensional vector $\overrightarrow{t_d}$. Let $tf(d,t)$ denote the frequency of term $t \in T$ in document $d \in D$. Then the vector representation of a document d is

$$\overrightarrow{t_d} = (tf(d,t_1), \dots, tf(d,t_m))$$

Although more frequent words are assumed to be more important as mentioned above, this is not usually the case in practice. For example, words like *a* and *the* are probably the most frequent words that appear in English text, but neither are descriptive nor important for the document's subject. In fact, more complicated strategies such as the *tfidf* weighting scheme as described below is normally used instead.

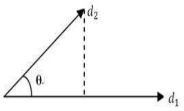

Figure 1: Angle between documents

With documents presented as vectors, we measure the degree of similarity of two documents as the correlation between their corresponding vectors, which can be further quantified as the cosine of the angle between the two vectors. Figure 1 shows the angle in two-dimensional space but in practice the document space usually has tens and thousands of dimensions. Some useful properties of the cosine measure are discussed in Section 3.3.

Terms are basically words. But we applied several standard transformations on the basic term vector representation. First, we removed *stop words*. There are words that are non-descriptive for the topic of a document, such as *a*, *and*, *are* and *do*. Following common practices, we used the one implemented in the Weka machine learning workbench, which contains 527 stop words.

Second, words were stemmed using Porter's suffix-stripping algorithm [14], so that words with different endings will be mapped into a single word. For example *production*, *produce*, *produces* and *product* will be mapped to the stem *produc*. The underlying assumption is that different morphological variations of words with the same root/stem are thematically similar and should be treated as a single word.

Third, we considered the effect of including infrequent terms in the document representation on the overall clustering performance and decided to discard words that appear with less than a given threshold frequency. The rationale by discarding infrequent terms is that in many cases they are not very descriptive about the document's subject and make little contribution to the similarity between two documents. Meanwhile, including rare terms can also introduce noise into the clustering process and make similarity computation more expensive. Consequently, we select the top 2000 words ranked by their weights and use them in our experiments.

In the clustering process, we also need to compare the dissimilarity/similarity between two clusters or between a cluster and an object. In hierarchical clustering this is normally computed as the *complete-link*, *single-link* or *average-link* distance [8]. However, in partitional clustering algorithms, a cluster is usually represented with a centroid object. For example, in the K-means algorithm the centroid of a cluster is the average of all the objects in the cluster—that is, the centroid's value in each dimension is the arithmetic mean of that dimension over all the objects in the cluster. Let C be a set of documents. Its centroid is defined as

$$\overrightarrow{t_C} = \frac{1}{|C|} \sum_{\overrightarrow{t_d} \in C} \overrightarrow{t_d},$$

which is the mean value of all term vectors in the set. Moreover, we normalize the vectors to a unified length to avoid long documents dominating the cluster.

As mentioned previously, the most frequent terms are not necessarily the most informative ones. On the contrary, terms that appear frequently in a small number of documents but rarely in the other documents tend to be more relevant and specific for that particular group of documents, and therefore more useful for finding similar documents. In order to capture these terms and reflect their importance, we transform the basic term frequencies $tf(d, t)$ into the $tfidf$ (term frequency and inversed document frequency) weighting scheme. $Tfidf$ weighs the frequency of a term t in a document d with a factor that discounts its importance with its appearances in the whole document collection, which is defined as:

$$tfidf(d,t) = tf(d,t) \times log(\frac{|D|}{df(t)}).$$

Here $df(t)$ is the number of documents in which term t appears. In subsequent experiments we use the $tfidf$ value instead of the absolute term frequency of each term to build term vectors. To generalize, we use $w_{t,d}$ to denote the weight of term t in document d in the following sections.

3. SIMILARITY MEASURES

Before clustering, a similarity/distance measure must be determined. The measure reflects the degree of closeness or separation of the target objects and should correspond to the characteristics that are believed to distinguish the clusters embedded in the data. In many cases, these characteristics are dependent on the data or the problem context at hand, and there is no measure that is universally best for all kinds of clustering problems.

