Term-Based Clustering and Summarization of Web Page Collections

Yongzheng Zhang, Nur Zincir-Heywood, and Evangelos Milios

Faculty of Computer Science, Dalhousie University
6050 University Ave., Halifax, NS, Canada B3H 1W5
{yongzhen,zincir,eem}@cs.dal.ca

Abstract. Effectively summarizing Web page collections becomes more and more critical as the amount of information continues to grow on the World Wide Web. A concise and meaningful summary of a Web page collection, which is generated automatically, can help Web users understand the essential topics and main contents covered in the collection quickly without spending much browsing time. However, automatically generating coherent summaries as good as human-authored summaries is a challenging task since Web page collections often contain diverse topics and contents. This research aims towards clustering of Web page collections using automatically extracted topical terms, and automatic summarization of the resulting clusters. We experiment with word- and term-based representations of Web documents and demonstrate that term-based clustering significantly outperforms word-based clustering with much lower dimensionality. The summaries of computed clusters are informative and meaningful, which indicates that clustering and summarization of large Web page collections is promising for alleviating the information overload problem.

1 Introduction

Effectively summarizing Web page collections becomes more and more critical as the online information continues to overload the World Wide Web. A concise and meaningful summary, which is generated automatically and consists of a few key phrases and key sentences, can help Web users understand the essential topics and main contents covered in the collection quickly without spending much browsing time [22].

However, automatically generating coherent summaries as good as human-authored summaries is a challenging task since Web page collections (e.g., Web sites of universities, organizations) often contain various topics and heterogenous contents. For example, a university's Web site may cover many topics such as courses, students, faculties and research. Summarizing the whole Web site often yields an incoherent summary or a summary that is heavily biased towards a subset of the topics included in the Web site. It is much more beneficial to first identify the essential topics in the Web site, and then summarize each individual topic.

A.Y. Tawfik and S.D. Goodwin (Eds.): Canadian AI 2004, LNAI 3060, pp. 60–74, 2004.

Text clustering, which has been extensively studied in many scientific disciplines, plays an important role in organizing large amounts of heterogenous data into a small number of semantically meaningful clusters [15]. In particular, Web collection clustering is useful for summarization, organization and navigation of semi-structured Web pages [21].

Clustering of documents, including Web pages, suffers from the very high dimensionality of the feature vector space required if the naive bag-of-words representation of documents is used. In a high dimensional space, the distance between any two documents tends to be constant, making clustering on the basis of distance ill-defined [11]. Therefore the issue of reducing the dimensionality of the space is critical. Automatic term extraction [9], based on a combination of linguistic and statistical constraints, has the promise of leading to reduced dimensionality representations by focusing on semantically meaningful word collocations [13].

This research aims towards K-Means and EM clustering of Web page collections using automatically extracted topical terms, and automatic summarization of the resulting clusters. Summarization of a cluster proceeds with the identification and extraction of keyterms and key sentences in the *narrative* text, which constitute the summary of the cluster [22].

The quality of clustering with term-based document representation is compared to that with word-based document representation, which is used as the baseline method. We demonstrate that term-based clustering significantly outperforms word-based clustering with much lower dimensionality. The significance of Web collection clustering for automatic Web collection summarization is also investigated.

The rest of the paper is organized as follows. Section 2 reviews published Web collection clustering and summarization approaches, and Section 3 explains how to construct a Web page collection for experimental purposes. Section 4 describes how to conduct K-Means and EM clustering with both word- and term-based document representations and Section 5 evaluates the clustering quality. Section 6 discusses automatic summarization of resulting clusters. Finally, Section 7 concludes our work and describes future research directions.

2 Related Work

A variety of approaches to text clustering have been developed [2,12,15]. Typically clustering approaches can be categorized as *agglomerative* or *partitional* based on the underlying methodology of the algorithm, or as *hierarchical* or *flat* (non-hierarchical) based on the structure of the final solution [23].

In general, text clustering involves constructing a vector space model and representing documents by feature vectors. First, a set of features (e.g., bag of words) is properly selected from the document corpus. Second, each document is represented by a feature vector, which consists of weighting statistics of all features. Finally, clustering proceeds by measuring the similarity (e.g., a function of

Euclidean distance) between documents and assigning documents to appropriate clusters.

