International Journal of Engineering Research-Online
A Peer Reviewed International Journal
Articles available online http://www.ijoer.in

Vol.3., Issue.2, 2015

RESEARCH ARTICLE

ISSN: 2321-7758

EXAMINE THE MEMO DRIVE USING CATCHWORDS IN SMARTPHONES

G.ISWARYA*,B.GAYATHIRI, T.GAYATHRI, MRS J.JOSEPHA MENANDASM.E
Department of Computer Science and Engineering
Panimalar Engineering College

Article Received: 07/03/2015 | Article Revised on14 /03/2015 | Article Accepted on:18/03/2015

ABSTRACT
Daily everyone receives a lot of SMS on their mobile from friends, Internet Banking related, subscription messages, confirmation message and lot many others. The size of your SMS grows day by day. There is no way to find the required message from the mobile in-box in a short period of time. This application provides short period of time to search the desired memo from the android device even though the messages are deleted in the inbox and also give the additional security to the user which will take the backup of all SMS from the android device and also available for the personal backup which will also give the search option to the user. It will delete the unwanted messages automatically which have the user defined Keywords without user interaction. In this paper, we design an algorithm which combines classical partitioning algorithms with probabilistic models in order to create an effective clustering approach.
Index Terms—classical partitioning,probablistic model.

I. INTRODUCTION

Android is an open source and Linux-based operating system for mobile devices such as smartphones and tablet computers. Android was developed by the *Open Handset Alliance*, led by Google, and other companies[1]. Android offers a unified approach to application development for mobile devices which means developers need only develop for Android, and their applications should be able to run on different devices powered by Android. With a user interface based on direct manipulation, Android is designed primarily for touch screen mobile devices such as smart phones and tablet computers, with specialized user interfaces. Android is popular with technology companies which require a ready-made, low-cost and customizable operating system for high-tech devices. Android's open nature has encouraged a large community of developers and enthusiasts to use the open-source code as a foundation for community-driven projects, which add new features for advanced users or bring Android to devices which were officially released running other operating systems. The operating system's success has made it a target for patent litigation as part of the so-called "Smartphone wars" between technology companies.

DATAMINING

Data mining (the analysis step of the "Knowledge Discovery in Databases" process, or KDD), an

interdisciplinary subfield of computer, is the computational process of discovering patterns in large data sets involving methods at the intersection of artificial, machine learning, statistics, and database systems[2]. The overall goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further use.

The Knowledge Discovery in Databases (KDD) process is commonly defined with the stages:

(1) Selection
(2) Pre-processing
(3) Transformation
(4) *Data Mining*
(5) Interpretation/Evaluation.

In many application domains, a tremendous amount of side information is also associated along with the documents. This is because text documents typically occur in the context of a variety of applications in which there may be a large amount of other kinds of database attributes or meta information which may be useful to the clustering process.

The main objective of this paper is to produce the application which gives the less time utilization to search the memo from their mobile and also flexible to provide the conversation backup and also give the search facility in backup to the user. It supports the additional internal memory to the user because it will delete the unwanted messages in android device [1]. In addition to that it makes automatic reply to the messages by using predefined replies.

In existing system, backup is available only with user interaction and there is no search facility in the backup. It is not possible to delete the unwanted messages in android device. Automatic replies are available only during calls not for messages.

Our proposed work provides the application this application store all messages in the database automatically and also stores the particular person's messages into the database. It makes the search option to the mobile in-box and also the application backup. It will also delete the unwanted messages in the android mobile. In addition to that it makes automatic reply to the messages by using predefined replies.

2 CLUSTERING WITH SIDE INFORMATION

In this section, we will discuss an approach for clustering text data with side information. We assume that we have a corpus *S* of text documents. The total number of documents is *N*, and they are denoted by *T*1 . . . *TN*. It is assumed that the set of distinct words in the entire corpus *S* is denoted by *W*. Associated with each document *Ti*, we have a set of side attributes *Xi*. Each set of side attributes *Xi* has *d* dimensions, which are denoted by *(xi*1 . . . *xid)*. We refer to such attributes as *auxiliary* attributes [6]. For ease in notation and analysis, we assume that each side-attribute *xid* is binary, though both numerical and categorical attributes can easily be converted to this format in a fairly straightforward way. This is because the different values of the categorical attribute can be assumed to be separate binary attributes, whereas numerical data can be discretized to binary values with the use of attribute ranges.

