Cross domain text categorization using K-means, Expectation-Maximization algorithm

[1]Dr M.Ramakrishna Murthy, [2]P.Madhu Kumar, [3]K.Haritha, [4]P.Anuhya,[5]P. Hariteja

[1]Proffeser, [2]Student, [3]Student ,[4]Student, [5]Student

Department of Computer Science and Engineering

Anil Neerukonda Institute of Technology and Sciences, Visakhapatnam, India.

Abstract : Text Analysis is important, emerging, research area, because lots of textual content resources developing rapidly through the net and virtual world. In the textual content records evaluation textual content categorization is one among the important techniques. Traditional text categorization methods are not capable of handle well with studying across specific domains. Cross-area type is more challenging hassle than single area class .In this mission cross area text categorization is implemented using EM(expectation maximization) algorithm.

***IndexTerms* -** : cross domain , text categorization, K-means, EM-Algorithm.

I. Introduction

Text mining is the task of routinely sorting a set of documents into categories. When two or greater domains are concerned in a selected text document then it is called CROSS DOMAIN. Internet is a tremendous repository of disparate information growing at an exponential rate. The dynamic growth of net generates not only huge wide variety of text documents but additionally wide sorts of text documents in a end result of documents being generated in various domain. Efficient and effective document retrieval and classification structure are required to turn the huge amount of facts into useful information and eventually to knowledge. Applications of textual content mining are publishing and media, telecommunications, energy and other offerings industries, data era region and internet, banks, coverage and financial markets, political institutions, political analysts, public administration and prison documents, pharmaceutical and research organizations and healthcare Clustering is a procedure which partitions a given facts set into homogenous businesses primarily based on given capabilities such that similar gadgets are kept in a set whereas dissimilar gadgets are in different agencies. It is the most essential unsupervised studying problem. It offers with finding structure in a group of unlabeled data. The number of text documents are growing with the arrival of the web and development of world wide web. The huge growth of text documents are incredible to manually classify. In general statistical approaches are applied in single domain for text classification. These approaches are based on the word occurrence i.e. frequency of one or more words during a given document. But these approaches don't work well with multiple domains. So to achieving this goal one of the most important challenges is the problem of learning topics in text documents that belong to different domains. In this paper cross domain text categorization is implemented using expectation maximization algorithm.

2. Literature study:

The enormous amount of data stored in unstructured texts can-not simply be used for further processing by computers, which usually handle text as simple sequences of character strings. Therefore, specific (pre-)processing methods and algorithms are required in order to extract useful patterns. Text mining refers generally to the process of extracting interesting information and knowledge from unstructured text. This article discuss text mining as a young and interdisciplinary field within the intersection of the related area information retrieval, machine learning, statistics, linguistics and particularly data mining. It describes the main analysis tasks pre-processing, classification, clustering, information extraction and visualization. In addition. In addition, it[3] briefly discuss a number of successful applications of text mining. The area of textual content mining seeks to extract useful statistics from unstructured textual statistics via the identification and exploration of interesting patterns. The strategies employed commonly do no longer contain deep linguistic analysis or parsing, but depend upon simple "bag-of-words" textual content representations based on vector space. Several approaches[4] to the identification of styles are discussed, such as dimensionality reduction, automated class and clustering. Pattern exploration is illustrated thru two programs from our latest work: a category-based Web meta-seek engine and visualization of co-authorship relationships robotically extracted from a semi-structured series of files. Document mining is the process of deriving splendid records from huge collections of documents like news feeds, databases, or the Web. Document mining tasks consist of cluster evaluation, category, era of taxonomies, data extraction, trend identity, sentiment analysis. The Challenges in Document Mining that customers can expect are as many clusters as they become aware of topics within the end result set, the files within each cluster are semantically similar to every other, every cluster is classified intuitively. In order to obtain a satisfying solution, the state-of-artwork of concepts and algorithms from facts retrieval, unsupervised learning, information extraction, and herbal language processing need to be combined[5] in a user - focused manner. Text category is one of the core applications in facts mining because of the big amount of uncategorized textual records available. Training a text classifier effects in a class version that reflects the characteristics of the area it was discovered on. However, if no training facts is available, labelled information from a related but distinctive area might be exploited to perform cross-domain class.

