ISSN: 2229-6956 (ONLINE)
DOI: 10.21917/ijsc.2022.0399

A DEEP LEARNING BASED IT SERVICE DESK TICKET CLASSIFIER USING CNN

S.P. Paramesh[1] and K.S. Shreedhara[2]

[1]Department of Computer Science and Engineering, School of Engineering, Central University of Karnataka, India
[2]Department of Studies in Computer Science and Engineering, University B.D.T College of Engineering, India

Abstract

Assignment of problem tickets to a proper resolver group is an important aspect and crucial step in any IT Service management tools like IT Service desk systems. Manual categorization of tickets may lead to dispatching of problem tickets to an inappropriate expert group, reassignment of tickets, delays the response time and interrupts the normal functioning of the business. Traditional supervised machine learning approaches can be leveraged to train an automated service desk ticket classifier by using the historical ticket data. Sparsity, non-linearity, overfitting and handcrafting of features are some of the issues concerning the traditional ticket classifiers. In this research work, a deep neural network based on Convolution Neural Network (CNN) is proposed for the automated classification of service desk tickets. CNN automatically extracts the most salient features of the ticket descriptions represented using word embeddings. The extracted features are further used by the output classification layer for efficient ticket category prediction. To corroborate the efficacy of the proposed ticket classifier model, we empirically validated it using a real IT infrastructure service desk data and compared the results with the traditional classifier models like Support Vector machines, Naive Bayes, Logistic Regression and K-nearest neighbour. The proposed CNN model with proper hyperparameters tuning outperforms the traditional classifiers in terms of overall model performance. Assignment of tickets to the correct domain groups, speedy resolution, improved productivity, increased customer satisfaction and uninterrupted business are some of the benefits of the proposed automated ticket classifier model.

Keywords:

Service desk, Machine learning, Deep neural networks, Convolution Neural Network, Word Embeddings

1. INTRODUCTION

In the modern service delivery business, it is very much important to ensure that the required services are delivered to the end users of the organization on time for the smooth functioning of the business [1]. Most of the service organizations depends on the Information Technology Service Management (ITSM) tools like IT service desk to resolve the end user's queries and to provide the organizational services. Service desk systems are the unified platforms where different stakeholders of the business can raise various kinds of issue tickets and get the solutions for their problems [2]. As a typical use case, employees of an IT organization may experience issues related to infrastructure, application, facilities, payroll, travel, medical etc. The employees use the IT service desk platform to submit the problem tickets and gets the requested support for the issue from the concerned domain expert team. Service desk systems are also called as customer helpdesk, support desk or incident management systems.

Nowadays, service desk systems are playing an important role in almost all the domain areas like banking, finance, education, healthcare, retail, telecommunications, manufacturing etc to resolve the customer queries. The users of the organization may submit the problem tickets through various service desk platforms like email, chat, web-based user interface (UI) or through telephones. The designated service desk agent then analyses the submitted tickets, categorises the tickets manually and assign the ticket to the concerned domain team on the basis of problem category for further resolution. The domain expert team then works on the assigned tickets to resolve the problem tickets within a stipulated time period as agreed in the Service Level Agreements (SLA's).

Currently, most of the organizations uses their own customized web-based IT service desk portal to support the end user queries. In such web-based system, the end users manually select the category, sub category, priority and severity of the tickets, enters a brief unstructured natural language description about the tickets and attaches the supporting files if any using the web user interface. Depending upon the selected category, the tickets are forwarded to the concerned domain team for the resolution.

Manual selection and categorization of problem tickets either by the service desk agent or by end users is time consuming process and it may result in wrong classification due to which the ticket is assigned to an inappropriate domain expert group. The lack of domain knowledge, wrong perception about the problem category, informal nature of the ticket descriptions, heavy inflow of tickets etc. are some of the reasons for wrong categorization. If a ticket is mistakenly routed to incorrect domain expert team, then it may result in resolution delay, interrupts the normal functioning of the business, unnecessary utilization of domain resources, customer satisfaction deterioration and finally it may worsen the business growth at the end of the day. Because of these limitations with the manual categorization of tickets, it is necessary to develop an intelligent model which auto categorises the support tickets [3], [11].

In the literature, traditional machine learning (ML) models are explored to automate the process of ticket classification. Some research works uses the combination of classification algorithms and natural language processing (NLP) techniques to develop an automated help desk ticket classifier model [5]-[9]. Unsupervised algorithms like clustering can be used to build the ticket classifier model when there exists unlabelled training data [10]. The traditional ticket classifiers are modelled using the previously resolved service desk tickets as the training data. Most of these approaches uses the prior ticket descriptions and the corresponding labels for modelling the ticket classifier system. The conventional ticket classifier models use a vector space model to represent each user's ticket descriptions which may result in complicated model because of the increased dimensionality and sparsity of the data [14]. Traditional classifier models do not take into consideration the semantic similarity exists among various features in the dataset [4]. Further, the handcrafted feature selection and extraction techniques are used in these traditional classifier models to extract the key features needed for the classification [12]-[13].