We note that our technique is not restricted to binary auxiliary attributes, but can also be applied to attributes of other types. When the auxiliary attributes are of other types (quantitative or categorical), they can be converted to binary attributes with the use of a simple We note that our technique is not restricted to binary auxiliary attributes, but can also be applied to attributes of other types. When the auxiliary attributes are of other types (quantitative or categorical), they can be converted to binary attributes with the use of a simple transformation process. For example, numerical data can be discretized into binary attributes. Even in this case, the derived binary attributes are quite sparse especially when the numerical ranges are discretized into a large number of attributes. In the case of categorical data, we can define a binary attribute for each possible categorical value. In many cases, the number of such values may be quite large. Therefore, we will design

our techniques under the implicit assumption that such attributes are quite sparse. The formulation for the problem of clustering with side information is as follows

Text Clustering with Side Information *Given a corpus S of documents denoted by T*1 . . . *TN, and a set of auxiliary variables Xi associated with document Ti, determine a clustering of the documents into k clusters which are denoted by C*1 . . . *Ck,based on both the text content and the auxiliary variables.*

We will use the auxiliary information in order to provide additional insights, which can improve the quality of clustering. In many cases, such auxiliary information may be noisy, and may not have useful information for the clustering process. Therefore, we will design our approach in order to magnify the coherence between the text content and the side-information, when this is detected. In cases, in which the text content and side-information do not show coherent behavior for the clustering process, the effects of those portions of the side-information are marginalized.

2.1 The COATES Algorithm

In this section, we will describe our algorithm for text clustering with side-information. We refer to this algorithm as *COATES* throughout the paper, which corresponds to the fact that it is a *COntent and Auxiliary attribute based Text cluStering* algorithm. We assume that an input to the algorithm is the number of clusters *k*. As in the case of all text-clustering algorithms, it is assumed that stop-words have been removed, and stemming has been performed order to improve the discriminatory power of the attributes [8]. The algorithm requires two phases.

Initialization We use a lightweight initialization phase in which a standard text clustering approach is used without any side-information. For this purpose, we use the algorithm described in . The reason that this algorithm is used, because it is a simple algorithm which can quickly and efficiently provide a reasonable initial starting point. The centroids and the partitioning created by the clusters formed in the first phase provide an initial starting point for the second phase. We note that the first phase is based on text only, and does not use the auxiliary information.

Main Phase The main phase of the algorithm is executed after the first phase. This phase starts off with these initial groups, and iteratively reconstructs these clusters with the use of *both* the text content and the auxiliary information. This phase performs alternating iterations which use the text content and auxiliary attribute information in order to improve the quality of the clustering. We call these iterations as *content* iterations and *auxiliary iterations* respectively. The combination of the two iterations is referred to as a *major iteration*. Each major iteration thus contains *two minor iterations*, corresponding to the auxiliary and text-based methods.

3. ARCHITECTURE DIAGRAM

In this diagram, application avert all unwanted messages and take automatic backup of all messages and personal backup is also available in this application. There is a scheduler module which will automatically reply for all the incoming messages. This application provides short period of time to search the desired memo from the android device eventhough the messages are deleted in the inbox and also give the additional security to the user which will take the backup of all SMS from the android device and also available for the personal backup which will also give the search option to the user.It will delete the unwanted messages automatically which have the user defined Keywords without user interaction.In Existing System,personalised backup is not available.An algorithm which combines classical partitioning algorithms with probabilistic models in order to create an effective clustering approach is used for the filtering process.

