Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 5.258

IJCSMC, Vol. 5, Issue. 4, April 2016, pg.556 – 569

POINT WISE TECHNIQUE FOR GENERATING OPTIMIZED INDEXES IN TEXT MINING

Mr. SathishKumar.A[1], Dr. Selvan.C[2], Mr. ArunKumar.S[3]
[1]Department Of CSE, Sri Eshwar College Of Engineering, Coimbatore, India
[2]Associate Professor Department Of CSE, Sri Eshwar College Of Engineering, Coimbatore, India
[3]Associate Software Engineer, India Global Delivery Center, Chennai, India
[1] acksathish@gmail.com; [2] dr.selvan.c@gmail.com; [3] arunkumar.srinivasan@cgi.com

Abstract— Text discovery is the process of assigning a document to one or more target categories, based on its contents. Classification is often posed as a supervised learning problem in which a set of labelled data is used to train a classifier. Classification includes different parts such as text processing, feature extraction, feature vector construction and final classification. In the proposed method, machine learning methods for text classification is used to apply some text preprocessing methods in different dataset, and then to extract a feature vector construction for each new document by using feature weighting and feature selection algorithms for enhancing the text classification accuracy. Further training the classifier by Naive Bayesian (NB) and K-nearest neighbor (KNN) algorithms so that, the predication can be made according to the category distribution among this k nearest neighbors. Experimental results show that the methods are favourable in terms of their effectiveness and efficiency when compared with other classifiers.

Keywords— clustering, information extraction and visualization

I. INTRODUCTION

Text Classification or Categorization, the problem of automatically assigning semantic categories to natural language text, has become one of the most important methods for organizing textual information. Since the classification by hand is costly and in most cases highly unpractical due to the increasing number of documents and categories in many corpora, most state of the art approaches employ machine learning techniques to automatically learn text classifiers from training examples. Unlike many other classification tasks, text classification involves also preprocessing steps, e.g., stemming and dimensionality reduction, which have an important influence on the effectiveness of the actual classification outcome.

Categorizing text documents means to discover their category or topic from a set of predefined categories, e.g. `sports' or `economics'. Text categorization is an important field within natural language processing. Its application areas are many and the need for them is increasingly important as the amounts of information continue to grow. Junk mail filtering has been an important area for text categorization the last decade, as have portals with hierarchies of web sites, digital libraries and more. But the general task of _ling a text document in the correct location {or spotting its correct topic {will exist as long as digital written texts are being produced. Other examples include publishing newspaper articles in the correct category or storing a digital document correctly in an archive or library. Automatic text categorization was first done as early as the sixties, though the lack of computer power made it infeasible for a long time. During the last decade or so however, we have seen a lot of efforts in the area. While computers today are capable of learning and performing text categorization within reasonable time limits, growing amounts of data makes TC challenging today as well.

The first feature extraction method based on feature clustering was proposed by Baker and McCallum which was derived from the "distributional clustering" idea of Pereira et al... Al-Mubaid and Umair used distributional clustering to generate an efficient representation of documents and applied a learning logic approach for training text classifiers. The Agglomerative Information Bottleneck approach was proposed by Tishby et al. The divisive information-theoretic feature clustering algorithm was proposed by Dhillon et al. which is an information-theoretic feature clustering approach, and is more effective than other feature clustering methods. In these feature clustering methods, each new feature is generated by combining a subset of the original words. However, difficulties are associated with these methods. A word is exactly assigned to a subset, i.e., hard-clustering, based on the similarity magnitudes between the word and the existing subsets, even if the differences among these magnitudes are small. Also, the mean and the variance of a cluster are not considered when similarity with respect to the cluster is computed. Furthermore, these methods require the number of new features be specified in advance by the user.

II. OVERVIEW OF DATA MINING

Data mining is an ambiguous term that has been used to refer to the process of finding interesting information in large repositories of data. More precisely, the term refers to the application of special algorithms in a process built upon sound principles from numerous disciplines including statistics, artificial intelligence, machine learning, database science, and information retrieval. Data mining (sometimes called data or knowledge discovery) is the process of analyzing data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both.

Data mining is the process of processing large volumes of data (usually stored in a database), searching for patterns and relationships within that data. There is no single standard algorithm for data mining, though statistical techniques such as regression are often used to identify relationships between items. While large-scale information technology has been evolving separate transaction and analytical systems, data mining provides the link between the two. Data mining software analyzes relationships and patterns in stored transaction data based on open-ended user queries. Several types of analytical software are available: statistical, machine learning, and neural networks. Generally, any of four types of relationships are sought:

Classes: Stored data is used to locate data in predetermined groups. For example, a restaurant chain could mine customer purchase data to determine when customers visit and what they typically order. This information could be used to increase traffic by having daily specials.

Clusters: Data items are grouped according to logical relationships or consumer preferences. For example, data can be mined to identify market segments or consumer affinities. Associations: Data can be mined to identify associations. The beer-diaper example is an example of associative mining.

Sequential patterns: Data is mined to anticipate behavior patterns and trends. For example, an outdoor equipment retailer could predict the likelihood of a backpack being purchased based on a consumer's purchase of sleeping bags and hiking shoes.

Data mining consists of five major elements:

- Extract, transform, and load transaction data onto the data warehouse system.
- Store and manage the data in a multidimensional database system.
- Provide data access to business analysts and information technology professionals.
- Analyze the data by application software.
- Present the data in a useful format, such as a graph or table.

Different levels of analysis are available:
Artificial neural networks: Non-linear predictive models that learn through training and resemble biological neural networks in structure.
Genetic algorithms: Optimization techniques that use processes such as genetic combination, mutation, and natural selection in a design based on the concepts of natural evolution.
Decision trees: Tree-shaped structures that represent sets of decisions. These decisions generate rules for the classification of a dataset. Specific decision tree methods include Classification and Regression Trees (CART) and Chi Square Automatic Interaction Detection (CHAID). CART and CHAID are decision tree techniques used for classification of a dataset. They provide a set of rules that you can apply to a new (unclassified) dataset to predict which records will have a given outcome. CART segments a dataset by creating 2-way splits while CHAID segments using chi square tests to create multi-way splits. CART typically requires less data preparation than CHAID.
Nearest neighbor method: A technique that classifies each record in a dataset based on a combination of the classes of the k record(s) most similar to it in a historical dataset (where k is 1). It is called the k-nearest neighbor technique.
Rule induction: The extraction of useful if-then rules from data based on statistical significance.
Data visualization: The visual interpretation of complex relationships in multidimensional data. Graphics tools are used to illustrate data relationships.

