Predicting Online Doctor Ratings from User Reviews Using Convolutional Neural Networks

Ranti D. Sharma, Samarth Tripathi, Sunil K. Sahu, Sudhanshu Mittal, and Ashish Anand

Abstract—**Individuals are increasingly turning to the web to seek and share healthcare information and this trend in online health information has resulted in a proliferation of user generated health centric content, especially online physician reviews. Physician rating websites can play a major role in empowering patients to make informed choices while selecting healthcare providers for advice and treatment. Given the wealth of information hidden in unstructured narratives such as online ratings, comments and clinical documents, there is a critical need for building efficient and accurate text classifiers for biomedicine corpus. In this paper, we analyze patient (dis)satisfaction using performance reviews of doctors and predict their ratings on various measures such as 'Knowledgeability', 'Staff' and 'Helpfulness'. We explore solutions for the same problem using Convolutional Neural Networks trained on various optimization and loss functions. We analyze the 35000 user reviews available at "www.ratemds.com" for more than 10000 doctors. The proposed model obtained an accuracy of 93% for positive/negative binary classification of patient reviews. Moreover, we obtained a mean absolute error of 0.525 in predicting rating on a 5-point scale, thu, significantly improving upon the state of the art's error rate of 0.71.**

Index Terms—**Text classification, sentiment analysis, convolutional neural networks, dropout, physician review.**

I. Introduction

Online health forums are becoming an increasingly popular platform for people to search for health-related information. Patients are seeking information not only about disease conditions but also about physicians and hospitals [1]. A major chunk of these online health related search queries deal with physicians and hospitals related information, in particular, physician information including performance reviews and ratings. Physician reviews are first-hand qualitative feedback provided by patients to their medical consultants. Platforms such as RateMDs.com, HealthGrades.com, vitals.com etc. are few of the popular websites that allow patients to share their experience with individual MDs on a variety of aspects such as medical knowledgeability, personal demeanor, and staff quality [2]. These physician-rating websites offer patients an open way to critique and obtain information about physician performance. Not only do such ratings provide information to make informed choices in a convenient and universally accessible manner, but they can also provide key insights on the factors that influence a patient's perception of his or her physician when analyzed en masse [3].

Predicting a user's sentiment from reviews and comments has been studied widely [2], [4]-[6]. Recent developments in machine learning [7], [8] has shown that *convolutional neural network* (CNN) model can yield high fidelity text classifiers and obtain state-of-the-art performance for several natural language processing (NLP) tasks including sentiment analysis. Most notable is the recent work of Yoon Kim [8] which uses CNN trained on top of pre-trained word vectors for sentence-level classification tasks to provide prime accuracy for generic sentiment analysis. Feature engineering can be greatly substituted using CNNs [9] demonstrating the strength of such models in capturing high level features that lead to superior text classifiers.

In our work, we seek to predict online doctor ratings from users' reviews available in RateMDs.com using CNN. Our research builds upon the work of Paul *et al*. [2], where they use *Joint-Topic Model* to help predict patients' ratings for doctors based on their reviews. We also compare various combinations of loss functions, learning rate optimizers, and word vector models and while training our model. Moreover, we discuss the performance of Dropout technique [10] on our model's learning capacity.

The proposed CNN model provides for state-of-the-art accuracy in classifying users' reviews into discrete labels and ratings. Our model yields 93% for positive/negative classification and 0.525 mean absolute error for 5 point scale classification of doctor ratings based on user reviews for various categories such as 'Knowledgeability', 'Staff' and 'Helpfulness' which convincingly improves the accuracy reported by Paul *et al*. [2] by 73%. Our model is simple, robust, fully predictive, and scalable which can be easily extended as it involves no feature engineering.

II. Related Works

Galizzi *et al*. [11] explored the extent to which doctor rating websites are known and used among a sample of respondents from London to understand the main predictors of what makes people willing to use doctor rating websites. They conducted a self-administered survey to assess the extent and determinants of awareness of doctor-rating websites; the level of actual usage of those websites; the intention to use. They concluded that online rating websites can play a major role in supporting patients' informed decisions on which healthcare providers to seek advice from,

Manuscript received October 30, 2015; revised February 11, 2016.