To address the limitations of traditional machine learning based ticket classifiers, a deep neural network model based on CNN is proposed in this research work that uses word embeddings, CNN and a fully connected classification layer combinations to model the automated IT service desk ticket classifier. The proposed ticket classifier auto categorises the service desk tickets into one of the predefined service categories by mining the unstructured natural language ticket description.

Deep neural network models do not support the use of sparse representation of the data and hence the proposed ticket classifier model uses the word embedding representation method to represent each ticket descriptions. Word embeddings represents the features in a distributed space and creates a dense, low dimensional representation by mapping each feature to a real valued vector [19]. In such representation, the vectors of the similar words are collinear to each other. Word embeddings can be pretrained (ex: Word2Vec or Glove) or learned as a part of neural network training [16]. In our research work, word embeddings are randomly initialized and are learnt while training the neural network model.

CNN based deep learning models are very powerful in the area of pattern recognition and image processing as they proven to be good at automatic extraction of most salient features needed for classification [21]. CNNs proved their effectiveness in solving many NLP tasks like Named Entity Recognition (NER), Part of Speech (PoS) tagging etc. [15]. CNNs are effectively used in sentence and document classification problems due to their ability to extract automatically the most salient and highly representative n-grams from the input sentences. [16], [17], [19].

Because of the dense representation of word embeddings and effective feature extraction function of CNN, it motivated us to use these combinations for building an effective automated ticket classifier. In this work, a single layer CNN model is trained on top of word embeddings corresponding to each input ticket descriptions. CNN extracts the most salient features from the input data and these features are then fed to the classification layer to predict the category of the input service desk tickets.

In this research work, a real IT infrastructure service desk dataset is used for experimental purposes. The dataset may contain details about ticket category, subcategory, priority, severity, ticket descriptions, attachments etc. but user's unstructured natural language ticket descriptions and the associated ticket label are mainly used for training the proposed ticket classifier model.

This research work has the following main contributions.

- Handling the data related issues and challenges found in the initial service desk ticket data.
- Representation of ticket descriptions using the word embeddings representation scheme.
- CNN model training, hyperparameters tuning and evaluation of the model for autoclassification of service desk tickets.
- Comparison of the proposed CNN model performance with traditional ML algorithms using various classifier evaluation measures.

Quick resolution time, effective resource utilization, improved productivity, uninterrupted service delivery and growth in organization business etc are some of the benefits of the proposed system.

2. PRIOR RESEARCH WORKS

Some of the prior research efforts in the development of automated service desk ticket classifiers systems are given below.

Al-Hawari et al. [3] discusses the development of IT service desk ticket classifier using the traditional ML methods like J-48, Rule based, Support Vector Machines (SVM) and Naive bayes. The performance of the models is validated against the 331 service desk test ticket and it is found that the SVM algorithm outperformed its counterparts.

A. Revina et al. [4] provides a comparative study of various ticket representation schemes during the development of a ticket classification model. Bag of Words using *tf-idf* term weighting scheme and linguistic domain features are used for ticket data representation. Rule based classifiers, standard K-NN, decision trees, SVM and logistic regression are used to build the ticket classification model. Performance results of the various chosen classifiers is measured using various evaluation measures. Results indicates that the classifier accuracy depends on the most representative features used for classification. Chosen classifiers with domain linguistic features achieved good performance when compared to the classifiers using the tf-idf features.

Paramesh et al. [5] developed an automated IT Help desk incident classifier based on ensemble of classification techniques like bagging and boosting. The performance of ensemble of classifiers is analysed against the base classifiers like SVM and Naive bayes using various performance evaluation metrics. Results indicates that ensemble classifiers outperformed the corresponding base classifiers.

Mandal A. et al [8] proposes an email ticket dispatcher system using a combination of ticket classifier ensemble and a rule engine. The email ticket classifier models are generated using the historical email data containing the email subject, descriptions and corresponding resolver group. The rules engine is used to capture the domain specific features from the training data.

Pandey, N. etc al. [9] applied classification algorithms like KNN, linear discriminant analysis, Random Forest, SVM etc. to categorize the issue reports pulled from different open-source issue tracking systems. Random forests ensemble classifier performed well and SVM also achieved good accuracy with certain kernels.

Roy et al. [10] developed a framework to organize the incident data into number of clusters followed by labelling of clusters. The clusters are generated based on the similar ticket descriptions and are further used to classify the incoming incident. The proposed model uses k-means clustering technique based on a novel distance measure that uses the combination of cosine and Jaccard distances to identify the similar ticket descriptions. Logical item sets derived from each generated cluster are further used to label each cluster.

Automated ticket classification is an extension of text document classification problem and hence literature review in the field of text mining and natural language processing is also done and some of these works are detailed below.

