Deep Learning based Intelligent E-mail Autoresponder

B.Lakshmi Priya1, S.Jayalakshmy2,D.Saraswathi3, V.Kumar4, E.Dinesh5
1,3Associate Professor, Department of Electronics and Communication Engineering, ManakulaVinayagar Institute of Technology, Puducherry.
2Associate Professor, Department of Electronics and Communication Engineering, IFET College of Engineering, Villupuram.
4,5UG Scholar,Department of Electronics and Communication Engineering ManakulaVinayagar Institute of Technology, Puducherry

*1lakshmipriyaece@mvit.edu.in, 2jayalakshmyece@ifet.ac.in, 3saraswathiece@mvit.edu.in, 4kumar30799@gmail.com, 5dineshkalai263.pdy@gmail.com

Abstract. Handling huge volume of emails is a very challenging task in the customer support applications and an automated email responding system will be of great help. In this paper, an intelligent email autoresponder system is developed which either attempts to respond to the incoming emails from various category of customers or generate token for service request to address the issue manually by an expert member. First, based on the content the system has to predict whether the mail belong to the category of auto responding or to invoke a service request. This classification of email is carried out using long short term (LSTM) and bi-directional LSTM networks (Bi-LSTM) networks and the classification performance is analyzed. The results presented in this work show that the Bi-LSTM classifier outperforms LSTM network.

1. Introduction
Volume of text data is increasing exponentially day by day in all organizations. Channels such as email, documents, social media, and others contribute increasing amounts of text data. This data carries valuable information that when extracted and acted upon helps to provide better products and services to the customers. Dealing with this ever-increasing data is often time-consuming and error prone and sometimes leads to missed business opportunities and costs. Text classification, one of the fundamental natural language processing (NLP) problems can be used in these scenarios to automatically interpret the nature and importance of the data. Automated text classification has been exploited as a tool to tackle the ever increasing data which in general manages and processes a vast amount of documents in digital forms. Text classification plays an important role in information extraction and summarization, text retrieval, and question-answering. Managing the unprecedented rise in the customer queries for a commercial organization is a challenging task. This requires one or more dedicated employees to read the email, forward it to the concerned department and to send a reply mail to the customer followed by the follow-up action for every received email. Furthermore, there is always a chance of making error in managing these actions and chance of missing any important mail. To overcome these issues, an automated task of interpreting the received email and the action to be taken can be implemented. With the help of NLP algorithms, understanding the content of email can be done. Following this, machine learning algorithms can be deployed for the purpose of forwarding the mail to respective departments; auto response mail and follow up action for every received mail can be done. With this motivation, this project aims to develop an Intelligent E-mail Auto responder model which uses recurrent neural networks (RNN) to automatically classify the incoming email and route it to the appropriate team for further process.

2. Related Works
2.1 Literature Survey
Text classification is an integral part of NLP. The basic form of NLP comes with categorizing the text based on the contents. Text classification is used in various tasks like segregating the spam mails from the personal ones, categorizing the news articles, understanding the customer feedbacks, social media monitoring etc. Initially machine learning algorithms are used for text classification which extracts the features from the contents of the text and then categorizes the same as belonging to one of the several associated classes. A hybrid approach for text classification is proposed in [1] using Naïve Bayes and support vector machine (SVM) algorithms for text document classification. Naïve Bayes algorithm basically does the job of vectorising the keywords extracted from the text document and
classification is performed using SVM. Precision/Recall-Driven Decision-Tree (PRDT) Algorithm is proposed in [2] for the classification of XML documents and demonstrated a better performance at that time. Alecsa: attentive learning for email categorization using structural aspects was proposed by Mostafa et al in [3] to explore the dynamics associated with the structural data in email with low time complexity. Boosting based universum algorithm with semi supervised learning using Adaboost classifier has been implemented for text classification in [4]. The algorithm was tested on four datasets namely 20 Newsgroups, CiteULike, WebKB3 and Reuters-21578 and demonstrated that the universum examples significantly contribute to the improved performance of text classification. The study on text classification by considering sentence vector space model and unigram representation has reported improvement in classification accuracy [5]. Two different neural networks have been exploited for extracting sentence vector and unigram representation and are fused based score levels. A supervised Hebb rule based feature selection (HRFS) proposed in [6] demonstrated that the HRFS strategy outperforms the then existing feature selection methods. It is obvious that the accuracy of text classification is entirely dependent on the types of features extracted and feature engineering plays a vital role. Recurrent neural networks were then deployed for the case of text classification to classify the text without feature engineering. Deep learning through long short term memory (LSTM) network was performed for gender prediction from word representation [7]. The authors demonstrated that LSTM records better performance with word embeddings namely word to vector representation. In [8], LSTM based spam-ham short messaging service (SMS) classification was performed Spam SMS Classification dataset available at UCI machine learning repository. With the help of tokenization based pre-processing, an accuracy of 98% was achieved using LSTM classifier. Spam detection using LSTM was proposed in [9] by converting the text in semantic vectors using word2vec, WordNet and ConceptNet and was implemented on Spam SMS Classification and Twitter dataset. Emotion recognition from the text data using bidirectional LSTM (Bi-LSTM) combined with transfer learning architecture was proposed in [10]. The improvement in the accuracy was acclaimed by a customized strategy of transfer learning over sentiment analysis.

