Journal of Theoretical and Applied Information Technology
30th June 2022. Vol.100. No 12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AUTOMATION OF INCIDENT RESPONSE AND IT TICKET MANAGEMENT BY ML AND NLP MECHANISMS

M.VENKATA SUBBARAO[1], KASUKURTY VENKATARAO[2], CH. SURESH[3]

[1]Research Scholar, Andhra University, Department of Computer Science & Engineering, Vizag, India

[2]Prefessor, Andhra University, Department of Computer Science & Engineering, Vizag, India

[3] Professor, Gitam University, Department of Computer Science & Engineering, Vizag, India

E-mail: [1]subbaraomudragada@gmail.com, [2]professor_venkat@yahoo.com, [3]sureshchitteneni@gmail.com

ABSTRACT

In today's digital world, Ticket Management Systems are widely utilized in various businesses and organizations since they are essential for efficiently resolving client requests and difficulties. Moreover, in real-world application contexts (e.g., technology support platforms and defect detection systems), properly labeling new requests is essential to improving ticket handling quality, grade and productivity. This research aims to discover the factors affecting customers' online ticket purchasing behaviors and proposes an effective technique for systematic ticket categorization that effectively clusters and labels. Information technology (IT) tickets quickly using Machine Learning and Natural Language Processing techniques. The framework's performance in numerous ticket categorization jobs has been indicated by experimental results based on a specific usage situation that involves data from ticket mining the ServiceNow platform and filtering a vast dataset, among other things. To develop a Machine Learning model that significantly predicts an incident resolution category with supervised ML algorithms. The models' performance was estimated using various NLP techniques LDA with TF-IDF and 3 Gram, and stop-word removal and lemmatization produced the best results, with a precision of 96.46 percent.

Keywords: *Natural language processing (NLP), Machine Learning, classification, Incident Response, Text Mining, Information Retrieval*

1. INTRODUCTION

Trust is the essential key factor in success and gaining a competitive advantage against rivals in e-commerce. Nowadays, we depend extensively on computers for our day-to-day activities, especially with increases in backup capacities, faster processing, and internet bandwidths. Humans can be replaced with little care but replacing a computer system is significant. Because of this, information technology (IT) has become a substantial concern in virtually every enterprise, encouraging the development of worldwide benchmarks in IT service management (ITSM) [1]. The ITSM is a vital business task that is accountable for increasing organizational productivity, harnessing innovation to generate value, and delivering end-to-end professional services, among other things. Front-end IT interactions are frequently characterized by extended, laborious conversations with support professionals (online or by phone). Whether you seek a new password, make software modifications, or merely seek assistance, you are in for a tedious experience. Nowadays, IT leaders struggle for staff and manage a complete help-desk management solution that helps the rest of the company, so the stigma remains. The incident management method governs service interruptions, system and application failures, and any problem affecting the entire stack of servers [2]. Incident management focuses on restoring regular service as quickly as possible to reduce the negative effect on the business. An issue's symptoms are described, as are the event's resolution and many structure fields, such as the date, resolver, servers, and impacted services. This information is found in the incident tickets. In this way, a classification of the incident tickets that have happened in a specific IT environment would provide a clear picture of the issues that are currently being experienced [3]. Focusing on the most vital or pervasive issue categories and specifying strategies to prevent them from happening again will allow efficient overall incident management.

IT service managers, for example, may concentrate on resolving the root causes of server downtime. Server unavailability is a critical incident category because it affects corporate income and productivity. Non-actionable incident tickets could be another area of concern. These are incident reports generated by systems that don't need to be fixed since the problems are either temporary or acceptable. Even though these tickets do not indicate service disruptions, they serve to deflect attention away from more pressing issues of the day. To boost productivity, it is necessary to discover and remediate these issues, for example, by fine-tuning monitoring systems [4].

Moreover, in most ITSM platforms, identifying the topic or categorization of the IT ticket content takes a manual effort by a maintenance engineer. An incident is defined as an unanticipated interruption or deterioration in the quality of an IT service. By addressing the problems of the raised tickets, a ticketing approach attempts to reduce the business effect of every incident. Any prior knowledge derived from ticket data can aid in the speedy resolution of the issue [5]. In a particular IT environment, every incident ticket is mainly categorized into separate clusters; therefore, it is possible to understand the system's troubles better. This can also help the system characterize warnings, events, and requests. This can lead to effective ticket administration capable of dealing with the most severe issues or common incidents. Clustering and classifying incident tickets, on the other hand, can be complex for a variety of reasons. First, more tickets would rise to 1000 per year in a large IT ecosystem, making classification unfeasible [6]. A ticket description could include a combination of user and machine-generated language (from the monitoring system) as a sound problematic-specific phrase.

Furthermore, the tickets are reported by numerous teams operating various management methods and wording. Therefore, the linguistic descriptions of the tickets can vary from one IT environment to the next. Therefore, this study aims to categorize IT incidents initiated by end-users based on the content in the IT ticket using machine learning and NLP techniques to reduce the requirement for human intervention in classifying IT incident tickets or help requests on their topic.