Data mining is the practice of automatically searching large stores of data to discover patterns and trends that go beyond simple analysis. Data mining uses sophisticated mathematical algorithms to segment the data and evaluate the probability of future events. Data mining is also known as Knowledge Discovery in Data (KDD). Generally, data mining (sometimes called data or knowledge discovery) is the process of analyzing data from different perspectives and summarizing it into useful information.

III. MOTIVATION WORK

A number of works have been done on the area of opinion mining especially for topic classification. This section mentions some of these works.
M. Paul says, this paper describes cross-collection latent Dirichlet allocation (ccLDA), a probabilistic topic model that captures meaningful word co-occurrences across multiple text collections. The model is applied to three different applications: discovering cultural differences in blogs and forums from different countries, discovering research topics across multiple scientific disciplines, and comparing editorial differences between multiple media sources. A variety of qualitative and quantitative evaluations of ccLDA are performed, including log-likelihood measurements and performance measurements of the model used as a generative classifier. Improvements over previous work are demonstrated. Finally, possible extensions and modifications to the model are presented with promising results.

Our model, ccLDA, shares with the LDA Collocation (Griffiths et al., 2007) and Topical NGrams models the assumption that each word can come from two different word distributions, one of which depends on another observable variable. In these models, a word can come from either its topic’s word distribution, or it can come from a word distribution associated with the previous word, in the case that the word is determined to be part of a collocation. The key difference here is that in these models, the alternative word distribution depends on the word preceding a token, while in ccLDA, this depends on the document’s collection. The model is also related to hierarchical variants of LDA, in particular the hierarchical Pachinko allocation (hPAM) model, in which both a topic and hierarchy depth are chosen, and there is a different word distribution at different levels in the hierarchy. A natural way to view our model is as a two-level hierarchy where the top level represents the collection-independent distributions and the bottom level represents the collection specific distributions. One of the main differences here is that the discovered hierarchies in hPAM can be arbitrary, whereas the graphical structure of our model is pre-determined such that each topic has exactly one “sub-topic” representing each collection.

A. Ahmed and E. Xing proposed a method , with the proliferation of user-generated articles over the web, it becomes imperative to develop automated methods that are aware of the ideological-bias implicit in a document collection. While there exist methods that can classify the ideological bias of a given document, little has been done toward understanding the nature of this bias on a topical-level. In this paper we address the problem of modeling ideological perspective on a topical level using a factored topic model. We develop efficient inference algorithms using Collapsed Gibbs sampling for posterior inference, and give various evaluations and illustrations of the utility of our model on various document collections with promising results. Finally we give a Metropolis-Hasting inference algorithm for a semi-supervised extension with decent results.

M. Paul and R. Girju says, this paper presents the Topic-Aspect Model (TAM), a Bayesian mixture model which jointly discovers topics and aspects. We broadly define an aspect of a document as a characteristic that spans the document, such as an underlying theme or perspective. Unlike previous models which cluster words by topic or aspect, our model can generate token assignments in both of these dimensions, rather than assuming words come from only one of two orthogonal models. We present two applications of the model. First, we model a corpus of computational linguistics abstracts, and find that the scientific topics identified in the data tend to include both a computational aspect and a linguistic aspect. For example, the computational aspect of GRAMMAR emphasizes parsing, whereas the linguistic aspect focuses on formal languages. Secondly, we show that the model can capture different viewpoints on a variety of topics in a corpus of editorials about the Israeli-Palestinian conflict. We show both qualitative and quantitative improvements in TAM over two other state-of-the-art topic models.

The structure of the Topic-Aspect Model is very malleable and can be easily altered to suit the needs of a particular application. For example, the background/topical level binomial could be shared across the entire corpus rather than being drawn per-document. Conversely, the binomial distribution over x could be made to be generated per-document. The dependencies of x on z and/or ` could be dropped if a more rigid model is desired, or for more flexibility x could also depend on the aspect y. We believe there are a number of applications in which TAM could potentially be used. LDA-style topic models have been shown to be very useful for document summarization, and TAM could be used similarly, for example to extract sentences to summarize the same information from different perspectives. TAM's outputs could be used to enrich the features used in certain systems. For example, if we wanted to train a system to extract the computational approaches used for a problem in a scientific paper, the aspect(s) assigned to a sequence of words might be useful features for distinguishing the method/approach from the problem.

Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei proposed a method called, we consider problems involving groups of data where each observation within a group is a draw from a mixture model and where it is desirable to share mixture components between groups. We assume that the number of mixture components is unknown a priori and is to be inferred from the data. In this setting it is natural to consider sets of Dirichlet processes, one for each group, where the well-known clustering property of the Dirichlet process provides a nonparametric prior for the number of mixture components within each group. Given our desire to tie the mixture models in the various groups, we consider a hierarchical model, specifically one in which the base measure for the child Dirichlet processes is itself distributed according to a Dirichlet process. Such a base measure being discrete, the child Dirichlet processes necessarily share atoms.

D. Andrzejewski, X. Zhu, and M. Craven says, Users of topic modeling methods often have knowledge about the composition of words that should have high or low probability in various topics. We incorporate such domain knowledge using a novel Dirichlet Forest prior in a Latent Dirichlet Allocation framework. The prior is a mixture of Dirichlet tree distributions with special structures. We present its construction, and inference via collapsed Gibbs sampling. Experiments on synthetic and real datasets demonstrate our model's ability to follow and generalize beyond user-specified domain knowledge.

I. Sato and H. Nakagawa says, one important approach for knowledge discovery and data mining is to estimate unobserved variables because latent variables can indicate hidden specific properties of observed data. The latent factor model assumes that each item in a record has a latent factor; the co-occurrence of items can then be modeled by latent factors. In document modeling, a record indicates a document represented as a "bag of words," meaning that the order of words is ignored, an item indicates a word and a latent factor indicates a topic. Latent Dirichlet allocation (LDA) is a widely used Bayesian topic model applying the Dirichlet distribution over the latent topic distribution of a document having multiple topics. LDA assumes that latent topics, i.e., discrete latent variables, are distributed according to a multinomial distribution whose parameters are generated from the Dirichlet distribution. LDA also models a word distribution by using a multinomial distribution whose parameters follows the Dirichlet distribution.