One important issue for effective clustering is feature selection. A good feature set is effective if it can discriminate dissimilar documents as much as possible and its dimensionality is as low as possible. Approaches to Web collection clustering have been either *word-based* or *link-based*. Word-based approaches [4,5] use a set of common words shared among documents as features, which suffers from the high space dimensionality while link-based approaches [6,20] analyze the hyperlinks between Web documents for feature selection, which depends on the availability of co-citations. Some Web page clustering systems [14,21] use a combination of above two.

Research work in [13] demonstrates that automatically extracted topical terms can reduce the space dimensionality significantly while providing comparable performance with link-based feature representation (citation graph) in an information retrieval task, i.e., document similarity analysis. We aim to use automatically extracted terms for Web collection clustering and compare the clustering quality to that with bag of words representation.

Zhang et al. [22] propose an effective content-based approach to automatic Web site summarization, which can be applied to automatic summarization of computed clusters. Traditional text summarization systems are mostly *extraction-based* via the identification of the most significant sentences in source text for summary generation [10]. However, Web pages often contain diverse textual fragments such as bullets (which are not complete phrases and therefore difficult to analyze linguistically) or short phrases, that carry no information (e.g., date page last revised / copyright note), so being able to identify narrative text apart from non-narrative text is very important when moving from traditional coherent text such as news stories to Web documents. By filtering out non-narrative text identified by decision tree rules, Web-based applications, including automatic term extraction and Web collection clustering, can make use of state-of-the-art techniques for cleaner text.

3 Web Page Collection

The Text REtrieval Conference (TREC[1]) provides a Web test collection .GOV[2] with 18G data crawled from the .GOV Web site[3], which is a huge site with 14 main topics[4]. Each topic has a *description* Web page which lists all the subtopics categorized in the current topic. In order to construct a Web page collection for testing the clustering quality using word- and term-based document representations, a Web crawler is designed to collect Web pages of subtopics from each of the 14 topics. The 14 Web pages of topic description are excluded.

[1] http://trec.nist.gov

[2] http://www.ted.cmis.csiro.au/TRECWeb/govinfo.html

[3] http://www.firstgov.gov

[4] http://www.firstgov.gov/Citizen/Topics/All_Topics.shtml

A total of 1123 Web pages of subtopics is collected. Among them, a subtotal of 141 link-broken and application files (.pdf, .ps, .doc and other binary) is removed, leading to a smaller set of 982 Web pages. Then plain text from all Web pages is extracted by the text browser Lynx[5], which is found to outperform several alternative text extraction tools such as *HTML2TXT*[6] and *html2txt*[7] [22]. Text files with more than $100K$ size are truncated to that size.

The final data set used for clustering experiments is summarized in Table 1, where i is the topic index and $|T_i|$ is the number of Web pages in each topic T_i.

Table 1. Number of Web pages in each of 14 topics.

i	**Topic** T_i	$\|T_i\|$
1	Benefits and Grants	58
2	Consumer Protection	56
3	Defense and International	86
4	Education and Jobs	89
5	Environment, Energy and Agriculture	101
6	Family, Home and Community	119
7	Health	94
8	History, Arts and Culture	75
9	Money and Taxes	55
10	Public Safety and Law	58
11	Public Service and Volunteerism	69
12	Science and Technology	36
13	Travel and Recreation	48
14	Voting and Elections	38

As shown in Table 1, the most populated topic is *Family, Home and Community* with 119 Web pages while the least populated topic is *Science and Technology* with only 36 Web pages.

4 *K*-means and EM Clustering

In this section we discuss how to conduct word- and term-based K-Means and EM clustering on the Web page collection constructed in Section 3.

4.1 *K*-means and EM Algorithms

The K-Means algorithm has been widely used due to its implementation simplicity. This non-hierarchical method first selects K data points as the initial

[5] http://lynx.isc.org

[6] http://user.tninet.se/~jyc891w/software/html2txt/

[7] http://cgi.w3.org/cgi-bin/html2txt

centroids, one for each of K clusters. Second, all data points are assigned to the cluster whose centroid is the closest to the current data point. Third, the centroid of each cluster is recalculated. Steps two and three are repeated until the centroids do not change [18].