M. Ikonomakis et al. [12] illustrates the end-to-end text classification process based on the machine learning techniques. The research paper discusses in detail the document representation using the Vector space model, various feature selection and extraction techniques, different supervised ML models to build the automated document classifier followed by explanation of various classifier evaluation metrics.

A good review on the various ML techniques involved in the text classification process are detailed in Mironczuk et al. [13] and Kowsari et al. [14]. Different data pre-processing methods, dimensionality reduction techniques, existing ML algorithms and techniques and various classifier evaluation metrics are discussed in detail. The review also covers the various limitations and applications of each ML technique in the real-world problems.

Collobert and Weston [15] proposed a unified architecture using a single layer Convolution Neural Network for modelling the sentence. Given an input sentence, this model implements the multiple NLP problems like Chunking, NER, PoS tagging, semantic roles labelling (SRL) etc.

T. Young et al. [19] discusses the popular deep neural architectures like CNN, Recurrent Neural Network (RNN) and Recursive neural networks and their applications in various NLP problems. The work also describes how to represent the input sentences using the dense, distributed word embedding representation to overcome the problem of curse of dimensionality that may occurs in sparse representation of data.

3. PROPOSED METHODOLOGY

At a first glance, process of service desk ticket classification looks like a typical text document classification problem with the ticket descriptions and its associated ticket category representing the text document and document label respectively. The research work becomes quite challenging during the actual design and implementation phase depending upon the complexity of the training data. The end-to-end solution diagram for training the proposed service desk ticket classifier using a CNN based deep learning model is depicted in Fig.1.

The key components involved in the ticket classification using CNN are explained as follows.

3.1 TRAINING DATA

In this work, real data from a reputed IT infrastructure service desk system is used for building the ticket classifier. The chosen dataset had many service categories belonging to various issues like Hardware, LAN, Operating System, printer, Outlook etc. and hence it is a typical multiclass classification problem. Already resolved tickets containing the unstructured ticket descriptions and the corresponding class labels were used to build the classifier model.

3.2 TICKET DATA PRE-PROCESSING

Data pre-processing is one of the most crucial phases in any typical data mining process to build an efficient ML model. Generally, huge amount of unwanted and other kinds of messy data exists in the initial raw ticket descriptions. The presence of noisy data increases the feature set dimension and degrades the performance of the ticket classifier model. Hence, it is required to identify, extract and eliminate unwanted features from the dataset and to retain only relevant features that aids in classification. The historical helpdesk ticket dataset used in this research work had lot of undesired features or attributes like stop words, numbers, special characters, email ids, phone numbers, functional words, names, places etc. To remove such undesired data, each ticket descriptions are converted into lower case and further tokenized into words, numbers, punctuations, email ids etc using the tokenization process.