2.1 Deep Learning for Natural Language Processing
Deep learning algorithms which play a vital role in predictive analytics in the fields of computer vision, signal processing etc are also contributing significantly in natural language processing. With the help of these algorithms, the trivial tasks of describing the contents of images and translating the text from one language to another are handled very efficiently. Recurrent neural networks (RNNs) are a class of deep neural network considers a series of input namely words or samples with no limit of predetermined length. These networks are with memory which remembers the past decision for a while and manipulates them for future decisions. RNNs, by the virtue of their memory are capable of processing variable length input sequences and are hence popular in text classification, handwriting recognition, time series prediction and so on. RNNs basically considers a single time step of the input and calculates the current state from the present input and previous decisions. This current state becomes the previous state for next time step and this process continues. LSTM network is a type of RNN used in sequence prediction tasks. A typical LSTM unit consists of a cell, an input gate, an output gate and a forget gate. The architecture of LSTM is shown in Fig. 1. Similar to LSTM, in Bi-LSTM there exists a bidirectional long term dependency between the input time series data or sequence. It is constructed by having two LSTMs together. This structure allows the propagation of information in both forward and backward directions. The Bi-LSTM cell works in two ways: from past to future and from future to past. Though Bi-LSTM is complex compared to LSTM architecture, it has recorded better performance for complicated input sequence. The architectures of LSTM and Bi-LSTM networks are presented in Figures 1 and 2.

![Fig. 1 LSTM Architecture](image_url)
With the motivation gained by the virtue of LSTM and Bi-LSTM networks’ greater performance without feature engineering, automatic email responding system is proposed in this work. The major objective of this work is to create an intelligent e-mail auto responder and to automatically segregate and forward the customer e-mails of a company to a concerned section by interpreting the e-mails through recurrent neural networks (RNN). The work aims at design and implementation of an intelligent system for extracting the semantic features hidden in the text to make the efficient topic classification.

3. System Model
3.1 Problem Description
The motivation for the work considered here is taken from TCSInframinds 2018 problem statement formulated by SkyCraper Solution, an audit firm. E-mail is the main mode of communication in information technology firms and 85% of the total email volume contacts the service desk. Among the mails intended for service desk, 70% of the issues reported to the service desk have simple task and can be automated so that the issue can be sorted out without any delay. The intelligent email auto responder has to perform the following tasks: i) read the e-mail sent by the user ii) create an appropriate ticket: if the issue is resolved automatically incident ticket should be issued otherwise a service request has to be initiated to resolve it manually with the help of an expert person.

3.2 System Architecture
To address the above issue, the following technologies are used: i) client / server technology ii) e-mail technology and iii) deep learning. Client / server technology does the task of separating the functions of an application into two or more distinct parts. It describes the relationship between two computer programs in which one program, the client, makes a service request from another program, the server, which fulfills the request. The client presents and manipulates data on the desktop computer. The server acts like a mainframe to store and retrieve protected data. Email is the main mode of communication. Here the tickets are assigned only through emails. In the support process, incoming new tickets / service request are analyzed and assessed by organization's support teams in order to fulfill the ticket request. Deep learning model leverages the data present in the database to classify the incoming new ticket or service request to appropriate queues. The novelty of this work lies in predicting the best category to which the ticket corresponding to the incoming e-mail can be fed. The overall architecture of the proposed system model is presented in Figure 3.

The forwarding tasks in the scarcely covered areas are increased by forcing the higher energy cluster nodes and next hop as few member nodes to achieve load balance among cluster heads. Each node has to send and receive data transmission process. The network node shifts between transmitting and receiving states during the process of transmission of data. At the time of data transmission, the sensor nodes consume energy.

Two different networks namely LSTM and Bi-LSTM are used for predicting the incoming emails as they belong either to incident or service request category. The accuracy of prediction using the above two RNNs are compared and are presented in section IV. When tickets are assigned to a group for the first time or reassigned from another group - for example, transactional tickets involving questions on recent orders, FAQs etc. - the Round-robin system will
assign them in a circular fashion to the agents who are online in that group. This guarantees that incoming tickets are equitably distributed among online agents, which not only saves you time but let your agents get to work immediately.

![Fig.3 System Architecture](image)

When tickets are assigned to a group for the first time, they are assigned to the least loaded agent first and are then distributed to other agents in the group until they reach the threshold you’ve set. Thereafter, tickets are assigned to an agent as and when they resolve one of the assigned tickets. Not all of your agents will log into the helpdesk at the same time. The prevailing scenario is, initially only one of your agents will be logged in and all the incoming tickets will get assigned to this one agent. This leads to the agent working on the previously assigned tickets and simultaneously managing the rush of incoming tickets.

