DETERMINING INITIAL STARTING CONDITIONS FOR DOCUMENTS CLUSTERING

Omar H. Karam
ohkaram@hotmail.com

Ahmed M. Hamad
amhamad13@yahoo.com

Sherin M. Moussa
shero@usa.com

Department of Information Systems
Faculty of Computer and Information Sciences
Ain Shams University, Cairo, 11566, Egypt
Phone: (202) 684 4284 Fax: (202) 682 8298

Abstract

Unlabeled document collections are becoming increasingly common and available; mining such data sets represents a major contemporary challenge. Using words as features, text documents are often represented as high dimensional and sparse vectors- a few thousand dimensions is typical.
Practical approaches to clustering such document vectors use an iterative procedure (e.g. k-means, EM) that is known to be especially sensitive to initial starting conditions (k and initial centroids).
In this paper, we introduce a hybrid clustering algorithm that determines these initial conditions automatically, depending on the required quality for the obtained clusters. The hybrid algorithm combines the agglomerative hierarchical approach with the k-means approach to provide k disjoint clusters. However, the textual, unstructured nature of documents makes the task considerably more difficult than other data sets. We present the results of an experimental study of our introduced algorithm.

Keywords: *data mining, text mining, document clustering, agglomerative clustering, and k-means clustering.*

1 Introduction

Data Mining is the exploration and analysis by automatic or semiautomatic means of large quantities of data, in order to discover meaningful patterns and rules. Due to the continuous growth of the volume of electronic data currently available, automated knowledge extraction techniques have become necessary to filter the huge amounts of stored data.

Text mining is an emerging field at the intersection of several research areas, including data mining, natural language processing, and information retrieval. There are several differences between data mining and text mining: Data mining usually deals with structured data sets - typically in the first normal form, in terminology of relational databases. By contrast, text mining deals with unstructured or semi-structured data; namely, the text found in articles, documents, etc. Furthermore, the occurrence of synonyms (different words with the same meaning) and homonyms (words with the same spelling but with distinct meanings) makes it difficult to detect valid relationships between different parts of the text [4],[8],[11].

Document clustering has been investigated for use in a number of different areas of text mining and information retrieval. Initially, document clustering was investigated for improving the precision or recall in information retrieval systems. It has been used to discover “latent concepts” in sets of unstructured text documents, and to summarize and label such collections. Clustering is inherently useful in organizing and searching large text collections [3].

Many techniques are used for document clustering [1],[6],[9],[10]. The most commonly used approach is the well-known k-means clustering algorithm. “k-means” refers to finding “k” number of clusters using initial “means”, that is, the mean or median of each created cluster. This approach is based on the idea that a center point (centroid), representing the average point of the data points grouped in one cluster, can represent a cluster. Given the number of required clusters (k) and the initial k center points (centroids), documents can be clustered by assigning each document to the cluster having the nearest centroid. This is done by measuring similarity through the distance function between the document and each of the k centroids. A new centroid is then re-computed for each newly created cluster, and the whole process is repeated until no change in the k centroids or the detected change is less than a determined threshold. Another approach is the Agglomerative hierarchical clustering

algorithm, which starts with each document as an individual cluster, and at each iteration, merges the most similar or closest pair of clusters until one cluster is obtained. Therefore, clustering should stop before obtaining the single cluster depending on a specified condition, such as the number of clusters required, or the quality of the clusters to obtain [1],[2],[3],[12].

Iterative techniques (e.g. k-means) are especially sensitive to initial starting conditions (i.e. number of clusters, k, and initial centroids). Most work does not address the issue of initialization or assumes either user-provided or randomly chosen starting points. A procedure is needed for computing these initial starting conditions, based on an efficient technique that allows the iterative algorithm to converge to a "better" local minimum [7].

In this paper, we examine how agglomerative techniques can be used in conjunction with the k-means algorithm as a method of specifying these initial conditions. We evaluate the different parameters involved in the agglomerative algorithm such as the pruning threshold, the agglomerative distance threshold, and sample sizes. We concentrate on building this hybrid algorithm for clustering text documents. Furthermore, we analyze the problem of clustering text data sets in relevance to the unstructured nature of documents. In addition, we discuss the results of an experimental study of our introduced algorithm.

The rest of the paper is organized as follows: In section 2, Clustering Text Documents is discussed, while section 3 presents The Initialization Problem. In section 4, The Hybrid Clustering Algorithm is introduced. In section 5, Experiments and Results are discussed. Section 6 is the Conclusion.