D. Mimno, H. Wallach, and A. McCallum proposed a method, previous work on probabilistic topic models has either focused on models with relatively simple conjugate priors that support Gibbs sampling or models with non-conjugate priors that typically require variational inference. Gibbs sampling is more accurate than variational inference and better supports the construction of composite models. We present a method for Gibbs sampling in non-conjugate logistic normal topic models, and demonstrate it on a new class of topic models with arbitrary graph-structured priors that reflect the complex relationships commonly found in document collections, while retaining simple, robust inference. For each model, the held-out likelihood was computed using a subset of the corpus, e.g., all papers from a particular venue or year. The remaining papers were used to infer the group-specific means. Note that if the held-out data and graph structure are aligned such that all documents from a particular group belong to the held-out set, then the mean for that group is inferred using only information from adjacent means. The held-out likelihood therefore provides a measure of the extent to which the inferred mean accurately represents the document-specific topic distributions for that group.

L. Du, W. Buntine, H. Jin, and C. Chen says, understanding how topics within a document evolve over the structure of the document is an interesting and potentially important problem in exploratory and predictive text analytics. In this article, we address this problem by presenting a novel variant of latent Dirichlet allocation (LDA): Sequential LDA (SeqLDA). This variant directly considers the underlying sequential structure, i.e. a document consists of multiple segments (e.g. chapters, paragraphs), each of which is correlated to its antecedent and subsequent segments. Such progressive sequential dependency is captured by using the hierarchical two-parameter Poisson–Dirichlet process (HPDP).

D. Newman, E. Bonilla, and W. Buntine says, topic models have the potential to improve search and browsing by extracting useful semantic themes from web pages and other text documents. When learned topics are coherent and interpretable, they can be valuable for faceted browsing, results set diversity analysis, and document retrieval. However, when dealing with small collections or noisy text (e.g. web search result snippets or blog posts), learned topics can be less coherent, less interpretable, and less useful. To overcome this, we propose two methods to regularize the learning of topic models. Our regularizers work by creating a structured prior over words that reflect broad patterns in the external data.

B. A. Frigyik, M. R. Gupta, and Y. Chen says, although the Dirichlet distribution is widely used, the independence structure of its components limits its accuracy as a model. The proposed shadow Dirichlet distribution manipulates the support in order to model probability mass functions (pmfs) with dependencies or constraints that often arise in real world problems, such as regularized pmfs, monotonic pmfs, and pmfs with bounded variation. We describe some properties of this new class of distributions, provide maximum entropy constructions, give an expectation-maximization method for estimating the mean parameter, and illustrate with real data.

IV. RELATED WORK

The proposed differential topic model is an instance of the general correlated topic model family, where we try to model different sources of correlation between documents. Correlation in topic models can be considered in two forms: (1) the correlation in topic distributions, the correlation between topics; and (2) the correlation in topic-word distributions, the correlation between words. Our model falls into the later case. There is considerable research from both perspectives, each with different motivation and algorithms. For the first case, representative work are on shared and hierarchical topic models. Blei and Lafferty proposed the correlated topic model [17], which replaces the Dirichlet prior with a logistic normal distribution. A Gibbs sampling method for this kind of model is described in [18]. Later, Paisley et al. extend the logistic normal distribution to a nonparametric setting and also use it for correlated topic modeling [19]. This generalizes the model of [17]. The nested Chinese restaurant process (nCRP) [14] models topic hierarchies by introducing a nested Chinese restaurant process prior on a tree. Documents are generated by drawing a set of words along the path of one branch in the tree, following the nCRP prior. Li and Mccallum proposed the Pachinko Allocation model (PAM) [12] to model topic correlations using a directed acyclic graph.

In the four-level PAM, they assume words in the documents are drawn by choosing a super-topic which generates the sub-topic word distributions. Sampling is performed on an extended version of LDA with multiple levels. Du et al. developed a series of models exhibiting sharing across segments in a document both hierarchically and sequentially [20], [21] that were very competitive against standard LDA. Note that the above works, while hierarchical, do not consider the problem of topic sharing between groups of data sets, nor do they consider correlations among words in the topic. On the other hand, there is also work on modeling topic word distributions. Andrzejewski et al. [13] use a Dirichlet forest prior for the topic-word matrices so that some must link and cannot-link constraints between words can be Fig. 1. Differential topic modeling using the TPYP. The top level is an abstract space that generates each sub-space for each group of documents. Each group's vocabulary subspace is formed by taking a transformation from the top abstract space.

CHEN ET AL.: DIFFERENTIAL TOPIC MODELS 231 introduced. These constraints are modeled as preferences so the technique is quite general, and in our view should see wider use in the community. While their model is a correlation model rather than a differential model of word use, we could have employed this technique to handle shared semantics. Sato and Nakagawa use the PYP to model word distributions [16], however, they do not consider word correlations for each topic and the topic sharing between groups. Our model is thus a sharing extension of theirs. Furthermore, sparsity constraints are introduced in [22], Markov constraints are introduced in [23] in which priors for the topic-word distribution are defined as Gaussian and encoded with domain knowledge. Petterson et al. [24] proposed an extension of LDA using an informative prior instead of the symmetric Dirichlet prior for the topic-word distribution matrices, again without considering the problem of topic sharing between groups. Their technique is comparable in goal to Newman et al. [25], and our technique is basically an application of the same approach to the context of hierarchical Bayesian modeling.

There are now several useful tools to model correlations in word use, and some we could explore in later work. However, our specific goal was to model differential word use. Similar to our goal, Paul and Girju's topic-aspect model [8] extends Paul's cross-collection topic model [2]. It models different aspects within the data set by using an extension of the LDA model. Later they combined this model with a random walk model to achieve summarizing contrastive viewpoints in opinionated text [9].

The basic idea here is that multiple collections have word usage in common but also word usage that is unique to each collection. By linking the common and unique words through a latent topic, and thus enforcing co-occurrence, the similarities and differences are discovered. In the machine learning community, a topic is defined as a collection of related words from the vocabulary. Most existing hierarchical techniques for modeling topic word distributions are based on the Dirichlet process (DP). LDA using an informative prior instead of the symmetric Dirichlet prior for the topic-word distribution matrices, again without considering the problem of topic sharing between groups. The Pitman-Yor process and the Dirichlet process as non-parametric Bayesian priors, have become increasingly popular in statistical machine learning with applications found in diverse fields such as topic modeling. Each draw from a PYP is a probability distribution with possibly infinitely many types, facilitating the use of the PYP as a prior in modeling topic-word distributions. Thus in topic modeling, the base measure H(.) is a probability distribution over a vocabulary space, samples xi are words, and p_k is the probability of observing word x^*_k in a topic.