Ranti Dev Sharma, Samarth Tripathi, Sunil Kumar Sahu, and Sudhanshu Mittal were with the Computer Science Department, Indian Institute of Technology Guwahati, India (e-mail: ranti.iitg@gmail.com, samarthtripathi@gmail.com, sunilitggu@gmail.com, sudhanshumittal1992@gmail.com).

Ashish Anand is with the Computer Science Department, Indian Institute of Technology Guwahati, India (e-mail: anand.ashish@iitg.ernet.in).

thus potentially fostering patients' choice in healthcare. Similarly according to Keckley *et al.* [1] in the United States, 47% people look up information about their providers online, 37% consult physician-rating sites, and 7% of people who seek information about their provider post a review online. Moreover, a separate study found that 15% of consumers compare hospitals before making a selection, and 30% of consumers compare physicians' online before making a selection [4]. Finally, Kadry *et al*. [5] bolsters the claim that a single overall rating to evaluate physicians may be sufficient to assess a patient's opinion of the physician. Hence, it is clear that there is a need to encapsulate sentiments from numerous user reviews on a physician's qualities and quantify them into numeric rating levels that are much easy to grasp.

An attempt to rate physicians based on patient reviews was made by Lopez *et al*. in [12], where they conducted qualitative content analysis of 712 online reviews from two rating websites. They purposively sampled reviews of 445 primary care doctors (internists and family practitioners) from four geographically dispersed U.S. urban locations. They observed that majority of internet reviews of primary care physicians are positive in nature. Their findings reaffirm that the care encounter extends beyond the patient–physician dyad; staff, access, and convenience all affect patient's reviews of physicians. In addition, negative interpersonal reviews underscore the importance of well-perceived bedside manner for a successful patient–physician relationship.

Our paper mainly seeks to address the same problem as Paul *et al*. [2]. They use a Joint-Topic Model to predict patients' ratings for doctors based on their reviews using factorial Latent Dirichlet Allocation for topic modelling; and obtain prime accuracy predicting the 5 point rating for rateMDs.com reviews using text classification. Our work closely follows the same problem but uses a completely different approach of convolutional neural networks and convincingly betters their results. Paul *et al*. [2] approach incorporated a small amount of pre labelled data to aid their f-LDA based model. Our approach is more generic and extendible as it does not need predefined labels. Our motivation also stems from the works of Rios and Kavuluru [7], who provide a platform to build high accuracy text classifiers for the field of biomedicine using convolutional neural networks. Their model effectively extracts information hidden in structured narratives in articles and clinical documents, and classifies them based on Medical Subject Headings (MeSH). Their results underscored the potential that CNNs hold in the field of biomedicine for the purposes of classification. We deploy a similar model for classifying sentiments that patients have towards their physician for various attributes.

Other notable works involving sentiment based textual classification includes Ganu *et al*. [6], who identify topical and sentiment information from free-form text reviews for restaurants, and use this knowledge to improve user experience in accessing reviews. Kalchbrenner *et al.* [13] describe a convolutional architecture (dubbed Dynamic Convolutional Neural Network (DCNN)) that they adopt for the semantic modelling of sentences. The network uses Dynamic k-Max pooling, a global pooling operation over linear sequences, handles input sentences of varying length and induces a feature graph over the sentence that is capable of explicitly capturing short and long-range relations. Their network achieves excellent performance in many NLP tasks such as sentiment analysis and a greater than 25% error reduction in the Twitter sentiment prediction with respect to the strongest baseline. Zhang *et. al*. [14] demonstrate that we can apply deep learning to text understanding from character level inputs all the way up to abstract text concepts, using temporal convolutional networks. They show that Convolutional Networks do not need any knowledge on the syntactic or semantic structure of a language to give good benchmarks text understanding especially in contrast with various previous approaches where a dictionary of words is a necessary starting point, and usually structured parsing is hard-wired into the model.

