A BRIEF SURVEY ON CLASSIFICATION, CLUSTERING AND PREPROCESSING TECHNIQUES USEGE IN TEXT MINING

Radha Mothukuri *1, DR. B. BASAVESWARA RAO BOBBA 2

1 DEPT OF CSE,RESEARCH SCHOLAR OF Acharya Nagarjuna University, GUNTUR, ANDHRA PRADESH, INDIA

2 DEPT OF CSE ,RESEARCH SUPERVISOR OF Acharya Nagarjuna University, GUNTUR, ANDHRA PRADESH, INDIA

ABSTRACT

The advancement of the World Wide Web it is not any more achievable for a client can see every one of the information originating from characterize into classes. The development of information and power programmed classification of information and textual information picks up progressively and give superior. The utilization of the information and knowledge separated from a lot of information benefits numerous applications like market investigation and business administration. In numerous applications database stores information in text frame so text mining is the standout amongst the most despise region for inquire about. To separate client required information is the testing issue. Text Mining is an imperative advance of knowledge discovery process. Text mining extricates concealed information from not-organized to semi-organized information. Text mining is the discovery via naturally extricating information from various composed assets and furthermore by PC for removing new, already obscure information. Text mining is the errand of removing important information from text, which has increased noteworthy considerations lately. In this paper, we portray a few of the most major text mining assignments and methods including text preprocessing, classification and clustering.

KEYWORDS:

Text mining, classification, clustering, information retrieval, Knowledge Discovery; Applications, TF/IDF algorithms, Word Net, Word Disambiguation.

I. INTRODUCTION

Text Mining (TM) field has picked up a lot of consideration as of late due the gigantic measure of text information, which are made in an assortment of structures, for example, interpersonal organizations, persistent records, social insurance protection information, news outlets, and so on. IDC, in a report supported by EMC, predicts that the information volume will develop to 40 zettabytes1 by 2020, prompting a 50-time development from the earliest starting point of 2010 [52].

Text mining process is as shown in following fig.1

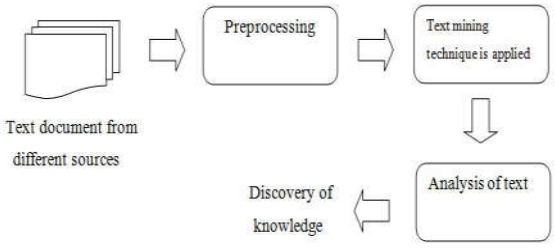


Fig. 1 Text mining process

Text information is a decent case of unstructured information, which is one of the least complex types of information that can be produced in many situations. Unstructured text is effortlessly handled and seen by people, yet is essentially harder for machines to get it. Obviously, this volume of text is a significant wellspring of in-arrangement and knowledge. Subsequently, there is a urgent need to outline techniques and algorithms keeping in mind the end goal to viably process this torrential slide of text in a wide assortment of applications.

Text mining approaches are identified with conventional information mining, and knowledge discovery techniques, with some specificity, as de-scribed beneath.

II. Literature Review

Anjali Ganesh Jivani [22] talked about that the motivation behind stemming is to lessen distinctive syntactic structures or word types of a word like its thing, descriptor, verb, and modifier and so on. The objective of stemming is to diminish inflectional structures and once in a while derivationally related types of a word to a typical base frame. This paper talks about various strategies for stemming and their correlations regarding use, favorable circumstances and additionally confinements. The fundamental contrast amongst stemming and lemmatization is additionally talked about.

Vishal Gupta et.al [23] has investigated the stemmer's execution and viability in applications, for example, spelling checker changes crosswise over dialects. A run of the mill basic stemmer calculation includes expelling postfixes utilizing a rundown of regular additions, while a more intricate one would utilize morphological knowledge to get a come from the words. The paper gives a point by point framework of regular stemming methods and existing stemmers for Indian dialects.