Authors aim to appropriately classify unlabelled weblogs into usually agreed upon newspaper categories the usage of labelled records from the information area. The labelled information and the unlabelled blog corpus are incredibly dynamic and hourly developing with a topic drift, so the class wishes to be efficient. Experiments showed that this algorithm achieves a comparable accuracy than k-Nearest Neighbour (k-NN) and Support Vector Machines (SVM), yet at linear time cost for education and category. We[2] inspect the classifier performance and generalization ability the usage of a unique visualization of classifiers. This technique is to apply a quick novel text class algorithm to carry out efficient cross-domain category

3.Proposed methodology ：

The Expectation-Maximization (EM) algorithm may be a thanks to find maximum-likelihood estimates for model parameters when your data is incomplete, has missing data points, or has unobserved (hidden) latent variables. It is an iterative thanks to approximate the utmost likelihood function. While maximum likelihood estimation can find the "best fit" model for a group of knowledge , it doesn't work particularly well for incomplete data sets. The more complex EM algorithm can find model parameters albeit you've got missing data. It works by choosing random values for the missing data points, and using those guesses to estimate a second set of knowledge . The new values are wont to create a far better guess for the primary set, and therefore the process continues until the algorithm converges on a hard and fast point.

Algorithm:

1: pick an initial set of parameters.
2: repeat
3: Expectation Step For each object, calculate the possibility that every object belongs to distribution. i.e., calculate prob(distribution j|xi,Ɵ).
4: Maximization Step Given the probabilities from the expectancy step, find The new estimates of the parameters that maximize the expected likelihood.
5: until The parameters do no longer change.

1) Expectation (E-step)

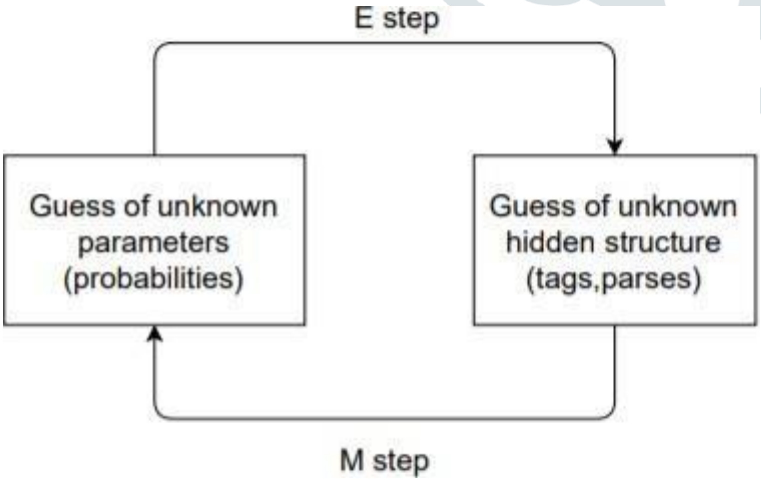
$$E[z_{i,j}] = \frac{p(x = x_i|\theta = \theta_j)}{\sum_{n=1}^{k} p(x = x_i|\theta = \theta_n)}$$

2) Maximization (M-step)

$$\theta_j = \frac{\sum_{i=1}^{m} E[z_{i,j}]x_i}{\sum_{i=1}^{m} E[z_{i,j}]}$$

If $z_{i,j}$ is known:

$$\theta_j = \frac{\sum_{i=1}^{m_j} x_i}{m_j}$$


E step
Guess of unknown parameters (probabilities)
Guess of unknown hidden structure (tags,parses)
M step