2 Clustering Text Documents

Since documents are in text formats, some preprocessing is necessary in order to represent these documents in a mathematical form [3],[4]. Documents are represented using the vector space model, where each document, d, is considered to be a vector, **d**, in the multidimensional space of document words called term space [2],[3],[4]. Each document is represented by the (TF) vector, the Term-Frequency vector:

$$\mathbf{d}_{tf} = (tf_1, tf_2, \ldots\ldots\ldots\ldots, tf_n) \qquad (1)$$

where tf_i is the frequency of the ith term in the document. Very common words, i.e. "a", "the", "for", "are", known as stop words, are stripped out completely and different forms of a word are reduced to a single stem form. In our work, Porter's suffix stripping algorithm is used [13]. For example, the words "compute", "computing", and "computed" are stemmed to "comput". Given a set, S, of documents and their corresponding vector representations, the centroid vector, **c**, is defined to be

$$\mathbf{c} = \frac{1}{|S|} \sum_{\mathbf{d} \in S} \mathbf{d} \qquad (2)$$

which is obtained by averaging the weights of the various terms present in the documents of S, where **d** is the document vector obtained from the set S.

Thus, we can regard the vector space model of a text data set as a word-by-document matrix whose rows are words and columns are document vectors. Typically, a large number of words, a few thousands or more, commonly exist in even a moderately sized set of documents. Hence, the document vectors are very high dimensional. High dimensionality is reduced by pruning document vectors by a given threshold, P, where words having frequency less than this threshold are removed completely from the document vectors. Document vectors lengths are then unified by creating a superset vector that contains all words in all the documents after pruning. This superset vector is then compared to each document vector, and if a word found in the superset vector is missing in the document vector, it is added to the document vector with term frequency zero. Thus, a dimension space is constructed, where each word in a document vector is considered to be a dimension.

The similarity between two document vectors or between a document vector and a centroid vector is computed using the Minkowski distance function,

$$d(i, j) = (| x_{i1} - x_{j1} |^q + | x_{i2} - x_{j2} |^q + \ldots\ldots\ldots\ldots\ldots + | x_{ip} - x_{jp} |^q)^{1/q} \qquad (3)$$

where i represents the ith document vector and j represents the jth document vector, x_{ip} is the word number p in the ith document vector; x_{jp} is the word number p in the jth document vector, q is the power of the distance function.

3 The Initialization Problem

The quality of clusters obtained from partitioning (or iterative) clustering approaches (i.e. k-means) is highly dependent on the initial starting conditions given to the algorithm, these are the number of clusters required, k, and the initial centroids, one for each cluster. The measure of a cluster's quality is taken as the sum of the distances between each document vector and the centroid vector of the cluster to which this document is attached [2],[3]. Lower sums represent higher cluster qualities. In other words, document vectors are very close to the centroid vector of this cluster, thus the quality of this cluster is high. Using the clustering algorithm with the right initial parameters provides clusters with high quality. By changing these parameters, clusters resulted differ dramatically.

$$\text{Quality of one cluster} = \sum_{i=1}^{n} (\mathbf{d}_i - \mathbf{c}) \qquad (4)$$

Where $\mathbf{d}_i$ is the document vector of the ith document, and $\mathbf{c}$ is the centroid vector of the cluster to which this document is attached.

A method is needed to deduce these initial parameters without prior information about the data set to cluster and the absence of any external knowledge about the class labels. These parameters are usually determined through trial and error iterations until proper values are selected that yield the most acceptable results. This approach is very exhaustive; as its cost of time and resources is very high. They could also be user-provided estimates depending on the user's experience with the data set, but in the absence of this expert user or when clustering a new different data set, user-provided assumptions are inadequate. Most of the researchers' efforts are directed towards refining existing clustering techniques and proposing new clustering methods to provide clusters with higher quality, in addition to the enhancement of clustering performance, as clustering techniques are known to consume a long time and require high resources especially for high-dimensional, large data sets which decrease the clustering performance [2],[3],[4],[5],[7],[9].

Hierarchical agglomerative clustering is often portrayed as the better quality clustering approach, but is limited because of its quadratic time complexity $O(n^2)$. In contrast, k-means and its variants have a time complexity which is linear in the number of documents O(n), but are thought to produce inferior clusters [3].

