Quantify customer perception using natural language reviews

Amit Garg
Stanford University
amit93@stanford.edu

Rahul Venkatraj
Stanford University
vrahul@stanford.edu

Abstract

This project focuses on the multi-label classification of product (BeerAdvocate dataset) and service (Yelp restaurants) reviews. We found Recurrent Neural Networks (RNN) with Long Short Term Memory (LSTM) and Convolutional Neural Networks (CNN) to be the most suitable Neural Network architecture for unstructured natural language content like reviews. We achieved validation accuracies of 56-58% with LSTM and CNN, against best-performing baselines 55% (from Support Vector Machines) and 47% (from the Random Forest approach). This is in line with similar best in class performance by neural networks in the RNN [5],[6] and CNN [13],[14] families. The typical skew towards higher scores in reviews has been removed, and equal numbers of reviews corresponding to each label were considered. While our architecture works far superiorly on binary classification tasks (82% accuracy), we have focused on 5-label classification since that is more representative of the actual application we target (predicting quantified summary of reviews).

1 Introduction

A marketing manager needs to have a constant pulse of the market's perception of his product or service. Consumer product companies receive human language reviews and comments about their product from surveys, customer complaints, store feedback and more. We have created neural network models to help a marketing manager condense this overwhelming natural language content, into simple numerical scores.

We formulated this as a multi-class labeling problem. We used the BeerAdvocate [19],[20] dataset, and the Yelp Academic Dataset [21] for training our model. These are 10,000 - 20,000 human language reviews, each of which contain 100-500 words of text and a corresponding score (ranging from 1 to 5) given by the reviewer. Some sample reviews are shown in Table 1 and 2. We used the Python based 'Theano' library [22],[24] to build our neural networks.

2 Background Research

The most common approach for Sentiment Analysis is the Bag of Words approach [1]. Other methods include modeling the sentiment compositionality through feature engineering like contextual valence shifters[2] and propagation of polarity for two sentence components [3]. Another approach is the successive application of Word Sense Disambiguation, Sense Level Polarity Assessment, training of Hidden Markov Models and then Sentence Level Polarity Detection [4].

More recent neural networks for this task have been Recurrent Neural Networks built on the LSTM architecture [5], [6]. The most basic Simple Recurrent Neural Network [7] was used a starting point to address sequence-dependent tasks like machine translation [8] and language modeling[9]. However, the practical difficulty with using RNNs for natural language tasks is that the gradients

Table 1: Yelp dataset - Review samples with their respective scores

Label	Sample Review Extract
5	... will travel the extra miles because the service and food at this location is the best...
4	... everyday, well prepared and taste bud pleasing home style cooking ...
3	... better than average, but I don't like seeing all the sauce resting at the bottom ...
2	... We will not be back . The iced tea is also terrible tasting ...
1	... worst pizza I've ever had What a mistake. I will never order from them again! ...

Table 2: Beeradvocate dataset - Review samples with their respective scores

Label	Sample Review Extract
5	... OUTSTANDING! Quite possibly one of the finest Rauch beers I have had ...
4	... I think it works well. Note it's not cloying, the sweetness is enjoyable ...
3	... I wish the flavor bore out the strength of the aroma, but still not a bad beer ...
2	... it's not something I would seek out again, but it doesn't fail ...
1	... yellow, fizzy, meant for washing dirt out of your mouth after mowing the lawn ...