We assume that the *k* clusters associated with the data are denoted by *C*1 . . . *Ck*. In order to construct

a probabilistic model of membership of the data points to clusters, we assume that each auxiliary iteration has a *prior* probability of assignment of ocuments to clusters (based on the execution of the lgorithm so far), and a *posterior* probability of assignment of documents to clusters with the use of auxiliary variables in that iteration. We denote the prior probability that the document *Ti* belongs to the cluster *Cj* by *P(Ti ∈ Cj)*. Once the pure-text clustering phase has been executed, the *a-priori* cluster membership probabilities of the auxiliary attributes are generated with the use of the last content-based iteration from this phase. The *apriori* alue of *P(Ti ∈Cj)* is simply the fraction of documents which have been assigned to the cluster *Cj*. In order to compute the *posterior* probabilities *P (Ti ∈ Cj|Xi)* of membership of a record at the end of the auxiliary Iteration, we use the auxiliary attributes *Xi* which are associated with *Ti*. Therefore, we would like to compute the conditional probability *P(Ti ∈ Cj|Xi)*. We will make the approximation of considering only those auxiliary attributes (for a particular document), which take on the value of 1. Since we are focussing on sparse binary data, the value of 1 for an attribute is a much more informative event than the default value of 0. Therefore, it suffices to condition only on the case of attribute values taking on the value of 1. For example, Let us consider an application in which the auxiliary information corresponds to users which are browsing specific web pages. In such a case, the clustering behavior is influenced much more significantly by the case when a user *does* browse a particular page, rather than one in which the user *does not* browse a particular page, because most pages will typically not be browsed by a particular user.

Furthermore, in order to ensure the robustness of the approach, we need to eliminate the noisy attributes. This is especially important, when the number of auxiliary attributes is quite large. Therefore, at the beginning of each auxiliary iteration, we compute the *gini-index* of each attribute based on the clusters created by the last content based iteration. This gini-index provides a quantification of the discriminatory power of each attribute with respect to the clustering process. The gini-index is computed as follows. Let *frj* be the fraction of the records in the cluster *Cj* (created in the last content-based iteration), for which the attribute *r* takes on the value of 1. Then, we compute the *relative presence prj* of the attribute *r* in cluster *j* as follows:

$$p_{rj} = \frac{f_{rj}}{\sum_{m=1}^{k} f_{rm}}.$$

The values of *prj* are defined, so that they sum to 1 over a particular attribute *r* and different clusters *j*. We note that when all values of *prj* take on a similar value of 1*/k*, then the attribute values are evenly distributed across the different clusters. Such an attribute is not very discriminative with respect to the clustering process, and it should not be used for clustering. While the auxiliary attributes may have a different clustering behavior than the textual attributes, it is also expected that informative auxiliary attributes are at least somewhat related to the clustering behavior of the textual attributes. This is generally true of many applications such as those in which auxiliary attributes are defined either by linkage-based patterns or by user behavior. On the other hand, completely noisy attributes are unlikely to have any relationship to the text content, and will not be very effective for mining purposes. Therefore, we would like the values of *prj* to vary across the different clusters. We refer to this variation as *skew*. The level of skew can be quantified with the use of the gini-index. The gini-index of attribute *r* is denoted by *Gr*, and is defined as follows:

$$G_r = \sum_{j=1}^{k} p_{rj}^2.$$

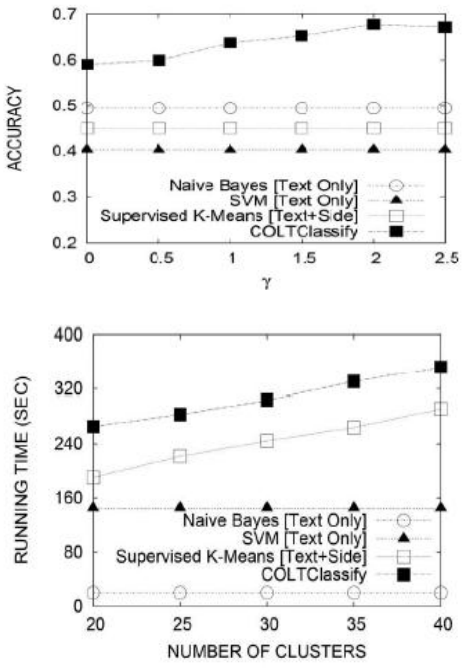
The value of *Gr* lies between 1*/k* and 1. The more discriminative the attribute, the higher the value of *Gr*. In each iteration, we use only the auxiliary attributes for which the gini-index is above a particular threshold *γ* . The value of *γ* is picked to be 1.5 standard deviations below the mean value of the gini-index in that particular iteration. We note that since the clusters may change from one iteration to the next, and the gini-index is defined with respect to the current clusters, the values of the gini-index will also change over the different iterations. Therefore, different auxiliary attributes may be used over different iterations in the clustering process, as the quality of the clusters become more refined, and the corresponding discriminative power of auxiliary attributes can also