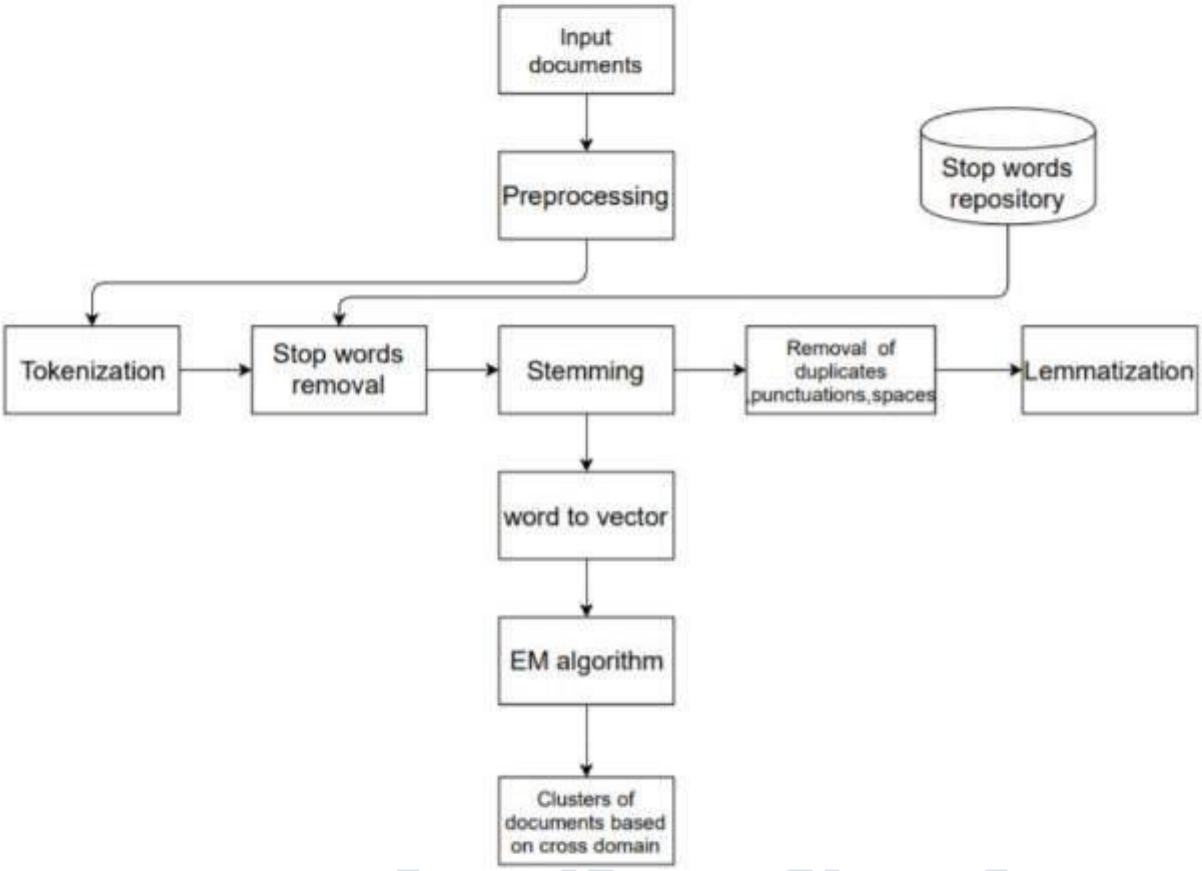


fig 1.1 System architecture

The flow of architecture is represented in the above figure 2.1. Input is taken as a set of documents. For each document pre-processing is done through the following steps. They are tokenizing the words, removal of stop words, stemming of words using porterstemmer algorithm, removal of duplicates and punctuations and lemmatization. The obtained words are converted into vectors. Then the vectors are used to implement the EM(expectation maximization) algorithm.

4.Dataset and Preprocessing:

In this module the dataset is chosen, which is a collection of text files for each State of the Union Address since the first in 1790 . Then preprocessing is performed to all files in the dataset. .In preprocessing, the text in the file is first converted into tokens then stopwords are removed .To the obtained words stemming is done through porter stemmer algorithm. Then duplicate words, punctuations, spaces are removed. Then lemmatization is performed.

Tokenization :

Tokenization is the task of chopping it up into pieces, called tokens.