K.K. Agbele [24] examined the procedure for creating inescapable figuring applications that are adaptable and versatile for clients. In this context, be that as it may, information retrieval (IR) is regularly characterized as far as area and conveyance of reports to a client to fulfill their information require. Much of the time, morphological variations of words have comparable semantic translations and can be considered as equal with the end goal of IR applications. The calculation Context-Aware Stemming (CAS) is proposed, which is an adjusted adaptation of the widely utilized Porter's stemmer. Considering just created important stemming words as the stemmer yield, the outcomes demonstrate that the altered calculation essentially decreases the mistake rate of Porter's calculation from 76.7% to 6.7% without bargaining the adequacy of Porter's calculation.

Hassan Saif [25] has researched in the case of evacuating stop words aides or hampers the viability of Twitter estimation classification strategies. For this examination he has connected, six distinctive stop word recognizable proof strategies to Twitter information from six diverse datasets and watch how evacuating stop words influences two surely understood administered assumption classification techniques. The outcome demonstrates that utilizing pre-assembled arrangements of stop words contrarily impacts the execution of Twitter assumption classification approaches. Then again, the dynamic age of stop word records, by evacuating those rare terms seeming just once in the corpus seems, by all accounts, to be the ideal technique for keeping up a high classification execution while diminishing the information sparsely and considerably contracting the component space.

III. CHALLENGING ISSUES

Multifaceted nature of characteristic dialect is principle testing issue in text mining. The characteristic dialect isn't free from the uncertainty issue. Single word may have various implications and different words can have same importance. The ability of being comprehended in at least two conceivable ways implies vagueness. This vagueness prompts clamor in removed information. Equivocalness can't be altogether dispensed with from the characteristic dialect as it gives adaptability and ease of use. There are different approaches to decipher one expression or sentence along these lines different implications can be gotten. In spite of the fact that various inquires about have been directed in settling the equivocalness issue, the work is as yet youthful and the proposed approach has been devoted

for a particular space. It is test to answer what client needs as semantic implications of numerous found words are dubious.

Merits of Text mining:

i) The names of different entities and relationship between them can easily be found from the corpus of documents set using the technique such as information extraction.

ii) The challenging problem of managing great amount of unstructured information for extracting patterns e is solved by text mining.

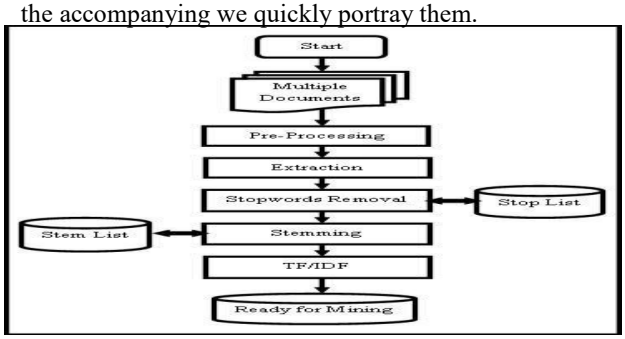
Demerits of Text mining:

i) The information which is initially needed is no where written.

ii) To mine the text for information or knowledge no programs can be made in order to analyze the unstructured text directly.

IV. Text Preprocessing

Preprocessing is one of the key segments in numerous text mining algorithms. For instance a customary text order structure contains preprocessing, highlight extraction, include determination and classification steps. In spite of the fact that it is affirmed that component extraction , highlight determination [and classification calculation have noteworthy effect on the classification procedure, the preprocessing stage may have detectable impact on this achievement. Uysal et al. have examined the effect of preprocessing undertakings especially in the territory of text classification. The preprocessing step more often than not comprises of the undertakings, for example, tokenization, sifting, lemmatization and stemming. In the accompanying we quickly portray them.



Tokenization: Tokenization is the task of breaking a character sequence up into pieces (words/phrases) called tokens, and perhaps at the same time throws away certain characters such as punctuation marks. The list of tokens then is used to further processing .