Moreover, choosing an appropriate similarity measure is also crucial for cluster analysis, especially for a particular type of clustering algorithms. For example, the density-based clustering algorithms, such as DBScan [4], rely heavily on the similarity computation. Density-based clustering finds clusters as dense areas in the data set, and the density of a given point is in turn estimated as the closeness of the corresponding data object to its neighboring objects. Recalling that closeness is quantified as the distance/similarity value, we can see that large number of distance/similarity computations are required for finding dense areas and estimate cluster assignment of new data objects. Therefore, understanding the effectiveness of different measures is of great importance in helping to choose the best one.

In general, similarity/distance measures map the distance or similarity between the symbolic description of two objects into a single numeric value, which depends on two factors—the properties of the two objects and the measure itself. In order to make the results of this study comparable to previous research, we include all the measures that were tested in [17] and add another one—the averaged Kullback-Leibler divergence. These five measures are discussed below. Different measure not only results in different final partitions, but also imposes different requirements for the same clustering algorithm, as we will see in Section 4.

3.1 Metric

Not every distance measure is a metric. To qualify as a metric, a measure d must satisfy the following four conditions. Let x and y be any two objects in a set and $d(x, y)$ be the distance between x and y.

1. The distance between any two points must be non-negative, that is, $d(x, y) \geq 0$.

2. The distance between two objects must be zero if and only if the two objects are identical, that is, $d(x, y) = 0$ if and only if $x = y$.

3. Distance must be symmetric, that is, distance from x to y is the same as the distance from y to x, ie. $d(x, y) = d(y, x)$.

4. The measure must satisfy the triangle inequality, which is $d(x, z) \leq d(x, y) + d(y, z)$.

3.2 Euclidean Distance

Euclidean distance is a standard metric for geometrical problems. It is the ordinary distance between two points and can be easily measured with a ruler in two- or three-dimensional space. Euclidean distance is widely used in clustering problems, including clustering text. It satisfies all the above four conditions and therefore is a true metric. It is also the default distance measure used with the K-means algorithm.

Measuring distance between text documents, given two documents d_a and d_b represented by their term vectors $\vec{t_a}$ and $\vec{t_b}$ respectively, the Euclidean distance of the two documents is defined as

$$D_E(\vec{t_a}, \vec{t_b}) = (\sum_{t=1}^{m} |w_{t,a} - w_{t,b}|^2)^{1/2},$$

where the term set is $T = \{t_1, \ldots, t_m\}$. As mentioned previously, we use the $tfidf$ value as term weights, that is $w_{t,a} = tfidf(d_a, t)$.

3.3 Cosine Similarity

When documents are represented as term vectors, the similarity of two documents corresponds to the correlation between the vectors. This is quantified as the cosine of the angle between vectors, that is, the so-called cosine similarity. Cosine similarity is one of the most popular similarity measure applied to text documents, such as in numerous information retrieval applications [21] and clustering too [9].

Given two documents $\vec{t_a}$ and $\vec{t_b}$, their cosine similarity is

$$SIM_C(\vec{t_a}, \vec{t_b}) = \frac{\vec{t_a} \cdot \vec{t_b}}{|\vec{t_a}| \times |\vec{t_b}|},$$

where $\vec{t_a}$ and $\vec{t_b}$ are m-dimensional vectors over the term set $T = \{t_1, \ldots, t_m\}$. Each dimension represents a term with its weight in the document, which is non-negative. As a result, the cosine similarity is non-negative and bounded between [0,1].