The Expectation Maximization (EM) algorithm was first introduced by Dempster et al. [7] as an approach to estimating the missing model parameters in the context of fitting a probabilistic model to data. It can be seen as an iterative approach to optimization. It first finds the expected value of the *log likelihood* with respect to the current parameter estimates. The second step is to maximize the expectation computed in the first step. These two steps are iterated if necessary. Each iteration is guaranteed to increase the log likelihood and the algorithm is guaranteed to converge to a local maximum of the likelihood function [1]. EM has many applications including text classification and text clustering. It is one of the most popular statistical unsupervised learning algorithms.

4.2 Word- and Term-Based Feature Selection

As discussed in Section 2, feature selection plays an important role in achieving high quality clustering. In our work, we experiment with two ways of feature selection, i.e., bag of words (word-based) and automatically extracted terms (term-based). In both cases, the text parts of Web pages are represented using the standard normalized TF-IDF weighting scheme, which is used in many text-based applications [8,16].

Word-based Representation of Documents. The naive bag of words method selects a set of common words shared among documents as features. First, a standard list of 425 stopwords [8] is removed from each text file. Second, each unique word is stemmed using the Porter stemmer, and its frequency in the current text file (TF part) and the number of documents where it appears (IDF part) are recorded. Finally, the word list is sorted in descending order of total frequency in all documents. It is observed there is a total of 39245 unique words in 982 text files. The most frequent word *state* appears 5816 times while about 6000 words appear only once. After we remove words which appear less than 6 times in the corpus, there are 5526 words left in the list.

In selecting the words to be used as features, it is observed that words appearing in most documents in the corpus are not useful features, because they do not help discriminate among clusters. So we re-rank the list of 5526 words according to their IDF in order to set proper upper and lower cutoffs for selecting words appearing with intermediate frequency [13]. After the words used as features are selected, their TF-IDF values in each document are calculated and normalized as in (1) and (2), respectively.

$$w_{i,j} = f_{i,j} \times \log \frac{N}{df_i}. \tag{1}$$

$$W_{i,j} = \frac{w_{i,j}}{\sqrt{\sum_{i=1}^{n} w_{i,j}^2}}. \quad (2)$$

where, $w_{i,j}$ is the TF-IDF weight of word i in document j; $f_{i,j}$ is the frequency of word i in document j; N is the total number of documents in the collection, i.e., 982; df_i is the number of documents containing word i; and $W_{i,j}$ is the normalized weight of word i in document j.

We experimented with different word cutoffs such as [10, 4000] (from the 10^{th} word to the 4000^{th} word ranked by IDF) and found that the best word cutoff with respect to clustering quality (evaluated by F-measure in Section 5) is [15, 3000].

Term-based Representation of Documents. Word-based representation involves thousands of features and subsequent clustering often suffers from the dimensionality curse [11]. Multi-word terms are known to be linguistic descriptors of documents. Automatic term extraction is a useful tool for many text related applications such as text clustering and document similarity analysis. It is beneficial to automatically extract multi-word terms as features in order to significantly reduce the high dimensionality [13].

In our work, we apply a state-of-the-art method, *C-value/NC-value* [9], to extract multi-word terms from the Web page collection automatically. The *C-value* is a domain-independent method used to automatically extract multi-word terms. It aims to get more accurate terms than those obtained by the pure frequency of occurrence method, especially terms that may appear as nested within longer terms. The *NC-value* is an extension to *C-value*, which incorporates context words information into term extraction. Context words are those that appear in the vicinity of candidate terms, i.e. nouns, verbs and adjectives that either precede or follow the candidate term. This term extraction approach consists of both linguistic analysis (linguistic filter, part-of-speech tagging [3], and stop list [8]) and statistical analysis (frequency analysis, *C-value/NC-value*). Experiments in [9,13] show that *C-value/NC-value* method performs well on a variety of special text corpora.

However, unlike traditional text documents with well-structured discourse, Web documents often contain diverse textual segments (e.g., bullets and short phrases) which are not suitable for extraction of semantically meaningful terms. We apply the same approach as proposed in [22] to identify *narrative* paragraphs in the Web collection and consequently terms are extracted from narrative text only. First, text parts of Web pages are automatically segmented into paragraphs by Lynx. Second, one classifier, LONG, is used to filter out *short* paragraphs. Third, the other classifier, NARRATIVE, is in turn used to extract *narrative* paragraphs from *long* paragraphs identified in the previous step. These two classifiers are trained by the decision tree tool C5.0[8] based on features extracted by shallow natural language processing.