Filtering: Filtering is usually done on documents to remove some of the words. A common filtering is stop-words removal. Stop words are the words frequently appear in the text without having much content information (e.g. prepositions, conjunctions, etc). Similarly words occurring quite often in the text said to have little information to distinguish different documents and also words occurring very rarely are also possibly of no significant relevance and can be removed from the documents.

Lemmatization: Lemmatization is the task that considers the morphological analysis of the words, i.e. grouping together the various inflected forms of a word so they can be analyzed as a single item. In other words lemmatization methods try to map verb forms to infinite tense and nouns to a single form. In order to lemmatize the documents we first must specify the POS of each word of the documents and because POS is tedious and error prone, in practice stemming methods are preferred.

Stemming: Stemming methods aim at obtaining stem (root) of derived words. Stemming algorithms are indeed language dependent. The first stemming algorithm introduced in [92], but the stemmer published in [110] is most widely stemming method used in English .

V. CLASSIFICATION in TEXT MINING

Text classification has been broadly studied in different communities such as data mining, database, machine learning and information retrieval, and used in vast number of applications in various domains such as image processing, medical diagnosis, document organization, etc. Text classification aims

to assign predefined classes to text documents .The problem of classification is defined as follows. We have a training set D = {d1, d2, ...dn} of documents,

Such that each document di is labeled with a label ℓi from the set

$\mathcal{L}$ = {ℓ1, ℓ2, . . . , ℓk }. The task is to find a classification model (clas-sifier) f where

F: D $\longrightarrow \mathcal{L}$ f (d) = ℓ (3)

Which can assign the correct class label to new document d (test instance). The classification is called hard, if a label is explicitly assigned to the test instance and soft, if a probability value is as-signed to the test instance. There are other types of classification which allow assignment of multiple labels to a test instance. For an extensive overview on a number of classification methods see. Yang et al. evaluates various kinds of text classification algorithms [14]. Many of the classification algorithms have been implemented in different software systems and are publicly available such as BOW toolkit ,Mallet and WEKA4.

To evaluate the performance of the classification model, we set a side a random fraction of the labeled documents (test set). After training the classifier with training set, we classify the test set and compare the estimated labels with the true labels and measure the performance. The portion of correctly classified documents to the total number of documents is called accuracy . The common evaluation metrics for text classification are precision, recall and F-1 scores. Charu et al. [1] defines the metrics as follows: “precision

is the fraction of the correct instances among the identified positive instances. Recall is the percentage of correct instances among all the positive instances. And F-1 score is the geometric mean of precision

and recall”.

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall} \quad (4)$$

5.1 Naive Bays Classifier

Probabilistic classifiers have picked up a great deal of fame as of late and have appeared to perform surprisingly well .These probabilistic methodologies make suppositions about how the information (words in records) are produced and propose a probabilistic model in view of these suspicions. At that point utilize an arrangement of preparing cases to gauge the parameters of the model. Bayes manage is utilized to arrange new illustrations and select the class that is undoubtedly has produced the case .

The Naive Bayes classifier is maybe the least difficult and the most broadly utilized classifier. It demonstrates the dispersion of reports in each class utilizing a probabilistic model expecting that the conveyance of various terms are autonomous from each other. Despite the fact that this supposed "gullible Bayes" supposition is plainly false in numerous true applications, credulous Bayes performs shockingly well.

There are two principle models normally utilized for credulous Bayes classifications [96]. The two models go for finding the back likelihood of a class, in light of the dissemination of the words in the report. They distinction between these two models is, one model considers the recurrence of the words though the other one doesn't.

(1) Multivariate Bernoulli Model: In this model a report is spoken to by a vector of paired highlights indicating the nearness or nonattendance of the words in the record. Along these lines, the recurrence of words is overlooked. The first work can be found in .

(2) Multinomial Model: We catch the frequencies of words (terms) in a record by speaking to the report as pack of words. A wide range of varieties of multinomial model have been presented in , McCallum et al. have completed a broad correlation amongst Bernoulli and multinomial models and inferred that

• If the measure of the vocabulary is little, the Bernoulli model may outflank multinomial model.

