Deep Neural Network Models for Question Classification in Community Question-Answering Forums

Akshay Upadhya B
Dept. of Information Technology
National Institute of Technology
Karnataka, India
akshayupadhya@gmail.com

Swastik Udupa
Dept. of Information Technology
National Institute of Technology
Karnataka, India
swastikudupa@gmail.com

Sowmya Kamath S
Dept. of Information Technology
National Institute of Technology
Karnataka, India
sowmyakamath@nitk.edu.in

***Abstract*—Automatic generation of responses to questions is a challenging problem that has applications in fields like customer support, question-answering forums etc. Prerequisite to developing such systems is a requirement for a methodology that classifies questions as yes/no or opinion-based questions, so that quick and accurate responses can be provided. Performing this classification is advantageous, as yes/no questions can generally be answered using the data that is already available. In the case of an opinion-based or a yes/no question that wasn't previously answered, an external knowledge source is needed to generate the answer. We propose a LSTM based model that performs question classification into the two aforementioned categories. Given a question as an input, the objective is to classify it into opinion-based or yes/no question. The proposed model was tested on the Amazon community question-answer dataset as it is reflective of the problem statement we are trying to solve. The proposed methodology achieved promising results, with a high accuracy rate of 91% in question classification.**

***Index Terms*—Classification, Supervised learning, Deep Learning, Soft Computing**

I. Introduction

Answering and replying to customer queries are an essential component in the organizational workflow of ecommerce sites, especially those that provide a varied and voluminous product catalogue. Such volume and variety inevitably invites significant number of customer queries and questions regarding the product specifications, which have to dealt with by the customer care team. Most such questions are often repeated with a request for information on issues that might have already been addressed in the past or can be responded to easily using the information that already exists in the system. Identifying such duplicate and near-duplicate questions automatically and later, automating answer generation for them, can be crucial in streamlining online customer support and community-based question answering forums. Only in certain cases when the question is opinion-based or is not currently addressed in the system will the involvement of a human or customer care professional is required.

Another motivating aspect for developing such automated systems, is the requirement for developing intuitive and customer-friendly product pages on e-commerce websites. Most products have a multitude of features and specifications, and asking questions regarding other users' experiences is highly encouraged in most e-commerce websites. When a large number of questions are posed for a popular product, very often users find it tedious to go through the FAQ's and forums to determine if anyone else has posted an issue similar to theirs. In such cases, users may genuinely pose a question of their own to satisfy their requirement. Here, automatic identification of duplicate and near-duplicate questions, and then a quick automated response can help in improving customer satisfaction as well as engagement with the service provider.

Towards building such automated response systems, a necessary prerequisite is a mechanism that can classify a given question as a opinion based or a yes/no question. This classification is critical in e-commerce community/customer support, as most yes/no questions can be answered immediately using prior known information. However, in case of an open-ended question, additional information needs to be captured or collected from an external source or a human customer support professional.

In this paper, a deep neural network based approach for addressing the task of classifying questions as fact based or opinion based is proposed. Given a question, the proposed model is built for first capturing the representative features of the question through semantic analysis of the question, comparing the extracted features of the newly submitted question to the extracted features of the model, and determining whether the newly submitted question is factual or opinion-based. This classification can be then used as an input for the subsequent system, where the objective is to determine if the answer to the query can be provided with the resources at hand or not. We present two different approaches - CNN based question classification model and LSTM based question classification model and benchmark their performance on a standard dataset, Amazon QA product dataset.

The rest of this paper is organized as follows - Section II presents a brief overview of existing work in the area of interest. In Section III, the proposed approaches are discussed in detail. Section IV presents the experimental results and observations on relative performance of the proposed models on a standard dataset, followed by conclusion and references.

II. Related Work

Question classification is a topic of common interest in text mining and question-answering and information retrieval communities. There are two main approaches for classification of questions - Machine learning and Rule Based approach. Most search engines and information retrieval systems process the submitted queries using different techniques in a quest to return the most relevant documents based on identified concepts (keywords, phrases, concepts etc). When a user poses a question, they often fail to analyze the information need behind the questions. Hence, understanding user context is a challenging task despite significant advancements in search engines and information retrieval systems.