[8] http://www.rulequest.com/see5-unix.html

Then the *C-value/NC-value* method is applied and 1747 terms are extracted. These terms are ranked by NC-value and an NC-value threshold 17.5, which is 5% of the maximum NC-value, is set. This in return produces a term list, C, of 924 terms.

It is observed that the term list C contains some noun phrases (e.g., *home page*), which appear frequently in Web pages. These noun phrases can be treated as *Web-specific* stop words [17] and must be removed. We experimented with 60 DMOZ[9] Web sites and manually identified a stop list, L, of 81 noun phrases (e.g., *Web site*, *home page*, *credit card*, *privacy statement*, ...). The term list C is filtered through the noun phrase stop list L and the resulting term list T contains 892 terms.

Next, all documents are scanned in order to record the TF and IDF values for all terms in T. As we did in the word-based approach, we re-rank all the terms in T according to their IDF in order to set proper upper and lower cutoffs for selecting terms. We experiment with different term cutoffs and the best term cutoff is found to be [1, 450].

5 Clustering Experiments and Evaluation

In this section we discuss the clustering experiments and evaluate the quality of the clustering results.

In our work, Java programs of K-Means and EM algorithms in the WEKA[10] package are used. For each cutoff (see previous section) of words and terms, we construct either a document-word matrix or a document-term matrix with Attribute-Relation File Format (ARFF)[11], where each document is represented with normalized TF-IDF values of either words or terms. Then we run the clustering software (we set $K = 14$ and force EM to create 14 clusters) and compare the clustering quality using word- and term-based document representations.

5.1 Evaluation Schemes

There are two main types of measures of clustering quality, i.e., *internal quality measure* and *external quality measure.* Internal quality measure uses the "overall similarity", which is based on the pairwise similarity of documents in a cluster, to evaluate the clustering quality. It compares different sets of clusters without reference to external knowledge. On the other hand, external quality measure such as *entropy* and *F-measure* examines the clustering quality by comparing the resulting clusters to known classes (topics) [18].

We use the F-measure to evaluate the clustering quality since topic information for all Web pages is available. F-measure combines *precision* and *recall.* For a particular cluster C_i, we have to decide which topic T_j it most probably

[9] http://dmoz.org

[10] http://www.cs.waikato.ac.nz/∼ml/weka/index.html

[11] http://www.cs.waikato.ac.nz/∼ml/weka/arff.html

belongs to. More specifically, we calculate the precision $P_{i,j}$, recall $R_{i,j}$, and F-measure $F_{i,j}$ of cluster C_i with respect to topic T_j as in (3).

$$P_{i,j} = \frac{n_{i,j}}{|C_i|}, \quad R_{i,j} = \frac{n_{i,j}}{|T_j|},$$
$$F_{i,j} = \frac{2 \cdot P_{i,j} \cdot R_{i,j}}{P_{i,j} + R_{i,j}} \quad (i, j \in [1..14]). \tag{3}$$

where, $n_{i,j}$ is the number of documents in topic T_j which appear in cluster C_i; $|C_i|$ is the number of documents in cluster C_i; and $|T_j|$ is the number of documents in topic T_j. Then the cluster C_i will belong to the topic T_j which maximizes $F_{i,j}$. This is formally represented in (4).

$$F_i = \max_{j=1}^{14} F_{i,j} \quad (i \in [1..14]). \tag{4}$$

Finally the overall value of F-measure F is calculated by taking a weighted average of all F_i values as represented in (5).

$$F = \sum_{i=1}^{14} \frac{|T_i|}{N} F_i, \quad N = \sum_{j=1}^{14} |T_j|. \tag{5}$$

5.2 Word-Based Clustering Quality

We use both K-Means and EM algorithms to experiment with various document-word matrices, which are constructed on different word cutoffs. Table 2 summarizes the top 5 overall F values obtained with both algorithms. It is observed that EM algorithm achieves better F values than K-Means algorithm in all 5 word cutoffs.

Table 2. The top 5 F values achieved by word-based K-Means and EM clustering.

Cutoff$_{word}$	[15, 3000]	[10, 3000]	[10, 3500]	[15, 4000]	[5, 3000]
$F_{K-Means}$	0.52	0.51	0.46	0.44	0.42
F_{EM}	0.59	0.57	0.50	0.49	0.43

As shown in Table 2, both K-Means and EM achieve better F values with the word cutoff [15, 3000] than with the word cutoff [10, 3000]. This indicates that the words at the top of the word list (ranked by IDF) appear too frequently and they are not appropriate for feature selection. Also cutoff [10, 3000] outperforms cutoff [10, 3500], which indicates that the words at the bottom of the word list appear too rarely and they are not good features either. This demonstrates our assumption that words with intermediate document frequency are better choices of features.