We present a hybrid algorithm that solves the initialization problem, where the agglomerative hierarchical approach is combined with the k-means to provide k-means with these initial parameters. The proposed algorithm's results for the initial k and the initial centroids are dependant upon the required cluster qualities. The quality measure for the agglomerative clusters is the distance at which agglomerative clustering stops, (i.e. agglomerative threshold).

4 The Hybrid Clustering Algorithm

The hybrid algorithm combines the agglomerative hierarchical clustering approach with the k-means clustering approach. It works as follows:

a) A number of N text documents is provided to the system. On reading each document, a document vector is created to represent all the words contained in this document and the number of occurrences of each word.

b) Words of each document are sampled by randomly selecting L words out of each 10 words so that each document is represented by a percentage of its words. Thus, the document vector will be reduced.

c) Document vectors are then pruned by a pruning threshold, P, where all words having their occurrence less than or equal to P, are totally removed from each document vector.

d) After this further vector reduction, document vectors are not equal in length, as words contained in each document vector are not unified.

e) Since words are considered to be the dimensions of text documents data set space, therefore, words are unified in all the document vectors by creating the superset vector that contains all the remaining words in all the N document vectors after pruning.

f) Each of the N document vectors is compared to the superset vector. If a word in the superset vector is missing in the document vector, it is added to the document vector

with a zero occurrence value. Thus the dimension space is constructed for the whole data set.

g) At this stage, document vectors are now ready for agglomerative clustering.

h) The applied agglomerative clustering approach uses the centroid similarity technique for distance measurements. The centroid similarity technique defines the distance between two clusters as the distance between the centroids of these two clusters.

$$d(\mathbf{c}_1, \mathbf{c}_2) = (| x_{11} - x_{21} |^q + | x_{12} - x_{22} |^q + \ldots\ldots + | x_{1p} - x_{2p} |^q)^{1/q} \quad (5)$$

Where $\mathbf{c}_1$ is the centroid vector of cluster 1and $\mathbf{c}_2$ is the centroid vector of cluster 2, x_{1p} is the word number p in $\mathbf{c}_1$; x_{2p} is the word number p in $\mathbf{c}_2$, q is the power of the distance function.

i) A sample of ratio R is randomly selected from the N document vectors to apply agglomerative clustering on them, where S document vectors are randomly selected out of each 10 document vectors; S = R/10.

j) The power of the distance function used for similarity measurement, q, is specified: at q = 1, it is the Manhattan distance function, at q = 2, it is the Euclidean distance function.

k) The agglomerative distance threshold, A, at which clustering should end is specified. It is the clustering quality measure for the clusters.

l) The hybrid algorithm then agglomerates the R individual document vectors in the sample till the quality of the created clusters be are greater than or equal to A.

m) At this stage, the number of created clusters is the "k" parameter, and the centroids of the created clusters are computed to represent the initial centroids.

n) Individual documents that have not been merged yet with other clusters are considered to be individual clusters.

o) These parameters can now be fed to the k-means clustering approach to cluster the whole data set, till variations in the computed centroids are less than a given threshold, M.

The algorithm outputs k disjoint clusters of English text documents. Figures (1), (2) describes the main procedures of the hybrid algorithm.

5 Experiments and Results

In order to apply the hybrid algorithm to cluster text documents, there are some important parameters that should be specified which affect the quality of the clusters obtained. These parameters are the pruning threshold, P, the sample size used in the agglomerative clustering approach, R, the agglomerative distance threshold, A, and the power of the distance function, q, used to compute distances between the document vectors, or between a document vector and the centroid vectors.

We studied the effect of these parameters on the clustering results of the NSF (National Science Foundation) data set obtained by downloading 725 abstracts of the grants awarded by the NSF. Thus, N = 725. These abstracts included subjects Such as astronomy, population studies, undergraduate education, materials, mathematics, biology, health, oceanography, computer science and chemistry. We applied different pruning thresholds on the data set with P = 2, 3, 4, 5, 6, 7 where most of the document vectors contained no words at P = 7. Table (1) represents the created document vector for a NSF abstract after stemming and stop words removal. Table (2) represents the document vector of the same NSF abstract after pruning with P = 7. We can notice the great reduction in length of the considered document vector after pruning.

After creating the corresponding superset vector and constructing the dimension space, samples of R = 10%, 20% were chosen to run the agglomerative clustering approach on these sample document vectors. Table (3) represents the created superset vector of the NSF data set after pruning with P = 7. Despite of pruning with high pruning threshold (P = 7), the created superset vector is still very large. Hence, all document vectors will be of the same length. At this stage, we used the Euclidian distance function (q = 2) to compute distances between the vectors. We also applied different agglomerative distance thresholds, A = 5, 10, 15, and 20 where the number of the created clusters at A = 20 was only two (k = 2). We monitored the effect of varying these parameters on the value of k, the number of the created agglomerative clusters.