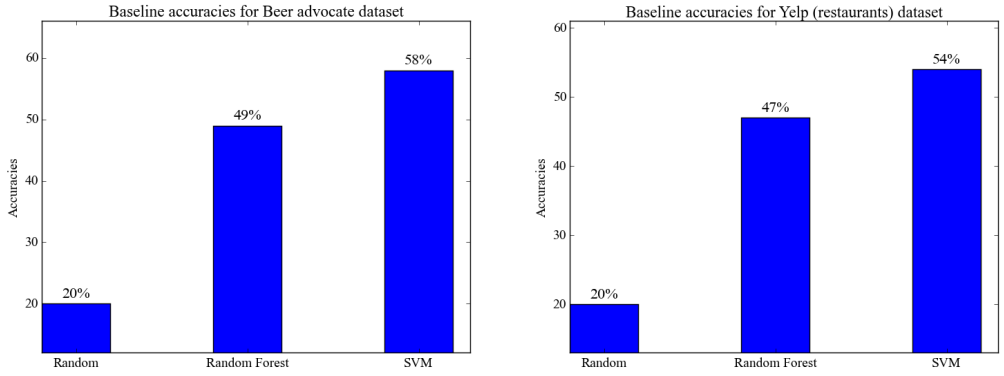
from the objective function vanish after a few steps [10]. The LSTM architecture is observed to overcome this problem [11] for natural language tasks including sentiment analysis. Studies targeted at a similar objective as ours, have successfully used LSTMs [5],[6] to obtain accuracies of 48-50% (on 5-label classification tasks). Similar to Socher et. al. [13], we have used cross-entropy error as the overall objective function to minimize.

Convolutional Neural Networks (CNNs) are another widely used architecture for sentiment analysis on unstructured text, since this induces a feature graph over the input sentence. Experiments towards a similar objective as ours, have led to accuracies of 48-50% on similar sentiment classification tasks [13], [14] in the movie reviews dataset.

In recent times, recursive neural networks [12] have delivered distinctly better performance on sentiment analysis tasks, thanks to the tree structure being processed. For example, multi-class labeling accuracy, when measured at each tree level, is 81%. The overall sentence accuracy (measured at the root node level) is 46%. We have not taken this approach since we wanted the neural network to process unstructured text in the natural form.

3 Experiments

3.1 Baselines



(a) Beeradvocate dataset

(b) Yelp dataset

Figure 1: Baseline Results

To verify the usefulness of the data and target a baseline to beat, we created two classifiers based on the Random Forest and Support Vector Machine (SVM) techniques. We used 12,500 reviews from each of the two datasets to create this baseline. The results are shown in Figure 1.

We created three types of Neural Networks (with increasing levels of complexity) - the Simple Neural Network, a Recurrent Neural Network with the LSTM architecture and a Convolutional Neural Network.

3.2 Simple Neural Network

We implemented a one hidden layer neural network as a starting point.

$$h = \sigma\left(xW_1 + b_1\right)$$
$$\hat{y} = \text{softmax}\left(hW_2 + b_2\right)$$

There was very little improvement observed upon varying the size of hidden dimension. The results are shown in Figure 2.

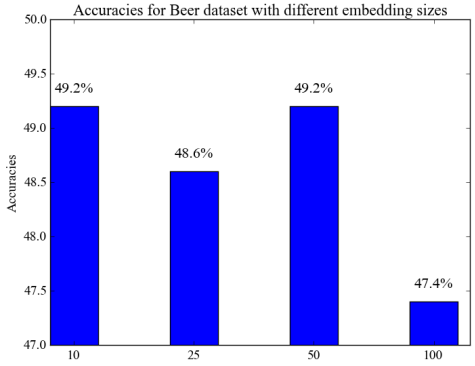
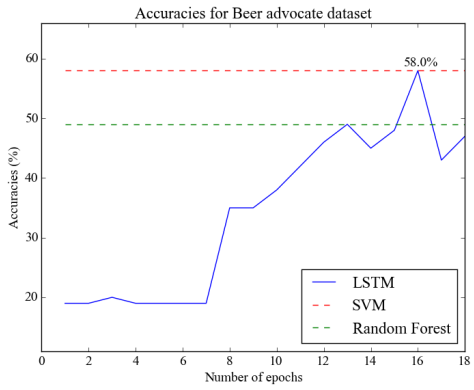