be computed more effectively. Let *Ri* be a set containing the indices of the attributes in *Xi* which are considered discriminative for the clustering process, and for which the value of the corresponding attribute is 1. For example, let us consider an application in which we have 1000 different auxiliary attributes. If the dimension indices of the attributes in the vector *Xi* which take on the value of 1 are 7, 120, 311, and 902 respectively, then we have *Ri* = {7, 120, 311, 902}. Therefore, instead of computing $P(Ti \in Cj|Xi)$, we will compute the conditional probability of membership based on a particular value of the set *Ri*. We define this quantity as the *attribute subset based conditional probability of cluster membership*.

4.PERFORMANCE

On comparing with other algorithms like divisive algorithm, co-clustering algorithm this classical partitioning algorithm gives the better results on mining the data. It is very effective in extracting the data from large collection of data set. other algorithms uses some predefined dictionary and prespecified rows and columns. Classical partitioning algorithm is more accurate and effective than other type of clustering algorithms [8] [9] [10].

This graph shows the effectiveness and accuracy of classical partitioning algorithm when compared with Naïve-bayes,support vector machine and supervised K-means algorithms.



5. CONCLUSION AND FUTUREWORK

In this paper, we presented methods for mining text data with the use of side-information. Many forms of textdatabases contain a large amount of side-information or meta-information, which may be used in order to improve the clustering process. In order to design the clustering method, we combined an iterative partitioning technique with a probability estimation process which computes the importance of different kinds of side-information. This general approach is used in order to design both clustering and classification algorithms. We present results on real data sets illustrating the effectiveness of our approach. The results show that the use of side-information can greatly enhance the quality of text clustering.

REFERENCES

[1]. C. C. Aggarwal and H. Wang, *Managing and Mining Graph Data*.New York, NY, USA: Springer, 2010.

[2]. C. C. Aggarwal, *Social Network Data Analytics*. New York, NY,USA: Springer, 2011.

[3]. C. C. Aggarwal and C.-X. Zhai, *Mining Text Data*. New York, NY,USA: Springer, 2012.

[4]. C. C. Aggarwal and C.-X. Zhai, "A survey of text classification algorithms," in *Mining Text Data.* New York, NY, USA: Springer, 2012.

[5]. C. C. Aggarwal and P. S. Yu, "A framework for clustering massive text and categorical data streams," in *Proc. SIAM Conf. DataMining*, 2006, pp. 477–481.

[6]. C. C. Aggarwal, S. C. Gates, and P. S. Yu, "On using partial supervision for text categorization," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 2, pp. 245–255, Feb. 2004.

[7]. C. C. Aggarwal and P. S. Yu, "On text clustering with side information," in *Proc. IEEE ICDE Conf.*, Washington, DC, USA,2012.

[8]. R. Angelova and S. Siersdorfer, "A neighborhood-based approach for clustering of linked document collections," in *Proc. CIKM Conf.*, New York, NY, USA, 2006, pp. 778–779.

[9]. A. Banerjee and S. Basu, "Topic models over text streams: A study of batch and online unsupervised learning," in *Proc. SDM Conf.*, 2007, pp. 437–442.

[10]. J. Chang and D. Blei, "Relational topic models for document networks," in *Proc. AISTASIS*, Clearwater, FL, USA, 2009, pp. 81–88.