V. PROPOSED SYSTEM

Text discovery is the process of assigning a document to one or more target categories, based on its contents. Classification is often posed as a supervised learning problem in which a set of labeled data is used to train a classifier. Classification includes different parts such as text processing, pointwise feature extraction, feature vector construction and final classification. In the proposed method, machine learning methods for text classification is used to apply some text preprocessing methods in different dataset, and then to extract a feature vector construction for each new document by using feature weighting and feature selection algorithms for enhancing the text classification accuracy. Further training the classifier by Naive Bayesian (NB) and K-nearest neighbor (KNN) algorithms so that, the predication can be made according to the category distribution among this k nearest neighbors. Experimental results show that the methods are favorable in terms of their effectiveness and efficiency when compared with other classifiers.

VI. SYSTEM ARCHITECTURE

6.1 LOAD DATA SET

The data set that is used for text classification are (1) self-made and (2) Reuters. In this process the data sets are loaded to the Corpus and further trained for better results.

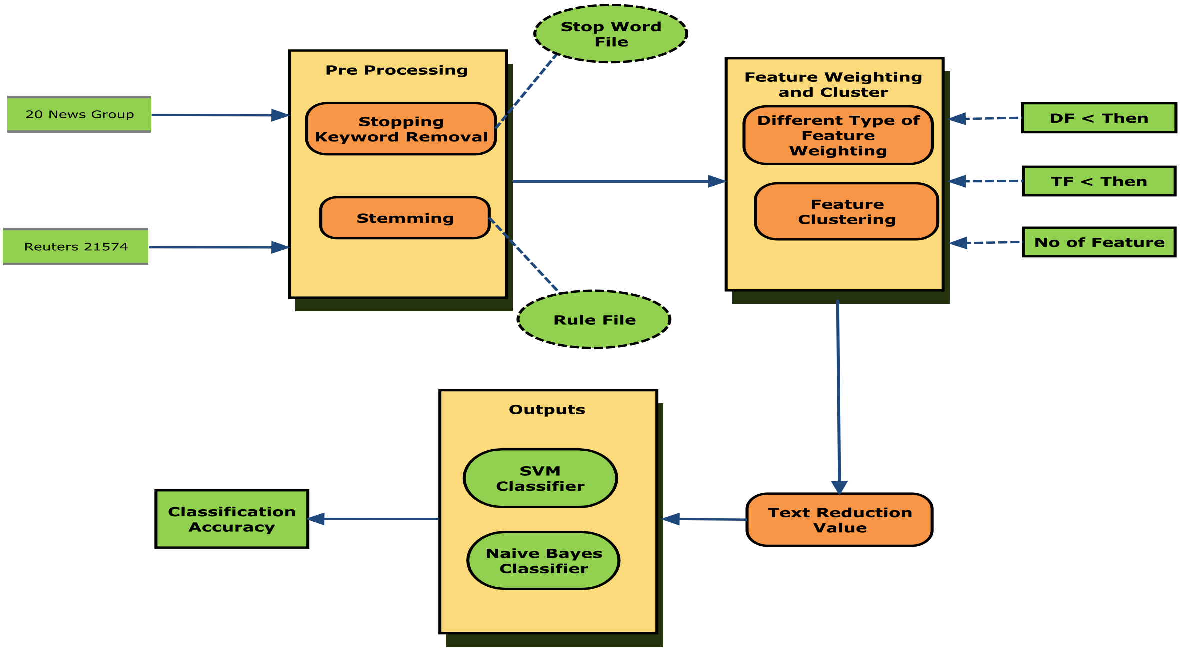


Figure 6.1 Architecture Diagram

6.1.1 Self Made

For the development used a small self-made corpus since the running time needed to be as short as possible. I collected articles online from the New York Times, Washington Post and CNN.com out of the standard categories" Science", "Business", "Sports", "Health", "Education", "Travel", and "Movies". This includes easy (e.g. Sports $ Business) and more difficult (Education $ Science $ Health) classification tasks. I collected 150 documents with the following categories: Sports {30 Training Documents}, Health {30}, Science {27}, Business {23}, Education {24}, Travel {6}, Movies {10}, with in average 702 words per document.

6.1.2 The Reuters 21578 corpus

The second corpus already included in the system is the frequently used Reuters 21578 corpus. The corpus is freely available on the internet. Uses an XML parser, it was necessary to convert the 22 SGML documents to XML, using the freely available tool SX. After the conversion I deleted some single characters which were rejected by the validating XML parser as they had decimal values below 30. This does not affect the results since the characters would have been considered as whitespaces anyway.

6.2 Text Preprocessing

Text processing is used for removing the stop words and stemming the text present as root word given in the document.

6.2.1 Removal of Stop Words

In most of the applications, it is practical to remove words which appear too often (in each or almost every document) and thus support no information for the task. Good examples for this kind of words are prepositions, articles and verbs like "be" and "go". If the box "Apply stop word removal" is checked, all the words in the file "swl.txt" are considered as stop words and will not be loaded. This file contains currently the 100 most used words in the English language which on average account for a half of all reading in English. If the box "Apply stop word removal" is unchecked, the stop word removal algorithm will be disabled when the corpus is loaded.

6.2.2 Stemming

Stemming or lemmatization is a technique for the reduction of words into their root. Many words in the English language can be reduced to their base form or stem e.g. agreed, agreeing, disagree, agreement and disagreement belong to agree. Furthermore are names transformed into the stem by removing the ”’s”. The variation ”Peter’s” in a sentence is reduced to ”Peter” during the stemming process. The result of the removal may lead to an incorrect root. However, these stems do not have to be a problem for the stemming process, if these words are not used for human interaction. The stem is still useful, because all other inflections of the root are transformed into the same stem. Case sensitive systems could have problems when making a comparison between a word in capital letters and another with the same meaning in lower case. Following a selection of suffixes and prefixes for removal during stemming

- **suffixes:** ly, ness, ion, ize, ant, ent , ic, al , ical, able, ance, ary, ate, ce, y, dom , ed, ee, eer, ence, ency, ery, ess, ful, hood, ible, icity, ify, ing, ish, ism, ist, istic, ity, ive, less, let, like, ment, ory, ty, ship, some, ure
- **prefixes:** anti, bi, co, contra, counter, de, di, dis, en, extra, in, inter, intra, micro, mid, mini, multi, non, over, para, poly, post, pre, pro, re, semi, sub, super, supra, sur, trans, tri, ultra, un.