5.3 Term-Based Clustering Quality

Document-term matrices are constructed on different term cutoffs. Then both K-Means and EM algorithms are applied to these matrices. Table 3 summarizes the top 5 overall F values achieved with various term cutoffs. We observe that EM achieves bigger F values than K-Means in all 5 term cutoffs.

Table 3. The top 5 F values achieved by term-based K-Means and EM clustering.

Cutoff$_{term}$	[1, 450]	[1, 500]	[2, 450]	[2, 500]	[3, 500]
$F_{K-Means}$	0.67	0.67	0.67	0.65	0.64
F_{EM}	0.68	0.68	0.68	0.67	0.65

As shown in Table 3, there is no difference between the F values using either K-Means or EM with the first 3 term cutoffs. This indicates that a proper number (e.g. 450) of terms at the top of the term list (ranked by IDF) can be readily used as features.

Furthermore, we do EM clustering by fixing the lower term cutoff to 1, and changing the upper term cutoff with different values varying from 300 to 892 (full size of the term list). As shown in Fig. 1, different upper term cutoffs lead to different F values. The best F value is achieved with the upper term cutoff 450. If we increase or decrease the upper term cutoff such as using 350 and 600, then the clustering performance becomes worse. This indicates that there exists an optimal dimensionality for term-based clustering. Moreover, it is observed that with cutoffs [1, 300] and [1, 315], EM clustering fails because *null* feature vector appears, i.e., there is some document which does not include any one of the terms. This indicates that there is a minimum value for the upper term cutoff for valid clustering.

Likewise, we do EM clustering by fixing the upper term cutoff to 450, and changing the lower term cutoff with values varying from 1 to 10. It is observed that there is no much difference between resulting F values and that a lower term cutoff of 6 will lead to invalid clustering, i.e., there is at least one null vector in this case. This indicates that with much lower dimensionality in term-based clustering, terms at the top of the term list are indispensable.

The experiments above are a rough approximation of the search for optimal upper and lower term cutoffs. We leave this two-dimension search problem for future research.

5.4 Comparison of Word- and Term-Based Clustering

The main objective of our work is to investigate if term-based clustering can significantly outperform word-based clustering on Web page collections.

Table 4 summarizes the details of the best F values in word- and term-based K-Means clustering, where w_i is the weight of each topic, i.e., $\frac{|T_i|}{N}$, and F is the

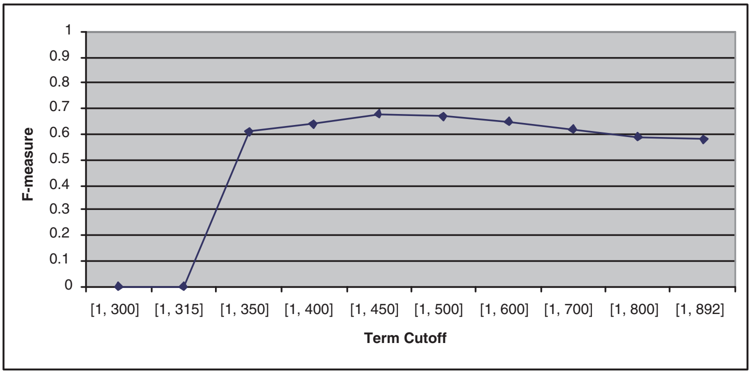

Fig. 1. Varying F values corresponding to different upper term cutoffs with fixed lower term cutoff 1 in EM clustering.

sum of $w_i \cdot F_i$ over all 14 topics, as calculated in (5). K-Means algorithm achieves the best overall F values, 0.52 using the word cutoff [15, 3000], and 0.67 using the term cutoff [1, 450], respectively.

Table 4. The best F values achieved in K-Means clustering using the word cutoff [15, 3000] and the term cutoff [1, 450], respectively.