One of the foremost methods for question classification is a rule based approach, that uses various combinations of constituent words in the questions, along with rules defined by experts. A critical limitation of such systems is that it is often practically unfeasible to automatically analyze these rules and near impossible to deduce every rule. Dodiya et al. [1] proposed a rule based approach for classifying medical questions by a discussion of various rules and patterns. They classified questions into different types like 'what', 'where', 'when', 'why', and 'how' and achieved an average accuracy of 53%. Xu et al. [2] classified questions related to tourism asked in a Chinese question answer forum using three algorithms, namely Bayes, SVM, and a two level method based on question semantic similarity and support vector machine. They reported good accuracy using these methods. Biswas et al. [3] improved upon Li and Roth's [4] methods of classifying questions using a rule based approach. They used a rich feature set, and assigned each question to categories namely, Definition, Factoid, and Descriptive. The question types again here are 'what', 'where', 'when', 'why', and 'how', and achieved an accuracy of approximately 98% on a question dataset given by Li and Roth.

Roth and Li [4] made use of a hierarchical classifier that uses the Winnow Algorithm within Sparse Network of Winnows (SNoW) learning Architecture to classify questions. Kadri and Wayne [5] introduced a statistical question answer based classifier and trained the classifier using the same dataset as Li and Roth. They used a combined system of Singular Value Decomposition (SVD) along with Named Entity tagging and got accuracies of approximately 80%. Suzuki et al. [6] used a hierarchical classifier based on Support Vector Machines (SVM) built on a HDAG Kernel. They classified 5011 questions using five-fold cross validation getting an accuracy of 88.2%. Zhang et al. [7] performed experiments with five machine learning algorithms namely Naive Bayes, Nearest Neighbours, Decision Tree, Support Vector Machines and Sparse Network of Winnows, for automated question classification. They concluded that syntactic structures are helpful for classification of questions. They also reported that SVM outperforms the other four classifiers when it comes to only surface text features.

Several deep neural network models [8] and representation learning [9] approaches have been proposed for addressing data sparsity problems. Learning word representations using many neural models have also been proposed [10] [11] [12]. Word embedding refers to the neural representation of a word is a vector which is real valued. By using the distance between two embedding vectors word embedding helps to understand relatedness between two words. When neural networks are used alongside with word embedding it becomes possible to demonstrate great performances in many NLP tasks. Mannin et al [13] performed sentimental analysis of phrases and sentences using recursive tensor network. Usage of recurrent neural network to build language models was proposed by Mikolov et al [14]. Kalchbrenner et al [15] proposed a novel recurrent network for dialogue act classification. Semantic labeling of roles using convolutional neural network was introduced by Collobert et al [16]. Elaneh et al [17] proposed a novel method to find questions similar to a question posted on a community question answer forum using deep LSTM neural networks and reported good accuracy. In a similar application, Paitan et al [18] proposed a Recurrent Neural Network (RNN) in the form of Long-short term memory to create an automated FAQ bot to answer user queries. Arefin et al [19] proposed a model which combines Convolution Neural Networks with Recurrent networks to provide automated answering.

Based on this overview, it can be concluded that most existing works utilize question parts to set rules for facilitating classification. Also, most proposed approaches are applied to small datasets. For large datasets, the process of defining rules is highly challenging due to variety and voluminosity of the data. Hence, rule based approaches are unsuitable, thus we propose deep learning approaches for semi-supervised question classification in this paper. The motivation behind our work is to use the proposed model to design a system that provides automated responses to user queries, by determining the nature of the question and using this to identify if it can be answered with existing data or not. To the best of our knowledge, there are no works that use the Amazon QA dataset for benchmarking such a system and hence we propose to develope such an automated response system.

