Analysis of Various Pre-processing and Clustering Algorithms on Text Data Sets for Finding Similarity through Computer Inspection

Mukkapati L Koteswararao [#1], Bommireddy Dinesh Reddy [*2]

M.Tech Scholar [#1], Associate Professor, M.Tech (PhD)[*2]

Department of Computer Science & Engineering,

Vignan Institute of Information Technology,

Visakhapatnam, AP, India.

Abstract

Forensic Data Analysis (FDA) is a branch of Digital forensics. It examines *structured* data with regard to incidents of financial crime. The aim is to discover and analyse patterns of fraudulent activities. Data from application systems or from their underlying databases is referred to as structured data. *Unstructured data* in contrast is taken from communication and office applications or from mobile devices. This data has no overarching structure and analysis thereof means applying keywords or mapping communication patterns. Analysis of unstructured data is usually referred to as forensics. In computer forensic analysis, hundreds of thousands of files are usually examined. Much of the data in those files consists of unstructured text, whose analysis by computer examiners is difficult to be performed. In this context, automated methods of analysis are of great interest. We present an approach that applies document clustering algorithms to forensic analysis of computers seized in police investigations. We illustrate the proposed approach by carrying out extensive experimentation with three well-known clustering algorithms (K-means, K-medoids, Single Link, Complete Link, Average Link, and Cosine Similarity Measure) applied on some of the text documents of having different categories like games, sports, living beings, animals, food items, books and so on. In addition, two relative validity indexes were used to automatically estimate the number of clusters. Related studies in the literature are significantly more limited than our study. Our experiments show that the proposed architecture is best suited for forensic analysis of unstructured documents.

Keywords

Clustering, Data Mining, Clustering, Structured Data, Un-Structured Data.

1. Introduction

Data mining is the discovery of the unknown patterns from both heterogeneous and homogeneous database. Secure Data Mining helps to discover association rules which are being shared by homogeneous databases (same schema but the data is present on different entities).The algorithm not only finds the union and intersection of association rules with support and confidence which hold in the total database, while ensuring the data held by players to be authenticated.

It is estimated that the volume of data in the digital world increased from 161 hexabytes in 2007 to 998 hexabytes in 2011 [1]—about 18 times the amount of information present in all the books ever written—and it continues to grow exponentially. This large amount of data has a direct impact in *Computer Data Inspection*, which can be broadly defined as the discipline that combines several elements of data and computer science to

collect and analyze data from computer systems in a way that is admissible as the data should have similarities between several collected data fields. In our particular application domain, it usually involves examining hundreds of thousands of files per computer. This activity exceeds the expert's ability of analysis and interpretation of data. Therefore, methods for automated data analysis, like those widely used for machine learning and data mining, are of paramount importance. In particular, algorithms for pattern recognition from the information present in **text** documents are promising, as it will hopefully become evident later in the paper.

Acronym	Algorithm	Attributes	Distance	Initialization	K-estimate
Kms	*K-means*	Cont. (all)	Cosine	Random	Simp. Sil.
Kms100	*K-means*	100 > TV	Cosine	Random	Simp. Sil.
Kms100*	*K-means*	100 > TV	Cosine	[18]	Simp. Sil.
KmsT100*	*K-means*	100 > TV	Cosine	[18]	Silhouette
KmsS	*K-means*	Cont. (all)	Cosine	Random	Rec. Sil.
Kms100S	*K-means*	100 > TV	Cosine	Random	Rec. Sil.
Kmd100	*K-medoids*	100 > TV	Cosine	Random	Silhouette
Kmd100*	*K-medoids*	100 > TV	Cosine	[18]	Silhouette
KmdLev	*K-medoids*	Name	Lev.	Random	Silhouette
KmdLevS	*K-medoids*	Name	Lev.	Random	Rec. Sil.
AL100	*AverageLink*	100 > TV	Cosine	-	Silhouette
CL100	*CompleteLink*	100 > TV	Cosine	-	Silhouette
SL100	*SingleLink*	100 > TV	Cosine	-	Silhouette
NC	CSPA	Name, Cont. (all)	CSPA	Random	Simp. Sil.
NC100	CSPA	Name,100 > TV	CSPA	Random	Simp. Sil.
E100	CSPA	Cont.100 random	CSPA	Random	Simp. Sil.