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	F
w_i	0.06	0.06	0.09	0.09	0.10	0.12	0.10	0.08	0.06	0.06	0.07	0.04	0.05	0.04	—
$F_{i_{word}}$	0.53	0.45	0.67	0.43	0.39	0.68	0.33	0.54	0.44	0.64	0.61	0.49	0.70	0.45	**0.52**
$F_{i_{term}}$	0.62	0.59	0.83	0.55	0.71	0.82	0.53	0.59	0.61	0.71	0.66	0.49	0.87	0.63	**0.67**

In order to measure if the difference between the quality of word- and term-based clustering (i.e., $w_i \cdot (F_{i_{term}} - F_{i_{word}})$ $(i = 1..14)$) is significant, we apply the standard two-tail paired t-test, which generally compares two different methods used for experiments carried in pairs.

The t-test carried on Table 4 obtains a t-statistic equal to 4.543. By checking the t-table, we have $t_{0.05,13} = 2.160$. Since $t > t_{0.05,13}$ (P-value < 0.001), there is a significant difference between the clustering quality of word- and term-based document representation using K-Means algorithm. More precisely, term-based K-Means clustering significantly outperforms word-based K-Means clustering with much lower dimensionality.

Table 5 shows the details of the best F values in word- and term-based EM clustering. EM algorithm achieves the best overall F values, 0.59 using the word cutoff [15, 3000], and 0.68 using the term cutoff [1, 450], respectively. Similar t-test analysis as above obtains a t-statistic equal to 3.237 and a P-value less than 0.01, which shows that term-based EM clustering significantly outperforms word-based EM clustering.

Table 5. The best F values achieved in EM clustering using the word cutoff [15, 3000] and the term cutoff [1, 450], respectively.

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	F
$F_{i_{word}}$	0.63	0.54	0.65	0.47	0.61	0.76	0.48	0.68	0.43	0.52	0.55	0.38	0.77	0.69	**0.59**
$F_{i_{term}}$	0.75	0.89	0.54	0.49	0.66	0.84	0.51	0.77	0.64	0.59	0.58	0.54	0.91	0.78	**0.68**

Moreover, with the term cutoff [1, 335], EM algorithm achieves an overall F value 0.59, which is equal to the best F value with the word cutoff [15, 3000]. This indicates that term-based document representation can significantly reduce the dimensionality by almost an order of magnitude while maintaining comparable performance with word-based document representation in EM clustering.

5.5 Comparison of K-means and EM Clustering

We also apply t-test to measure if there is a significant difference between clustering quality of K-Means and EM using the best word and term cutoffs.

The t-test for comparison of EM and K-Means algorithms using the best word cutoff [15, 3000] obtains a t-statistic equal to 2.199 and a P-value less than 0.05, so there is a significant difference between the quality of word-based EM clustering and word-based K-Means clustering. More precisely, the EM algorithm significantly outperforms the K-Means algorithm using word-based document representation. However, a similar t-test shows that there is no significant difference (t-statistic = 0.042, P-value = 0.967) between clustering quality of K-Means and EM when using the best term cutoff [1, 450].

Regarding the computational complexity, it is observed that K-Means clustering is roughly seventeen times (CPU time: 37 seconds vs. 630 seconds) faster than EM clustering with the best term cutoff when running in a Linux machine.

6 Automatic Cluster Summarization

In this section we explain how to summarize each individual cluster obtained by EM clustering using the best term cutoff [1, 450] described above and evaluate cluster summaries.

6.1 Cluster Summary Generation

The automatic cluster summarization is a multi-step process similar with the keyword-based approach proposed by Zhang et al. [22].

1. First a list of multi-word terms is extracted by the *C-value*/*NC-value* method from the narrative text of Web pages in each cluster. Then the term list is filtered through the noun phrase stop list L and the top 25 terms (ranked by *NC-value*) are kept as keyterms.
2. Narrative text of the Web pages in the current cluster is scanned to extract the most significant sentences. The significance of a sentence is measured by calculating a weight value, which is the maximum of weights of *word clusters* within the sentence. A word cluster is defined as a list of consecutive words which starts and ends with a keyterm and at most 2 non-keyterms must separate any two neighboring keyterms. The weight of a word cluster is calculated as the sum of keyterm weights divided by the number of keyterms in the word cluster.
3. A summary is generated, which consists of the top 25 keyterms and the top 5 key sentences. The numbers 25 and 5 are determined by both the informativeness of the keyterms and the key sentences and the size of the cluster summaries.

As an example, Table 6 gives the cluster summary corresponding to Topic 7: Health. As we can see, this summary provides informative contents of the Health topic.