III. Proposed Methodology

Figure 1 and 2 depict the proposed CNN and LSTM based models for question classification. We describe the processes involved in detail in this section. The dataset used for the experimental evaluation of the proposed approaches contains a collection of Amazon product Question Answers, called the Amazon QA dataset [20] [21]. The dataset was created by crawling Amazon.com's website for user questions and associated answers, which are labelled for training machine learning models. Each data item is provided as a jsonarray, containing a json object with various features like question, answer, question type etc. We considered the question and questionType elements of the item. A sample data item from the Amazon QA dataset is shown below.

```
{
    "asin": "B00004U9JO",
    "questionType": "yes/no",
    "answerType": "Y",
    "answerTime": "Oct 31, 2013",
    "unixTime": 1.3832e+09,
    "question": "will this one work
        to replace the badger 1 1/3 HP?",
    "answer": "Yes it will. The sink
        connector is even the same"
}
```

The dataset is almost equally divided into two classes, i.e., *yes/no* and *open-ended*. A yes/no question is one that can be answered with a simple *yes* or *no*. An open-ended question is one that requires a description for an answer, or requires stating certain facts or properties of the product as the answer. Our aim is to classify the questions into the above mentioned question types. The dataset has multiple sets of questions of various product categories sold in Amazon. Some of them are as listed below. We