2. RELATED WORK

Automatic text categorization has been proposed using a variety of supervised machine-learning algorithms, including support vector machines(SVM), Naive Bayes(NB), maximum entropy (multinomial logistic regression), and gradient boosting machine (GBM) [7]. Some of these methodologies and rule-based approaches have been used for maintenance and incident ticket classification. It is vital to note that topic modeling is a branch of text mining in which each document is supposed to be a collection of abstract themes. An unsupervised approach can be used to cluster a collection of documents based on the most prevalent subjects in this manner. Recent works in this area are based on Latent Dirichlet Allocation (LDA), a probability distribution type. A strategy for producing a hierarchical topic structure is provided in [7], and a background topic is introduced in to detect uninformative terms that appear across all topics described in [8]. The construction of a document-document graph and the subsequent clustering of documents by identifying communities in the network [9] are two alternative approaches to text classification.

Finally, minimizing the labeling effort required for various classification jobs is a significant issue that has been addressed in many publications. Activated learning enables this by automatically picking the most informative unlabeled examples for manual annotation [10], which machine learning achieves

In this area, [11] conducted a similar study. Data has been gathered from Istanbul Technical University's Issue Tracking System, and their research intended to categorize over 10000 IT issues into four categories and twenty subcategories. They worked on a Turkish dataset. The findings demonstrated that the training datasets have an impact on algorithm accuracy. SVM Naive Bayes and Decision Tree techniques outperformed on more extensive training datasets, while smaller training datasets benefited from the Naive Bayes [10] and Decision Tree approach [12]. A positive matrix factorization technique called k-means has been used to group the tickets into clusters rather than depending on labeled training sets, as described in [13]. A 60 to 70% similarity between the clusters was discovered using k-means and Nonnegative Matrix Factorization, with the proportion altering depending on which groups were to be removed. The higher the similarity of the findings, the fewer clusters they used. Decision tree algorithms have been used in a study [14] to classify IT events based on the resolved category. As a result, researchers acquired over 90% efficiency, including all decision tree approaches for allocating issues to resolving groupings. Their research used a dataset of 14440 items, separated into five categories. The study shows that decision trees can be highly successful even with a small dataset. A similar approach is utilized by the researchers in this study [15], although the information used to categorize the occurrences differs. In their experiments, words must be associated with two categories, and the strength of the link will decide the power of the association. An unbalanced dataset and the limited number of classes make their trials rely on characteristics such as precision, recall, and F1 score because of the restricted categories.

Issue tickets can be mined and clustered using the Trouble Miner mechanism [16]. Network events and maintenance operations can be analyzed using trouble tickets from the network affected by the problem. It was developed in [17] to use clustering to classify event data. The authors' initial stages are carried out using graph clustering and topic modeling, followed by hierarchical clustering or active learning. "Three real-world datasets evaluate the technique's human labeling effort, accuracy, and efficiency. Unsupervised learning is used in another work [18] to classify warnings and occurrences according to linguistic information.

The prime objective of our technique is to limit the amount of time spent manually categorizing tickets while keeping high levels of prediction accuracy. In addition, to making this possible, the system administrator groups are given the ability to classify tickets independently [19]. Given that they are the subject matter experts with the most in-depth understanding of the target area, this assures reliable ticket categorization, which is critical to the overall quality of the final classification.

There are a variety of techniques for classifying text and different documents. Using the LDA (Latent Dirichlet Allocation) technique, Blei divides the magazine Science's 17.000 articles into 100 categories [41]. The discovered categories offered an excellent and comprehensive overview of the various topics. LDA and NMF are part of the topic models employed in this study. [42] details addition to LDA. A hierarchical version of this was used to categorize news stories on the disappearance of Malaysia Airlines Flight 370. Theories like a plane crash or terrorism were then broken down into subcategories. Using NMF (Nonnegative Matrix Factorization), Kuang et al. demonstrate how to classify documents. A baseline k-means clustering is also used to compare the outcomes of NMF on five distinct data sets, including the well-known ones from 20Newsgroups and Reuters. NMF performed better in four out of five cases. According to Shahnaz et al. [43], clustering accuracy is highly dependent on the dataset and cluster size. For example, the Reuters dataset has a 99 percent accuracy rate with just two categories. Only 54% of the categories have been included. However, they could correctly identify more than 80 percent of the groups on another dataset. Another approach uses clustering algorithms and keyword lists to produce training data before training an SVM or Naive Bayes classifier in the second step. Afterward, the classifier is utilized to categorize various texts. So they use both unsupervised and supervised learning techniques. In [44], Ko and Seo use a list of keywords to sort the materials into sentences. A naive Bayes classifier is trained with these sentences. A pure supervised learning approach yielded similar results. Nonetheless, it might be helpful when creating training data. [45] describes a method for generating training data by utilizing automatically generated keywords. After evaluating their approach using a dataset of 20Newsgroups.com, they found that employing only a modest number of keywords was the most effective. In addition, they saw a slight boost if humans filtered out the terms that were automatically generated.

2.0 Problem Identification and Motivation

Relying solely on a service agent's subjective personal experience creates an undue dependency when responding to incidents and the high volume of incidents and the incorrect assignment of a resolver group, which harms the incident route. But requires more resources and leads to wasted time. Therefore, it should be considered (Because many incidents aren't caused by something that hasn't been seen before and has a known solution in its knowledge base, this can lead to wasted time and resources for resolvers. In addition, the incident resolution process is time-consuming and inefficient. By automating some of these steps, this waste can be reduced.

2.1 The objective of a Solution

Using Machine learning techniques, IE, and NLP, this study tries to discover a way to automatically extract and propose appropriate resolution actions for new incident tickets.