• The multinomial model dependably outflanks Bernoulli display for vast vocabulary sizes, and quite often per-frames superior to anything Bernoulli if the

span of the vocabulary picked ideally for the two models.

Both of these models assume that the documents are generated by a mixture model parameterized by θ. We use the framework McCallum el at. Defined as follows:

The mixture model comprises mixture components cj ∈ C = {c1, c2, ...ck}. Each document di = {w1, w2, . . . , wni } is generated by first selecting a component according to priors, P(cj |θ)

And then use the component to create the document according to its own parameters, P (di |cj; θ). Hence, we can compute the likelihood of a document using the sum of probabilities over all mixture components:

$$P(d_i|\theta) = \sum_{j=1}^{k} P(c_j|\theta)P(d_i|c_j;\theta)$$

We assume a one to one correspondence between classes L = {ℓ1, ℓ2, . . . , ℓk } and mixture components, and therefore cj indicates both the jth mixture component and the jth class. Consequently, Given a set of labeled training examples, D = {d1, d2, . . . , d | D |}, we first learn (estimate) the parameters of the probabilistic classification model, θˆ, and then using the estimates of these parameters, we perform the classification of test documents by calculating the posterior probabilities of each class cj, given the test document, and select the most likely class (class with the highest probability).

$$P(c_j|d_i;\hat{\theta}) = \frac{P(c_j|\hat{\theta})P(d_i|c_j;\hat{\theta}_j)}{P(d_i|\hat{\theta})} = \frac{P(c_j|\hat{\theta})P(w_1, w_2, \ldots, w_{n_i}|c_j;\hat{\theta}_j)}{\sum_{c\in C} P(w_1, w_2, \ldots, w_{n_i}|c;\hat{\theta}_c)P(c|\hat{\theta})} \quad (6)$$

Where based on naive Bays assumption, words in a document are independent of each other, thus:

$$P(w_1, w_2, \ldots, w_{n_i}|c_j;\hat{\theta}_j) = \prod_{i=1}^{n_i} P(w_i|c_j;\hat{\theta}_j) \quad (7)$$

5.2 Nearest Neighbor Classifier

Nearest neighbor classifier is a proximity-based classifier which use distance-based measures to perform the classification. The main idea is that documents which belong to the same class are more likely "similar" or close to each other based on the similarity measures such as cosine defined in (2.2). The classification of the test document is inferred from the class labels of the similar documents in the training set. If we consider the k-nearest neighbor in the training data set, the approach is called k-nearest neighbor classification and the most common class from these k neighbors is reported as the class label,.

5.3 Decision Tree classifiers

Decision tree is basically a hierarchical tree of the training instances, in which a condition on the attribute value is used to divide the data hierarchically. In other words decision tree [50] recursively partitions the training data set into smaller subdivisions based on asset of tests defined at each node or branch. Each node of the tree is a test of some attribute of the training instance, and each branch

Descending from the node corresponds to one the value of this attribute. An instance is classified by beginning at the root node, testing the attribute by this node and moving down the tree branch corresponding to the value of the attribute in the given instance. And this process is recursively repeated [10].

In case of text data, the conditions on the decision tree nodes are commonly defined in terms of terms in the text documents. For instance a node may be subdivided to its children relying on the presence or absence of a particular term in the document. For a detailed discussion of decision trees.

Decision trees have been used in combination with boosting techniques. [9] Discuss boosting techniques to improve the accuracy of the decision tree classification.

5.4 Support Vector Machines

Support Vector Machines (SVM) are supervised learning classification algorithms where have been extensively used in text classification problems. SVM are a form of Linear Classifiers. Linear classifiers in the context of text documents are models that making a classification decision is based on the value of the linear combinations of the documents features. Thus, the output of a linear predictor is defined to be y = a® • x® + b, where x® = (x1, x 2, . . . , xn) is the normalized document word frequency vector, a® = (a1, a2, . . . , an) is vector of coefficients and b is a scalar. We can interpret the predictor y = a® • x® + b in the categorical class labels as a separating hyper plane between different classes.