In particular, we analyze and build upon the work of Yoon Kim [8], who reports on a series of experiments with convolutional neural networks (CNN) trained on top of pre-trained word vectors for sentence-level classification tasks. Their research shows CNN with little hyperparameter tuning and static vectors achieves excellent results on multiple benchmarks. Learning task-specific vectors through fine-tuning offers further gains in performance. They additionally propose a simple modification to the architecture to allow for the use of both task-specific and static vectors. In their work Kim uses and compares many variants of convolutional neural networks including random, static, non-static, and multichannel based on how word vectors are trained. They treat a sentence as a concatenation of words and apply a filter to each possible window of words to produce a feature map. They then apply a max-overtime pooling operation [9] to capture the most important feature for each feature map. The word vectors have two channels - one from static training and other fine-tuned via backpropogation. Their approach eclipses many standard natural language processing problems including sentiment analysis and question classification. Our approach employs a similar deep learning model with Dropout as regularizer.

Our work also involves training a convolutional neural network on top of word vectors generated by unsupervised learning. Initializing word vectors with those obtained from an unsupervised neural language model [9] is a popular method to improve performance in the absence of a large supervised training set. Word vectors, wherein words are projected from a sparse, 1-of-V encoding (where V is the vocabulary size) onto a lower dimensional vector space via a hidden layer, are essentially feature extractors that encode semantic features of words in their dimensions. In such dense representations, semantically close words are likewise close—in Euclidean or cosine distance—in the lower dimensional vector space [8].

III. Experimental Setup

A. Dataset

The website RateMDs.com provides a platform for patients to review doctors and give ratings on a 5-point scale on three different categories - helpfulness, staff quality, and knowledgeability. We have downloaded more than 35000

reviews covering approximately 10000 physician specialties. A typical patient review consists of free text and corresponding numerical scores for each of the main categories as depicted in Table I. The dataset is uniformly sampled across a wide geographical coverage. We use RateMDs's dataset mainly for two important reasons: Firstly, it serves as the standard dataset for user reviews and sentiment in the field of healthcare and has been widely used by researchers with consistent reliability. Secondly, it helps us to directly compare our results with the earlier reported state-of-the-art accuracy for the same problem by Paul *et al*. [2].

TABLE I: SAMPLE REVIEWS OF DOCTORS FROM RATEMDS.COM

Review	Knowledge	Helpful	Staff
Dr. Romero is wonderful. I can always know that he will be honest and not money hungry. If i ask for a service that he feels I do not need he will tell me no. He has great bedside manner. I have been coming to him for 4 years and I plan on staying with him for a long time. I do routine Botox with him. Once every 4 months.	5	5	5
The worst dentist/ clinic I've ever experienced- Just used to 100% better than I recieved. I ended up with an infection- which didn't surprise me at all after the treatment I recieved- I got the antibiotic from a family physician. I feel offended that I'm forced to give 1 point. If the staff keep smiling while you are feeling mistreated is that good?	1	1	1

B. Experiment Design

For our experiment's setup we use the mapping of the text reviews to their respective ratings to train our convolutional neural network model, and then predict ratings for test reviews. We formulate two classification problems for this setting. The ratings of reviews for each category are discretized into two classes of positive and negative sentiments in the first problem, whereas, in the second, we keep the original ratings on a 5-point scale as it is and train our model to learn the mapping of texts to ratings. In the process, our model also learns the key characteristic words which most important in determining the net sentiment for each category of doctor ratings.

C. Word Vectors

Word vectors are low dimensional vector representations that encode semantic features of words [15] learned in unsupervised neural language framework [16] using a large text corpus. In our work, we use the publicly available word2vec [17] word vectors that were trained on 100 billion words from Google News. Each word is represented as a 300 dimensional vector. We call this as pre-trained word vectors and model using them without updating their element are called *static word vector model*. Words which are not present in the set of pre-trained words are initialized randomly.