Table 6. An example of cluster summary corresponding to the *Health* topic.

Part I. top 25 keyterms
health administration, veterans health, health care, drug administration, food safety, veterans health administration, severe acute respiratory syndrome, health maintenance organization, mine safety, veterans health administration facilities, health insurance, vital events, drug abuse, veterans integrated service networks, veterans health administration program, health information, acute respiratory syndrome, respiratory syndrome, severe acute, women health, other communication disorder, health care professionals, health administration facilities, health maintenance, health service
Part II. top 5 key sentences
1. The Veterans Health Administration (VHA) provides a broad spectrum of medical, surgical, and rehabilitative care to its customers. 2. VHA Organizations A number of offices work together to make the Veterans Health Administration (VHA) an efficient and patient-centered health care system. 3. The Mine Safety and Health Administration - Look here for information about the dangers of playing near or in mines. 4. Data on health status, healthy lifestyles, illness and disability, the use of health care and vital events. 5. The Food and Drug Administration (FDA) today is advising women and health care professionals about important new safety changes to labeling of all estrogen and estrogen with progestin products for use by postmenopausal women.

6.2 Cluster Summary Evaluation

In this subsection, we describe how to evaluate the quality of cluster summaries in a way which has been extensively used in related work [13,19]. Human subjects are asked to read each cluster summary and judge the relatedness of keyterms and key sentences to the corresponding topic as follows:

1. For each cluster, browse the description Web page (see Section 3) of its corresponding topic (which maximizes the F value of the current cluster) for a sufficient time in order to understand the subtopics categorized in the current topic.
2. Read the cluster summary and rank each *summary item*, i.e., keyterm or key sentence, into *good*, *fair* or *bad* using the following rules:
   - If it is strongly pertinent to the current topic, rank it *good.*
   - If it is pertinent to the topic, rank it *fair.*
   - If it is not pertinent to the topic, rank it *bad.*
3. Count n_g, n_f, and n_b, which is the number of good, fair, and bad items in each summary, respectively.

Related research in [19] defines *acceptable* terms as good and fair terms. Let p_t and p_s be the percentage of acceptable keyterms and key sentences, respectively. We formally define p_t and p_s in (6).

$$p_t, p_s = \frac{n_g + n_f}{n_g + n_f + n_b}. \tag{6}$$

For example, in the summary example shown in Table 6, there are 17 good, 7 fair, and 1 bad keyterms; 2 good, 3 fair, and 0 bad key sentences. So the percentage of acceptable keyterms is calculated as $\frac{17+7}{25} = 96\%$, and the percentage of acceptable key sentences is 100%.

Table 7 summarizes the percentage of acceptable keyterms and key sentences for all cluster summaries. Our approach achieves an average of 83.7% acceptable keyterms and 88.6% acceptable key sentences, which indicates that the cluster summaries are acceptable to human readers.

Table 7. Details of summary quality values.

Summary	1	2	3	4	5	6	7	8	9	10	11	12	13	14	**Average**
$p_t(\%)$	100	96	64	96	56	92	96	68	96	84	84	72	80	88	83.7
$p_s(\%)$	100	100	80	100	100	60	100	100	100	80	80	60	80	100	88.6

7 Conclusion and Discussion

In this paper, we experiment with word- and term-based representations of Web documents for K-Means and EM clustering. The term-based approach relies on a start-of-the-art automatic term extraction method on narrative content of the Web page collection. We evaluate the difference between word- and term-based clustering and demonstrate that term-based clustering significantly outperforms word-based clustering with much lower dimensionality. We also show that summarization of individual clusters provides a good overview of the various essential topics covered in the Web page collection and this approach is critical for alleviating the information overload problem on the World Wide Web.

Future research involves several directions: 1) Investigation of automatic Web-specific stop words generation [17] to achieve better term extraction from Web page collections; 2) Search for optimal upper and lower term cutoffs for best clustering quality; 3) Bisecting K-Means and EM clustering to produce a hierarchical decomposition of the essential topics and their subtopics covered in the data collection; 4) Application on the whole .GOV collection to estimate the effectiveness and efficiency of our approach on the huge data set.

Acknowledgements. We are thankful to reviewers for many valuable suggestions on this work. The research has been supported by grants from the Natural Sciences and Engineering Research Council of Canada.