Figure 2: SRN - Dependence on embedding size

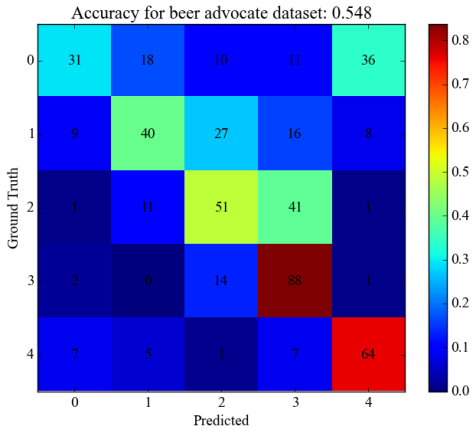
3.3 Long short term memory

Since we were working with unstructured text and had to incorporate contextual information from the 'recent past' of the sentence, we decided to create a Recurrent Neural Network (RNN) using the Long Short Term Memory (LSTM) architecture.

$$i_t = \tanh\left(W^{(i)}x_t + U^{(i)}h_{t-1}\right) :: \text{Input Gate}$$
$$f_t = \tanh\left(W^{(f)}x_t + U^{(f)}h_{t-1}\right) :: \text{Forget gate}$$
$$o_t = \tanh\left(W^{(o)}x_t + U^{(o)}h_{t-1}\right) :: \text{Output}$$
$$\tilde{c}_t = \tanh\left(W^{(c)}x_t + U^{(c)}h_{t-1}\right) :: \text{New memory}$$
$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t :: \text{Final memory}$$
$$h_t = o_t \circ \tanh\left(c_t\right) :: \text{Final hidden state}$$
$$\hat{y}_t = \text{softmax}\left(Uh_t + b\right) :: \text{Prediction}$$



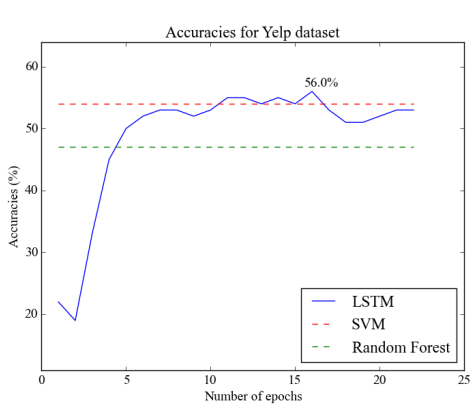
(a) Validation accuracy compared to baseline



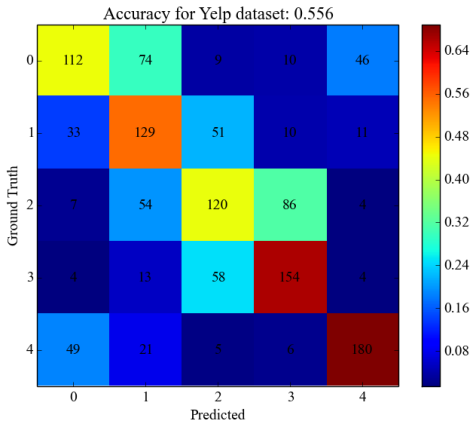
(b) Confusion matrix

Figure 3: Beer Advocate Dataset - best performing LSTM results

We again represent every unique word by a unique integer ID (about 10,000-12,000 unique words). We got the best performance using an embedding layer of dimension 25 on the Yelp dataset, and dimension 50 on the Beeradvocate dataset. The results are shown in Figure 3 and 4.



(a) Validation accuracy compared to baseline

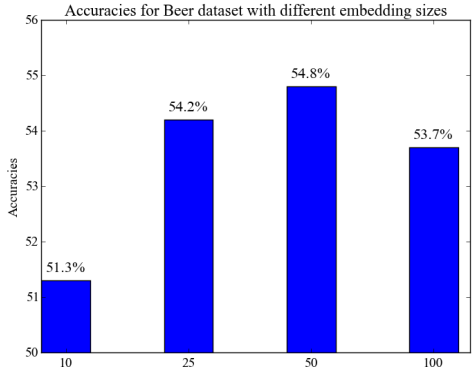


(b) Confusion matrix

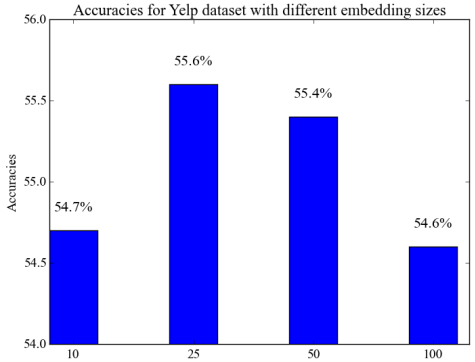
Figure 4: Yelp Dataset - best performing LSTM results