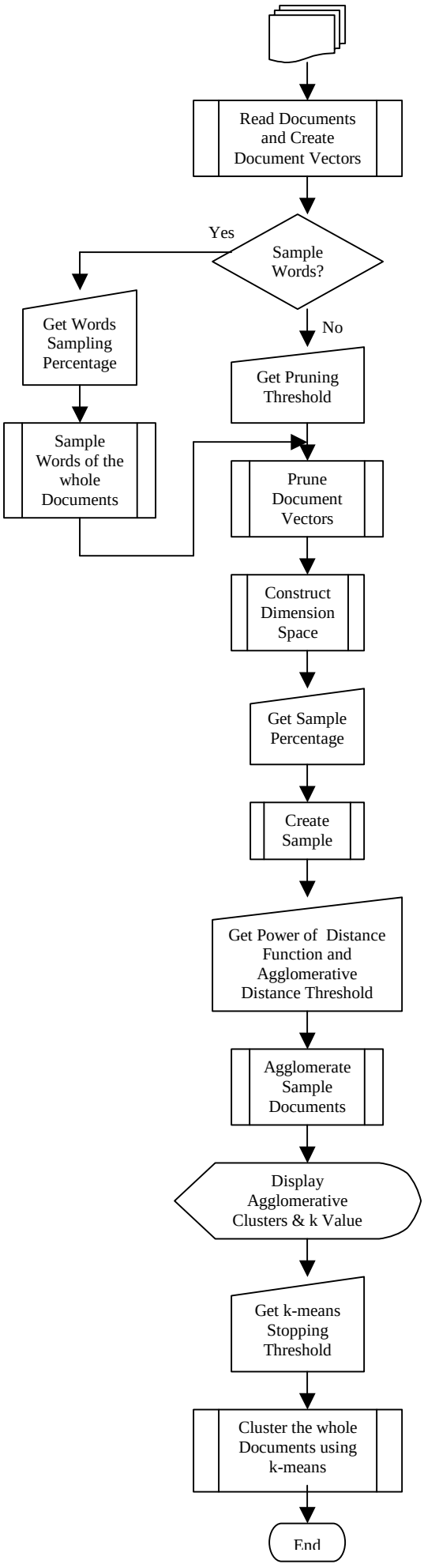

Figure (1) – The Hybrid Algorithm.