100 > TV: 100 attributes (words) that have the greatest variance over the documents
Cont. 100 random: 100 randomly chosen atributes from document content
Cont. (all): all features from document content
Lev.: Levenshtein distance
Simp. Sil.: Simplified Silhpuette
Rec. Sil.: "Recursive" Silhouette
*: Initialization on distant objects
Name: file name

TABLE I
SUMMARY OF ALGORITHMS AND THEIR PARAMETERS

Clustering algorithms are typically used for exploratory data analysis, where there is little or no prior knowledge about the data [2], [3]. This is precisely the case in several applications of *Computer Data Inspection*, including the one addressed in our work. From a more technical viewpoint, our datasets consist of unlabeled objects—the classes or categories of documents that can be found are *a priori* unknown. Moreover, even assuming that labeled datasets could be available from previous analyses, there is almost no hope that the same classes (possibly learned earlier by a classifier in a supervised learning setting) would be still valid for the upcoming data, obtained from other computers and associated to different investigation processes. More precisely, it is likely that the new data sample would come from a different population.

In this context, the use of clustering algorithms, which are capable of finding latent patterns from text documents found in seized computers, can enhance the analysis performed by the expert examiner. Clustering algorithms have been studied for decades, and the literature on the subject is huge. Therefore, we decided to choose a set of several representative algorithms in order to show the potential of the proposed approach, namely: the partitional K-means [3] and K-medoids [4], the hierarchical Single/Complete/Average Link [5], and the cluster ensemble algorithm known as CSPA [6] and also Cosine similarity function. These algorithms were run with different combinations of their parameters, resulting in various different algorithmic instantiations, as shown in Table I. Thus, as a contribution of our work, we compare their relative performances on the studied application domain—using different sample text data sets containing information like sports, food habits, culture and animals.

2. Related Work

In this section we will describe the assumptions that are used in the proposed paper.

2.1 Text Mining

Text mining, also referred to as *text data mining*, roughly equivalent to text analytics, refers to the process of deriving high-quality information from text. High-quality information is typically derived through the devising of patterns and trends through means such as statistical pattern learning. Text mining usually involves the process of structuring the input text (usually parsing, along with the addition of some derived linguistic features and the removal of others, and subsequent insertion into a database), deriving patterns within the

structured data, and finally evaluation and interpretation of the output. 'High quality' in text mining usually refers to some combination of relevance, novelty, and interestingness. Typical text mining tasks include text categorization, text clustering, concept/entity extraction, production of granular taxonomies, sentiment analysis, document summarization, and entity relation modeling (*i.e.*, learning relations between named entities).

Text analysis involves information retrieval, lexical analysis to study word frequency distributions, pattern recognition, tagging/annotation, information extraction, data mining techniques including link and association analysis, visualization, and predictive analytics. The overarching goal is, essentially, to turn text into data for analysis, via application of natural language processing (NLP) and analytical methods.

A typical application is to scan a set of documents written in a natural language and either model the document set for predictive classification purposes or populate a database or search index with the information extracted.

2.2 Data Preprocessing Technique

Data pre-processing is an often neglected but important step in the data mining process. The phrase "Garbage In, Garbage Out" is particularly applicable to data mining and machine learning. Data gathering methods are often loosely controlled, resulting in out -of-range values (e.g., Income: 100), impossible data combinations (e.g., Gender: Male, Pregnant: Yes), missing values, etc. Analyzing data that has not been carefully screened for such problems can produce misleading results. Thus, the representation and quality of data is first and foremost before running an analysis.

If there is much irrelevant and redundant information present or noisy and unreliable data, then knowledge discovery during the training phase is more difficult. Data pre-processing includes cleaning, normalization, transformation, feature extraction and selection, etc. The product of data pre -processing is the final training set.