3.3.1 Effect of Embedding Dimension

Embedding dimension represents how complex a word's representation would be. Every word is represented by a 'd' dimensional vector, if the embedding dimension is 'd'. This is another parameter we used to ensure that the model is trained in a general manner. Since our tasks contain 10,000-12,000 distinct words, we tried embedding dimensions between 10 to 100. The results are shown in Figure 5.



(a) Beeradvocate dataset



(b) Yelp dataset

Figure 5: Effect of Embedding dimension on Validation accuracy

3.3.2 Effect of Dropout

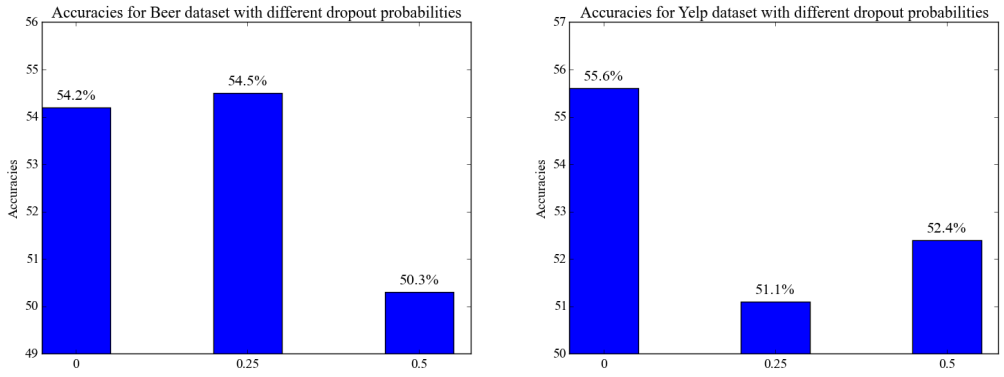
We found that the 'Dropout' mechanism [15] did not work well with the LSTM. "The reason might be that dropout corrupted its memory, thus making training more difficult" [5]. The results are shown in Figure 6.

3.3.3 Effect of Regularization

Regularization helps the neural network learn a generalized model, and not a model very specific to the training data. The regularization parameter controls how much of this effect needs to be imposed. The results are shown in Figure 7.

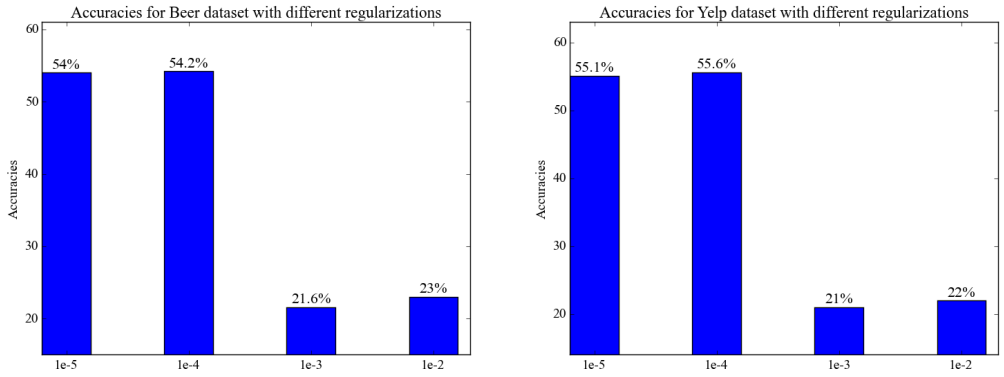
3.3.4 Effect of output node

We created two configurations of output from the LSTM architecture - one with output averaged from all the nodes, and another with output from only the final node. We observed that the LSTM with output averaged from all the nodes performs marginally better (minor improvement of 0.1%).