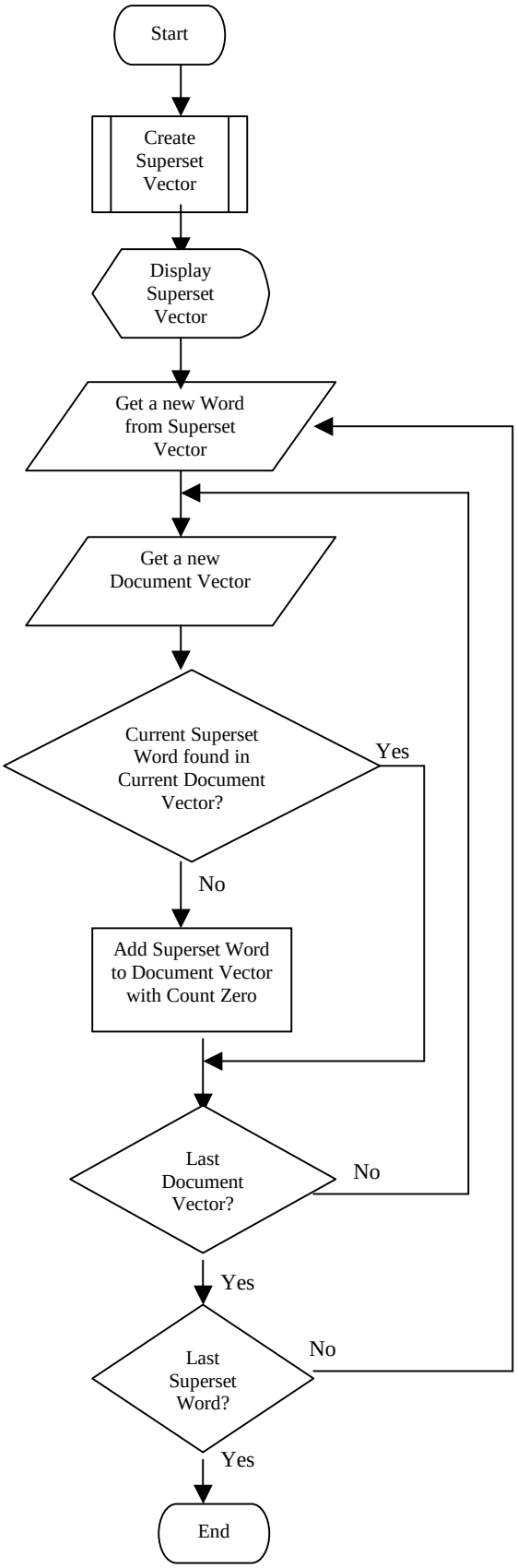

Figure (2) – Dimension Space Construction.

Word Name	F	Word Name	F	Word Name	F	Word Name	F	Word Name	F	Word Name	F
feas	1	studi	1	cryptographi	2	assist	1	app	2	year	1
squar	1	arithmet	4	ris	1	trivi	1	degr	2	computer	2
diagonal	1	geometri	7	carri	1	develop	2	nineteen	1	theorist	2
matrix	1	contain	1	propos	2	integer	2	formul	1	comput	1
chang	1	easi	1	research	2	odd	1	r	2	princip	1
spectrum	3	class	2	basi	1	bound	1	id	1	suggest	1
fract	1	ago	1	discrimin	1	rol	1	abelian	2	includ	1
combinator	1	play	1	purpos	1	virtu	1	consider	1	depend	1
hand	2	algorithm	10	ti	1	preciou	1	add	1	tantal	1
mathem	1	appear	1	combin	1	group	2	wiless	1	context	1
knowledg	1	form	2	altogether	1	diagon	1	exc	1	featur	1
gradu	2	defin	1	serv	1	funct	1	rar	1	express	1
stud	3	second	1	interest	1	perspect	1	futur	1	properti	1
thesi	1	estim	2	product	2	pract	1	suit	1	plac	1
indefin	1	gorenstein	1	symmetr	1	breakthrough	1	area	4	learn	1
differ	1	centr	1	pur	1	expert	1	quest	4	object	2
aspir	1	theorem	1	way	1	inspir	2	unimodular	1	acquir	1
commod	1	sampl	1	number	18	set	2	answer	1	belong	2
compos	1	geometr	2	algebra	12	commut	2	rich	1	investig	5
repres	1	fair	1	problem	6	cur	1	not	1	attract	1
numbertheoret	2	theori	17	19	1	narrow	1	adv	1	inner	2
nic	1	view	1	homolog	1	upper	1	broad	1	spac	1
elementari	1	subject	1	collect	1	appl	4	unend	1	equiv	1
analyt	1	mainstream	1	choos	1	inv	1	pos	1	better	1
project	3	quadr	1	expect	1	theoret	1	known	1	varieti	1
method	1	field	2	order	5	der	1	display	1	strength	1
interact	1	typ	2	nee	1	standard	1	good	2	requir	1
work	1	sum	1	involv	1	andrew	1	techniqu	3	decad	1

Table (1) – The Document Vector of a NSF abstract document.

As shown in figures (3) and (4), for R = 10% and 20% respectively and for different values of A, the more we increase the pruning threshold, P, in other words, the more we reduced the document vectors, the less the number of created agglomerative clusters obtained (k value). This is due to the removal of many words from the document vectors that distinguish documents from one another; therefore, document vectors containing no words at high pruning thresholds will be agglomerated together in one cluster although they might be of different subjects. At lower values of P, k is so high, hence it loses its significance since all the document's words are included in its vector except for the least occurring. Hence, the probability of words common to several documents being present is high. It is natural to assume that the most frequent words in a document are those which best describe it. Therefore, at higher pruning distances between documents decrease resulting in lower k values.