The SVM initially introduced in [3]. Support Vector Machines try to find"good" linear separators between various classes [4]. A single SVM can only separate two classes, a positive class and a negative class [6]. SVM algorithm attempts to find a hyper plane with the maximum distance ξ (also called margin) from the positive and negative examples. The documents with distanceξ from the hyper plane are called support vectors and specify the actual location of the hyper plane. If the document vectors of the two classes are not linearly separable, a hyper plane is determined such that the least number of document vectors are located in the wrong side.

One advantage of the SVM method is that, it is quite robust to high dimensionality, i.e. learning is almost independent of the dimensionality of the feature space. It rarely needs feature selection since it selects data points (support vectors) required for the classification [6]. Joachims et al. [4] has described that text data is an ideal choice for SVM classification due to sparse high dimensional nature of the text with few irrelevant features. SVM methods have been widely used in many application domains such as pattern recognition, face detection and spam filtering . For a deeper theoretical study of SVM method sees .

VI. Clustering IN TEXT MINING

Clustering method can be used in order to find groups of documents with similar content. The outcome of clustering is typically a partition called clusters P and each cluster consists of a number of documents d. The contents of the documents within one cluster are more similar and between the clusters more dissimilar then the quality of clustering is considered better. Even though clustering technique used to group similar documents it differs from categorization because in clustering documents are clustered on the fly instead of use of predefined topics. As documents can appear in multiple subtopics clustering ensures that a useful document will not be omitted from search results [7].

In data mining K-means is frequently used clustering algorithm; in text mining field also it obtains good results. A basic clustering algorithm creates a vector of topics for each document and measures the weights of how well the document fits into each cluster. The organization of management information systems makes use of clustering technology as organizational database contain thousands of documents.

6.1 Hierarchical Clustering algorithms

Hierarchical clustering algorithms received their name because they build a group of clusters that can be depicted as a hierarchy of clusters. The hierarchy can be constructed in top-down (called divisive) or bottom-up (called agglomerative) fashion. Hierarchical clustering algorithms are one of the Distanced-based clustering algorithms, i.e. using a similarity function to measure the closeness between text documents. An extensive overview of the hierarchical clustering algorithms for text data is found in .In the top-down approach we begin with one cluster which includes all the documents. We recursively split this cluster into sub-clusters. In the agglomerative approach, each document is initially considered as an individual cluster. Then successively the most similar clusters are merged together until all documents are embraced in one cluster. There are three different merging methods for agglomerative algorithms: 1) Single Linkage Clustering: In this technique, the similarity between two groups of documents is the highest similarity between any pair of documents from these groups. 2) Group-Average Linkage Clustering: In group-average clustering, the similarity between two clusters is the average similarity between pairs of

documents in these groups. 3) Complete Linkage Clustering: In this method, the similarity between two clusters is the worst case similarity between any pair of documents in these groups. For more information about these merging techniques see [1].

6.2 k-means Clustering

K-means clustering is one the partitioning algorithms which is widely used in the data mining. The k-means clustering, partitions n documents in the context of text data into k clusters. Representative around which the clusters are built. The basic form of k-means algorithm is: Finding an optimal solution for k-means clustering is computation-ally difficult (NP-hard), however, there are efficient heuristics such as [18] that are employed in order to converge rapidly to a local optimum. The main disadvantage of k-means clustering is that it is indeed very sensitive to the initial choice of the number of k. Thus, there are some techniques used to determine the initial k, e.g. using another lightweight clustering algorithm such as agglomerative clustering algorithm. More efficient k-means clustering algorithms can be found in [7, 79].

ALGORITHM 1: *k*-means clustering algorithm

Input : Document set $\mathcal{D}$, similarity measure $\mathcal{S}$, number k of cluster

Output: Set of k clusters

initialization

Select randomly k data points as starting centroids.

while *not converged* **do**

  Assign documents to the centroids based on the closest similarity.