The following is just a breakdown of the publication's structure: The proposed extracting features approaches and explain the present gaps in this field of study and potential future directions are described in Section 3. Section 4 contains thorough explanations of standard datasets and different preprocessing techniques. Section 5 discusses a new model for clustering with TF-IDF. Section 6

Journal of Theoretical and Applied Information Technology
30th June 2022. Vol.100. No 12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

explains the results. Finally, the conclusion summarizes the automation of incident response and it ticket management by ML and NLP mechanisms from this research in section 7.

3. PROPOSED APPROACH

To categorize support tickets, they are first preprocessed, and then to cluster, the data is transformed into feature vectors using algorithms such as TF-IDF, Weighted gram, and Trigram on which the clustering algorithms such as K-Means [20], LDA [21], and NMF [22] are applied to categorize them. To compare the efficiency of our approach, we have reached our results with the categories in the dataset. Accuracy and similarity are used as metrics for this comparison. In addition, it's possible to use a pre-trained version of the algorithm to predict the class of new tickets in the future.

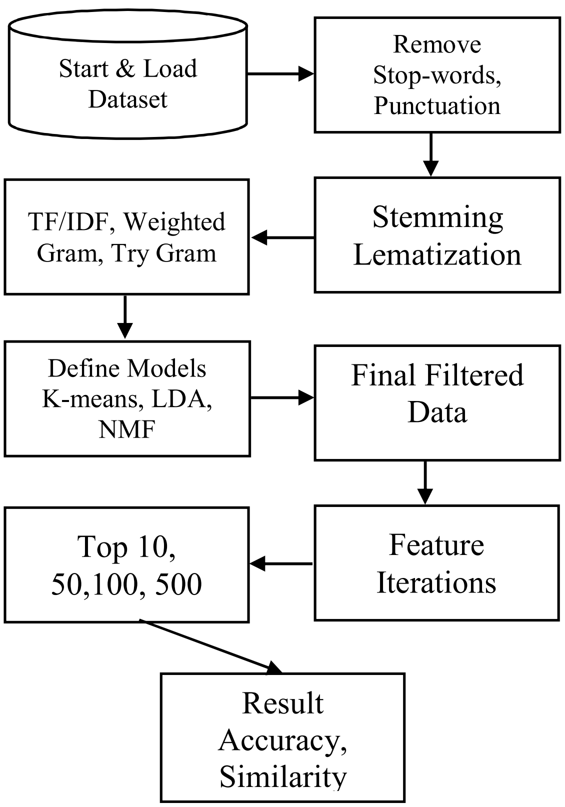

Figure 1: Proposed Architecture

4. DATASET

The dataset for this research is taken from the ServiceNow Platform [23]. We obtained the IT incident/request data from the service Now platform [24] using the Table API [25] technology. TableAPI provides an inbuilt API Explorer. We have curated this dataset by creating bespoke API queries from the ground up, breaking down query parameters, fields, and other details into manageable chunks. The curated dataset contains 47664 incidents/request records and nine attributes.

Table 1: Attributes in the data set.

Attribute	Description
Title	It is an agent who entered a title for the ticket.
Body	It contains the agent entry of the ticket description.
Ticket type	It contains a Numerical Value of, 0 refers to e-mail, and 1refers to phone.
Category	The quantity in this area ranges from 0 to 12.
Sub category1	This field includes a number from 0 to 58.
Sub category2	This field holds a number from 0 to 118.
Business service	This field includes a number from 0 to 102.
Urgency	There is a range of 0 to 3 in this field.3 denotes a very urgent ticket, while 0 indicates no urgency.
Impact	A numerical value between zero and four can be found in this area. 5 signifies the highest impact, and zero is the lowest.

4.0 Pre-processing

In step1, the data to consider for this study must be normalized and preprocessed, including blank/null data removal and several other preprocessing steps. Next, the title and the message are the two sections of a support ticket. Finally, the title and message are merged and transformed to lower case for further processing attributes.

4.1 Removing Stop Words

Stop words are unnecessary since they carry almost no information and can thus be eliminated. So instead, we employed the NLTK framework's stop word list1, supplemented with a list of words that could impair cluster naming, including salutation and greeting expressions.

4.2 Removing Punctuations

Once the stop words are removed, we have also carried out the punctuation removal, wherein the symbols are removed as they too do not carry any meaningful information [26].

4.3 Stemming

Stemming is a technique for eliminating prefixes and suffixes from words to retrieve the base form. We have used a Snowball stemmer [27]. The Snowball stemming is a method of removing grammatical and flexional suffixes from Lexical items

5. FEATURE VECTORS

5.0 Tfidf

Two assumptions underpin these feature vectors. First, words that frequently appear in a document are more significant or better characterized; second, terms that frequently appear in all documents are unimportant. Here mathematical measures 3,4, and 5 are waiting for the scheme for document term weight.

$$f_{t,d}.log\frac{N}{n_t} \quad (3)$$

$$\log(1+f_{t,d}) \quad (4)$$

$$(1+logf_{t,d}).log\frac{N}{n_t} \quad (5)$$

Pseudocode:

1. V←Sort by key(B.id)
2. C←Reduce by key(V.id)
3. for i←1to k do
4. id =Bi.id
5. T Fi=Bi.counter / Get_total(C, id)
6. end for
7. for i←1to k do
8. Ci.total = 1
9. end for
10. D←Count(C.id)
11. V←Sort by key (B.t)
12. for I←1to k do
13. Ci.t =Vi.t
14. Ci. total = 1
15. end for
16. C←Reduce by key (C.t)
17. for i←1to k do
18. t=Bi.t
19. IDFi=log (D/ Get_total(C, t))
20. end for
21. for i←1to k do
22. T F I DFi←T Fi∗IDFi
23. end for

5.1 Weighted Gram

Weighted N-gram is a symbol sequence extracted from a long string. A character, byte, or word is a sequence of symbols. The semantics of text are better conveyed when word sequencing is factored in. As a result, we are looking at N-grams at the word level, with adjacent N-words acting as N-grams [31]. That means bigrams, trigrams, and other combinations can be found. As a result, N-gram representation would be less susceptible to linguistic and typos problems.

$$P(w_n \mid w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1}w)} \quad (6)$$

5.2 Trigram

We use [32] to build vectors based on word trigrams for the subject and its candidate labels. There are three aspects to a word trigram vector, symbolizing one of the letter trigrams (e.g., ABC or acd). First, we generate a letter Trigram set for every word l. It is a word trigram set when you have all the sentence's word trigrams in one place. Every dimension of the phrase trigram vector of phrase l is assigned an integer value representing the frequency of the corresponding letter trigram in the multiset. Finally, the counts are normalized to add to one.

$$P^{\wedge}(w_n \mid w_{n-2}w_{n-1}) = \lambda_1 P(w_n \mid w_{n-2}w_{n-1}) + \lambda_2 P(w_n \mid w_{n-1}) + \lambda_3 P(w_n) \quad (7)$$

6. MODELS

To compare a standard cluster algorithm with a topic model, we utilize k-means. It is a simple baseline algorithm and NMF because it belongs to the topic models and gives descriptive findings.

6.0 k-means

K-means is a straightforward technique for dividing data into groups. A three-step process is possible. There is initial randomization of the cluster centroids in the first step of the clustering process. The cluster with the shortest distance is assigned to each data point x in the second phase. A third stage involves recalculating the cluster centroids. The average of all the data points allocated to the centroid is used to determine its location at any given point in space and time. It is repeated until cluster centroids no longer change.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Clustering data with K-Means [28] is a simple method. There are three steps to this process. The cluster centroids µ1, µK, and µK are randomly initialized in the first phase. Then, each data point x is grouped into clusters with the shortest distance in the second phase. In the third phase, the cluster centroids are recalculated. The centroid is always determined by averaging the values of all the data points that have been allocated to it. Steps 2 and 3 are performed up to no continuously changing cluster centroids.

1.Prepare cluster centroids µ1, µ2,….. µk ∈Rn randomly.

2. Repeat until convergence: {
   For every I, set
   c(i) = argminj||x(i)- µj||2
   For each j, set
   }

6.1 Nonnegative matrix factorization (NMF)

Nonnegative matrix factorization (NMF) is a method for representing a matrix as the product of two smaller matrices. This method can be used to group openers as well. NMF can be allocated to the topic available instead of traditional cluster techniques like k-means [30]. Like cluster methods, the Topic Model may detect patterns in documents. They divide the texts into topics, overlapping clusters rather than distinct groups.

Pseudocode:

Initialize W and H nonnegative. W and H are then updated by calculating the following using the iteration index as a reference.

$$H^{n+1}_{[i,j]} \leftarrow H^{n}_{[i,j]} \frac{((W^n)^T V)_{[i,j]}}{((W^n)^T W^n H^n)_{[i,i]}} \quad (1)$$

7.0

and

$$W^{n+1}_{[i,j]} \leftarrow W^{n}_{[i,j]} \frac{\left(V(H^{n+1})^T\right)_{[i,j]}}{(W^n H^{n+1}(H^{n+1})^T)_{[i,j]}} \quad (2)$$

8.0

Until W and H are stable.

Humans find it challenging to extract information from grouped papers without descriptive cluster names. As a result, one of the significant demanding challenges in record clustering is to develop a human-readable report. Thanks to the training data, this is simple to achieve with supervised learning algorithms. During clustering, however, these are absent. Cluster labeling is a term used in the literature to describe designating clusters. It could be accomplished in a variety of methods.

9.0

NMF has been widely utilized as a clustering approach, particularly for text data, and as a topic modeling method because it produces semantically relevant results easily interpretable in clustering applications. In-text mining, the NMF algorithm is used for document clustering based on a given term-document matrix (TDM), which the user provides. One of its most valuable characteristics is its ability to provide a sparse representation of the data, which allows for more accurate analysis. Furthermore, it is possible to develop positive factors for the TDM matrix. If this matrix can be used to analyze text document collections more effectively, the measure is given in equation (7) it can be used to analyze them more effectively.

$$\mathrm{J} = \frac{1}{2}||V - WH|| \quad (7)$$

NMF is used in various text mining applications, including text document clustering, and is very effective. NMF is used to analyze text collections in this work, and the results are presented. One such error measure is given in equation (8). The objective is to minimize this error function.

$$E(W,H) = \left||V - WH|\right|^2 = \sum_{i,j}(V_{ij} - (WH)_{ij})^2 \quad (8)$$

9.1 Latent Dirichlet allocation

It is a frequently utilized probability bag-of-words version called LDA [29]. Text corpus, groupings of discrete data, are generated using a probabilistic generative technique. Every collection element is a discrete mixture across an underlying set of themes in the LDA three-level hierarchical Bayesian paradigm. An endless blend of possibilities for each issue is therefore shown.