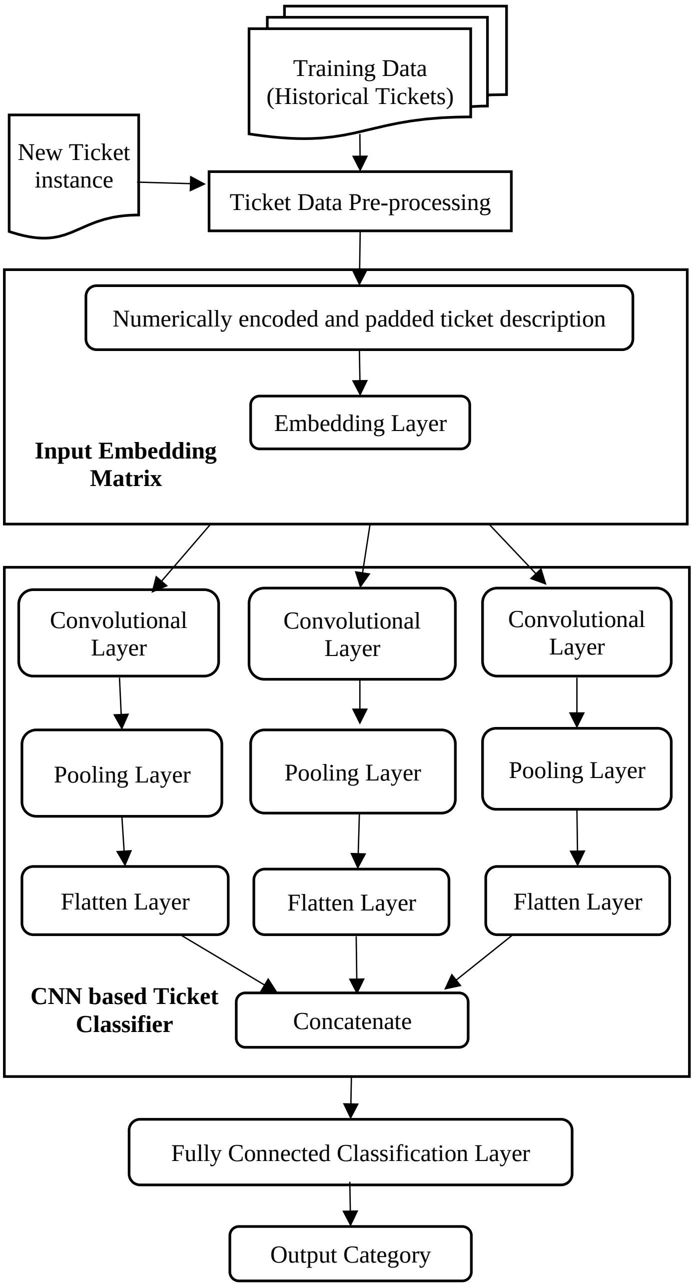

Fig.1. Proposed solution diagram of CNN based Service desk ticket classification

The following techniques are then applied to remove various kinds of noise in the data.

- Removal of Stop words i.e., commonly used English words are eliminated from the feature set as they do not contribute for classification.
- Entities such as email ids, date, time etc present in the ticket descriptions are removed by developing specific pattern recognizers.
- Special characters and numbers are removed as they do not aid in classification.
- Some of the functional words like verbs, nouns, pronouns, conjunctions, prepositions and quantifiers present in the ticket descriptions are removed by tagging each word with a Parts of Speech (PoS) tagging.
- Named Entity Recognizers (NER) are used to remove the entities like names, places etc.
- Finally, stemming operation is performed, which reduces each feature to its base form.

3.3 NUMERICAL ENCODING AND PADDING OF TICKET DESCRIPTIONS

CNN model accepts only fixed length numerical input data. Each pre-processed ticket descriptions are tokenized into words followed by integer encoding of the words based on the word index in the entire corpus of training data. The encoded ticket descriptions are then padded using padding mechanism to ensure every input is equal in size. In this research work, each input ticket description is converted into fixed length of size n =250 words and this fixed length 'n' is determined by finding the ticket instance in the training data containing the maximum number of unique features.

3.4 CNN BASED DEEP LEARNING MODEL FOR SERVICE DESK TICKET CLASSIFICATION

The proposed CNN model for the service desk ticket classification comprises of an input embedding layer, single dimension convolution layer, pooling and a fully connected classification layer. Each of these layers are described below.

3.4.1 Embedding Layer:

Numerically encoded and padded input sentence (i.e., a ticket description in our case) is fed as an input to the embedding layer. Let a ticket instance 'S' consisting of 'n' words be the input to the model. Let $x_i \in \mathbb{R}^d$ be the d-dimentional real valued feature vector of the i^{th} token of the sentence and is represented using the following Eq.(1)

$$x_i=[r_1,r_2,\ldots.r_d] \quad (1)$$

In this research work, a word embedding vector dimension of d=100 is used and these vectors are initialized with random weights and are learned during the neural network model training. The sentence with 'n' words is represented using an embedding matrix or input sentence representation matrix $S\in\mathbb{R}^{(n\times d)}$. The embedding matrix S is formed by concatenating the d-dimensional real valued vector of x_i for every i^{th} token in the sentence S and is represented using the below Eq.(2):

$$S = x_{1:n}=[x_1\oplus x_2\oplus x_3\ldots\oplus x_n] \quad (2)$$

Here, $\oplus$ is the concatenation symbol. So, the dimension of our input embedding matrix is $n\times d$=250×100.

3.4.2 Convolution Layer:

Convolution layer is used as feature extraction layer which extracts the most salient features from the input ticket descriptions represented using the word embeddings. Input embedding matrix of size $n \times d$ corresponding to each ticket description S is given as an input to the convolution layer. Convolution operation is performed on the embedding matrix using a linear filter of some fixed height 'h'. A convolution involves a filter parameterized by a weight matrix 'm' belongs to $\mathbb{R}^{h\times d}$ and is applied over 'h' input words to generate a new feature. An element c_i created using a window of features $w_{i:i+h-1}$ is given using the below Eq.(3)

Convolution layer is used as feature extraction layer which extracts the most salient features from the input ticket descriptions represented using the word embeddings. Input embedding matrix of size $n\times d$ corresponding to each ticket description S is given as an input to the convolution layer. Convolution operation is performed on the embedding matrix using a linear filter of some fixed height 'h'. A convolution involves a filter parameterized by a weight matrix 'm' belongs to $R^{h\times d}$ and is applied over 'h' input words to generate a new feature. An element c_i created using a window of features $w_{i:i+h-1}$ is given using the below Eq.(3)

$$c_i = f(w_{i:i+h-1}\cdot m+b) \quad (3)$$

Here '·' is element wise product operator, $b\in\mathbb{R}$ is the bias value and f denotes a non-linear activation function like Rectified Linear Unit (ReLU). The filter vector 'm' convolves over all possible input features in the sentence to produce a new single dimension feature map as given in Eq.(4)

$$c=[c_1,c_2,\ldots,c_{n-h+1}] \quad (4)$$

with $c\in\mathbb{R}^{n-h+1}$.