  Calculate the the cluster centroids for all the clusters.

end

return *k clusters*

6.3 Probabilistic Clustering and Topic Models Topic modeling is one of the most popular the probabilistic clustering

Algorithms which has gained increasing attention recently. The main idea of topic modeling [16] is to create a probabilistic generative model for the corpus of text documents. In topic models, documents are mixture of topics, where a topic is a probability distribution over words.

The two main topic models are Probabilistic Latent Semantic Analysis (pLSA) [6] and Latent Dirichlet Allocation (LDA) [16].

Hofmann (1999) introduced pLSA for document modeling. pLSA model does not provide any probabilistic model at the document level which makes it difficult to generalize it to model new unseen documents. Blei et al. [16] extended this model by introducing a Dirichlet prior on mixture weights of topics per documents, and called the model Latent Dirichlet Allocation (LDA). In this section we describe the LDA method.

The latent Dirichlet allocation model is the state of the art unsu-pervised technique for extracting thematic information (topics) of a collection of documents. [16]. The basic idea is that documents are represented as a random mixture of latent topics, where each topic is a probability distribution over words. The LDA graphical representation is shown is Fig. 1.

Let D = {d1, d2, . . . , d | D | } is the corpus and V = {w1, w2, . . . , w |V | } is the vocabulary of the corpus. A topic zj , 1 ≤ j ≤ K is rep-

resented as a multinomial probability distribution over the |V| words, p(wi |zj), Íi|V | p(wi |zj) = 1. LDA generates the words in a two-stage process: words are generated from topics and topics are generated by documents. More formally, the distribution of words given the document is calculated as follows:

$$p(w_i|d) = \sum_{j=1}^{K} p(w_i|z_j)p(z_j|d) \qquad (8)$$

The LDA assumes the following generative process for the corpus D:

(1) For each topic k ∈ {1, 2, . . . ,K}, sample a word distribution φk ∈ Dir(β)

(2) For each document d ∈ {1, 2, . . . ,D},

(a) Sample a topic distribution θd Dir(α)

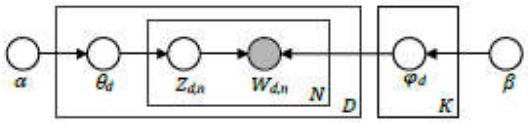


Figure 1: LDA Graphical Model

$$P(\phi_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D}) = \prod_{j=1}^{K} P(\phi_j|\beta) \prod_{d=1}^{D} P(\theta_d|\alpha) \times \left(\prod_{n=1}^{N} P(z_{d,n}|\theta_d) P(w_{d,n}|\phi_{1:K}, z_{d,n}) \right) \quad (9)$$

(b) For each wordwn, where n {1, 2, . . . , N}, in document d,

i. Sample a topic zi Mult(θd)

ii. Sample a word wn Mult(φzi)

$$P(\varphi_{1:K}, \theta_{1:D}, z_{1:D}|w_{1:D}) = \frac{P(\varphi_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D})}{P(w_{1:D})} \quad (10)$$

The joint distribution of the model (hidden and observed variables) is: for more complex goals. Furthermore, LDA has been extensively used in a wide variety of domains. Chemudugunta et al.combined LDA with concept hierarchy to model documents. [2, 5] developed ontology-based topic models based on LDA for automatic topic labeling and semantic tagging, respectively. [4] proposed acknowledge-based topic model for context-aware recommendations. defined more complex topic models based on LDA for entity disambiguation, [3] and has proposed a entity-topic models for discovering coherence topics and entity linking, respectively. Additionally, many variations of LDA have been created such as supervised LDA (sLDA) [15], hierarchical LDA (hLDA) [14] and Hierarchical pachinko allocation model (HPAM) [100].