By observing the effect of the agglomerative distance threshold, A, on the k value for R = 10% and 20% respectively and for different P values, we found that the more we increase the agglomerative distance threshold, A, the less the number of created agglomerative clusters obtained (k value). This is because we increased the accepted distance between the centroid vectors of the clusters to be merged together, and so the k value will be decreased. Figures (5) and (6) show the obtained results. At A = 20 and P = 7 results lead to virtually the same k as these resulting from other values of P. As we decrease A, the variance in k increases. At A = 15 the variance is not so large therefore, we can obtain a good estimate of k while using P of 6 or higher leading to greater decreases in runtime values. Table (4) contains the sizes of the created matrices of the dimension spaces constructed, where rows represent the number of contained words (obtained from the superset vector) and columns represent the number of considered document vectors. The following can now be stated:

1. At larger values of the agglomerative threshold, A, the effect of pruning threshold, P, diminishes, thus allowing for high values of P to be used and hence, smaller document vectors.
2. On changing the pruning threshold, P, the agglomerative threshold, A, is still effective for both the sample sizes, R = 10%, 20%, but still its variations at large A's are minimal.

Word Name	Freq.
algorithm	10
theori	17
Number	18
Algebra	12

Table (2) – The Document Vector of the same NSF abstract document after pruning by 7.

Word Name	Freq	Word Name	Freq	Word Name	Freq	Word Name	Freq	Word Name	Freq	Word Name	Freq	Word Name	Freq
impl	10	lego	11	laser	19	plateau	10	deform	27	receptor	8	new	26
mathem	10	fold	9	endosperm	13	video	18	molecular	8	cluster	19	glob	10
drug	8	star	39	auxin	11	cours	16	physiologi	10	pollen	22	guid	9
ida	8	switch	9	knox1	10	film	22	ultrasound	10	el	12	tissu	14
atom	9	scientif	11	diffus	9	stress	22	theori	88	fram	9	develop	19
cloud	18	dna	30	chlorid	26	radio	23	propos	10	sit	10	fir	11
joid	8	qlc	8	children	10	oxid	8	solv	19	human	18	univers	11
nuclear	8	nonlinear	10	queri	13	metabol	16	mistak	8	aqm	12	process	39
unc7	11	studi	31	disinfect	8	algorithm	18	vessel	14	network	33	smet	9
perceptu	24	manipul	10	test	30	random	17	dhp	8	react	38	assist	17
risk	12	biochem	8	sect	8	met	8	erupt	8	support	20	isozym	10
confer	11	train	8	superlatt	9	bia	9	vent	13	speci	9	mt2	19
til	22	cell	61	electr	10	recogn	23	drill	26	n	20	mrna	10
statist	19	mid	9	model	67	aleuron	12	testosteron	11	genom	19	cooper	8
chang	27	galaxi	9	equ	31	regul	8	mut	26	chemistri	8	equip	8
indulg	8	predict	8	treat	8	map	9	nt	16	ccn	9	perform	8
patch	8	root	12	geometri	8	axon	20	languag	10	surfac	16	sensor	13
optim	27	scienc	43	obliqu	9	plasmid	8	ncar	10	polymer	69	teacher	11
qualityofserv	8	method	25	electron	18	rainfal	12	research	71	number	18	math	8
gender	9	diamond	12	soybean	13	steel	8	measur	35	algebra	33	zeol	8
tim	12	combust	8	agp	8	gravel	10	energi	9	problem	70	srnas	8
coast	26	project	70	linear	12	ligo	8	schedul	9	magnet	9	vacuol	15
brain	8	repres	20	estim	17	commun	23	block	8	layer	9	gen	32
ionotrop	8	mo	8	adsorpt	12	quantum	17	pol	16	storag	10	lim	34
environ	10	stat	8	part	9	antarct	10	serv	31	sea	8	nat	8
strip	9	mhd	10	center	8	ecosystem	8	si	24	instru	18	hysteresi	9
crust	28	intern	8	river	32	defens	9	ocean	19	discours	9	exhib	10
stud	52	behavior	10	modul	11	prepared	8	sequ	16	mechan	26	assemb	14
anisotrop	8	wireless	10	organ	19	fern	8	wall	13	hydrophob	9	ferroelectr	9
group	22	molecul	9	appl	9	act	18	liquid	33	charg	10	er	12
pattern	38	immigr	9	cft	8	polarimetr	9	data	48	cost	8	program	74
calcium	8	ga	8	Indepth	8	polyelectrolyt	14	subst	8	texa	12	aerosol	11
beneath	8	tomato	8	ic	69	chain	10	internet	27	clim	8	stochast	9
lithospher	8	penetr	8	manifold	12	compli	9	opt	16	lmc	8	area	8
mangan	8	loc	27	control	8	id	10	chem	8	futur	11	prep	8
graph	9	law	8	scal	30	issu	8	immun	17	damper	8	power	11
imag	36	transform	9	temperatur	8	market	9	concret	20	numer	9	disk	10
lectin	11	cycl	13	materi	49	publ	10	nutri	26	learn	32	nitrou	8
mobil	25	system	34	complex	9	interspecif	8	neurofila	8	concern	10	calgebra	10
sequenc	9	team	8	protein	46	express	8	radar	9	tsunami	8	impuls	9
telomer	9	flow	11	comput	18	leaf	12	properti	18	jakobid	9	congest	10
judg	18	structur	19	oper	35	water	8	aug	10	prefer	9	peopl	11
rank	9	arc	9	media	13	dev	23	contact	8	design	20	let6	10
lak	23	cat	12	engineer	50	soil	14	experi	36	hifu	13	north	9
endocytosi	11	dynam	18	2	9	investig	10	erad	14	enzym	12	spac	21