Although we use also random word vectors for each word but they get updated during training. In other words, at the end of training, we also learn the vector representation of the words present in our training data. We call this model as R*andom Word-Vector model*.

IV. MODEL

A. Architecture Overview

Our model is similar to the CNN architecture used by Collobert *et al*. in [9]. In this architecture, we can think of a sequence of words in the sentence or in the context window as input to the network. Each word is being represented as vectors of same length. Thus a sentence is being represented as a 2-dimensional matrix. Here, one can draw analogies between text classification and image recognition, a task for which convolutional neural networks have shown to be exceptionally effective [18]. Apart from the INPUT layer, we deploy four main types of layers to build neural architecture namely, Convolutional Layer (CONV), ReLu Layer (RELU), Pooling Layer (POOL), and Fully-Connected Layer (FC) in that particular order. Fig. 1 depicts the structure of our CNN, where '*n*' represents the total number of reviews in the dataset.

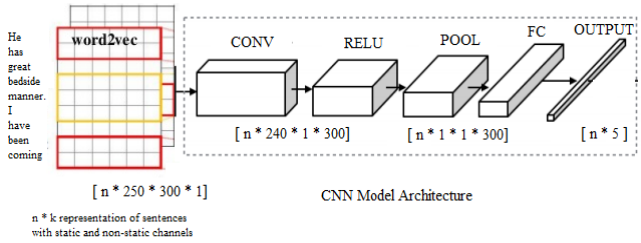

Fig. 1. Architecture of our CNN implementation.

CONV layer computes the output of neurons that are connected to local regions in the input, each computing a dot product between their weights and the region they are connected to the input volume. RELU layer applies an element wise activation function, such as the max(0, *x*) thresholding at zero. This leaves the size of the volume unchanged. POOL layer performs a downsampling operation along the spatial dimensions (width, height), resulting in the said volume. FC layer computes the class scores, resulting in volume of size, where each of the 5 numbers correspond to a class score [19].

B. Input Layer

INPUT layer holds all user reviews. All the sentences within a review are considered as one mega sentence. We regard this mega sentence of length *k* as a vector of words of dimensionality of *k*. As explained earlier, each word is represented by a word vector of dimensionality *word_vector_dimension* = 300. Hence regarding a sentence as a concatenation of words, and treating words as word vector, a sentence becomes a 2- dimensional matrix. As all reviews may not be of the same length, we considered all mega sentences as of length *max_word_count*, and mega sentences with shorter length were padded to make them of

the same length. Thus *INPUT* [number_of_reviews * max_word_count * word_vector_dimension * 1] layer will hold the sentences of the user review, where 'number_of_reviews' represents the total number of reviews in the dataset, 'max_word_count' represents the maximum count of words in a mega sentence (250 for our implementation) representing a review, and 'word_vector_dimension' represents the maximum length of a word vector which is 300 in our case.

C. Convolution

The Convolution layer is the core building block of a Convolutional Network. A single convolution operation involves applying filters of dimension 'number_of_reviews*feature_dimension*word_vector_dimension' to a window of 'feature_dimension' words. Every filter is spatially small (along width and height), but extends through the full depth of the input volume. A feature is defined as a function on a window of words using bias terms and a non-linear function. During the forward pass, we slide (more precisely, convolve) each filter across the width and height of the input volume, producing a 2-dimensional activation map of that filter. As we slide the filter, across the input, we are computing the dot product between the entries of the filter and the input. Once a filters have been glazed over the complete input, we find the single most important (adding most sentimental value) feature using max-over-time pooling operation (*POOL Layer*). This allows us to correctly identify one feature for each filter. The model repeats this for each filter in the sentence, to obtain best features for a rating, in each convolution. Stacking these features for all filters along the depth dimension forms the full output volume. Thus, every entry in the output volume can also be interpreted as an output of a neuron that looks at only a small region in the input and shares parameters with neurons in the same activation map. Our convolution layer's output has a dimensionality of CONV [number_of_reviews * 240 * 1 * word_vector_dimension], and the input is the INPUT layer.