CONCLUSION

In this article we endeavored to give a short prologue to the field of text mining. We gave a review of the absolute most central algorithms and procedures which are broadly utilized as a part of the text area. This paper likewise outlined some of vital text mining approaches in the biomedical area. Despite the fact that, it is difficult to depict every extraordinary technique and algorithms altogether with respect to the furthest reaches of this article, it should give a harsh diagram of current advances in the field of text mining. Text mining is fundamental to logical research given the specific high volume of logical writing being delivered each year These extensive chronicles of online logical articles are developing essentially as a lot of new articles is included a consistent schedule. While this development has empowered specialists to effortlessly get to more logical information, it has likewise made it very troublesome for them to recognize articles more related to their interests. In this manner, preparing and mining this enormous measure of text is of incredible enthusiasm to scientists.

REFERENCES

[1] Charu C Aggarwal and ChengXiang Zhai. 2012. Mining text data. Springer.

[2] Mehdi Allahyari and Krys Kochut. 2015. Automatic topic labeling using ontology-based topic models. In Machine Learning and Applications (ICMLA), 2015 IEEE 14th International Conference on. IEEE, 259–264.

[3] Mehdi Allahyari and Krys Kochut. 2016. Discovering Coherent Topics with Entity Topic Models. In Web Intelligence (WI), 2016 IEEE/WIC/ACM International Conference on. IEEE, 26–33.

[4] Mehdi Allahyari and Krys Kochut. 2016. Semantic Context-Aware Recommenda-tion via Topic Models Leveraging Linked Open Data. In International Conference on Web Information Systems Engineering. Springer, 263–277.

[5] Mehdi Allahyari and Krys Kochut. 2016. Semantic Tagging Using Topic Models Exploiting Wikipedia Category Network. In Semantic

Computing (ICSC), 2016 IEEE Tenth International Conference on. IEEE, 63–70.

[6] M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E. D. Trippe, J. B. Gutierrez, and K. Kochut. 2017. Text Summarization Techniques: A Brief Survey. ArXiv e-prints (2017). arXiv:1707.02268

[7] Khaled Alsabti, Sanjay Ranka, and Vineet Singh. 1997. An efficient k-means clustering algorithm. (1997).

[8] Sophia Ananiadou, Sampo Pyysalo, Jun'ichi Tsujii, and Douglas B Kell. 2010. Event extraction for systems biology by text mining the literature. Trends in biotechnology 28, 7 (2010), 381–390.

[9] Peter G Anick and Shivakumar Vaithyanathan. 1997. Exploiting clustering and phrases for context-based information retrieval. In ACM SIGIR Forum, Vol. 31. ACM, 314–323.

[10] Sofia J Athenikos and Hyoil Han. 2010. Biomedical question answering: A survey. Computer methods and programs in biomedicine 99, 1 (2010), 1–24.

[11] L Douglas Baker and Andrew Kachites McCallum. 1998. Distributional clustering of words for text classification. In Proceedings of the 21st annual internationalACM SIGIR conference on Research and development in information retrieval. ACM, 96–103.

[12] Ron Bekkerman, Ran El-Yaniv, Naftali Tishby, and Yoad Winter. 2001. On feature distributional clustering for text categorization. In Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 146–153.

[13] Daniel M Bikel, Scott Miller, Richard Schwartz, and Ralph Weischedel. 1997. Nymble: a high-performance learning name-finder. In Proceedings of the fifth conference on Applied natural language processing. Association for Computational Linguistics, 194–201.

[14] David M Blei, Thomas L Griffiths, Michael I Jordan, and Joshua B Tenenbaum. 2003. Hierarchical Topic Models and the Nested Chinese Restaurant Process.. In NIPS, Vol. 16.

[15] David M Blei and Jon D McAuliffe. 2007. Supervised Topic Models.. In NIPS, Vol. 7. 121–128.

[16] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. the Journal of machine Learning research 3 (2003), 993–1022.

[17] Olivier Bodenreider. 2004. The unified medical language system (UMLS): integrat-ing biomedical terminology. Nucleic acids research 32, suppl 1 (2004), D267–D270.