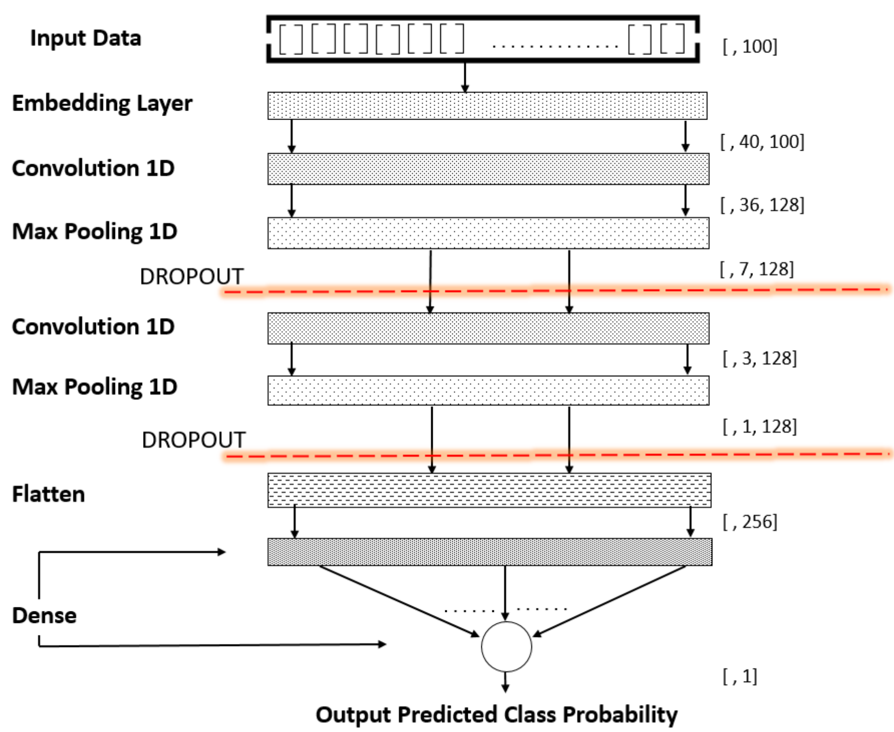

Fig. 1. Proposed CNN based model for Question Classification

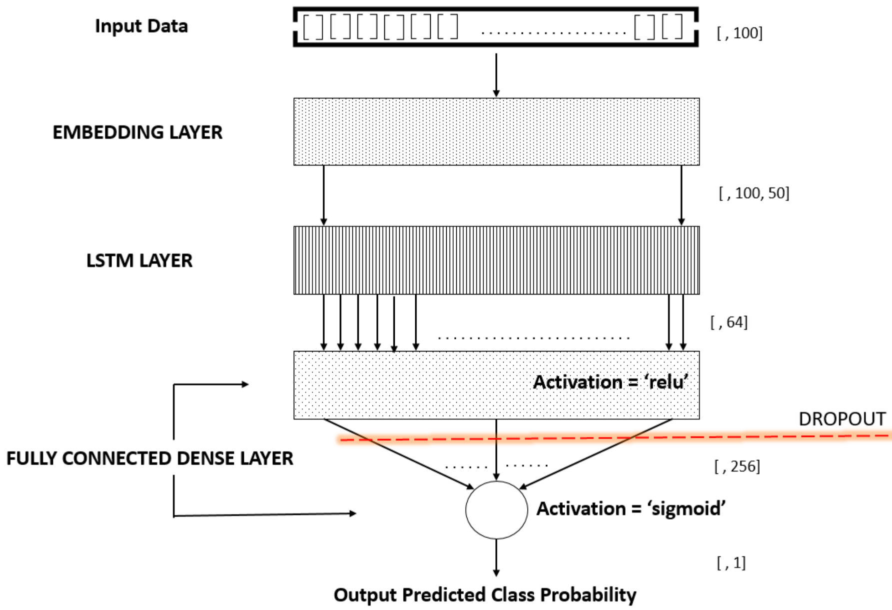

Fig. 2. Proposed LSTM Architecture for Question Classification

have classified the sets of questions into three categories - Appliances (9,011 questions), Beauty (42,422 questions) and Electronics (314,263 questions).

A. Data Preprocessing

For the purpose of training, we converted the JSON into a Pandas dataframe and extracted the 'question' and 'questionType' column from the dataframe. Basic text preprocessing was performed on each 'question' in the dataframe. Firstly, each question sentence was split into a list of words using a process of tokenization, after which every word was converted to its lower case form and numbers were discarded, thus normalizing the data. Finally, stop words like 'i', 'me', 'a', 'an', 'the' etc was removed, using the NLTK library of Python. This data is passed to the Keras Tokenizer which converts the words into specific numbers, and then into vectors which are the input vectors for the models.

B. Question Classification Models

To classify a given question as an open-ended or yes/no type question, a Convolution Neural Network (CNN) based architecture is proposed (depicted in Fig. 1). CNNs are deep neural networks composed of several layers of convolutions with activation functions such as tanh, sigmoid and ReLU applied to the results. We designed a ConvNet which is 5 layers deep with 3 convolutional layers and 2 fully-connected layers. Each question is represented as a word embedding and is input to the model. The architecture consists of an embedding layer followed by a convolution layer and a max pooling layer. Another convolution layer is added following which the data is flattened and fed to two dense layers and the desired output is obtained. Dropout layers are used after every maxpool and dense layers in order to prevent over fitting. We also inserted 2 dropouts [22] between the three convolution layers to regularize the output, with a dropout probability of 0.5.

Figure 2 illustrates the architecture of the Long Short Term Memory (LSTM) based model designed for question classification. LSTMs are a type of RNNs (Recurrent Neural Network) that can store information for any arbitrary duration, and system parameters are trainable in a reasonable amount of time. The proposed LSTM model consists of 4 layers. The third layer is a fully connected layer having 'relu' as the activation function, following a dropout layer. The dropout layer drops cells at a probability of 0.3. Following this dropout layer is a fully connected output layer which has 'sigmoid' as the activation function. This provides a binary prediction of the question type. The model is compiled with 'rmsprop' optimizer and uses 'binary cross entropy' loss function.The binary cross entropy loss is defined as per Eq. (1).

$$loss = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)] \quad (1)$$

where y_i stands for the actual value of the class, and $\hat{y}$ stands for the prediction made by the model. This is seen to produce the best results for our model and hence we use it for the loss function. The experimental results observed are discussed in Section IV.