LDA presupposes that for each item w in a corpus D, the following creative process will take place:

1. Choose N ∼ Poisson(ξ).
2. Choose θ ∼ Dir(α).
3. For each of the N-words wn:
(a) Choose a topic in ∼ Multinomial(θ).
(b) Choose a word wn from p(wn |zn,β), a multinomial probability conditioned on the topic zn.

Pseudo code:

1. initialize φ 0 ni := 1/k for all i and n
2. initialize γi := αi +N/k for all i
3. repeat
4. for n = 1 to N

5. for i = 1 to k
6. φ t+1 ni := βiwn exp(Ψ(γ t i))
7. normalize φ t+1 n to sum to 1.
8. γ t+1:= α+∑ N n=1 φ t+1 n
9. until convergence

Text documents are considered as mixes of latent themes, which are essential concepts given in the text, according to the Latent Dirichlet allocation (LDA) model [33]. Using a conjugate Dirichlet prior that is the same for all documents, the topic mixture is done. The four steps are to discuss the topic modeling for e-mail text collecting using LDA. In the first phase, a multinomial t distribution for each topic t is chosen from a Dirichlet distribution with parameters. This distribution is used in the second and third steps. It is decided which multinomial distribution b to use for each ticket in the second phase by selecting it from a Dirichlet distribution with parameter b [34]. In a third stage, for each word w in the ticket tc, a topic t from the list b is randomly selected and used. Finally, in the fourth phase, the word "w" from the word "t" is chosen to indicate the topic of the text message content. The chance of creating a corpus is provided by the following equation: Eqn (1)

$$\iint \prod_{i=1}^{k} P\left(\theta_t|\beta\right) \prod_{b=1}^{N} P\left(\theta_b|\alpha\right) \left(\prod_{t=1}^{Nb} \sum_{b=1}^{K} P(t_i|\theta)P(w_i|t,\emptyset)\right) d\theta d\emptyset \quad (9)$$

When dealing with an unlabeled collection of documents, LDA estimates the topic-term distribution and the document topic distribution using Dirichlet priors for distributions over a fixed number of topics [35].

7. RESULTS

The category attribute in the dataset consists of 13 categories. All the tickets in the dataset are clustered under these 13 categories. Therefore, we have first created 13 clusters using K-Means, NMF, and LDA algorithms to compare our study, as depicted in Table 3 and Table 4.

Table 2: Performance of Ticket Labeling and Clustering.

Classifier	Feature Extraction	Accuracy
SVM	TF-TDF	90%
Naïve Bayes	TF-TDF	52%
KNN	TF-TDF	71%
Decision Tree	TF-TDF	84%
LDA	3 Gram	96.46%

Table 3: Performance of Ticket Labeling and Clustering algorithms with No.of Feature Vectors10,50. No.of Labels 130,650

No.of Feature Vectors = 10 No.of Labels = 130			
Labelling	Clustering	#Matching labels	Accuracy
TF-IDF	KMeans	72	55.38
	LDA	69	53.07
	NMF	71	54.61
Weighted Gram	KMeans	85	65.38
	LDA	54	41.53
	NMF	115	88.46
3Gram	KMeans	106	81.53
	LDA	126	96.92
	NMF	116	89.23
No.of Feature Vectors = 50 No.of Labels = 650			
Labeling	Clustering	#Matching labels	Accuracy
TF-IDF	KMeans	164	25.23
	LDA	206	31.69
	NMF	177	27.23
Weighted Gram	KMeans	215	33.07
	LDA	204	31.38
	NMF	463	71.23
3Gram	KMeans	240	36.92
	LDA	588	90.46
	NMF	496	76.3

Table 4: Performance of Ticket Labeling and Clustering algorithms with No.of Feature Vectors100,500. No.of Labels 1300,6500.

No.of Feature Vectors = 100 No.of Labels = 1300			
Labelling	Clustering	#Matching labels	Accuracy
TF-IDF	KMeans	223	17.15
	LDA	343	26.38
	NMF	287	22.07
Weighted Gram	KMeans	249	19.15
	LDA	416	32
	NMF	865	66.53
3Gram	KMeans	300	23.07
	LDA	1102	84.76
	NMF	1029	79.15
No.of Feature Vectors = 500 No.of Labels = 6500			
Labeling	Clustering	#Matching labels	Accuracy
TF-IDF	KMeans	80	1.23
	LDA	1161	17.86
	NMF	797	12.26
Weighted Gram	KMeans	56	0.86
	LDA	2336	35.93
	NMF	3724	57.29
3Gram	KMeans	209	3.21
	LDA	4441	68.32
	NMF	4620	71.07

Journal of Theoretical and Applied Information Technology
30th June 2022. Vol.100. No 12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