An important property of the cosine similarity is its independence of document length. For example, combining two identical copies of a document d to get a new pseudo document d', the cosine similarity between d and d' is 1, which means that these two documents are regarded to be identical. Meanwhile, given another document l, d and d' will have the same similarity value to l, that is, $sim(\overrightarrow{t_d}, \overrightarrow{t_l}) = sim(\overrightarrow{t_{d'}}, \overrightarrow{t_l})$. In other words, documents with the same composition but different totals will be treated identically. Strictly speaking, this does not satisfy the second condition of a metric, because after all the combination of two copies is a different object from the original document. However, in practice, when the term vectors are normalized to a unit length such as 1, and in this case the representation of d and d' is the same.

3.4 Jaccard Coefficient

The Jaccard coefficient, which is sometimes referred to as the Tanimoto coefficient, measures similarity as the intersection divided by the union of the objects. For text document, the Jaccard coefficient compares the sum weight of shared terms to the sum weight of terms that are present in either of the two document but are not the shared terms. The formal definition is:

$$SIM_J(\overrightarrow{t_a}, \overrightarrow{t_b}) = \frac{\overrightarrow{t_a} \cdot \overrightarrow{t_b}}{|\overrightarrow{t_a}|^2 + |\overrightarrow{t_b}|^2 - \overrightarrow{t_a} \cdot \overrightarrow{t_b}}.$$

The Jaccard coefficient is a similarity measure and ranges between 0 and 1. It is 1 when the $\overrightarrow{t_a} = \overrightarrow{t_b}$ and 0 when $\overrightarrow{t_a}$ and $\overrightarrow{t_b}$ are disjoint, where 1 means the two objects are the same and 0 means they are completely different. The corresponding distance measure is $D_J = 1 - SIM_J$ and we will use D_J instead in subsequent experiments.

3.5 Pearson Correlation Coefficient

Pearson's correlation coefficient is another measure of the extent to which two vectors are related. There are different forms of the Pearson correlation coefficient formula. Given the term set $T = \{t_1, \ldots, t_m\}$, a commonly used form is

$$SIM_P(\overrightarrow{t_a}, \overrightarrow{t_b}) = \frac{m \sum_{t=1}^{m} w_{t,a} \times w_{t,b} - TF_a \times TF_b}{\sqrt{[m \sum_{t=1}^{m} w_{t,a}^2 - TF_a^2][m \sum_{t=1}^{m} w_{t,b}^2 - TF_b^2]}}$$

where $TF_a = \sum_{t=1}^{m} w_{t,a}$ and $TF_b = \sum_{t=1}^{m} w_{t,b}$.

This is also a similarity measure. However, unlike the other measures, it ranges from $+1$ to -1 and it is 1 when $\overrightarrow{t_a} = \overrightarrow{t_b}$. In subsequent experiments we use the corresponding distance measure, which is $D_P = 1 - SIM_P$ when $SIM_P \geq 0$ and $D_P = |SIM_P|$ when $SIM_P < 0$.

3.6 Averaged Kullback-Leibler Divergence

In information theory based clustering, a document is considered as a probability distribution of terms. The similarity of two documents is measured as the distance between the two corresponding probability distributions. The *Kullback-Leibler divergence* (KL divergence), also called the relative entropy, is a widely applied measure for evaluating the differences between two probability distributions.

Given two distributions P and Q, the KL divergence from distribution P to distribution Q is defined as

$$D_{KL}(P\|Q) = Plog(\frac{P}{Q}).$$

In the document scenario, the divergence between two distribution of words is:

$$D_{KL}(\overrightarrow{t_a}\|\overrightarrow{t_b}) = \sum_{t=1}^{m} w_{t,a} \times log(\frac{w_{t,a}}{w_{t,b}}).$$