Data Preprocessing Methods

Raw data is highly susceptible to noise, missing values, and inconsistency. The quality of data affects the data mining results. In order to help improve the quality of the data and, consequently, of the mining results raw data is pre-processed so as to improve the efficiency and ease of the mining process. Data preprocessing is one of the most critical steps in a data mining process which deals with the preparation and transformation of the initial dataset. Data preprocessing methods are divided into following categories:

1. **Data Cleaning**
2. **Data Integration**
3. **Data Transformation**
4. **Data Reduction**

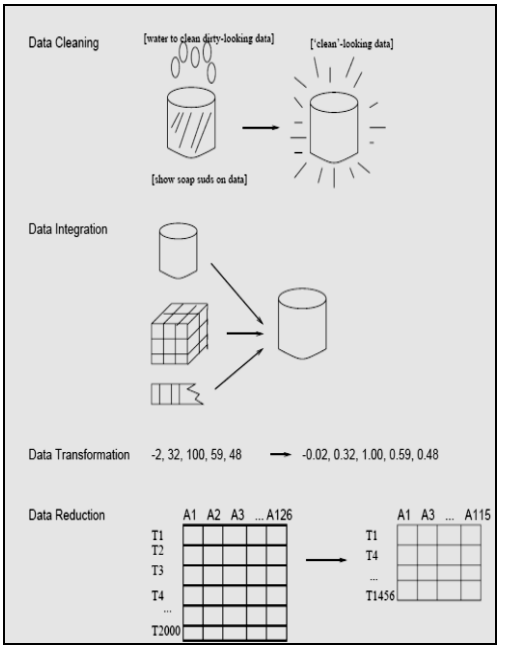

Figure 1.Various Forms of Pre-Processing

3. Proposed Methodologies

In this section we will discuss the proposed methodologies what we have used in the current paper and its pre-processing techniques that are used for making the data set free from errors for clustering**.**

3.1 Pre-Processing Steps

Before running clustering algorithms on text datasets, we performed some preprocessing steps. In particular, ***stopwords*** (prepositions, pronouns, articles, and irrelevant document metadata) have been removed. Also, the **Snowball *stemming* algorithm** for Portuguese words has been used. Then, we adopted a traditional statistical approach for text mining, in which documents are represented in a vector space model [7]. In this model, each document is represented by a vector containing the frequencies of occurrences of words, which are defined as delimited alphabetic strings, whose number of characters is between 4 and 25. We also used a dimensionality reduction technique known as Term Variance (TV) [8] that can increase both the effectiveness and efficiency of clustering algorithms. TV selects a number of attributes (in our case 100 words) that have the greatest variances over the documents. In order to compute distances between documents, two measures have been used, namely: cosine-based distance [7] and Levenshtein-based distance [9]. The later has been used to calculate distances between file (document) names only.

3.2 Estimating the Number of Clusters from Data

In order to estimate the number of clusters, a widely used approach consists of getting a set of data partitions with different numbers of clusters and then selecting that particular partition that provides the best result according to a specific quality criterion (e.g., a relative validity index [2]–[5]). Such a set of partitions may result directly from a hierarchical clustering dendrogram or, alternatively, from multiple runs of a partitional algorithm (e.g., K-means) starting from different numbers and initial positions of the cluster prototypes (e.g., see [7] and references therein).
For the moment, let us assume that a set of data partitions with different numbers of clusters is available, from which we want to choose the best one—according to some relative validity criterion. Note that, by choosing such a data partition, we are performing model selection and, as an intrinsic part of this process, we are also estimating the number of clusters.A widely used relative validity index is the so-called *silhouette* [4], which has also been adopted as a component of the algorithms employed in our work. Therefore, it is helpful to define it even before we address the clustering algorithms used in our study.

Let us consider an object belonging to cluster **A.** The average dissimilarity of *i* to all other objects of **A** is denoted by *a(i).* Now let us take into account cluster **C** . The average dissimilarity of *i* to all objects of **C** will be called **d(i,C)** . After computing **d(i,C)** for all clusters **C** ≠ **A** , the smallest one is selected, i.e. b(i) = min d(i,C), **C** ≠ **A**. This value represents the dissimilarity of *i* to its neighbor cluster, and the silhouette for a give object, s(i), is given by:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

It can be verified that -1 ≤ s(i) ≤ 1 . Thus, the higher s(i) the better the assignment of object *i* to a given cluster. In addition, if s(i) is equal to zero, then it is not clear whether the object should have been assigned to its current cluster or to a neighboring one [4]. Finally, if cluster **A** is a singleton, then s(i) is not defined and the most neutral choice is to set s(i)=0.Once we have computed s(i) over i= 1,2,……..N, where N is the number of objects in the dataset, we take the average over these values, and the resulting value is then a quantitative measure of the data partition in hand. Thus, the best clustering corresponds to the data partition that has the maximum average silhouette.