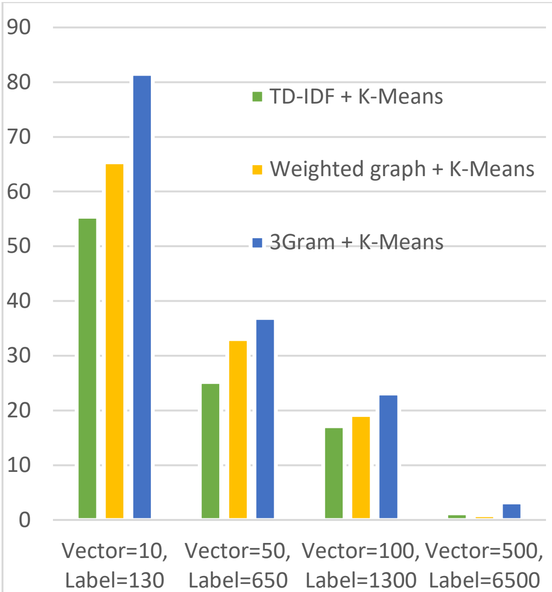

Figure 2: K-means Accuracy with Ticket Labeling and Clustering Algorithms

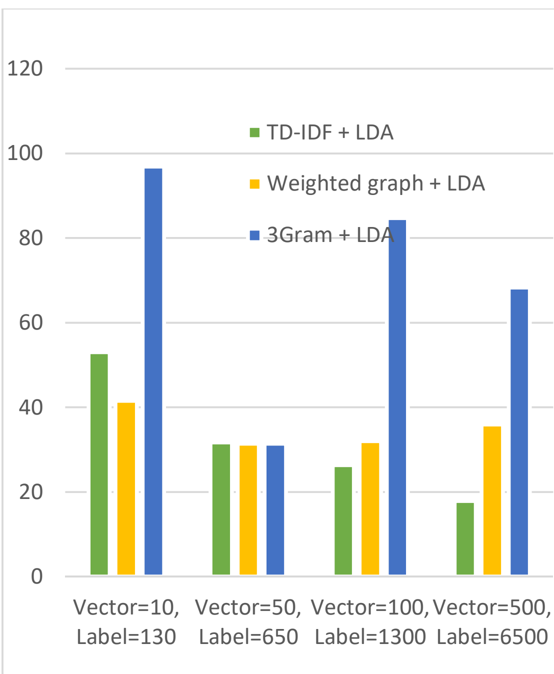

Figure 3: LDA Accuracy with Ticket Labeling and Clustering Algorithms

Then, based on the number of feature vectors from those clusters, we have generated cluster labels using the TFIDF, Weighted Gram, and Trigram's methods. We have considered feature vectors of 10, 50, 100, and 500 iterations, i.e., if the feature vector count is set to 10, the 13 clusters with 10 top labels are considered for each group [40]. Finally, these labels are compared with the Human identified labels for the same dataset, and accuracy is ascertained. The following table 2,3,4 shows the ticket labeling accuracy using our approach.

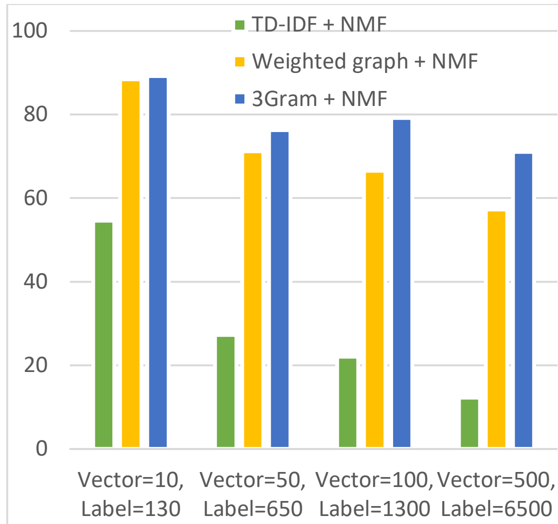

Figure 4: NMF Accuracy with Ticket Labeling and Clustering Algorithms

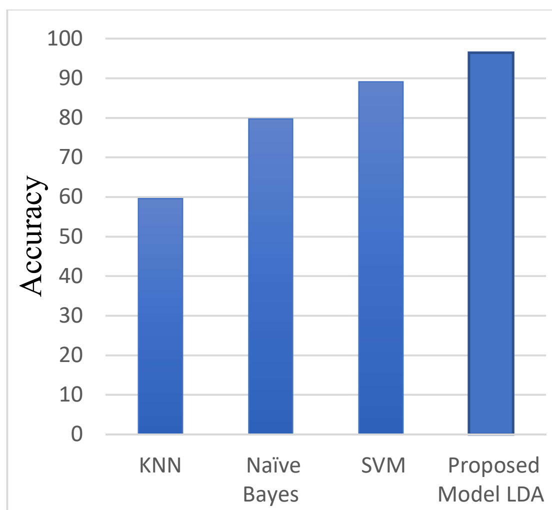

Figure 5: Compare Existing Classifier Models with the new proposed model.

Four distinct supervised machine learning approaches to categorize tickets with the best techniques: unpruned decision tree, SVM with the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

poly kernel, which supports non-linear models, nave Bayes, and k nearest neighbors with k equal 1. Figure 5 shows the results of several classifiers' performance comparisons. The average performance of machine learning algorithms is measured using ten-fold cross-validation.

Five classical supervised machine learning techniques, including an unpruned decision tree, SVM with the poly kernel (which allows non-linear models), naive Bayes, k nearest neighbors with k equal 1, and LDA, are applied using feature extracting techniques TI-IDF and 3Grams to classify tickets using the best learning techniques [40]. Figure 1 shows the results of comparing the performances of several classifiers, as shown in table 4. The average performance of machine learning algorithms is measured using ten-fold cross-validation. Fourteen thousand four hundred forty documents are contained within five categories in this collection. As a proof of concept, the study shows how effective decision trees maybe when dealing with a simple dataset.