However, unlike the previous measures, the KL divergence is not symmetric, ie. $D_{KL}(P\|Q) \neq D_{KL}(Q\|P)$. Therefore it is not a true metric. As a result, we use the averaged KL divergence instead, which is defined as

$$D_{AvgKL}(P\|Q) = \pi_1 D_{KL}(P\|M) + \pi_2 D_{KL}(Q\|M),$$

where $\pi_1 = \frac{P}{P+Q}$, $\pi_2 = \frac{Q}{P+Q}$ and $M = \pi_1 P + \pi_2 Q$. For documents, the averaged KL divergence can be computed with the following formula:

$$D_{AvgKL}(\overrightarrow{t_a}\|\overrightarrow{t_b}) = \sum_{t=1}^{m} \left(\pi_1 \times D(w_{t,a}\|w_t) + \pi_2 \times D(w_{t,b}\|w_t)\right),$$

where
$\pi_1 = \frac{w_{t,a}}{w_{t,a}+w_{t,b}}$, $\pi_2 = \frac{w_{t,b}}{w_{t,a}+w_{t,b}}$, and $w_t = \pi_1 \times w_{t,a} + \pi_2 \times w_{t,b}$.

The average weighting between two vectors ensures symmetry, that is, the divergence from document i to document j is the same as the divergence from document j to document i. The averaged KL divergence has recently been applied to clustering text documents, such as in the family of the Information Bottleneck clustering algorithms [18], to good effect.

4. CLUSTERING ALGORITHM

For all subsequent experiments, the standard K-means algorithm is chosen as the clustering algorithm. This is an iterative partitional clustering process that aims to minimize the least squares error criterion [15]. As mentioned previously, partitional clustering algorithms have been recognized to be better suited for handling large document datasets than hierarchical ones, due to their relatively low computational requirements [16, 9, 3].

The standard K-means algorithm works as follows. Given a set of data objects D and a pre-specified number of clusters k, k data objects are randomly selected to initialize k clusters, each one being the centroid of a cluster. The remaining objects are then assigned to the cluster represented by the nearest or most similar centroid. Next, new centroids are re-computed for each cluster and in turn all documents are re-assigned based on the new centroids. This step iterates until a converged and fixed solution is reached, where all data objects remain in the same cluster after an update of centroids.

The generated clustering solutions are locally optimal for the given data set and the initial seeds. Different choices of initial seed sets can result in very different final partitions. Methods for finding good starting points have been proposed [1]. However, we will use the basic K-means algorithm because optimizing the clustering is not the focus of this paper.

The K-means algorithm works with distance measures which basically aims to minimize the within-cluster distances. Therefore, similarity measures do not directly fit into the algorithm, because smaller values indicate dissimilarity. The Euclidean distance and the averaged KL divergence are distance measures, while the cosine similarity, Jaccard coefficient and Pearson coefficient are similarity measures. We apply a simple transformation to convert the similarity measure to distance values. Because both cosine similarity and

Jaccard coefficient are bounded in $[0, 1]$ and monotonic, we take $D = 1 - SIM$ as the corresponding distance value. For Pearson coefficient, which ranges from -1 to $+1$, we take $D = 1 - SIM$ when $SIM \geq 0$ and $D = |SIM|$ when $SIM < 0$.

5. EXPERIMENT

It is very difficult to conduct a systematic study comparing the impact of similarity metrics on cluster quality, because objectively evaluating cluster quality is difficult in itself. In practice, manually assigned category labels are usually used as a baseline criteria for evaluating clusters. As a result, the clusters, which are generated in an unsupervised way, are compared to the pre-defined category structure, which is normally created by human experts. This kind of evaluation assumes that the objective of clustering is to replicate human thinking, so a clustering solution is good if the clusters are consistent with the manually created categories. However, in practice datasets often come without any manually created categories, and this is the exact point where clustering can help. In this case, measures like cluster coherence in terms of the within-cluster distances and the well-separateness between clusters in terms of the between-cluster distances can be used for evaluation [13]. In order to make the result of this investigation comparable to previous researches, we choose datasets that have been commonly used for evaluating clustering as the experiment datasets. All of these datasets have appropriate pre-assigned category labels. The rest of this section first describes the characteristics of the datasets, then explains the evaluation measures, and finally presents and analyzes the experiment results.