Stop words Removal:

Stop words are the most common words in any natural language. For the cause of analyzing text facts and constructing NLP models, these stop words won't upload much value to the that means of the document. Consider this newsletter string– "There is a pen at the table". Now, the phrases "is", "a", "on", and "the" add no which means to the declaration even as parsing

it. Whereas words like "there", "book", and "table" are the key phrases and tell us what the statement is all about.

Stemming:

Stemming is the procedure of producing morphological editions of a root/base word. Stemming applications are commonly referred to as stemming algorithms or stemmers. A stemming set of rules reduces the words "run", "ran", "runs" to the root word, "run" and "book", "booked", "booking" lessen to the stem "book" There are mainly errors in stemming Over stemming and Under stemming. Over stemming occurs when two phrases are stemmed to equal root which can be of various stems. Under-stemming occurs when words are stemmed to identical root that are not of different stems. Applications of stemming are:

Stemming is used in records retrieval structures like search engines. used to determine domain vocabulary in domain analysis. Example: Words may be stemmed the usage of porter's stemmer, to map words with comparable roots to a single word. Root:

Automate Words that map with the foundation are automates, automation, automatic.

Lemmatization:

Lemmatization is the system of grouping together the inflected sorts of a word so they can be analyzed as a single item, diagnosed via the phrase's lemma, or dictionary form. Unlike stemming, lemmatization relies upon on correctly identifying the intended a part of speech and meaning of a word in a sentence, as nicely as within the large context surrounding that sentence, such as neighboring sentences or even a whole document. Word to vector conversion and csv file creation: Word vectors are simply vectors of numbers that represent the meaning of a word. Its input is a text corpus and its output is a set of vectors. Machine learning models take vectors(arrays of numbers) as input. Word vectors are in reality vectors of numbers that represent the which means of a phrase. Its input is a text corpus and its output is a fixed of vectors. Machine gaining knowledge of fashions take vectors (arrays of numbers) as input. Word vectors represent words as multidimensional non-stop floating factor numbers where in semantically comparable words are mapped to proximate factors in geometric space. In simpler terms, a word vector is a row

of actual valued numbers (in place of dummy numbers) where in each factor captures a size of the phrase's meaning and where semantically similar phrases have comparable vectors. This manner that phrases consisting of wheel and engine ought to have comparable phrase vectors to the phrase car (due to the similarity of their meanings), while the phrase banana have to be pretty distant. Put differently, phrases that are used in a similar context could be mapped to a proximate vector space (we are able to get to how those word vectors are created below). The splendor of representing phrases as vectors is they lend themselves to mathematical operators. For example, we can upload For instance, we can add and subtract vectors — the canonical instance here is displaying that via using phrase vectors we are able to determine that: The numbers in the word vector represent the phrase's dispensed weight across dimensions. In a simplified sense every measurement represents a meaning and the phrase's numerical weight on that dimension captures the closeness of its affiliation with and to that meaning. Thus, the semantics of the phrase are embedded across the dimensions of the vector.The words which are acquired after pre-processing are saved into a csv document and then those are transformed into vectors.

5.IMPLEMENTATION

MODULES

The modules included in our implementation are as follows

Module 1: Dataset and Pre-processing

In this module the dataset is chosen, which is a collection of news data from BBC with different category as CSV file. Then pre-processing is performed to all files in the dataset. .In pre-processing, the text in the file is first converted into tokens then stop words are removed .To the obtained words stemming is done through porter stemmer algorithm. Then duplicate words, punctuations, spaces are removed. Then lemmatization is performed.

Tokenization

Tokenization is the task of chopping it up into pieces, called tokens

Stop words Removal

Stop words are the most common words in any natural language. For the cause of analyzing text facts and constructing NLP models, these stop words won't upload much value to the that means of the document. Consider this newsletter string– "There is a pen at the table". Now, the phrases "is", "a", "on", and "the" add no which means to the declaration even as parsing it. Whereas words like "there", "book", and "table" are the key phrases and tell us what the statement is all about.

Stemming:

Stemming is that the process of manufacturing morphological variants of a root/base word. Stemming programs are commonly mentioned as stemming algorithms or stemmers. A stemming algorithm reduces the words "run", "ran", "runs" to the basis word, "run" and "book", "booked", "booking" reduce to the stem "book" There are mainly two errors in stemming – Over stemming and Under stemming.