Table (3) – The Superset Vector of the NSF data set after pruning by 7.

Matrices Size	Sample size = 10%	Sample size = 20%	k-means Dimension Space matrices size
Pruning Threshold = 2	2287 X 72	2287 X 145	2287 X 725
Pruning Threshold = 3	1450 X 72	1450 X 145	1450 X 725
Pruning Threshold = 4	986 X 72	986 X 145	986 X 725
Pruning Threshold = 5	696 X 72	696 X 145	696 X 725
Pruning Threshold = 6	464 X 72	464 X 145	464 X 725
Pruning Threshold = 7	348 X 72	348 X 145	348 X 725

Table (4) – The matrices size of the constructed dimension spaces, where rows represent the number of words (obtained from the superset vector) and columns represent the number of document vectors

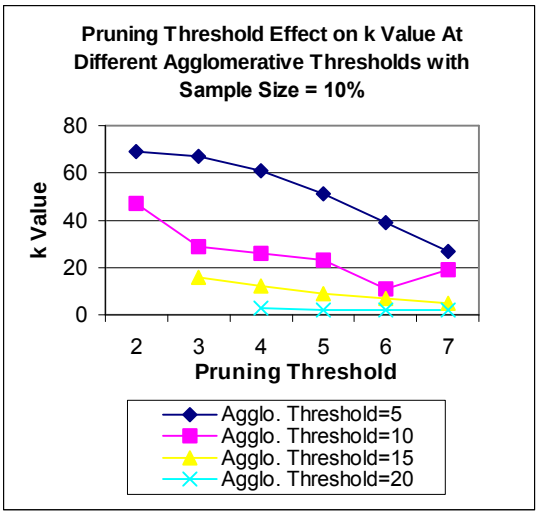

Figure (3)

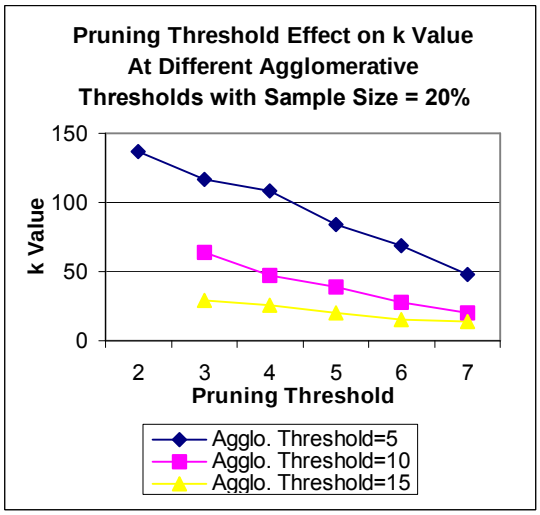

Figure (4)

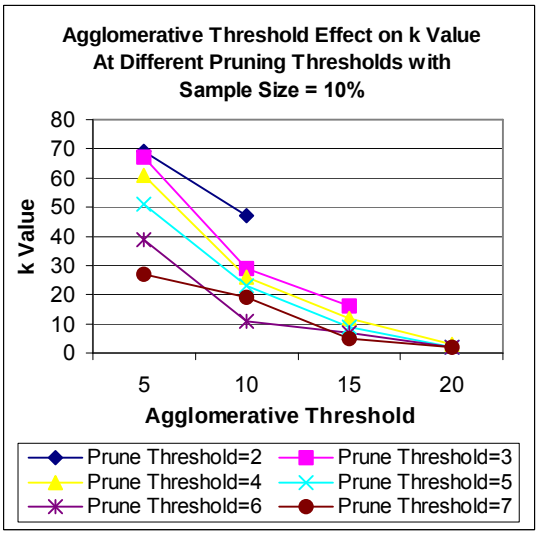

Figure (5)