[18] Paul S Bradley and Usama M Fayyad. 1998. Refining Initial Points for K-Means Clustering.. In ICML, Vol. 98. Citeseer, 91–99.

[19] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. 1984. Classification and regression trees. CRC press.

[20] Markus Bundschus, Mathaeus Dejori, Martin Stetter, Volker Tresp, and Hans-Peter Kriegel. 2008. Extraction of semantic biomedical relations from text using conditional random fields. BMC bioinformatics 9, 1 (2008), 207.

[21] Christopher JC Burges. 1998. A tutorial on support vector machines for pattern recognition. Data mining and knowledge discovery 2, 2 (1998), 121–167.

[22] Igor Cadez, David Heckerman, Christopher Meek, Padhraic Smyth, and Steven White. 2003. Model-based clustering and visualization of navigation patterns on a web site. Data Mining and Knowledge Discovery 7, 4 (2003), 399–424.

[23] Bob Carpenter. 2010. Integrating out multinomial parameters in latent Dirichlet al-location and naive bayes for collapsed Gibbs sampling. Technical Report. Technical report, LingPipe.

[24] Soumen Chakrabarti, Byron Dom, Rakesh Agrawal, and Prabhakar Raghavan. 1997. Using taxonomy, discriminants, and signatures for navigating in text databases. In VLDB, Vol. 97. 446–455.

[25] Yee Seng Chan and Dan Roth. 2010. Exploiting background knowledge for relation extraction. In Proceedings of the 23rd International Conference on Computational Linguistics. Association for Computational Linguistics, 152–160.

[26] Yee Seng Chan and Dan Roth. 2011. Exploiting syntactico-semantic structures for relation extraction. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1. Association for Computational Linguistics, 551–560.

[27] Chaitanya Chemudugunta, America Holloway, Padhraic Smyth, and Mark Steyvers. 2008. Modeling documents by combining semantic concepts with unsupervised statistical learning. In The Semantic Web-ISWC 2008. Springer, 229–244.

[28] Ming-Syan Chen, Jiawei Han, and Philip S. Yu. 1996. Data mining: an overview from a database perspective. IEEE Transactions on Knowledge and data Engineering 8, 6 (1996), 866–883.

[29] Hai Leong Chieu and Hwee Tou Ng. 2003. Named Entity Recognition with a Maximum Entropy Approach. In Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4 (CONLL '03). Association for Computational Linguistics, Stroudsburg, PA, USA, 160–163. https://doi.org/10. 3115/1119176.1119199

[30] Aaron M Cohen and William R Hersh. 2005. A survey of current work in biomedical text mining. Briefings in bioinformatics 6, 1 (2005), 57–71.

[31] K Bretonnel Cohen and Lawrence Hunter. 2008. Getting started in text mining. PLoS computational biology 4, 1 (2008), e20.

[32] Nigel Collier, Chikashi Nobata, and Jun-ichi Tsujii. 2000. Extracting the names of genes and gene products with a hidden Markov model. In Proceedings of the 18th conference on Computational linguistics-Volume 1. Association for Computational Linguistics, 201–207.

[33] Peter Corbett and Ann Copestake. 2008. Cascaded classifiers for confidence-based chemical named entity recognition. BMC bioinformatics 9, Suppl 11 (2008), S4.

[34] Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. Machine learning 20, 3 (1995), 273–297.

[35] Jim Cowie and Wendy Lehnert. 1996. Information extraction. Commun. ACM 39, 1 (1996), 80–91.

[36] Douglass R Cutting, David R Karger, and Jan O Pedersen. 1993. Constant interaction-time scatter/gather browsing of very large document collections. In Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 126–134.

[37] Douglass R Cutting, David R Karger, Jan O Pedersen, and John W Tukey. 1992. Scatter/gather: A cluster-based approach to browsing large document collections. In Proceedings of the 15th annual international ACM SIGIR conference on Research