6.2.2.1 Porter Stemming

The Porter Stemming algorithm was published in 1980. The idea of this algorithm is the removal of all pre- and suffixes to get the root of a word. The main field of application for the Porter Stemmer is languages with simple inflections, such as English. The algorithm makes a distinction between consonants and vowels in a word. Therefore the selection of the applying rules during the stemming process is based on the sequence of consonants and vowels. A word is represented by the form

[C]VCVC ... [V]

Where the notation of a sequence of VC is written as (VC) {m}, with VC repeated m times. An example for a repetition with m = 0 is sea, for m = 1 is cat, for m = 2 is garden and so on. The further processing of the suffix stripping is decided by several conditions. One of the conditions was mentioned in the sentences before, the repletion of VC in a word. The other conditions for the Porter Stemming are:

- *S - the stem ends with S (and similarly for the other letters).
- *v* - the stem contains a vowel.
- *d - the stem ends with a double consonant (e.g. -TT, -SS).
- *o - the stem ends CVC, where the second c is not W, X or Y (e.g. -WIL, -HOP).

Furthermore, combinations of these conditions are possible (using and, or and not). Following, the rules with some examples, divided into 5 steps. Only the application of one rule for a step is allowed. This rule has to remove the longest matching suffix.

6.2.2.2 Lancaster Stemming

Stemming is a well-known technique for information retrieval. The use of stems for searching has the advantage of increasing recall by retrieving terms that have the same roots but different endings. A major disadvantage of stemming is a decrease of precision as compared to the use of expanded terms. When searching with stems, it is not uncommon to retrieve many irrelevant terms that have similar roots but which are not related to the object of the search. For accurate retrieval, the search stems should be as long as necessary to achieve precision, but short enough to increase recall. Several commonly-used stemming programs and algorithms were evaluated to try to select a stemmer suitable for information retrieval of large databases.

6.3 Feature Weighting and Reduction

6.3.1 Odds Ratio

Odds Ratio compares the odds of a feature occurring in one category with the odds for it occurring in another category. It gives a positive score to features that occur more often in one category than in the other, and a negative score if it occurs more in the other. A score of zero means the odds for a feature to occur in one category is exactly the same as the odds for it to occur in the other, since ln (1) = 0.

The original Odds Ratio algorithm for binary categorization:

$$\mathbf{OR\ (F,\ C_k) = ln}\frac{P(F|C_k)(1-P(F|\overline{C_k}))}{P(F|\overline{C_k})(1-P(F|C_k))} = \mathbf{ln}\frac{\left(\frac{N_{F,C_k}}{NC_k}\right)\left(1-\frac{N_{F,\overline{C_k}}}{N\overline{C_k}}\right)}{\left(\frac{N_{F,\overline{C_k}}}{N\overline{C_k}}\right)\left(1-\frac{N_{F,C_k}}{NC_k}\right)}$$

$$\mathbf{P\ (F|C_k)} = \frac{N_{F,C_k}}{N_{C_k}}$$

Let P (t|c) be the probability of a randomly chosen word being t, given that the document it was chosen from belongs to a class c. Then odds (t|c) is defined as P (t|c)/ [1–P (t|c)] and the Odds Ratio equals to,

OR (t) = ln [odds (t|c+) / odds (t|c)]

6.3.2 Information Gain

Here both class membership and the presence/absence of a particular term are seen as random variables, and one computes how much information about the class membership is gained by knowing the presence/absence statistics (as is used in decision tree induction. Indeed, if the class membership is interpreted as a random variable C with two values, positive and negative, and a word is likewise seen as a random variable T with two values, present and absent, then using the information-theoretic definition of mutual information we may define Information Gain as:

IG(t) = H(C)

Here, τ ranges over {present, absent} and c ranges over {c+, c–}. As pointed out above, this is the amount of information about C (the class label) gained by knowing T (the presence or absence of a given word).

6.3.3 Document Frequency (DF) Thresholding

One of the simplest methods of vocabulary reduction, and hence vector dimensionality reduction, is the Document Frequency Thresholding,

$$\mathbf{DF\ (F) = N_F}$$

The number of documents containing a feature in the training set is counted. This is done for every feature in the training set, before removing all features with a document frequency less than some specified threshold and features with a frequency higher than some other threshold.

Example Setup

The document frequency values for our e-mail example can be read directly from Tables 1.1.Ranks the e-mail example features according to their document frequency value. Note that document frequency values are naturally global, so there is no need to aggregate them in any way.

Table 6.1 Document Frequency

Feature	Document frequency value
Science	5.0
Sports	4.65
Bus	5.8
News	4.32
Education	5.5

6.3.4 Term Frequency Document Frequency (Tfdf)

A method based on the term frequency combined with the document frequency threshold (Section 3.6.4) is presented. They call it Term Frequency Document Frequency, and prove it better than DF thresholding.

$$\textbf{TFDF (F)} = (n_1 \text{ x } n_2 + c\ (n_1 \text{ x } n_3 + n_2 \text{ x } n_3))$$

Where c is a constant c >=1, n1 is the number of documents without the feature, n2 is the number of documents where the feature occurs exactly once, n3 is the number of documents where the feature occurs twice or more. Use c = 10 in their experiments, and we follow this decision in our experiments.

6.3.5 Pointwise Mutual Information

Point wise Mutual Information can be proven equal to Information Gain for binary problems. For multi-class problems (with global feature lists) like we present in this report however, the two are not equal (although rather similar). Thus we present Mutual Information with its own equation as a separate feature selection algorithm here.

$$\textbf{MI}\ (F, C_k) = \sum_{\upsilon f \in \{1,0\}} \sum_{\upsilon f \in \{1,0\}} \quad P\ (F = {}_{f}, C_k = \upsilon_{C_k})\ \ln \frac{P(\ F = \upsilon_f, C_k = \upsilon_{C_k}}{P(F = \upsilon_f) P(C_k = \upsilon_{C_k})}$$

$$\textbf{MI}\ (F, C_k) = \frac{N_{F,C_k}}{N} \ln \frac{N N_{F,C_k}}{N_F N_{C_k}} + \frac{N_{F,\overline{C_k}}}{N} \ln \frac{N N_{F,\overline{C_K}}}{N_F N_{\overline{C_k}}} + \frac{N_{\overline{F},C_k}}{N} \ln \frac{N N_{\overline{F},C_k}}{N_{\overline{F}} N_{C_k}} + \frac{N_{\overline{F},\overline{C_k}}}{N} \ln \frac{N N_{\overline{F},\overline{C_k}}}{N_{\overline{F}} N_{\overline{C_k}}}$$

Then the values can be weighted and summarized to create a global ranked list of features:

$$\textbf{MI}\ (F) = \sum_{k=1}^{|C|} \frac{N_{C_k}}{N} \textbf{MI}\ (F, C_k)$$

6.3.6 Chi Square (Chi)

Feature Selection by X2 testing is based on Pearson's X 2 (chi square) tests. The X2 test is often used to test the independence of two variables. The null-hypothesis is that the two variables are completely independent of each other. The higher value of the X2 test, the closer relationship the variables have. In feature selection, the X2 test measures the independence of a feature and a category. The null-hypothesis here is that the feature and category are completely independent, i.e. that the feature is useless for categorizing documents.