Kernel size represents the number of n-grams to consider during convolution i.e., filter size. One may use number of filters of different kernel sizes where each filter extracts a particular pattern of n-gram.

In this research work, a multi-channel CNN model with different kernel sizes h = 3, 4, and 5 is used which convolves respectively over 3,4 and 5 words of the input at once to generate distinct feature maps having size $(n-h+1)\times 1$. The number of convolution filters for each kernel size is set to 32. The best kernel size and the optimal number of features maps to use for each kernel size is determined after fine tuning the hyper parameters of CNN model during the training process.

3.4.3 Pooling layer:

The feature maps generated by the convolution layer are fed to the pooling layer to extract the most relevant features from each map. A one-dimension max (1-max) pooling function is applied on every feature map which extracts the feature with the highest value $\hat{c} = \max\{c\}$ from each map. Since '32' different feature maps are used for each kernel sizes h = 3, 4 and 5, a total of '96' most important features are extracted during pooling operation. The advantages of max pooling are it extracts most relevant features across the whole ticket instance, decreases the computational overhead of the network and controls the overfitting.

3.4.4 Fully Connected Output Classification Layer:

The salient features generated using the 1-max pooling are flattened and are concatenated into a single dimensional feature vector 'v' which is fed directly into output classification layer for ticket category prediction. The classification layer gives the output in the form of a vector, the size of which is equal to the number of target classes in the training dataset. In this research work, because of the multiple classes in the dataset, a softmax activation function is used on each unit of the output layer. At this layer, one may apply 'dropout' as a means of regularization [20]. The softmax function gives the probability distribution of each input ticket instance belonging to different classes and is given using the below Eq.(5):

$$p_i = \frac{\exp(y(i))}{\sum_{k=1}^{n}\exp(y(k))} \quad (5)$$

The error between the actual and predicted output is found using a suitable loss function. Since, the proposed research work is based on multiple classes, categorical_crossentropy is used as the loss function and is formally given using the below Eq.(6):

$$L(\hat{y}, y) = -\sum_{i=1}^{N}\sum_{j=1}^{C} y_i^j \log(\hat{y}_i^j) \quad (6)$$

Here, y_i^j and $\hat{y}_i^j$ represents the actual and predicted output, N is the sample size and C is the number of classes.

The intent of the training is to optimize the loss function by updating the network parameters like filter weights, bias and weight vectors of the activation function using a suitable optimizer. A stochastic gradient descent optimizer and a backpropagation learning is used for updating the network parameters [22].

3.5 PERFORMANCE EVALUATION OF THE CNN MODEL

The performance of the proposed CNN based ticket classifier is evaluated using various classifier evaluation measures like accuracy, precision, recall and f-score and these metrics are described as follows.

Consider a service desk ticket classification with binary class problem having only positive and negative ticket categories. Let TP is the positive ticket instances that were classified correctly, FP is the positive ticket instances that were incorrectly classified, TN is the correctly classified negative ticket instances and FN is the incorrectly predicted negative ticket instances.

Accuracy is the number of correct predictions out of the total predictions made by the classifier and is mathematically given using Eq.(7).

$$\text{Accuracy}=(TP+TN)/(TP+TN+FN+FP) \quad (7)$$

Precision evaluates the number of correctly classified positive ticket instances among all the positive ticket instances predicted by the ticket classifier and is given by Eq.(8)

$$\text{Precision}=TP/(TP+FP) \quad (8)$$

Recall is the fraction of relevant positive ticket instances that are correctly predicted by the ticket classifier and is calculated using Eq.(9)

$$\text{Recall}=TP/(TP+FN) \quad (9)$$

Since our research work uses a dataset containing multiple ticket categories, precision and recall metrics are computed for each ticket label and then these values are averaged to get the final result. The precision and recall scores can be combined into another metric called f-score which expresses both properties and is calculated using the below Eq.(10)

$$f\text{-}score=((2\times precision\times recall))/((precision+recall)) \quad (10)$$

In order to check the efficacy of proposed CNN based service desk ticket classifier, the performance of CNN is compared with the traditional ML algorithms like SVM, Naive Bayes, Logistic regression and KNN. After the comparative study of all the chosen classifiers, the ticket classifier model which outperforms all other chosen classifiers in terms of accuracy and other evaluation metrics is selected as the best trained ticket classifier. The best trained model can be further used for making prediction on the new or unseen service desk tickets. Whenever a new unlabelled ticket arrives, the trained model auto categorizes it into one of the predefined service categories and based on the predicted category, the ticket will be assigned to the concerned domain expert group for the resolution of service desk tickets.

4. RESULTS AND DISCUSSIONS

The current research work has been done using a computer system equipped with Intel core i5 processor, 8 GB SD RAM, 512GB SSD, NVIDIA GeForce MX450 Graphics card and Windows 10 with 64-bit OS. Python's Kera's 2.3.1 with TensorFlow as backend, Scikit-learn, Pandas and Matplotlib libraries are used for implementation. The experimental results at different stages of the proposed ticket classification process are discussed as below.