3.3 Clustering Algorithms

The clustering algorithms adopted in our study—the partitional K-means [2] and K-medoids [4], the hierarchical Single/Complete/Average Link [5], and the cluster ensemble based algorithm known as CSPA [6]—are popular in the machine learning and data mining fields, and therefore they have been used in our study. Nevertheless, some of our choices regarding their use deserve further comments. For instance, K-medoids [4] is similar to K-means. However, instead of computing centroids, it uses medoids, which are the representative objects of the clusters. This property makes it particularly interesting for applications in which (i) centroids cannot be computed; and (ii) distances between pairs of objects are available, as for computing dissimilarities between names of documents with the Levenshtein distance [9].

Considering the partitional algorithms, it is widely known that both K-means and K-medoids are sensitive to initialization and usually converge to solutions that represent local minima. Trying to minimize these problems, we used a nonrandom initialization in which distant objects from each other are chosen as starting prototypes [10]. Unlike the partitional algorithms such as K-means/medoids, hierarchical algorithms such as Single/ Complete/Average Link provide a hierarchical set of nested partitions [3], usually represented in the form of a dendrogram, from which the *best* number of clusters can be estimated. In particular, one can assess the quality of every partition represented by the dendrogram, subsequently choosing the one that provides the best results [7].

For the hierarchical algorithms (Single/Complete/Average Link), we simply run them and then assess every partition from the resulting dendrogram by means of the silhouette [4]. Then, the best partition (elected according to the relative validity index) is taken as the result of the clustering process. For each partitional algorithm (K-means/medoids), we execute it repeatedly for an increasing number of clusters. The CSPA algorithm [6] essentially finds a consensus clustering from a cluster ensemble formed by a set of different data partitions. More precisely, after applying clustering algorithms to the data, a similarity (**coassociation**) matrix [11] is computed. Each element of this matrix represents pair-wise similarities between objects. The similarity between two objects is simply the fraction of the clustering solutions in which those two objects lie in the same cluster. Later, this similarity measure is used by a clustering algorithm that can process a proximity matrix—e.g., K-medoids—to produce the final consensus clustering. The sets of data partitions (**clusterings**) were generated in two different ways: (a) by running K-means 100 times with different subsets of attributes (in this case CSPA processes 100 data partitions); and (b) by using only two data partitions, namely: one obtained by K-medoids from the dissimilarities between the file names, and another partition achieved with K-means from the vector space model. In this case, each partition can have different weights, which have been varied between 0 and 1 (in increments of 0.1 and keeping their sum equals to 1).

3.4 Outliers Removal Process

We assess a simple approach to remove *outliers*. This approach makes recursive use of the *silhouette*. Fundamentally, if the best partition chosen by the silhouette has singletons (i.e., clusters formed by a single object only), these are removed. Then, the clustering process is repeated over and over again—until a partition without singletons is found. At the end of the process, all singletons are incorporated into the resulting data partition (for evaluation purposes) as single clusters. Table I summarizes the clustering algorithms used in our work and their main characteristics.

4. System Architecture

A system architecture or systems architecture is the conceptual design that defines the structure and/or behavior of a system. An architecture description is a formal description of a system, organized in a way that supports reasoning about the structural properties of the system. It defines the system components or building blocks and provides a plan from which products can be procured, and systems developed, that will work

together to implement the overall system. This may enable one to manage investment in a way that meets business needs. The fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution. The composite of the design architectures for products and their life cycle processes. A representation of a system in which there is a mapping of functionality onto hardware and software components, a mapping of the software architecture onto the hardware architecture, and human interaction with these components. An allocated arrangement of physical elements which provides the design solution for a consumer product or life-cycle process intended to satisfy the requirements of the functional architecture and the requirements baseline. Architecture is the most important, pervasive, top-level, strategic inventions, decisions, and their associated rationales about the overall structure (i.e., essential elements and their relationships) and associated characteristics and behavior. This is clearly shown in figure 2.