8. CONCLUSION

This paper presents an extensive ticket-categorization technique, which employs Machine Learning and Natural Language Processing algorithms to cluster and label information technology tickets. The main advantage of our approaches is that they accomplish excellent prediction features on the ticket classification task with minimal labeling effort. Experimental findings based on a specific usage circumstance involving ticket-mining the service Now platform and curating an extensive dataset have proved the framework's efficiency in several ticket classification activities. When the labeling algorithm such as TF-IDF, weighted gram, and Trigram is combined with K means, LDA, and NMF clustering algorithms, Trigram has shown better performance over TF-IDF and Weighted Gram. When feature vectors increased from 10 to 500, TF-IDF showed inferior performance. We have found that 3 Gram, when combined with LDA or NMF, has shown better performance than K-Means. With a very high number of feature vectors, NMF has demonstrated better performance over LDA when combined with 3 Gram algorithms.

In contrast, for smaller feature vectors, LDA has performed better than NMF, but when the size of feature vectors grew, NMF showed better similarity than LDA. We estimate the performance of our methods and compare them with other text classification methods on three real-world datasets. Our existing techniques are particularly effective for minimal classes, such as Server unavailable, which is a crucial ticket category as it translates into unavailable services and directly impacts business revenue. To develop a Machine Learning model that significantly predicts an incident resolution category with supervised ML algorithms was used. The models' performance was evaluated using a variety of NLP techniques. LDA with TF-IDF and 3 Gram and stop-word removal and lemmatization produced the best results, with a precision of 96.46 percent.

REFERENCES:

[1] L. Tang, T. Li, F. Pinel, L. Shwartz, and G. Grabarnik, "Optimizing system monitoring configurations for non-actionable alerts," Proc. of the IFIP/IEEE Network Operations and Management Symposium (NOMS) 2012, pp. 34–42, 2012.

[2] Z. Wang, X. Sun, D. Zhang, and X. Li, "An optimal SVM-based text classification algorithm," Proc. of the 5th IEEE International Conference on Machine Learning and Cybernetics, pp. 1378–1381, 2006.

[3] B. Zhang, J. Su, and X. Xu, "A class-incremental learning method for multi-class support vector machines in text classification," Proc. of the 5th IEEE International Conference on Machine Learning and Cybernetics, pp. 2581–2585, 2006.

[4] B. Rujiang and L. Junhua, "A novel conception-based text classification method," Proc. of the IEEE International e-Conference on Advanced Science and Technology, pp. 30–34, 2009.

[5] S. Kim, K. Han, H. Rim, and S. H. Myaeng, "Some effective techniques for na¨ıve Bayes text classification," IEEE Transactions on Knowledge and Data Engineering, vol. 18, pp. 1457–1466, 2006.

[6] M. J. Meena and K. R. Chandran, "Na¨ıve Bayes text classification with positive features selected by statistical method," Proc. of the IEEE International Conference on Advanced Computing, pp. 28–33, 2009.

[7] K. Nigam, J. Lafferty, and A. McCallum, "Using maximum entropy for text classification," IJCAI-99 Workshop on Machine Learning for Information Filtering, pp. 61–67, 1999.

[8] J. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," Annals of Statistics, vol. 29, pp. 1189–1232, 2000.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[9] G. A. D. Lucca, M. D. Penta, and S. Gradara, "An approach to classify software maintenance requests," Proc. of IEEE ICSM, pp. 93–102, 2002.

[10] R. Gupta, H. Prasad, L. Luan, D. Rosu, and C. Ward, "Multi-dimensional Knowledge Integration for Efficient Incident Management in a Services Cloud," Proc. of IEEE SCC, pp. 57–64, 2009.

[11] Y. Diao, H. Jamjoom, and D. Loewenstern, "Rule-based Problem Classification in IT Service Management," Proc. of IEEE CLOUD, pp. 221– 228, 2009.

[12] C. Kadar, D. Wiesmann, J. Iria, D. Husemann, and M. Lucic, "Automatic classification of change requests for improved IT service quality," Proc. of SRII Global Conference, pp. 430–439, 2011.

[13] J. Bogojeska, I. Giurgiu, D. Lanyi, G. Stark, and D. Wiesmann, "Impact of HW and OS Type and Currency on Server Availability Derived from Problem Ticket Analysis," Proc. of the IFIP/IEEE Network Operations and Management Symposium (NOMS) 2014, pp. 34–42, 2014.

[14] D. M. Blei, Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," Journal of Machine Learning Research, vol. 3, pp. 993–1022, 2003.

[15] D. M. Blei, T. L. Griffiths, and M. I. Jordan, "The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies," Journal of the ACM (JACM), vol. 57, no. 2, p. 7, 2010.

[16] C. Chemudugunta, P. Smyth, and M. Steyvers, "Modeling General and Specific Aspects of Documents with a Probabilistic Topic Model," Proc. of the Twentieth Annual Conference on Neural Information Processing Systems (NIPS), pp. 241–248, 2006.

[17] Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," Physical Review E, vol. 70, 2004.

[18] D. A. Cohn, L. Atlas, and R. E. Ladner, "Improving generalization with active learning," Machine Learning, vol. 15, pp. 201–221, 1994.

[19] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," Physical Review E, vol. 69, 2004.