4.1 DATA COLLECTION AND PRE-PROCESSING

A real time IT infrastructure service desk ticket data containing multiple number of classes is used for our experimental purposes. Issues related to hardware, operating system, network, printer, software etc. are some of the ticket categories associated with the chosen infrastructure dataset. The exploratory data analysis of the dataset chosen revealed the following statistics as given in the Table.1.

Table.1. Description of the dataset

Number of tickets in the chosen service desk dataset	10742
Number of target classes in the service desk dataset	18

It is observed that the historical service desk dataset used for this research work had huge amount of undesired and other kinds of messy data. Ticket data pre-processing is done to eliminate all the undesired entities like stop words, special characters, numbers, email id's, undesired functional words, date and time etc. from the dataset. The details about the word/feature count prior to and post the data pre-processing phase is given in Table.2.

Table.2. Word count details at various phases

Number of distinct words in the dataset before pre-processing of ticket descriptions	6302
Number of unique words in the dataset after eliminating the stop words from the dataset	6228

Vocabulary size after eliminating all other unwanted data like numbers, special characters, email ids, names, functional words etc.	3572

4.2 NUMERICAL ENCODING AND PADDING OF TICKET INSTANCES

Each pre-processed ticket descriptions of the training dataset are encoded as integers and are further subjected to padding to ensure each ticket instances are of fixed length i.e., n=250 in our case. The sample output representation screenshot after encoding and pre-padding of each service desk ticket instances is shown in Fig.2 and Fig.3 respectively.

```
[[12, 20, 592, 13, 263, 129, 18, 1, 13, 222, 20, 592, 13, 551, 181], [1, 1579, 366, 1580, 141, 3, 1581, 60, 216, 9, 4, 650], [3
4, 1], [6, 227], [707, 482, 5, 74, 4, 19, 188, 483, 707], [132, 19], [1, 651, 3, 13], [14, 17, 91, 188], [43, 1, 926, 3, 72, 32
1, 9, 107], [50, 129], [63, 20, 1], [6, 142, 5], [18, 58, 23, 593, 594, 56], [182, 312, 13, 5], [4, 101, 1582, 56, 286], [4, 10
1, 1583, 1584, 56, 286], [101, 294, 1585], [431, 1, 75, 1586, 161, 75, 42, 927, 708, 246], [344, 118, 104, 5], [12, 10, 117, 5
1, 6, 216, 1587, 811, 301, 7, 928, 7, 253, 709, 6, 4, 271, 259, 710, 89, 1588, 1589], [118, 50, 5, 103, 313, 240, 595, 115, 159
0, 457, 1591], [264, 136, 5, 204], [2, 314, 205, 6, 13], [7, 50, 20], [8, 59, 138, 136], [150, 140], [67, 36, 1, 3, 1592, 216,
186, 1143, 117, 1, 1593, 3, 151, 711, 72, 1594, 396, 110, 75, 4, 16, 197, 187], [12, 10, 200, 65, 24, 72, 5, 4, 44, 27, 201, 3
6, 121, 484, 1595, 321, 65, 24, 23, 24, 265, 1596, 65, 24, 31], [12, 10, 315, 9, 21, 206, 512, 183, 260, 37, 9, 712, 37], [812,
246, 412], [2, 232, 513, 26], [86, 162, 47, 31, 100], [33, 91, 21, 1144], [9, 5, 713, 367, 9, 1145, 397, 16, 105, 240, 106, 55
2, 115, 1597, 1598, 714], [25, 254, 137], [172, 5, 255, 596, 202, 167, 295, 50, 312, 17], [929, 5], [11, 71, 5, 9], [32, 5, 68,
```

Fig.2. Numerically encoded ticket instances of Dataset

```
[[   0    0    0 ...   13  630  183]
 [   0    0    0 ...    9    1  764]
 [   0    0    0 ...    0   30   37]
 ...
 [   0    0    0 ...    0   62   37]
 [   0    0    0 ...   89   67   10]
 [   0    0    0 ... 1356   63 1959]]
```

Fig.3. Encoded and Padded ticket instances of Dataset

Each encoded and padded ticket instance is then converted into an embedding matrix wherein each feature is mapped to real valued vector of dimension d=100 so that the size of the input sentence matrix is $n \times d$= 250x100. Python's Kera's embedding function is used in this research work to dynamically create the word embedding vector corresponding to each word of the input sentence. The input embedding matrix corresponding each input ticket is then fed to the CNN based classifier model as an input.