• Over stemming occurs when two words are stemmed to same root that are of various stems.

• Under-stemming occurs when two words are stemmed to same root that aren't of various

stems.

Applications of stemming are: 3

• Stemming is employed in information retrieval systems like search engines.

• It is used to determine domain vocabulary in domain analysis. Example: Words will be

stemmed using porter's stemmer, to map words with similar roots to a single word. Root:

Automate Words that map with the root are automates, automation, automatic.

Lemmatization:

Lemmatization is the process of grouping together the inflected forms of a word so they can be analyzed as a single item, diagnosed via the phrase's lemma, or dictionary form. Unlike stemming, lemmatization relies upon on correctly identifying the intended a part of speech and meaning of a word in a sentence, as nicely as within the large context surrounding that

sentence,such as neighboring sentences or even a whole document.

Module 2: Word to vector conversion and csv file creation

Word vectors are in reality vectors of numbers that represent the which means of a phrase. Its input is a text corpus and its output is a fixed of vectors. Machine gaining knowledge of fashions take vectors (arrays of numbers) as input. Word vectors represent words as multidimensional non-stop floating factor numbers where in semantically comparable words are mapped to proximate factors in geometric space. In simpler terms, a word vector is a row of actual valued numbers (in place of dummy numbers) where in each factor captures a size of the phrase's meaning and where semantically similar phrases have comparable vectors. This manner that phrases consisting of wheel and engine ought to have comparable phrase vectors to the phrase car (due to the similarity of their meanings), while the phrase banana have to be pretty distant. Put differently, phrases that are used in a similar

context could be mapped to a proximate vector space (we are able to get to how those word vectors are created below). The splendor of representing phrases as vectors is they lend themselves to mathematical operators. For example, we can upload For instance, we can add and subtract vectors — the canonical instance here is displaying that via using phrase vectors we are able to determine that: The numbers in the word vector represent the phrase's dispensed weight across dimensions. In a simplified sense every measurement represents a meaning and the phrase's numerical weight on that dimension captures the closeness of its affiliation with and to that meaning. Thus, the semantics of the phrase are embedded across the dimensions of the vector. The words which are acquired after pre-processing are saved into a csv document and then those are transformed into vectors.

Tf -Idf Vectorizer

One issue with simple counts is that some words like "the" will appear generally and their massive counts will no longer be very meaningful in the encoded vectors. Thus an opportunity is calculate phrase frequencies, and by a long way the most popular technique is known as TF-IDF (Term Frequency – Inverse Document Frequency) that are the components of the ensuing scores assigned to each phrase. Term Frequency: This summarizes how regularly a given word seems inside a document. Inverse Document Frequency: This downscales phrases that appear a lot throughout documents.The TfidfVectorizer will tokenize documents, research the vocabulary and inverse document frequency weightings, and can assist you encode new documents.

.Here we used stop _words='english', max_df=0.7, which removes the stops phrases from dataset textual content and gets rid of the words which appears in extra than 70% of documents.

Module 3: Implementation of K-means algorithm

K-means is most simplest learning algorithm to unravel the clustering problems. The process is simple and straightforward , it classifies given data set into certain number of clusters. It defines k centroids for each cluster. They must be placed the maximum amount as possible distant from one another . Then take each point belonging to given data set and relate into the nearest centroid. If no point is pending then an group age is done. Then we re-calculate k new centroid for the cluster resulting from previous steps. When we get the k centroid a replacement binding is to be done between same data points and nearest centroid. A loop is

been generated due to this loop key centroid change the location step by step until no more changes are done.

Implementation of EM algorithm:

The Expectation-Maximization (EM) set of rules may be a way to locate maximumlikelihood estimates for model parameters whilst your statistics is incomplete, has missing facts points, or has unobserved (hidden) latent variables. It is an iterative way to approximate the utmost probability function. While maximum chance estimation can find the "pleasant fit" version for a group of knowledge ,it doesn't work mainly nicely for incomplete facts sets. The more complicated EM set of rules can find model parameters albeit you've got missing records. It works by choosing random values for the missing data points, and the usage of the ones guesses to estimate a second set of knowledge . The new values are wont to create a much better guess for the number one set, and therefore the procedure continues till the set of rules converges on a fixed point.