D. Pooling Layer

Its function is to progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network, and hence to also control overfitting. The Pooling Layer operates independently on every depth slice of the input and resizes it spatially, using the MAX operation. As our pooling layer links between CONV and FC layers, its output dimensionality is POOL [number_of_reviews * 1 * 1 * word_vector_dimension].

E. Fully Connected Layer

Neurons in a Fully Connected layer have full connections to all activations in the previous layer, as seen in regular neural network models. Their activations can hence be computed with a matrix multiplication followed by a bias offset. The final FC outputs layer outputs [number_of_reviews *5] or [number_of_reviews *2] depending on whether the required output is for binary classification or for 5-scale classification.

F. Optimization and Activation Functions

To speed up the learning rates we experiment with and compare three learning techniques, namely RMSProp [20], Stochastic Gradient Descent (SGD) and Adadelta. RMSProp divides the learning rate for a weight by a running average of the magnitudes of recent gradients for that weight. We have used a learning rate of 0.001 and a gradient direction of 0.9 for our RMSProp learning. On the other hand, SGD method first divides the dataset into small batches of examples, compute the gradient using a single batch and make an update, then move to other batches of examples [21]. For SGD technique we use the learning rate of 0.01 with momentum and learning rate decay over each update. Adadelta is a method that uses the magnitude of recent gradients and steps to obtain an adaptive step rate by using exponential moving average over the gradients and steps (sliding window) [22]. We have used a learning rate of 1.0 and a gradient direction of 0.95.

We deploy Rectified Linear Units (ReLU) as our non-linear activation function, among various non-linear activation functions like sigmoid, tan hyperbolic etc. ReLU has also been argued to be more biologically plausible and practical [23, 24].

G. Dropout Technique

Deep learning generally uses huge datasets, in order of millions, for its neural layers to effectively learn. However as dataset size is constrained in our problem, we use dropout technique to obtain good results. The dropout technique [10] probabilistically drops a proportion p of the hidden units during forward-backpropagation from the neural network to avoid co-adaptation of hidden units. Thus dropout acts as a regularization to prevent overfitting [8]. While using the dropout technique it is important to use large number of epochs. An epoch is a measure of the number of times all of the training vectors are used once to update the weights. For batch training all of the training samples pass through the learning algorithm simultaneously in one epoch before weights are updated.

H. Details of Implementations

Our experiment uses the same model to predict doctor ratings as positive or negative, and also numerical rating on a scale of 1 to 5 (which is used by with rateMDs.com). We have implemented the proposed model on Keras [25], a Theano based deep learning library for python. We train our model in batches of 128, with an epoch of 500 to allow sufficient learning. For our input layer we use a dropout probability of 0.2, while for the successive hidden layers we use a dropout factor of 0.5. These probabilities have been empirically shown to yield best results [10], which we also reaffirm in our own tests. Our model trains on a single pass in the order [INPUT - CONV - RELU - POOL - FC - Softmax].

V. Results

Our model supports two classification models for the three categories, first as classifying reviews as positive or negative and secondly into classifying ratings into 5 ratings. Out of the total 35000 unique reviews found in RateMDs.com, we use the first 30000 as our initial training dataset, and the remaining 5000 (approximately 14 %) as our test dataset.

Results of the first binary classification problem of positive and negative sentiments are summarized in the Table II. As is evident from our results, the CNN model trained with only the pre-trained word vectors has obtained good accuracy. Further, our results suggest CNN model is sensitive to loss function and optimization method. We can see from the TABLE II that *Adadelta* with *Categorical cross entropy (CCE) consistently* provides the best combination to give best result across the three categories. All our experiments were performed with this combination of *CCE* as loss function and *Adadelta* as optimizer.