5.1 Datasets

The seven data sets in Table 1 were chosen for the empirical experiment. These have been widely used for evaluating feature selection techniques, classification and clustering. Except for the *20news* and *webkb* datasets, all from the CLUTO package. [1] These datasets differ in terms of document type, number of categories, average category size, and subjects. In order to ensure diversity, the datasets are from different sources, some containing newspaper articles, some containing newsgroup posts, some being web pages and the remaining being academic papers.

The characteristics and sources of these datasets are summarized in Table 1. The smallest contains 1504 documents and the largest contained 8282 documents. The number of classes in each dataset varies from 4 to 20. As mentioned previously, we removed stop words and applied stemming as described in Section 2, and only the top 2000 words are selected.

More specifically, the *20news* dataset contained newsgroup articles from 20 newsgroups on a variety of topics including politics, computers, etc. The *classic* dataset contains abstracts of scientific papers from four sources: CACM, CISI, CRAN and MED. It has been widely used to evaluate information retrieval systems. The *hitech* dataset consists of San Jose Mercury newspaper articles on six topics—computers, electronics, health, medical, research, and technology, and was part of the TREC collection [19]. The *tr41* data set is also derived from the TREC-5, TREC-6 and TREC-7 collections. The *wap* and the *webkb* datasets both consists of web pages. The *wap* dataset is from the *WebAce* project and contains web pages from the Yahoo! Subject hierarchy; the *webkb* was from the *Web Knowledge Base* project and contains web pages from several universities about courses, students, staffs, departments, projects and the like. The *re0* dataset contains newspaper articles and has been widely used for evaluating clustering algorithms.

For each data set, we experimented with different similarity measures and we had $5 \times 7 = 35$ experiments in total. Moreover, each experiment was run 10 times and the results are the averaged value over 10 runs. Each run has different initial seed sets.

5.2 Evaluation

For each of the above datasets, we obtained a clustering result from the K-means algorithm. The number of clusters is set as the same with the number of pre-assigned categories in the data set. The quality of a clustering result was evaluated using two evaluation measures—*purity* and *entropy*, which are widely used to evaluate the performance of unsupervised learning algorithms [23, 22].

To begin with, each cluster is labeled with the majority category that appears in that cluster. Moreover, if a category label has been assigned to a cluster, it still can be assigned to other clusters if it is the dominant category in that cluster. Based on the cluster labels, the purity and entropy measures are computed as follows.

The *purity* measure evaluates the coherence of a cluster, that is, the degree to which a cluster contains documents from a single category. Given a particular cluster C_i of size n_i, the purity of C_i is formally defined as

$$P(C_i) = \frac{1}{n_i} \max_h (n_i^h)$$

where $max_h(n_i^h)$ is the number of documents that are from the dominant category in cluster C_i and n_i^h represents the number of documents from cluster C_i assigned to category h.

Purity can be interpreted as the classification rate under the assumption that all samples of the cluster are predicted to be members of the actual dominant class for the cluster. For an ideal cluster, which only contains documents from a single category, its purity value is 1. In general, the higher the purity value, the better the quality of the cluster is.

The *entropy* measure evaluates the distribution of categories in a given cluster. The entropy of a cluster C_i with size n_i is defined to be

$$E(C_i) = -\frac{1}{\log c} \sum_{h=1}^{k} \frac{n_i^h}{n_i} \log(\frac{n_i^h}{n_i})$$

where c is the total number of categories in the data set and n_i^h is the number of documents from the hth class that were assigned to cluster C_i.