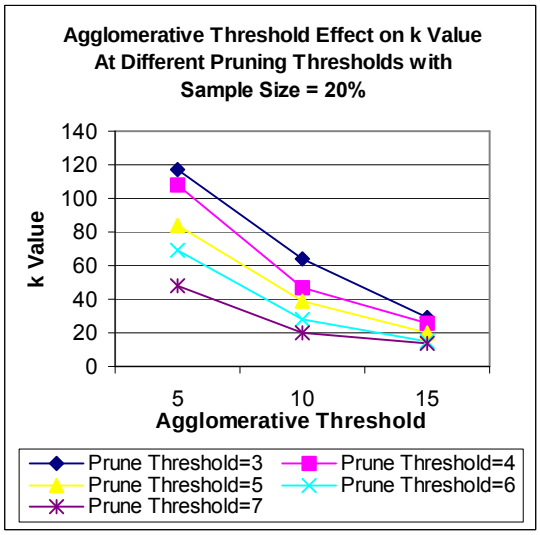

Figure (6)

6 Conclusion

Unlabeled document collections are becoming increasingly common and available; mining such data sets represents a major contemporary challenge. Practical approaches to clustering such document vectors use an iterative procedure (e.g. k-means) that is known to be especially sensitive to initial starting conditions (k and initial centroids).

In this paper, we introduced a hybrid clustering algorithm that determines these initial conditions automatically, depending on the required quality for the obtained clusters. The hybrid algorithm combines the agglomerative hierarchical approach with the k-means approach to provide k disjoint clusters. An experimental study was conducted to evaluate the different parameters affecting both the document vector sizes and the choice of k: specifically the pruning threshold, the agglomerative threshold for the sample sizes of 10% and 20%. It was shown that the effect of pruning thresholds is dominant over agglomerative thresholds for providing better convergence towards determining k.

References

[1] J. Han, M. Kamber. *Data Mining Concepts and Techniques.* Morgan Kaufmann Publishers. Pages 335 – 388 and 428 – 435, 2001.

[2] I. S. Dhillon and D. S. Modha. *Concept Decompositions for Large Sparse Text Data using Clustering*. Technical Report RJ 10147, IBM Almadan Research center, 2000.

[3] M. Steinbach, G. Karypis, ans V. Kumar. *A Comparison of Document Clustering Techniques.* In KDD workshop on Text Mining, 2000.
[4] J.L. Neto, A.D. Santos, C.A.A. Kaestner, and A.A. Freitas. *Document Clustering and Text Summarization.* In Proceedings, 4th International Conference on Practical Applications of Knowledge Discovery and Data Mining (PADD-2000), 41-55. London: The Practical Application Company, 2000.
[5] P.S. Bradley, U.M. Fayyad, C. A. Reina. *Scaling EM (Expectation Maximization) Clustering to Large Databases*. (Technical Report 98-35), 1999.
[6] N. Turenne and F. Rousselot. *Evaluation of Four Clustering Methods used in Text Mining*, 1999.
[7] P.S. Bradley, and U.M. Fayyad, *Refining Initial Points for K-Means Clustering.* Proceedings of the15th Intertional Conference on Machine Learning, 91-99. Morgan Kaufmann Publishers, Inc., San Francisco, CA.,1998.
[8] Rajman and Besanon, *Text Mining –Knowledge extraction from unstructured textual data*, 6th Conference of International Federation of Classification Societies (IFCS-98), Rome, 1998.
[9] M. Meila, and D. Hackerman, *An Experimental Comparison of Several Clustering and Initialization Methods,* (Technical Report 98-06). Microsoft Research Redmond, WA, 1998.
[10] C. Wei, Y. Lee and C. Hsu. *Empirical Comparison of Fast Clustering Algorithms for Large Data Sets*. Proceedings of the 33rd Hawaii International Conference on System Sciences, 1998.
[11] H. Ahonen, O. Heinonen, M. Klemettinen, and A. L. Verkamo. *Applying Data Mining Techniques in Text Analysis.* In Report C-1997-23, Dept. of Computer Science, University of Helsiniki, 1997.
[12] M. J. A. Berry, G. Linoff. *Data Mining Techniques for Marketing, Sales, and Customer Support.* John Wiley & Sons. Pages 187 – 215, 1997.
[13] M.F. Porter, *An Algorithm for Suffix Stripping*, Program 14 (3), Pages 130-137, July 1980.