TABLE II: ACCURACY FOR BINARY CLASSIFICATION
(A): ACCURACY FOR HELPFULNESS

	SGD	RMSProp	Adadelta
Mean Squared Error	88.33	93.73	93.52
Binary Cross Entropy	68.94	93.38	93.24
Categorical Cross Entropy	93.64	93.36	93.76

(B): ACCURACY FOR STAFF QUALITY

	SGD	RMSProp	Adadelta
Mean Squared Error	76.79	85.88	87.74
Binary Cross Entropy	82.39	86.73	87.51
Categorical Cross Entropy	78.63	85.39	88.81

(C): ACCURACY FOR KNOWLEDGE ABILITY

	SGD	RMSProp	Adadelta
Mean Squared Error	74.15	91.60	91.10
Binary Cross Entropy	82.27	91.93	91.79
Categorical Cross Entropy	79.65	91.60	92.08

Further, we evaluate the usefulness of pre-trained word vectors (static word vector model) compared to random vector model. In both the cases, we have kept the same combination of *CCE* and *Adadelta* as loss-function and optimizer respectively. Table III summarizes the result obtained for the three different categories and it is quite evident that the static word vector model has given better performance. Our result is in accordance to the results of Kim [9] and to our intuition as well. The pre-trained word vectors were trained on much bigger and semantically rich corpus than the training data used for this study and hence these word-vectors were able to capture more lexico-semantic properties of words than the random word-vectors trained only on training data.

TABLE III: COMPARISON OF ACCURACY FOR BINARY CLASSIFICATION BETWEEN STATIC AND RANDOM WORD VECTORS

Category	Helpfulness	Staff	Knowledge
Random	86.31	82.77	85.29
Static	93.76	88.81	92.08

We further checked how our model relates sentiment for each category to word vectors by analyzing words which word vectors were closest to important sentiment determining words. Table IV clearly depicts some of the closest words (using cosine distance) in vector space to important words for the category of "helpfulness", we derived using random word vector models. The table provides valuable insight into how our neural model assigns sentiment to words, for example the words closest to "helpful" in user reviews were "loving", "ingenious", "humor"; while for "professional" were "stern", "lively", and "marvelous". This helps us understand that generally users that found a doctor helpful also felt strongly related to an emotion of love and humor from the doctor. Our results, like Paul at. al., also finds strong correlation between "helpfulness" and words like “caring”, “helpful”, “indifferent”; the technical "staff" category contains words about “surgeries”, “energizing”, “disgusting”.

TABLE IV: CLOSEST WORDS TO SENTIMENT DETERMINING KEYWORDS

Key Word	Closest	words		
Good	virtue	transparent	terrific	Moral
Bad	superficial	indifferent	scathing	Guilt
Professional	marvelous	lively	zeal	stern
Racist	impulsive	disjointed	military	whine
Helpful	loving	ingenious	humor	firm

We also compare the effect of the dropout technique on the performance of static word vector model. We find, as expected, dropout improves the accuracy of our model by around 5% for binary predictions, and around 6.5% for the 5 rating predictions, as shown in the Table V.

TABLE V: COMPARISON OF ACCURACY FOR CLASSIFICATION WITH AND WITHOUT DROPOUT TRAINING SCHEMA. FIRST ENTRY IN EACH PAIR INDICATE ACCURACY OBTAINED USING DROPOUT AND SECOND WITHOUT DROPOUT

Category	Helpfulness	Staff	Knowledge
Binary	93.73 , 89.24	93.73 , 89.24	92.08 , 89.66
5-scale	83.72 , 78.31	74.20 , 65.07	79.73 , 73.08

In the second sets of our experiments, we evaluate the performance of our model on the prediction task of reviews. Each review in the training data was given a rating on a 5 point scale. While training our model, we used static word vector model with *CCE* as loss function, *Adadelta* as optimizer, supplemented by the Dropout technique. We compare our results to the results of the study [2] by Paul et al. Results are summarized in the Table VI. Since the results in [2] are given as *Mean Absolute Error*, we have also given our result in terms of the measure. As is evident, our approach yields substantially better results for all categories. It provides a better and more generalized model than the latent Dirichlet Allocation based Joint Topic model used in the earlier study.