The entropy measure is more comprehensive than purity because rather than just considering the number of objects in

[1] http://glaros.dtc.umn.edu/gkhome/fetch/sw/cluto/datasets.tar.gz

Table 1: Summary of datasets to evaluate the various similarity measures

Data	Documents	Classes	Terms	Average Class Size	Source	Description
20news	18828	20	28553	1217	20news-18828	Newsgroup posts
classic	7089	4	12009	1774	CACM/CISI/CRANFIELD/MEDLINE	Academic papers
hitech	2301	6	13170	384	San Jose Mercury (TREC, TIPSTER Vol. 3)	Newspaper articles
re0	1504	13	2886	131	Reuters-21578 ([10])	Newsgroup posts
tr41	878	10	7454	88	TREC5 and TREC6 (TREC 1999)	Newspaper articles
wap	1560	20	8460	78	WebACE ([6])	Web pages
webkb	8282	7	20682	1050	Web Knowledge Base ([2])	Web pages

and not in the dominant category, it considers the overall distribution of all the categories in a given cluster. Contrary to the purity measure, for an ideal cluster with documents from only a single category, the entropy of the cluster will be 0. In general, the smaller the entropy value, the better the quality of the cluster is.

Moreover, the averaged entropy of the overall solution is defined to be the weighted sum of the individual entropy value of each cluster, that is,

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

In many cases the two measures seem very similar, however, their meanings actually differ. For example, the two matrix below represents two clustering solutions with each row being a cluster and each column being a pre-defined category, so $v(i, j)$ indicates the number of documents in cluster i that are from category j. Intuitively, the solution as represented with the first matrix is more balanced and therefore better than the other solution as represented by the second matrix. However, the two solutions have the same purity value, which is 0.5. Meanwhile, the entropy for each individual cluster and the overall solution is 0.40 for the first matrix and 0.57 for the second matrix. This indicates that the first solution is better than the other, which is consistent with our observation.

$$\begin{bmatrix} 3 & 1 & 1 & 1 \\ 1 & 3 & 1 & 1 \\ 1 & 1 & 3 & 1 \\ 1 & 1 & 1 & 3 \end{bmatrix} \begin{bmatrix} 3 & 0 & 2 & 1 \\ 2 & 3 & 1 & 0 \\ 2 & 0 & 3 & 1 \\ 1 & 0 & 2 & 3 \end{bmatrix}$$

5.3 Results

Table 2 and Table 4 show the average purity and entropy result for each similarity/distance measure on the seven datasets.

As shown in Table 2, Euclidean distance performs worst while the performance of the other four measures are quite similar. On average, the Jaccard and Pearson measures are slightly better in generating more coherent clusters, which means the clusters have higher purity scores. The best result is achieved with the *classic* dataset. A closer look at this dataset found that the categories in this dataset is well separated. Table 3 shows one partition as generated by the KLD measure, which has the lowest purity score on this dataset (except for the Euclidean distance measure). The shape of the pre-defined category structure is still clear in the partition. Meanwhile, the Jaccard coefficient and the averaged KL divergence outperform with considerable difference on the *wap* dataset and the *webkb* dataset respectively.

Table 2: Purity Results

Data	Euclidean	Cosine	Jaccard	Pearson	KLD
20news	0.1	**0.5**	**0.5**	**0.5**	0.38
classic	0.56	0.85	**0.98**	0.85	0.84
hitech	0.29	0.54	0.51	**0.56**	0.53
re0	0.53	**0.78**	0.75	**0.78**	0.77
tr41	0.71	0.71	0.72	**0.78**	0.64
wap	0.32	0.62	**0.63**	0.61	0.61
webkb	0.42	0.68	0.57	0.67	**0.75**

Table 3: Clustering Results from the KL Divergence measure

Clusters	CRAN	MED	CACM	CISI	Label
cluster[1]	1	**823**	40	0	MED(823)
cluster[2]	14	33	661	**1444**	CISI(1444)
cluster[3]	9	171	**2375**	15	CACM(2375)
cluster[4]	**1374**	6	127	1	CRAM(1374)

The overall entropy value for each measure is shown in Table 4. As shown in table 4, the overall purity values are close and sometimes with only 1% difference. However, there is still some results worth noticing. For example, similar as above, the Euclidean distance is again proved to be an ineffective metric for modeling the similarity between documents. The averaged KL divergence and Pearson coefficient tend to outperform the cosine similarity and the Jaccard coefficient. Except for the *classic* dataset, either the KL divergence measure or the Pearson coefficient has the best result on a given dataset. This means that these two measures are more effective in finding more balanced cluster structures.