$$X^2(F, C_k) = \frac{N\ X\left(\left(N_{F,C_k}\ X\ N_{\overline{F},\overline{C_k}}\right) - \left(N_{F,\overline{C_k}}\ X\ N_{\overline{F},C_k}\right)\right)^2}{N_F\ X\ N_{\overline{F}}\ \ X\ N_{C_k}\ X\ N_{\overline{C_k}}}$$

6.3.7 NGL Coefficient

The NGL coefficient presented is a variant of the Chi square metric. It was originally named a `correlation coefficient', but we follow Sebastian [Seb02] and name it `NGL coefficient' after the last names of the inventors Ng, Goh, and Low.

$$NGL(F, C_k) = \frac{\sqrt{N}\left(N_{F,C_k} N_{\overline{F},\overline{C_k}} - N_{F,\overline{C_k}} N_{\overline{F},C_k}\right)}{\sqrt{N_F N_{\overline{F}} N_{C_k} N_{\overline{C_k}}}}$$

6.3.8 GSS Coefficient

The GSS coefficient was originally presented as a `simplified chi square function'. We follow and name it GSS after the names on the inventors Galavotti, Sebastiani, and Simi.

$$GSS(F, C_k) = N_{F,C_k} N_{\overline{F},\overline{C_k}} - N_{F\ ,\overline{C_k}} N_{\overline{F},C_k}$$

The experiments in [GSS00] showed far better results when using max as a globalizing strategy rather than average, hence we follow them on that:

$$\boldsymbol{GSS(F)} = \max_{\boldsymbol{k=1}} |\boldsymbol{C}|\, \boldsymbol{GSS(F, C_K)}$$

6.4 Text Classification

6.4.1 KNN Classification

In pattern recognition, the k-nearest neighbor algorithm (k-NN) is a method for classifying objects based on closest training examples in the feature space. k-NN is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification. It is also possible to compute the decision boundary itself explicitly, and to do so in an efficient manner so that the computational complexity is a function of the boundary complexity.

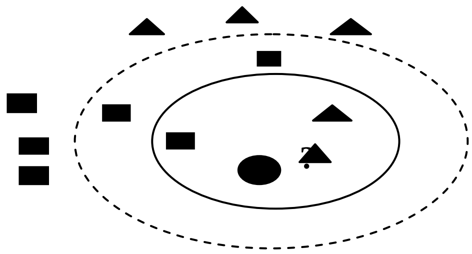


Figure 6.4.1. KNN Classification

The above diagram is an example of k-NN classification. The test sample (green circle) should be classified either to the first class of blue squares or to the second class of red triangles. If k = 3 it is assigned to the second class because there are 2 triangles and only 1 square inside the inner circle. If k = 5 it is assigned to the first class (3 squares vs. 2 triangles inside the outer circle). The training examples are vectors in a multidimensional feature space, each with a class label. The training phase of the algorithm consists only of storing the feature vectors and class labels of the training samples.

Properties

The naive version of the algorithm is easy to implement by computing the distances from the test sample to all stored vectors, but it is computationally intensive, especially when the size of the training set grows. Many nearest neighbor search algorithms have been proposed over the years; these generally seek to reduce the number of distance evaluations actually performed.

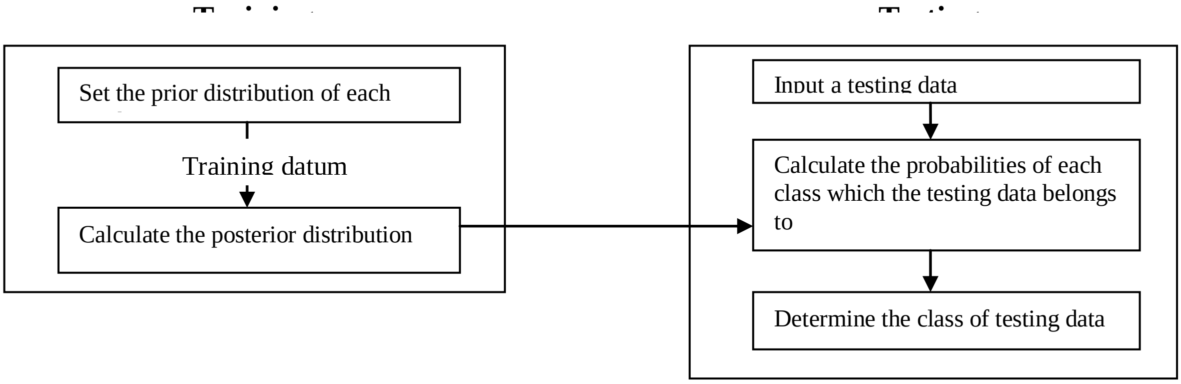
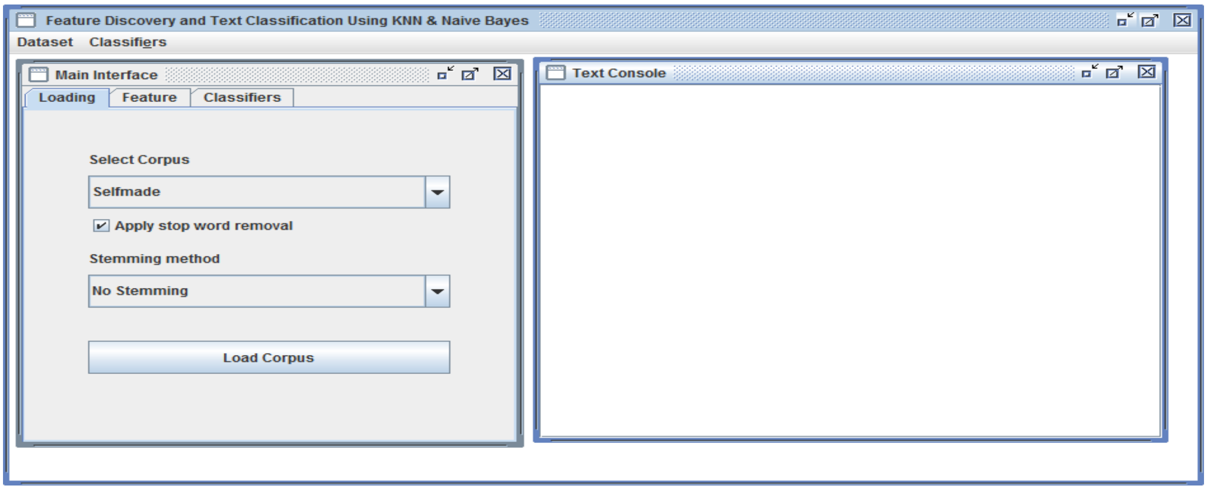