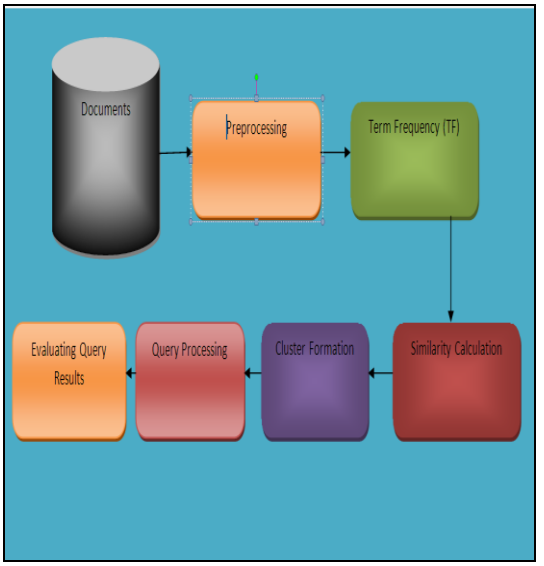

Figure 2.Architecture Flow of Proposed Computer Data Inspection

5. Conclusion

In this paper we have implemented various pre-processing techniques and clustering algorithms in order to cluster the text data sets for computer inspection. Of all these clustering algorithms our application is very accurate in finding clusters of text document by using K-Means and Incremental Clustering algorithms. By conduction several experiments on this proposed text data sets our application clearly tells that with this suite of algorithms we can inspect the computer having a lot of text documents in order to find the similarity between two or more that documents easily.

6. References

[1] J. F. Gantz, D. Reinsel, C. Chute, W. Schlichting, J. McArthur, S. Minton, I. Xheneti, A. Toncheva, and A. Manfrediz, "The expanding digital universe: A forecast of worldwide information growth through 2010," *Inf. Data*, vol. 1, pp. 1–21, 2007.

[2] B. S. Everitt, S. Landau, and M. Leese, *Cluster Analysis*. London, U.K.: Arnold, 2001.

[3] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[4] L. Kaufman and P. Rousseeuw, *Finding Groups in Gata: An Introduction to Cluster Analysis*. Hoboken, NJ: Wiley-Interscience, 1990.

[5] R. Xu and D. C.Wunsch, II, *Clustering*. Hoboken, NJ: Wiley/IEEE Press, 2009.

[6] A. Strehl and J. Ghosh, "Cluster ensembles: A knowledge reuse framework for combining multiple partitions," *J. Mach. Learning Res.*, vol. 3, pp. 583–617, 2002.

[7] G. Salton and C. Buckley, "Term weighting approaches in automatic

text retrieval," *Inf. Process. Manage.*, vol. 24, no. 5, pp. 513–523, 1988.

[8] L. Liu, J. Kang, J. Yu, and Z. Wang, "A comparative study on unsupervised feature selection methods for text clustering," in *Proc. IEEE Int. Conf. Natural Language Processing and Knowledge Engineering*, 2005, pp. 597–601.

[9] V. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Physics Doklady*, vol. 10, pp. 707–710, 1966.

[10] B. Mirkin, *Clustering for Data Mining: A Data Recovery Approach*. London, U.K.: Chapman & Hall, 2005.

[11] A. L. N. Fred and A. K. Jain, "Combining multiple clusterings using evidence accumulation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 6, pp. 835–850, Jun. 2005.

7. About the Authors

Mukkapati L Koteswararao is currently pursuing his 2 Years M.Tech (Software Engineering) in Department of Computer Science and Engineering at Vignan Institute of Information Technology, Visakhapatnam. His area of interests includes Data Mining.

Bommireddy Dinesh Reddy is currently working as Associate Professor in Department of Computer Science and Engineering at Vignan Institute of Information Technology, Visakhapatnam. He is currently doing his PhD in the relevant field of Computer Science. His research interests include Networks Security, Information Security and Data Mining.