4.3 CNN MODEL BUILDING AND EVALUATION

The proposed CNN based ticket classifier model is trained using 80% of the entire dataset and rest of the 20% of data is used for evaluating the model performance. During the modelling of the CNN based ticket classifier, the choice of hyper parameters of the CNN such as filter size, number of kernels, epochs etc. may influence the performance of the ticket classifier [18]. Several experiments were conducted to choose the best values of the filter sizes, the number of feature maps, epochs, batch size and other hyper parameters. Different classifier models are generated using the training data and with different filter sizes while keeping the number of feature maps equal to 32. The average accuracy performance of each generated model is recorded using the K fold cross validation method with K=5. The effect of combining different kernel sizes on the CNN model performance is also explored. The average performances of the training models obtained using different filter sizes is shown in Table.3.

The performance of the training model is also evaluated using different number feature maps for ex. 4, 16, 32, 64 and 128 and it is found that feature map with size=32 giving good results and hence the same is chosen for training the model.

Table.3. Performance evaluation of CNN model using different sized filters

Filter Sizes	Mean Accuracy (%)
1	84.6
2	88.4
3	85.8
4	88.0
1,2	87.9
2,3	88.1
3,4	88.5
1,2,3	88.2
2,3,4	88.4
3,4,5	**91.5**
4,5,6	88.8

From the Table.3, it indicates that the best filter size combination is (3,4,5) with an average training accuracy of 91.5%. The hyperparameters of the best CNN model used in this research work are given in Table.4.

Table.4. Optimal hyperparameters of CNN based ticket classifier model

Hyperparameter	Choice
Word Embedding size for each feature	100
Filter/Kernel size	(3,4,5)
Number of filters	32
Activation function used in CNN layer	ReLU
Pooling stride	1
Pooling strategy	1-Maxpooling
learning rate	0.001
loss function	categorical_crossentropy
Optimization algorithm	Stochastic Gradient Descent
Epochs	30
Batch size	64
Activation function used in the output layer	Softmax

The proposed CNN based ticket classifier is trained using the optimal hyperparameters as shown in Table.4. The loss function and accuracy of the CNN model between the training and validation set (20% of the training data) is recorded. The following Table.5. shows the performance evaluation results of the proposed ticket classifier during the training phase.

Table.5. Performance results of CNN based ticket classifier during training phase

Model	Training		Validation	
	Accuracy (%)	loss	Accuracy (%)	loss
CNN	92.0	0.15	91.5	0.38

From the Table.5, it indicates that CNN based ticket classifier model performed well with a training and validation accuracy of 92% and 91.5% and a quite low training and validation loss of 0.15 and 0.38. However, it looks like the proposed model exhibits a little bit of overfitting. A dropout layer can be added to the network to reduce this little overfitting problem [20].

The proposed CNN model performance is also validated against the 20% test data using various classification evaluation measures. The CNN model achieved 91.2% accuracy with a loss of 0.2. The proposed model effectiveness is also compared with traditional machine learning classifiers like Support Vector Machines (SVM), Multinomial Naive Bayes (MNB), Logistic Regression (LR) and K-Nearest Neighbour (K-NN) using the test set and the obtained results are shown in Table.6.

Table.6. Performance comparison of the traditional ML models and the proposed CNN model

Model	Accuracy (%)	Precision (%)	Recall (%)	F-score (%)
LR	80.87	81.64	80.87	80.53
KNN	68.35	73.75	68.35	68.53
MNB	68.40	73.53	68.40	65.40
SVM	87.43	87.46	87.43	87.36
CNN	**91.20**	**91.18**	**90.78**	**91.02**

From the Table.6, it indicates that the proposed CNN based ticket classifier with 91.20% accuracy over the test set outperforms the chosen traditional machine learning models in classifying the service desk tickets.

5. CONCLUSIONS

To overcome the limitations of manual classification of tickets in the existing IT service desk systems, an automated service desk ticket classifier based on deep neural network using CNN is proposed in this research work. The proposed model uses a combination of word embeddings, Convolution Neural Network and a fully connected output classification layer for the autocategorization of service desk tickets. NLP techniques like tokenization, stop words removal, PoS tagging, stemming etc. are leveraged to pre-process the unstructured ticket descriptions of the initial raw ticket data. The pre-processed ticket descriptions are then represented using the dense word embedding representation. The proposed model uses this word embedding representation of the input ticket data followed by automatic extraction of key features by the CNN layer. The salient features extracted from the CNN model are fed directly into the output classification layer for prediction of output ticket category.

The proposed CNN based ticket classifier is validated using a real-world IT infrastructure service desk dataset and it achieved a training and validation accuracy of 92% and 91.5% respectively during the training phase. The CNN model performance is also compared with the traditional ticket classifier models using the test data and the results shows that CNN based ticket classifier achieved a good accuracy of 91.2% in comparison with SVM (87.43%), LR (80.87%), MNB (68.4%) and KNN (68.35%).

The proposed approach greatly reduces the manual efforts and human errors by automating the task of service desk agent. Effective resource utilization, quicker response time, improved end user experience, growth in organization business etc. are some of the other benefits of the proposed automated ticket classifier model.