Not all tickets that enter the helpdesk are the same - each ticket might require a different expertise, and so you’d want the agent most experienced in handling such issues working on them. Skill based ticket assignment allows you to easily match tickets to the agent most proficient in dealing with them in your group, while simultaneously maintaining the load that the agent has to work with. You can associate distinct skills to each agent on your team and create rules to make sure that tickets matching those skills are always assigned to them.

4. Results and Discussions
Results obtained in this work are presented in this section. As the problem statement taken for this work is unique and no standard dataset is available, the dataset is created manually for this task. The classification task performed in task is basically a binary classification. The labels associated with the classification task are incident ticket and service request. The dataset required for the customer support task involved in this work is created with 2 to 6 words for every data as csv format and are presented in Figure. 4.
The hyper parameter settings for the classifier and the performance of the LSTM and Bi-LSTM classifier are tabulated in Tables 1 and 2. For fair comparison, the hyper-parameter settings are kept the same for both the classifiers. It is evident from Table 2 that both architectures perform well in classifying the incoming emails. However, Bi-LSTM shows a better performance compared to LSTM.

Table 1. Hyper-parameter Settings

| Parameter       | Value |
|-----------------|-------|
| No. of epochs   | 40    |
| Mini batch size | 27    |
| Optimizer       | Adam  |
| Learn Drop Rate | 0.1   |

Table 2. Performance Comparison of Classifier Performance

| Classifier | Accuracy | Loss  |
|------------|----------|-------|
| LSTM       | 98.4%    | 0.046 |
| BiLSTM     | 99.05%   | 0.0347|
In addition to the classification task to segregate the incoming emails, a web application is developed to respond automatically to the incident email categories. For the case of service request types, service request by expert is invoked. These are implemented using web applications. The screenshots of webpage developed and that for every output type is presented in Figures 5 to 7.

![Service Form](image)

**Fig.5** Home screen to reach the service desk

![User Details](image)

**Fig.6** Webpage displaying the status of token issued for different customers
5. Conclusion
In this work, an automated and intelligent email auto responder is developed using RNN based classifier. The modules presented in this work are RNN based email classification and development of web application for email responder and invoking service request. As there is no standard database available, a two class classification database is developed and is used in this work. The classification is performed using LSTM and Bi-LSTM networks and obtained classification accuracies of 98.4% and 99.05% respectively. The proposed intelligent email responder system will undoubtedly address the issue of handling huge volume of emails in customer support applications. Further, this work can be extended to incorporate natural language interpreters to understand the mail contents in native language and responding back in the same language in which the mail was drafted.

References
[1]. D. Isa, L. H. Lee, V. P. Kallimani and R. RajKumar, "Text Document Preprocessing with the Bayes Formula for Classification Using the Support Vector Machine," in IEEE Transactions on Knowledge and Data Engineering, vol. 20, no. 9, pp. 1264-1272, Sept. 2008, doi: 10.1109/TKDE.2008.76.
[2]. J. Wu, "A Framework for Learning Comprehensible Theories in XML Document Classification," in IEEE Transactions on Knowledge and Data Engineering, vol. 24, no. 1, pp. 1-14, Jan. 2012, doi: 10.1109/TKDE.2011.158.
[3]. Dehghani, M., Shakery, A., & Mirian, M. S. (2016). Alecsa: attentive learning for email categorization using structural aspects. Knowledge-Based Systems, 98, 44-54.
[4]. Liu, C. L., Hsiao, W. H., Lee, C. H., Chang, T. H., & Kuo, T. H. (2015). Semi-supervised text classification with universum learning. IEEE transactions on cybernetics, 46(2), 462-473.
[5]. Bhushan, S. B., & Danti, A. (2017). Classification of text documents based on score level fusion approach. Pattern Recognition Letters, 94, 118-126.
[6]. Wang, H., & Hong, M. (2019). Supervised Hebb rule based feature selection for text classification. Information Processing & Management, 56(1), 167-191.
[7]. Bhagvati, C. (2018). Word representations for gender classification using deep learning. *Procedia computer science, 132*, 614-622.

[8]. Chandra, A., & Khatri, S. K. (2019, November). Spam SMS Filtering using Recurrent Neural Network and Long Short Term Memory. In 2019 4th International Conference on Information Systems and Computer Networks (ISCON) (pp. 118-122). IEEE.

[9]. Jain, G., Sharma, M., & Agarwal, B. (2019). Optimizing semantic LSTM for spam detection. *International Journal of Information Technology, 11*(2), 239-250.

[10]. Kratzwald, B., Ilić, S., Kraus, M., Feuerriegel, S., & Prendinger, H. (2018). Deep learning for affective computing: Text-based emotion recognition in decision support. *Decision Support Systems, 115*, 24-35.