TABLE VI: COMPARISON OF MEAN ABSOLUTE ERRORS BETWEEN OUR AND PAUL *ET AL.* [2] RESULTS

Category	f-LDA	CNN
Helpfulness	0.59	0.389
Staff	0.91	0.685
Knowledge	0.64	0.501

Our results bolster the claim that convolutional neural networks can model text sentiments better than shallow learning models when used effectively. Our results also support that using predefined yet static word vectors for sentiment analysis can greatly help in improving the resulting accuracy for a sentiment analysis model.

VI. Conclusion

In this paper, we analyze patient sentiments using their reviews of doctors and predict doctors' ratings for various categories such as 'Knowledge', 'Staff' and 'Helpfulness'. For this we successfully demonstrate how convolutional neural networks can be used effectively for sentiment based rating prediction for doctors using user reviews. In the process, we improve upon the performance of earlier works of Paul *et al*. [2] and achieve state-of-the-art accuracy for a very important problem for healthcare services, while keeping the model simple, robust and easily extendible. Our research conducts exhaustive and complete experiments accounting for many popular approaches for the purpose of sentiment analysis using CNN and compared them. We finally conclude that a Convolutional Neural Network with pre-trained word vectors trained using Adadelta based optimizer, categorical cross-entropy as a loss function and dropout technique as a regularizer achieves the best result for both binary and 5 point rating classification problems.

References

[1] Deloitte Center for Health Solutions. (2011). Survey of Health Care Consumers in the United States: Key Findings, Strategic Implications. [Online]. Available: http://www.deloitte.com/assets/Dcom-UnitedStates/Local%20Assets/Documents/US_CHS_2011ConsumerSurveyinUS_062111.pdf

[2] M. J. Paul, B. C. Wallace, and M. Dredze, "What affects patient (dis) satisfaction? Analyzing online doctor ratings with a joint topic-sentiment model," presented at AAAI Workshop on Expanding the Boundaries of Health Informatics Using AI (HIAI), June 2013.

[3] D. A. Hanauer, K. Zheng, D. C. Singer, A. Gebremariam, and M. M. Davis, "Public awareness, perception, and use of online physician rating sites," *JAMA*, vol. 311, no. 7, pp. 734-735, 2014.

[4] Fox, Susannah, and Maeve Duggan, "Health online 2013," 2013.

[5] B. Kadry, L. F. Chu, B. Kadry, D. Gammas, and A. Macario, "Analysis of 4999 online physician ratings indicates that most patients give physicians a favorable rating," *Journal of Medical Internet Research*, vol. 13, no. 4, 2011.

[6] G. Ganu, Y. Kakodkar, and A. Marian, "Improving the quality of predictions using textual information in online user reviews," *Information Systems*, vol. 38, no. 1, pp. 1-15, 2013.

[7] A. Rios and R. Kavuluru, *Convolutional Neural Networks for Biomedical Text Classification: Application in Indexing Biomedical Articles*.
Y. Kim, "Convolutional neural networks for sentence classification," arXiv preprint arXiv:1408.5882, 2014.

[8] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuglu, and P. Kuksa, "Natural Language Processing (Almost) from Scratch," *Journal of Machine Learning Research*, vol. 12, pp. 2493–2537, 2011.

[9] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[10] M. M. Galizzi, M. Miraldo, C. Stavropoulou, M. Desai, W. Jayatunga, M. Joshi, and S. Parikh, "Who is more likely to use doctor-rating websites, and why? A cross-sectional study in London," *BMJ Open*, vol. 2, no. 6, e001493, 2012.

[11] L. Andrea, *et al*., "What patients say about their doctors online: A qualitative content analysis," *Journal of General Internal Medicine*, vol. 27, no. 6, pp. 685-692, 2012.

[12] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," arXiv preprint arXiv: 1404.2188, 2014.