ALGORITHM

Steps for EM algorithm:

1: select an initial set of parameters.

2: repeat

3: Expectation Step for every object, calculate the probability that every object

belongs to each distribution, i.e., calculate prob(distribution j|xi,ϴ).

4: Maximization Step Given the possibilities from the expectation step, find

The new estimates of the parameters that maximize the expected likelihood.

5: until The parameters do not change.

k means clustering algorithm:

The advantages of k means clustering algorithm are simplicity and speed.

1) Select k center from the problem(random)

2) Divide data into k clusters by grouping points.

3) Calculate the mean of k cluster to seek out new centers.

4) Repeat steps 2 and 3 until centers don't change.

6.RESULTS AND DISCUSSION AND PERFORMANCE ANALYSIS

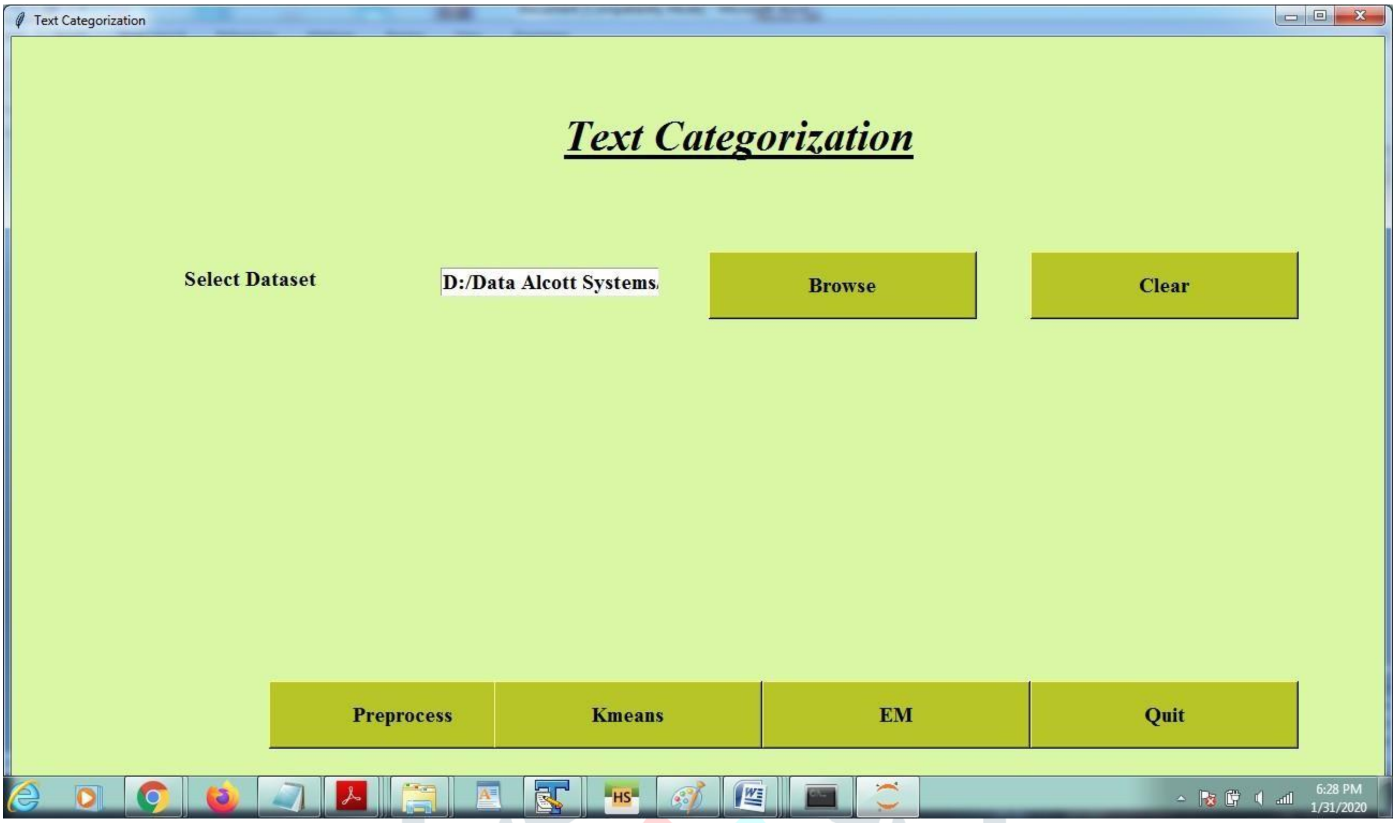
The proposed work is applied in Python 3.6.4 with libraries scikit-learn, pandas , matplotlib and other obligatory libraries. The BBC news dataset is taken for Clustering set of rules is applied consisting of Expectation maximization and K_means algorithm. We used these system gaining knowledge of set of rules and categorized cluster