REFERENCES

[1] P. Kubiak and S. Rass, “An Overview of Data-Driven Techniques for IT-Service-Management”, *IEEE Access*, Vol.6, pp. 63664-63688, 2018.

[2] M. Jantti, A. Cater-Steel and A. Shrestha, “Towards an Improved IT Service Desk System and Processes: A Case Study”, *International Journal on Advances in Systems and Measurements*, Vol. 5, No. 3-4, pp. 203-215, 2012.

[3] F. Al-Hawari and H. Barham, “A Machine Learning Based Help Desk System for IT Service Management”, *Journal of King Saud University-Computer and Information Sciences*, Vol. 33, No. 6, pp. 702-718, 2021.

[4] A. Revina, K. Buza and V.G. Meister, “IT Ticket Classification: The Simpler, the Better”, *IEEE Access*, Vol. 8, pp. 193380-193395, 2020.

[5] S.P. Paramesh and K.S. Shreedhara, “IT Helpdesk Incident Classification Using Classifier Ensembles”, *ICTACT Journal of Soft Computing*, Vol. 9, No. 4, pp. 1980-1987, 2019.

[6] N.A. Harun, S.H. Huspi and N.A. Iahad, “Question Classification Framework for Helpdesk Ticketing Support System using Machine Learning”, *Proceedings of International Conference on Research and Innovation in Information Systems*, pp. 1-7, 2021.

[7] S.P. Paramesh and K.S. Shreedhara, “Automated IT Service Desk Systems using Machine Learning Techniques”, *Proceedings of International Conference on Data Analytics and Learning*, pp. 331-346, 2018.

[8] A. Mandal, N. Malhotra, S. Agarwal and G. Sridhara, “Cognitive System to Achieve Human-Level Accuracy in Automated Assignment of Helpdesk Email Tickets”, *Proceedings of International Conference on Service-Oriented Computing*, pp. 332-341, 2018.

[9] N. Pandey and Amitava Sen, “Automated Classification of Software Issue Reports using Machine Learning Techniques: an Empirical Study”, *Innovations in Systems and Software Engineering*, Vol. 13, No. 4, pp. 279-297, 2017.

[10] S. Roy, D.P. Muni, J.Y.T. Yan, N. Budhiraja and F. Ceiler, “Clustering and Labeling IT Maintenance Tickets”, *Proceedings of International Conference on Service-Oriented Computing*, pp. 829-845, 2016.

[11] Cristina Kadar, Dorothea Wiesmann, Jose Iria, Dirk Husemann and Mario Lucic, “Automatic Classification of Change Requests for Improved it Service Quality”, *Proceedings of Annual SRII Global Conference*, pp. 430-439, 2011.

[12] M. Ikonomakis, S. Kotsiantis and V. Tampakas, “Text Classification using Machine Learning Techniques”, *WSEAS Transactions on Computers*, Vol. 4, No. 8, pp. 966-974, 2005.

[13] M.M. Mironczuk and J. Protasiewicz, “A Recent Overview of the State-of-the-Art Elements of Text Classification”, *Expert Systems with Applications*, Vol. 106, pp. 36-54, 2018.

[14] K. Kowsari, K.J. Meimandi, M. Heidarysafa, S. Mendu, L.E. Barnes and D.E. Brown, “Text Classification Algorithms: A Survey”, *Proceedings of International Conference on Computation and Language*, pp. 1-7, 2019.

[15] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu and P. Kuksa, “Natural Language Processing (Almost) from Scratch”, *Journal of Machine Learning Research*, Vol. 12, No. 3, pp. 2493-2537, 2011.

[16] Yoon Kim, “Convolutional Neural Networks for Sentence Classification”, *Proceedings of International Conference on Empirical Methods in Natural Language Processing*, pp. 1746-1751, 2014.

[17] N. Kalchbrenner, E. Grefenstette and P. Blunsom, “A Convolutional Neural Network for Modelling Sentences”, *Proceedings of Annual Meeting of the Association for Computational Linguistics*, 2014.

[18] Y. Zhang and B. Wallace, “A Sensitivity Analysis of (and Practitioners’ Guide to) Convolutional Neural Networks for Sentence Classification”, *Proceedings of International Joint Conference on Natural Language Processing*, pp. 253-263, 2017.

[19] T. Young, D. Hazarika, S. Poria and E. Cambria, “Recent Trends in Deep Learning Based Natural Language Processing”, *IEEE Computational Intelligence Magazine*, Vol. 13, No. 3, pp. 55-75, 2018.

[20] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”, *The Journal of Machine Learning Research*, Vol.15, No. 1, pp. 1929-1958, 2014.

[21] A. Krizhevsky, I. Sutskever, and G.E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks”, *Proceedings of International Conference on Neural Information Processing Systems*, pp.1097-1105, 2012.

[22] David E. Rumelhart and James L. McClelland, “*Learning Internal Representations by Error Propagation*”, MIT Press, 1986.