IV. Experiments and Results

We performed experiments for classifying questions from three different categories as mentioned earlier - Appliances (9,011 questions), Beauty (42,422 questions) and Electronics (314,263 questions). The questions are subjected to preprocessing processes as discussed in Section III-B. After this process, the data is split into a training set and a test set with a ratio of 85:15. The questions of the training set are passed to a Tokenizer of Keras whose properties of maximum allowed words are set to 1000. The questions now are converted into a sequence of tokens of length 100. If the length is insufficient, appropriate padding is applied.

The proposed CNN Model is applied to the Amazon QA dataset for 50 epochs. We hypertuned the parameters to obtain the best possible acurracy. It was observed that the data overfit when using this model as an average accuracy of 95% is obtained while producing comparatively low validation accuracy. Table I provides details on the observed performance during question classification using the CNN model. Table I shows the accuracy obtained by training the model on each component of the dataset. Although there is a significant increase in the size of the dataset between the three, we noticed that the accuracy remained almost at the same value across the various categories.

TABLE I
Question classification performance of the CNN Model

Dataset category	Test Accuracy
Appliances	69.55%
Beauty	69.77%
Electronics	72.30%

Similarily, the preprocessed training data is used to train the LSTM model. This classifier is trained separately on the aforementioned three categories of Amazon QA dataset for ten epochs each. We noted that the validation accuracy increased with every epoch's accuracy along with a decrease in the loss. Thus, we can conclude that there is no case of overfitting of the data. For the largest among the 3 categories, Electronics dataset, an accuracy of 93.4% was observed. The results of LSTM model are tabulated in Table II. It can be seen that, as the size of the dataset increases, there is a significant increase in the accuracy and the test loss also decreases. The accuracy peaked at 93.4% for the dataset component with the highest number of questions, i.e. the electronics category.

TABLE II
Question classification performance of the LSTM Model

Dataset category	Test Loss	Test Accuracy
Appliances	0.410	85.5%
Beauty	0.223	91.6%
Electronics	0.190	93.4 %

On comparing the performance of the two models, it can be observed that the LSTM classifier achieved a higher accuracy than the CNN classifier. The LSTM classifier outperformed the CNN classifier by approximately 23%, 31% and 29% for the dataset component Appliances, Beauty and Electronics respectively. Figure 3 illustrates the comparative performance of the two models.

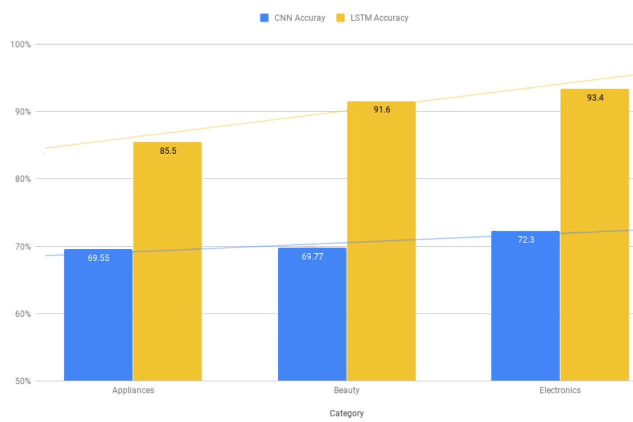

Fig. 3. Comparative performance of the CNN and LSTM Models on the Amazon QA Dataset

V. Conclusion and Future Work

In this paper, two approaches for classifying questions into yes/no and open-ended questions for supporting automated customer query answering in online question-answering forums and e-commerce sites was presented. The proposed models were benchmarked on a standard dataset containing labelled questions, called the Amazon QA dataset, from which three categories, Appliances, Beauty and Electronics was considered for the experimental evaluation. Two deep learning techniques, CNN and LSTM were trained on the word embeddings generated from the questions in the dataset. The CNN model achieved a maximum accuracy of 72.37% for the electronics category of the dataset, but suffered from overfitting problems. The LSTM model was able to fit the data really well and outperformed the CNN by a significant margin by achieving an accuracy of 93.4% for the electronics category of the dataset.