[20] J. Zhu, H. Wang, E. Hovy, and M. Ma, "Confidence-Based Stopping Criteria for Active Learning for Data Annotation," ACM Transactions on Speech and Language Processing, vol. 6, 2010.

[21] L. Berger, S. A. Della Pietra, and V. J. Della Pietra, "A maximum entropy approach to natural language processing." Computational Linguistics, vol. 22, pp. 39–71, 1996.

[22] D. M. Blei, A. Y.Ng, and M. I. Jordan, "Latent Dirichlet Allocation", Journal of Machine Learning Research, vol. 3, pp. 993-1022, 2003

[23] Zinner T., Lemmerich F., Schwarzmann S., Hirth M., Karg P., Hotho A. (2015) Text Categorization for Deriving the Application Quality in Enterprises Using Ticketing Systems. In:

[24] Maria S., Hara T. (eds) Big Data Analytics and Knowledge Discovery. DaWaK 2015. Lecture Notes in Computer Science, vol 9263. Springer, Cham.

[25] Altintas, M., & Tantug, A.C. (2014) Machine Learning-Based Ticket Classification in Issue Tracking Systems.

[26] Beneker, D., Gips, C. (2017). Using Clustering for Categorization of Support Tickets. LWDA

[27] Sakolnakorn, P.P., Meesad, P., Clayton, G. (2008). Automatic Resolver Group Assignment of IT Service Desk Outsourcing in Banking Business.

[28] C.J. van Rijsbergen, S.E. Robertson and M.F. Porter, 1980. New models in probabilistic information retrieval. London: British Library. (British Library Research and Development Report, no. 5587).

[29] Huang, P.S., He, X., Gao, J., Deng, L., Acero, A., Heck, L.: Learning deep structured semantic models for web search using clickthrough data.

[30] Proceedings of the 22nd ACM Conference on Information and Knowledge Management (CIKM 2013), pp. 2333–2338. San Francisco, USA (2013)

[31] Shahnaz, F., Berry, M.W., Pauca, V.P., Plemmons, R.J.: Document clustering using nonnegative matrix factorization. Information Processing & Management 42(2), 373–386 (Mar 2006) Weblink to ServiceNow platform accessed last on 8-Aug-2021 https://www.servicenow.com/

[32] G. A. Di Lucca, M. Di Penta, and S. Gradara, "An approach to classify software maintenance requests," in Proceedings of the International Conference on Software Maintenance. Piscataway, NJ: IEEE, 2002, pp. 93–102.

[33] Weblink to APIexplorer in ServiceNow last accessed on 8-Aug-2021https://developer.servicenow.com/dev.do#!/learn/learningplans/rome/servicenow_applicati

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

on_developer/app_store_learnv2_rest_rome_introduction_to_the_rest_api_explorer

[34] S. Godbole and S. Roy, "Text classification, business intelligence, and interactivity: automating c-sat analysis for the services industry," in Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Y. Li, B. Liu, and S. Sarawagi, Eds. New York, NY: ACM, 2008, pp. 911–919.

[35] Weblink to TableAPI last accessed on 8-Aug-2021https://developer.servicenow.com/dev.do#!/reference/api/rome/rest/c_TableAPI

[36] Leskovec, J., Rajaraman, A., Ullman, J.: Mining of Massive Datasets, 2nd and. Cambridge University Press, Cambridge (2014)

[37] Lin, D., Raghu, R., Ramamurthy, V., Yu, J., Radhakrishnan, R., Fernandez, J.: Unveiling alerts and incident management clusters in large-scale enterprise IT. In: The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2014, New York, NY, USA, 24–27 August 2014, pp. 1630–1639 (2014)

[38] Maksai, A., Bogojeska, J., Wiesmann, D.: Hierarchical incident ticket classification with minimal supervision. In: IEEE International Conference on Data Mining, ICDM 2014, pp. 923–928 (2014)

[39] Amelie, Medem & Akodjenou, Marc-Ismael & Teixeira, Renata. (2009). TroubleMiner: Mining network trouble tickets. 113 - 119. 10.1109/INMW.2009.5195946.

[40] Erdal Sever Iscen, M.Zahid Gurbuz, A Comparison of Text Classifiers on IT Incidents Using WEKA, in (UBMK'19) 4rd International Conference on Computer Science and Engineering – 510, IEEE, 2019.

[41] Blei, D.: Probabilistic Topic Models. Communications of the ACM 55(4), 77–84 (2012)

[42] Smith, A., Hawes, T., Myers, M.: Hiérarchie: Interactive Visualization for Hierarchical Topic Models. In: Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces. pp. 71–78. Association for Computational Linguistics (2014)

[43] Shahnaz, F., Berry, M.W., Pauca, V.P., Plemmons, R.J.: Document clustering using nonnegative matrix factorization. Information Processing & Management 42(2), 373–386 (Mar 2006)

[44] Ko, Y., Seo, J.: Automatic Text Categorization by Unsupervised Learning. In: Proceedings of the 18th Conference on Computational Linguistics. vol. 1, pp. 453459. Association for Computational Linguistics (2000)

[45] Ko, Y., Seo, J.: Learning with Unlabeled Data for Text Categorization Using a Bootstrapping and a Feature Projection Technique. In: Proc. 42nd Meeting of the Association for Computational Linguistics (ACL'04). pp. 255–262. Association for Computational Linguistics (2004)