Set the prior distribution of each
Training datum
Calculate the posterior distribution
Input a testing data
Calculate the probabilities of each class which the testing data belongs to
Determine the class of testing data

Figure 6.4.2 Naïve Bayesian classification

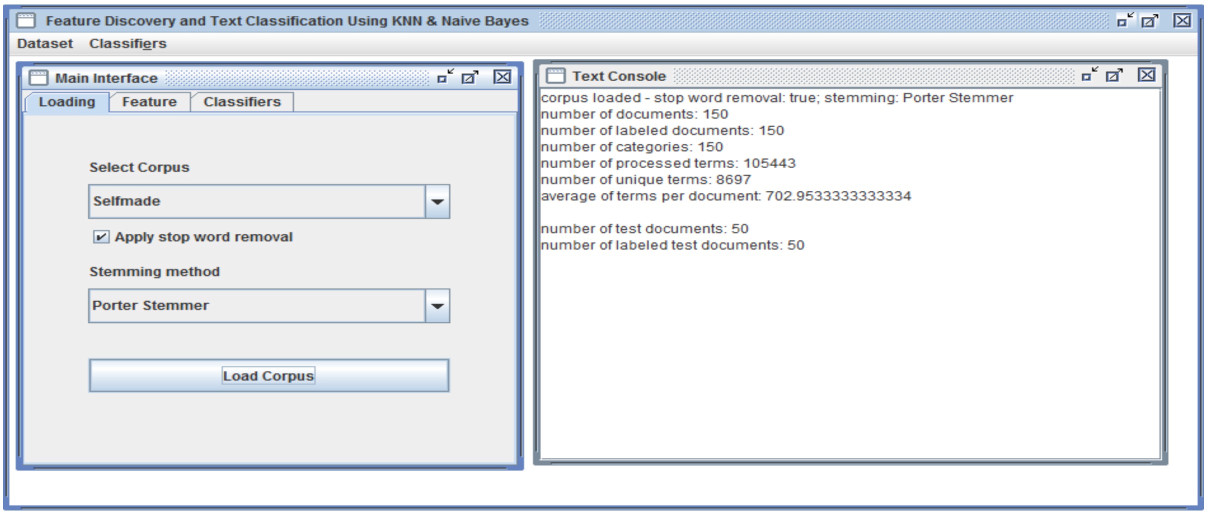


Select Corpus: Selfmade

☑ Apply stop word removal

Stemming method: No Stemming

Figure 6.4.3 Main Form



Select Corpus: Selfmade

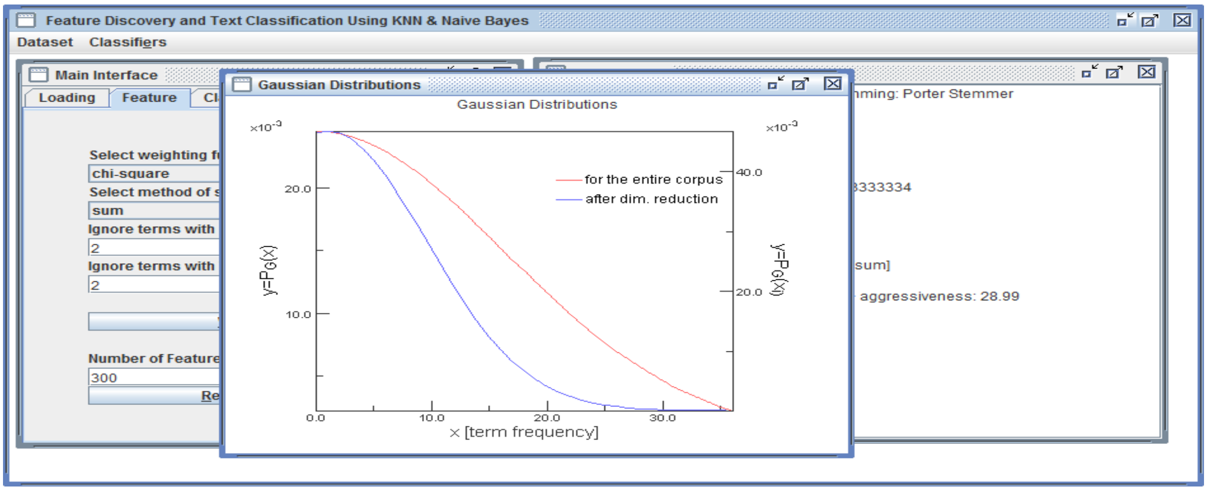
☑ Apply stop word removal

Stemming method: Porter Stemmer

```
corpus loaded - stop word removal: true; stemming: Porter Stemmer
number of documents: 150
number of labeled documents: 150
number of categories: 150
number of processed terms: 105443
number of unique terms: 8697
average of terms per document: 702.9533333333334

number of test documents: 50
number of labeled test documents: 50
```

Figure 6.4.4 Feature Discovery



Select weighting f: chi-square

Select method of s: sum

Ignore terms with: 2

Ignore terms with: 2

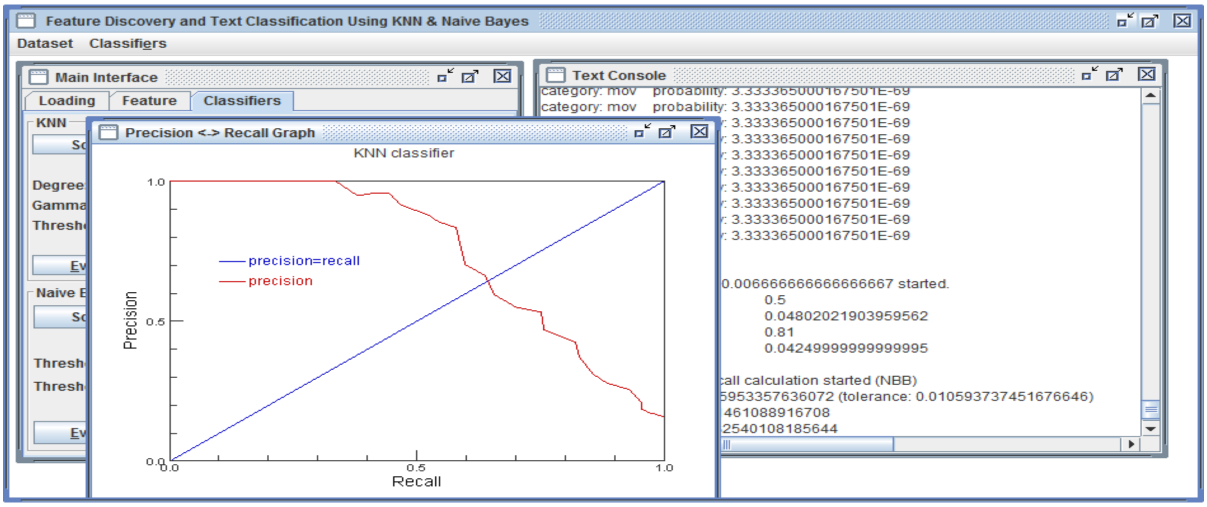
Number of Feature: 300

```
nming: Porter Stemmer

3333334

sum]
 aggressiveness: 28.99
```