(a) Beeradvocate dataset (b) Yelp dataset

Figure 6: Effect of Dropout on Validation accuracy



(a) Beeradvocate dataset (b) Yelp dataset

Figure 7: Effect of Regularization on Validation accuracy

3.3.5 Effect of multiple layers

We added another hidden layer to the existing LSTM architecture described above, and found that it does not lead to an improvement in validation accuracy.

3.3.6 Effect of bidirectional architecture

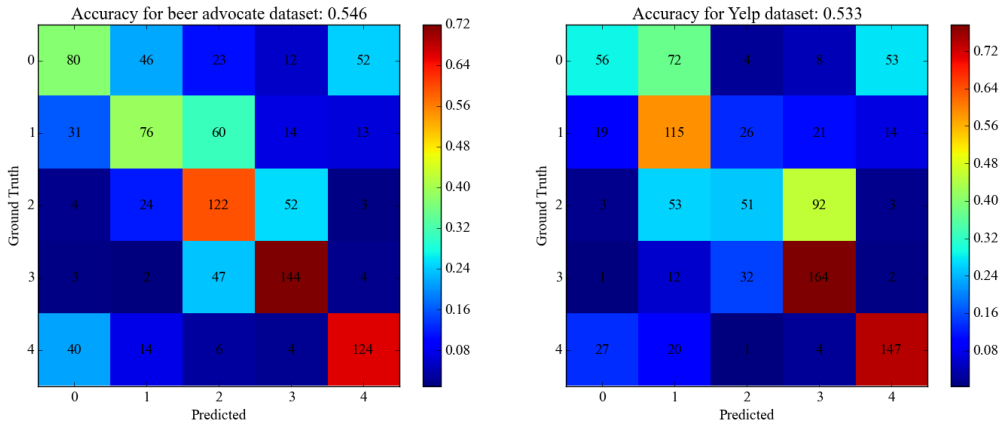
The LSTM architecture described above was modified such that each review is read both forwards and backwards, and the output vectors of both these are pooled towards the final output. Once again, we found that this does not lead to an improvement in validation accuracy.

3.3.7 Effect of regression architecture

To completely eliminate the mis-classification of a small number of 'very negative' (label 0) reviews as 'very positive' (label 4), we implemented a linear regressor in place of the softmax layer. This attempts to capture the inter-relation between the labels ($0 < 1 < 2 < 3 < 4$), rather than consider them as discrete unrelated named labels. The negative entropy objective function was also changed to max-margin. However, there was no improvement in validation accuracy.

3.4 Convolutional Neural Networks

From our background research, we found the CNN to be useful for similar tasks with unstructured data. Since CNNs are commonly applied for computer vision applications, we modified our task to look like a computer vision task involving word vectors, instead of image representations.



(a) Beeradvocate dataset

(b) Yelp dataset

Figure 8: CNN - best results on the respective test datasets

We created a pre-processing layer that replaces every word in a sentence with the corresponding word vector, created by the word2vec [17] algorithm. This is then passed through a single CNN layer, with a filter size of 5, and batch pooling of every 2 words [CNN-poolingLayer]. This is then fed to a hidden layer and the following activation layer is a softmax function. We found that this architecture yields the best results. In addition, we also tried using double CNN layers, that is [CNN-poolingLayer-CNN-poolingLayer], which gave slightly better (about 1.5%) results.

We observe that the LSTM predicts some very negative review (label 0) as very positive (label 4) on the Beeradvocate dataset, and this problem is addressed partially by the CNN. This is due to the presence of locally acting negations (like 'not smoked' etc) that are not captured by either of these architectures.

3.5 Other Interesting Observations

In all our experiments above, we have removed the inherent skew towards higher ratings in review datasets. However, if the dataset is used a it is (with the skew towards high ratings), we observe validation accuracies of 65-68% compared to the highest SVM baselines of 60-64%.