References

[1] J. Bilmes. A Gentle Tutorial on the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models. Technical Report ICSI-TR-97-021, University of California, Berkeley, 1997.

[2] D. Boley. Principal Direction Divisive Partitioning. *Data Mining and Knowledge Discovery*, 2(4):325–344, 1998.

[3] E. Brill. A Simple Rule-based Part of Speech Tagger. In *Proceedings of the 3rd Conference on Applied Natural Language Processing (ANLP-92)*, pages 152–155, 1992.

[4] A. Broder, S. Glassman, M. Manasse, and G. Zweig. Syntactic Clustering of the Web. In *Proceedings of the 6th International World Wide Web Conference*, pages 391–404, 1997.

[5] D. Cutting, D. Karger, J. Pedersen, and J. Tukey. Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 318–329, 1992.

[6] J. Dean and M. Henzinger. Finding Related Pages in the World Wide Web. In *Proceedings of the 8th International World Wide Web Conference*, pages 389–401, 1990.

[7] A. Dempster, N. Laird, and D. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[8] W. Frakes and R. Baeza-Yates. *Information Retrieval: Data Structures and Algorithms*. Prentice Hall, Englewood Cliffs, NJ, 1992.

[9] K. Frantzi, S. Ananiadou, and H. Mima. Automatic Recognition of Multiword Terms. *International Journal of Digital Libraries*, 3(2):117–132, 2000.

[10] J. Goldstein, V. Mittal, J. Carbonell, and J. Callan. Creating and Evaluating Multi-document Sentence Extract Summaries. In *Proceedings of the 9th ACM International Conference on Information and Knowledge Management (CIKM'00)*, pages 165–172, 2000.

[11] E. Han, G. Karypis, V. Kumar, and B. Mobasher. Clustering in a High-dimensional Space Using Hypergraph Models. Technical Report TR-97-063, Department of Computer Science and Engineering/Army HPC Research Center, University of Minnesota, 1997.

[12] A. Hotho, A. Maedche, and S. Staab. Ontology-based Text Clustering. In *Proceedings of the IJCAI Workshop on "Text Learning: Beyond Supervision"*, 2001.

[13] E. Milios, Y. Zhang, B. He, and L. Dong. Automatic Term Extraction and Document Similarity in Special Text Corpora. In *Proceedings of the 6th Conference of the Pacific Association for Computational Linguistics (PACLing'03)*, pages 275–284, 2003.

[14] D. Modha and W. Spangler. Clustering Hypertext with Applications to Web Searching. In *Proceedings of the 11th ACM Conference on Hypertext and Hypermedia*, pages 143–152, 2001.

[15] H. Schutze and H. Silverstein. Projections for Efficient Document Clustering. In *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'97)*, pages 74–81, 1997.

[16] F. Sebastiani. Machine Learning in Automated Text Categorization. *ACM Computing Surveys*, 34(1):1–47, March 2002.

[17] M. Sinka and D. Corne. Towards Modernized and Web-Specific Stoplists for Web Document Analysis. In *Proceedings of IEEE/WIC International Conference on Web Intelligence (WI'03)*, pages 396–402, 2003.

[18] M. Steinbach, G. Karypis, and V. Kumar. A Comparison of Document Clustering Techniques. In *Proceedings of KDD Workshop on Text Mining*, 2000.

[19] P. Turney. Coherent Keyphrase Extraction via Web Mining. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI'03)*, pages 434–439, 2003.

[20] Y. Wang and M. Kitsuregawa. Use Link-based Clustering to Improve Web Search Results. In *Proceedings of the 2nd International Conference on Web Information Systems Engineering (WISE'01)*, pages 115–124, 2001.

[21] Y. Wang and M. Kitsuregawa. Evaluating Contents-link Coupled Web Page Clustering for Web Search Results. In *Proceedings of the 11th ACM International Conference on Information and Knowledge Management (CIKM'02)*, pages 499–506, 2002.

[22] Yongzheng Zhang, Nur Zincir-Heywood, and Evangelos Milios. Summarizing Web Sites Automatically. In *Proceedings of the 16th Conference of the Canadian Society for Computational Studies of Intelligence (AI'03)*, pages 283–296, 2003.

[23] Y. Zhao and G. Karypis. Evaluation of Hierarchical Clustering Algorithms for Document Datasets. In *Proceedings of the 11th ACM International Conference on Information and Knowledge Management (CIKM'02)*, pages 515–524, 2002.