Considering that the above results all seem very close, in

Table 4: Entropy results

Data	Euclidean	Cosine	Jaccard	Pearson	KLD
20news	0.95	**0.49**	**0.51**	**0.49**	0.54
classic	0.78	0.29	**0.06**	0.27	0.3
hitech	0.92	0.64	0.68	0.65	**0.63**
re0	0.6	0.27	0.33	0.26	**0.25**
tr41	0.62	0.33	0.34	**0.3**	0.38
wap	0.75	**0.39**	0.4	**0.39**	0.4
webkb	0.93	0.6	0.74	0.61	**0.51**

order to test the statistical significance between different solutions, we used the t-test to compare the solutions. First, each solution is represented with a set of clusters and each individual cluster is in turn represented by its centroid vector. Meanwhile, the original dataset is represented as a set of pre-defined categories, and each category is also represented with its centroid. Then both the original dataset and the generated clustering solution are transformed into matrices, with each row being a centroid object and each column is a distinct term from the term set. Finally, the two matrices are tested with the t-test function in the *R* package.[2] The t-test result of any given two matrix shows that there is a true difference between any two matrices, because all the *p-value*s are less than 0.9.

6. RELATED WORK

The most similar work to this paper is [17], which compares four similarity measures on a collection of Yahoo! news pages. The present study differs in two aspects. First, we extended the experiments by including the averaged KL divergence. Our results broadly agree with Strehl et al's [17]. We both found that the performance of the cosine similarity, Jaccard correlation and Pearson's coefficient are very close, and are significantly better than the Euclidean distance measure. In addition, we found that the KL divergence measure is comparable and in some cases better than the others. Second, we also experimented with other types of data sets in addition to the web page documents.

As for the averaged Kullback-Leibler divergence, Lin gave a clear explanation in [11]. This measure was more frequently used to assess the similarity between words, especially for such applications as word sense disambiguation. It was not until recently that this measure has been utilized for document clustering. Information theoretic clustering algorithms such as the Information Bottleneck method [18] rely on this measure and have shown considerable improvement in overall performance.

Meanwhile, enhanced representation of documents has been a promising direction recently, especially the incorporation of semantic information and taking account of the semantic relatedness between documents. A number of researchers have reported results on these aspects. For example, Hotho et al. propose to extend the conventional bag of word representation with relevant terms from WordNet [7]. Experiments on document clustering task show the effectiveness of the extended representation. Moreover, the effectiveness of different representation strategies also depends on the type of task at hand. For example, when clustering journalistic text, proper names have been found to be a more appropriate representation for the text content [5]. This investigation differs from these strategies in that we use only the basic bag of words representation. However, combining these extended representation is likely to improve performance and this is planned for future work.

7. CONCLUSIONS

To conclude, this investigation found that except for the Euclidean distance measure, the other measures have comparable effectiveness for the partitional text document clustering task. Pearson correlation coefficient and the averaged KLD divergence measures are slightly better in that their resulting clustering solutions are more balanced and has a closer match with the manually created category structure. Meanwhile, the Jaccard and Pearson coefficient measures find more coherent clusters. Despite of the above differences, these measures' overall performance is similar. Considering the type of cluster analysis involved in this study, which is partitional and require a similarity or distance measure, we can see that there are three components that affect the final results—representation of the objects, distance or similarity measures, and the clustering algorithm itself. This lead us to two directions for future work as follows.

First, I plan to investigate the impact of using different document representation on clustering performance, and combine the different representations with similarity measures. In particular, I will use Wikipedia as a background knowledge base, and enrich the document representation by adding related terms identified by the relationships between terms in Wikipedia. Wikipedia provides rich semantic relations between words and phrases, with a extensive coverage [12]. This will also help to alleviate the problems with the bag of word document model, that words must co-occur literally and semantic relationships between words are neglected.