The following table shows the our experimental study.

Algorithm	Clusters categorized
K-means	6
EM -algorithm	6

Table: Clusters in proposed system

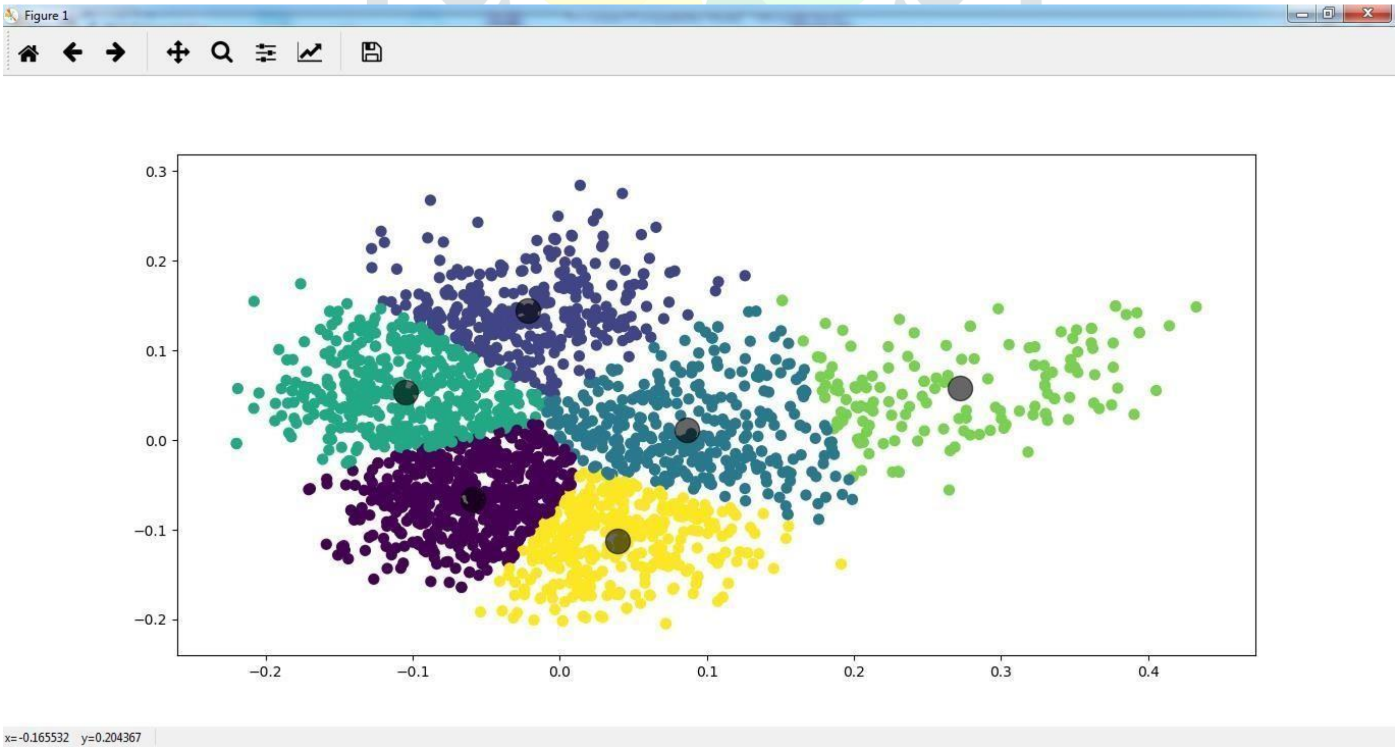
The following screen shows the main page of our application