As part of future work, we intend to apply the LSTM model on the community question-answering forum data and fine tune its parameters for further improvement in performance. Next, our intention is to enhance the question classification model for generation of automatic responses to the users' queries, with reasonable accuracy, so that the cognitive burden of dealing with duplicate and near-duplicate questions on community question-answering forums is reduced to a great extent.

REFERENCES

[1] T. Dodiya and S. Jain, "Question classification for medical domain question answering system," in *2016 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE)*, pp. 204–207, Dec 2016.

[2] J. Xu, Y. Zhou, and Y. Wang, "A classification of questions using svm and semantic similarity analysis," in *2012 Sixth International Conference on Internet Computing for Science and Engineering*, pp. 31–34, April 2012.

[3] P. Biswas, A. Sharan, and R. Kumar, "Question classification using syntactic and rule based approach," in *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 1033–1038, Sep. 2014.

[4] X. Li and D. Roth, "Learning question classifiers," in *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1*, COLING '02, (Stroudsburg, PA, USA), pp. 1–7, Association for Computational Linguistics, 2002.

[5] K. Hacioglu and W. Ward, "Question classification with support vector machines and error correcting codes," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: Companion Volume of the Proceedings of HLT-NAACL 2003–short Papers - Volume 2*, NAACL-Short '03, (Stroudsburg, PA, USA), pp. 28–30, Association for Computational Linguistics, 2003.

[6] J. Suzuki, H. Taira, Y. Sasaki, and E. Maeda, "Question classification using hdag kernel," in *Proceedings of the ACL 2003 Workshop on Multilingual Summarization and Question Answering - Volume 12*, MultiSumQA '03, (Stroudsburg, PA, USA), pp. 61–68, Association for Computational Linguistics, 2003.

[7] D. Zhang and W. S. Lee, "Question classification using support vector machines," in *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, (New York, NY, USA), pp. 26–32, ACM, 2003.

[8] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[9] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, pp. 1798–1828, Aug 2013.

[10] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, Mar. 2003.

[11] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural Language Processing (almost) from Scratch," *ArXiv e-prints*, Mar. 2011.

[12] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *ArXiv e-prints*, Jan. 2013.

[13] J. . M. C. . A. Y. N. Socher, Richard Bauer, "A neural probabilistic language model," *J. Mach. Learn. Res.*, pp. 455–465, 2003.

[14] R. Pascanu, T. Mikolov, and Y. Bengio, "Understanding the exploding gradient problem," *CoRR*, vol. abs/1211.5063, 2012.

[15] N. Kalchbrenner and P. Blunsom, "Recurrent continuous translation models," *EMNLP 2013 - 2013 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, vol. 3, pp. 1700–1709, 01 2013.

[16] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. P. Kuksa, "Natural language processing (almost) from scratch," *CoRR*, vol. abs/1103.0398, 2011.

[17] E. Karimi, B. Majidi, and M. T. Manzuri, "Relevant question answering in community based networks using deep lstm neural networks," in *2019 7th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS)*, pp. 1–5, Jan 2019.

[18] P. Muangkammuen, N. Intiruk, and K. R. Saikaew, "Automated thai-faq chatbot using rnn-lstm," in *2018 22nd International Computer Science and Engineering Conference (ICSEC)*, pp. 1–4, Nov 2018.

[19] M. M. A. Zaman and S. Z. Mishu, "Convolutional recurrent neural network for question answering," in *2017 3rd International Conference on Electrical Information and Communication Technology (EICT)*, pp. 1–6, Dec 2017.

[20] M. Wan and J. McAuley, "Modeling ambiguity, subjectivity, and diverging viewpoints in opinion question answering systems," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pp. 489–498, Dec 2016.

[21] J. McAuley and A. Yang, "Addressing complex and subjective product-related queries with customer reviews," in *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, (Republic and Canton of Geneva, Switzerland), pp. 625–635, International World Wide Web Conferences Steering Committee, 2016.

[22] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *CoRR*, vol. abs/1207.0580, 2012.