[13] X. Zhang and Y. LeCun, "Text understanding from scratch," arXiv preprint arXiv: 1502.01710, 2015.

[14] J. Turian, L. Ratinov, and Y. Bengio, "Word representations: A simple and general method for semi-supervised learning," in *Proc. the 48th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, July 2010, pp. 384-394.

[15] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," *Proceedings of the Empiricial Methods in Natural Language Processing*, vol. 12, pp. 1532-1543, 2014.

[16] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in Neural Information Processing Systems*, pp. 3111-3119, 2013.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, pp. 1097-1105, 2012.

[18] Convolutional neural networks for visual recognition. [Online]. Available: http://cs231n.stanford.edu/Convolutional Neural Networks for Visual Recognition

[19] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural Networks for Machine Learning*, vol. 4, 2012.

[20] L. Bottou, "Stochastic gradient descent tricks," *Neural Networks: Tricks of the Trade*, Springer Berlin Heidelberg, pp. 421-436, 2012.

[21] M. D. Zeiler, "ADADELTA: An adaptive learning rate method," arXiv preprint arXiv: 1212.5701, 2012.

[22] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. International Conference on Artificial Intelligence and Statistics*, pp. 315–323

[23] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. M ̈uller, "Efficient backprop," *LNCS*, vol. 1524, pp. 9-48, 1998.

[24] Keras: Deep Learning library for Theano and TensorFlow. [Online]. Available: https://github.com/fchollet/keras/blob/master/README.md

Ranti D. Sharma was born in Amritsar on Jan. 1, 1991. Mr. Sharma completed his bachelor of technology degree in computer science at Indian Institute of Technology, Guwahati in July 2014. He completed his internship as a software developer in Microsoft Research Hyderabad. He currently works as a software developer in Microsoft Research Hyderabad. His interests include machine learning, especially applications of deep and convolutional neural networks along with natural language processing.

Samarth Tripathi was born in Allahabad on February 11, 1992. Mr. Tripathi completed his bachelor of technology degree in computer science at Indian Institute of Technology, Guwahati in July 2014. He completed his internship as a software developer in Capital Dynamics Singapore. He currently works as a senior software developer in Samsung Research Bangalore. His interests include machine learning, especially applications of recursive neural networks and its applications in natural language processing.

Sunil K. Sahu was born in Bilaspur on Jan. 26, 1987. Mr. Sahu completed his bachelor of technology degree in Guru Ghasidas Vishwavidyalaya Bilaspur . He is currently pursuing his Ph.D. in computer science at Indian Institute of Technology Guwahati under Dr. Ashish Anand. His interests include machine learning and data analysis.

Sudhanshu Mittal was born in Jaipur on February 6, 1992. Mr. Mittal completed his bachelor of technology degree in computer science at Indian Institute of Technology, Guwahati in July 2014. He completed his internship as a software developer in Arista Networks in Bangalore. He currently works as a software developer in oracle hyderabad. His interests include machine learning, natural language processing and computational biology.

Ashish Anand is an assistant professor with the Department of Computer Science at Indian Institute of Technology, Guwahati. He did his masters (Int-MSc, 5 years) in mathematics and scientific computing from Indian Institute of Technology Kanpur. Thereafter, he joined Androgen Receptor Laboratory (University of Helsinki, Finland) as a visiting research student. Later he joined a collaborative project of Prof. Pradip Sinha and Prof. K Deb (at IIT Kanpur) to understand neoplastic cancer in the model organism D. Melanagastor. He did his PhD from Nanyang Technological University on "Computational Intelligence methods for problems in computational biology". In particular he worked on multi-class classification, template clustering for short time series data and imbalanced binary classification problems. And prior to joining IIT G, he was a part of European Consortium, BaSySBio at Systems Biology Lab (Group Leader: Dr Benno Schwikowski) Institut Pasteur, Paris. His post-doc work was mainly concentrated on regulatory network reconstruction and pathway analysis..