Select Dataset D:/Data Alcott Systems/

fig.2.1

The above figure Fig.2.1 is the home page in which we upload an input and perform various tasks like pre-processing , k-means , Em algorithm **.**



x=-0.165532 y=0.204367

fig.2.2

The above figure Fig.2.2 shows the output of K means clustering .It shows classifying given data set into certain number of clusters. It defines k centroids for each cluster. They must be placed as much as possible far away from each other.

7.Conclusion:

The dynamic nature of the web generates massive number of the text documents belongs to new and wide variety of domains. We started survey on cross domain text document classification techniques has many real world applications. Clustering aims at grouping data objects into classes so that the objects within a class are similar while the objects in different classes are dissimilar. Conventional clustering algorithms compute the distances between objects in the entire space of dimensions. K-means algorithm is used with Euclidean distance for obtaining the clusters. K-means algorithm is not very efficient, Choosing k manually, outliers might get their own cluster ,trouble clustering data where clusters have varying sizes. K-means is a hard clustering algorithm while EM is a soft clustering algorithm. It is a good basis for information retrieval systems.

References :

1. M.RamaKrishna Murthy, J.V.R Murthy, Prasad Reddy PVGD, S.C.Satapathy " A Survey of cross-Domain Text Categorization Techniques" in 2012 1st International Conference on Recent Advances in Information Technology (RAIT)

2. Elisabeth Lex, Christin Seifert, Michael Granitzer and Andreas Juffinger, "Efficient Cross-Domain Classification of Weblogs", International Journal of Intelligent Computing Research (IJICR), Volume 1,Issue ½,March/June 2010.

3. Jiang. Domain Adaptation in Natural Language Pro-cessing. PhD thesis, Computer Science in the Grad-uate College of the University of Illinois at Urbana-Champaign, 2008.

4. W. Y. Dai, G. R. Xue, Q. Yang, and Y. Yu. Co-clustering based classification for out of-domain doc-uments. In Proceedings of the 13th ACM SIGKDD,San Jose, California, pages 210–219, 2007

5. F. Wang, T. Li, and C. S. Zhang. Semi-supervised clustering via matrix factorization. In Proceedings of the 8th SDM, 2008

6. D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. Journal of Machine Learning Research, pages 993–1022, 2003

7. B. Li, Q. Yang, and X. Y. Xue. Can movies and books collaborate? cross-domain collaborative filtering for sparsity reduction. In Proceedings of the 21rd IJCAI, pages 2052–2057, 2009

8. Upasana and S. Chakravarty, "A Survey of Text Classification Techniques forE-mail filtering, in Second International Conference on Machine Learning and Computing", 2010.

9. Chang Wan, Rong Pan and Jiefei Li, "Bi-Weighting Domain Adaptation for CrossLanguage Text Classification", in Proceedings of the twenty second International Joint Conference on Artificial Intelligence, August 1, 2010.

10. Gui-Rong Xue, Wenyuan Dai, Qiang Yang and Yong Yu, "Topic-bridged PLSA for Cross-Domain Text Classification", in SICIR'D8, July 20-24, 2008, Singapore.

11. Johan Hovold, "Naïve Bayes Spam Filtering Using Word Position Based Attributes", in International Conference of E-mail and Anti spam, 2005.

12.Himanshu S. Bhatt, Manjira Sinha and Shourya Roy "Cross-domain Text Classification with Multiple Domains and DisparateLabel Sets" in Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics , Berlin, Germany, August 7-12, 2016. c 2016