Figure 6.4.5 Gaussian Distributions



```
category: mov    probability: 3.333365000167501E-69
category: mov    probability: 3.333365000167501E-69
                 : 3.333365000167501E-69
                 : 3.333365000167501E-69
                 : 3.333365000167501E-69
                 : 3.333365000167501E-69
                 : 3.333365000167501E-69
                 : 3.333365000167501E-69
                 : 3.333365000167501E-69
                 : 3.333365000167501E-69


0.006666666666666667 started.
        0.5
        0.04802021903959562
        0.81
        0.04249999999999995

call calculation started (NBB)
5953357636072 (tolerance: 0.010593737451676646)
461088916708
2540108185644
```

Figure 6.4.3 Precision and Recall graph

VII. PERFORMANCE ANALYSIS

Table 7.1 Compare classification accuracy existing algorithm with proposed

Algorithm	No of Reduce Feature			
	100	200	300	400
LDA	67	53	49	45
TPYP	75	66	51	45
Proposed	89	74	68	64

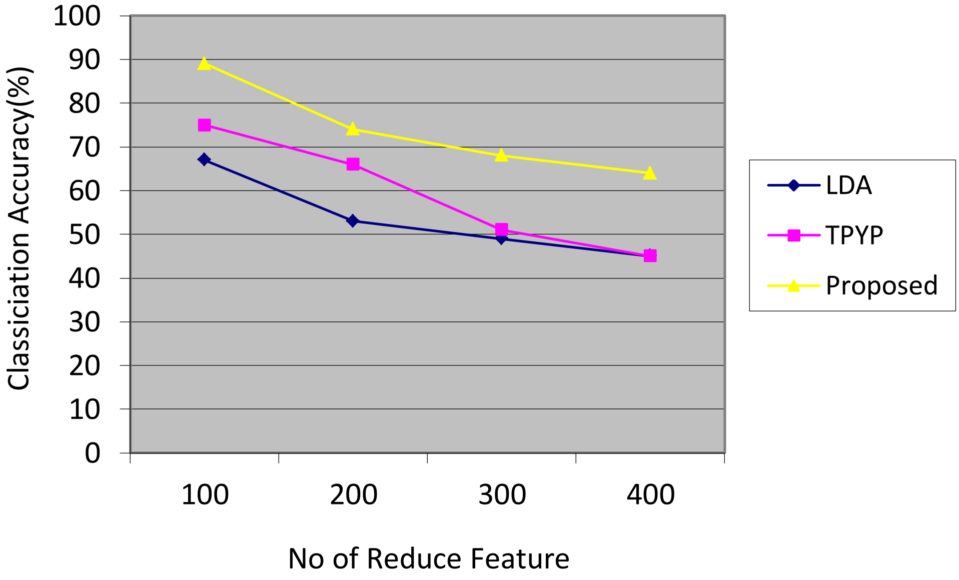


Figure 7.1 Comparison of classification

VIII. CONCLUSION

8.1 CONCLUSION

Analyzed the text classification using the Naive Bayesian and K-Nearest Neighbor classification. the methods are favorable in terms of their effectiveness and efficiency when compared with other classifier such as TPYP. The advantage of the proposed approach is classification algorithm learns importance of attributes and utilizes them in the similarity measure. In future the classification model can be build that analyzes terms on the sentence, document.

8.2 SCOPE FOR FUTURE WORK

The work leads to some interesting avenues of future work that we would like to explore. We would like to theoretically understand cross-training better and devise formal ways of studying related label-sets. We would like to extend our work in detecting evolving label-sets to larger scales and devise ways to track other kinds of evolution in label-sets apart from detecting new classes.

It uses a greedy rule-based strategy to learn the relation between documents and labels.

ACKNOWLEDGEMENT

I am thankful to Dr.Selvan.C for his guidance, support and supervision. And also providing the information and ready to help anytime in completion of this paper.

REFERENCES

[1] M. Paul, “Cross-collection topic models: Automatically comparing and contrasting text,” Master’s thesis, Univ. Illinois Urbana- Champaign, IL, USA, 2009.

[2] A. Ahmed and E. Xing, “Staying informed: Supervised and semi-supervised multi-view topical analysis of ideological perspective,” in Proc. Conf. Empirical Methods Natural Language Process., 2010, pp. 1140–1150.

[3] M. Paul and R. Girju, “A two-dimensional topic-aspect model for discovering multi-faceted topics,” AAAI Conf. Artificial Intell., pp. 545–550, 2010.

[4] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, “Hierarchical Dirichlet processes,” J. Amer. Statist. Assoc., vol. 101, no. 476, pp. 1566–1581, 2006.

[5] D. Andrzejewski, X. Zhu, and M. Craven, “Incorporating domain knowledge into topic modeling via Dirichlet Forest priors,” in Proc. 26th Annu. Int. Conf. Mach. Learning, 2009, pp. 25–32.

[6] I. Sato and H. Nakagawa, “Topic models with power-law using Pitman-Yor process,” in Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2010.

[7] D. Mimno, H. Wallach, and A. McCallum, “Gibbs sampling for logistic normal topic models with graph-based priors,” in Proc. NIPS Workshop Analyzing Graphs, pp. 1–8, 2008.

[8] L. Du, W. Buntine, H. Jin, and C. Chen, “Sequential latent Dirichlet allocation,” Knowl. Inform. Syst., vol. 31, no. 3, pp. 475–503,2012.

[9] D. Newman, E. Bonilla, and W. Buntine, “Improving topic coherence with regularized topic models,” in Proc. Advances Neural Inform. Process. Syst., 2011, pp. 496–504

[10] B. A. Frigyik, M. R. Gupta, and Y. Chen, “Shadow Dirichlet for restricted probability modeling,” in Proc. Advances Neural Inform. Process. Syst. 23, 2010, pp. 613–621.