Meanwhile, I plan to investigate the effectiveness of these similarity measures with a multi-view clustering approach. In many cases we can view a given document from more than one perspective. For example, web pages intuitively provide at least three views—the content text appear in the web page itself, the anchor text of the outgoing links that are embedded in the page and the anchor texts from incoming links. Conventional clustering normally combines these different views (if they are taken into account at all) into a single mixed representation and use it for clustering. We assume that by dividing the mixed representation up and using the different aspects individually can provide relevant information that is well separated according to its characteristics, therefore benefiting subsequent clustering.

8. REFERENCES

[1] D. Arthur and S. Vassilvitskii. k-means++ the advantages of careful seeding. In *Symposium on Discrete Algorithms*, 2007.

[2] M. Craven, D. DiPasquo, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery. Learning to extract symbolic knowledge from the world wide web. In *AAAI-98*, 1998.

[3] D. R. Cutting, J. O. Pedersen, D. R. Karger, and J. W. Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proceedings of the ACM SIGIR*, 1992.

[4] 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 2nd International Conference on KDD*, 1996.

[5] N. Friburger and D. Maurel. Textual similarity based on proper names. In *Proceedings of Workshop on Mathematical Formal Methods in Information Retrieval at th 25th ACM SIGIR Conference*, 2002.

[6] E. Han, D. Boley, M. Gini, R. Gross, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, and J. Moore. Webace: A web agent for document categorization and

[2] http://www.r-project.org/

exploartion. In *Proceedings of the 2nd International Conference on Autonomous Agents.*, 1998.

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

[8] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys (CSUR)*, 31(3):264–323, 1999.

[9] B. Larsen and C. Aone. Fast and effective text mining using linear-time document clustering. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1999.

[10] D. D. Lewis. Reuters-21578 text categorization test collection distribution 1.0. *http://www.research.att.com/ lewis*, 1999.

[11] J. Lin. Divergence measures based on the shannon entropy. *IEEE Transaction on Information Theory*, 37(1):145–151, 1991.

[12] D. Milne, O. Medelyan, and I. H. Witten. Mining domain-specific thesauri from wikipedia: A case study. In *Proc. of the International Conference on Web Intelligence (IEEE/WIC/ACM WI'2006)*, 2006.

[13] J. M. Neuhaus and J. D. Kalbfleisch. Between- and within-cluster covariate effects in the analysis of clustered data. *Biometrics*, 54(2):638–645, Jun. 1998.

[14] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[15] G. Salton. *Automatic Text Processing.* Addison-Wesley, New York, 1989.

[16] M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. In *KDD Workshop on Text Mining*, 2000.

[17] A. Strehl, J. Ghosh, and R. Mooney. Impact of similarity measures on web-page clustering. In *AAAI-2000: Workshop on Artificial Intelligence for Web Search*, July 2000.

[18] N. Z. Tishby, F. Pereira, and W. Bialek. The information bottleneck method. In *Proceedings of the 37th Allerton Conference on Communication, Control and Computing*, 1999.

[19] E. Voorhees and D. Harman. Overview of the fifth text retrieval conference (trec-5). In *Proc. of the Fifth Text REtrieval Conference (TREC-5)*, 1998.

[20] P. Willett. Recent trends in hierarchic document clustering: a critical review. *Information Processing and Management: an International Journal*, 24(5):577–597, 1988.

[21] R. B. Yates and B. R. Neto. *Modern Information Retrieval.* ADDISON-WESLEY, New York, 1999.

[22] Y. Zhao and G. Karypis. Evaluation of hierarchical clustering algorithms for document datasets. In *Proceedings of the International Conference on Information and Knowledge Management*, 2002.

[23] Y. Zhao and G. Karypis. Empirical and theoretical comparisons of selected criterion functions for document clustering. *Machine Learning*, 55(3), 2004.