Also, training a word2vec [17] model on the Yelp reviews dataset yields many interesting insights by itself. Sorting words by their cosine distance to the keywords gives a intuitive sense of what aspects of a restaurant influence a customer. For example, the 20 words closest to 'Pizza' include 'crust', 'pepperoni', 'knots', 'calzone' and 'wings' - suggesting that these are aspects on which a customer's mind forms an impression of a pizza. Similarly, the 20 words closest to 'Burger' include 'patty', 'fries', 'bun', 'juicy' which are very clearly the most expected features of a burger. At a higher level, the 20 words closest to 'Ambience' include 'decor', 'service', 'interior', 'setting', 'relaxing', 'cozy' and 'upscale', suggesting to a manager that these are some aspects his restaurant should focus on conveying.

4 Future Work

As a next step, the program could identify which keywords contribute the most towards a high review score. These keywords would give the restaurant owner a good idea of what facilities / features / food options create a perception of 'high review score' in the customer's mind. This can be used to improve the restaurant's service.

To take it the next level, we could implement a Named Entity Recognition module to identify specific brands in the reviews, and rank them based on the potential review scores.

A couple of other modifications to the existing system could be attempted. A new architecture could be created such that the CNN output feeds into the LSTM, thereby extracting the benefits of both a built-up tree structure and the memory nature of LSTM. Also, a model trained on one dataset, could

be tested on a slightly different dataset. For example, a model trained on restaurant reviews could be tested on a bars review dataset.

5 Conclusion

In this project, we create an LSTM architecture to capture history and long distance interplays through gated memory nodes. Our architecture's performance (56 - 58%) is in line with state of the art [5],[6],[12],[13],[14] for a similar task of fine-grained multi-label classification. For a dataset of 10,000 - 12,000 reviews (and 10,000 - 12,000 unique words), we find that an embedding dimension of 25-50, with a regularization parameter of 0.0001 delivers best validation accuracies. Dropout seems to give only a minor improvement of 0.3% with the Beeradvocate dataset (with probability of dropout 0.25), and does not improve accuracy with the Yelp dataset. On CNNs, we observe that additional an additional CNN layer improves validation accuracy by 1.5%.

References

[1] B. Pang and L. Lee. 2008. Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval, 2(1-2):1135.

[2] L. Polanyi and A. Zaenen. 2006. Contextual valence shifters. In W. Bruce Croft, James Shanahan, Yan Qu, and Janyce Wiebe, editors, Computing Attitude and Affect in Text: Theory and Applications, volume 20 of The Information Retrieval Series, chapter 1.

[3] K. Moilanen and S. Pulman. 2007. Sentiment composition. In In Proceedings of Recent Advances in Natural Language Processing

[4] V. Rentoumi, S. Petrakis, M. Klenner, G. A. Vouros, and V. Karkaletsis. 2010. United we stand: Improving sentiment analysis by joining machine learning and rule based methods. In Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10), Valletta, Malta.

[5] Le, Phong, and Willem Zuidema. "Compositional Distributional Semantics with Long Short Term Memory." arXiv preprint arXiv:1503.02510 (2015).

[6] Zhu, Xiaodan, Parinaz Sobhani, and Hongyu Guo. "Long Short-Term Memory Over Tree Structures." arXiv preprint arXiv:1503.04881 (2015).

[7] Jeffrey L. Elman. 1990. Finding structure in time. Cognitive science, 14(2):179211.

[8] Ilya Sutskever, Oriol Vinyals, and Quoc VV Le. 2014. Sequence to sequence learning with neural networks. In Advances in Neural Information Processing Systems, pages 31043112.

[9] Tomas Mikolov, Martin Karafia t, Lukas Burget, Jan Cer- nocky', and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In INTERSPEECH, pages 10451048.

[10] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhu- ber. 2001. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In Kremer and Kolen, editors, A Field Guide to Dynamical Recurrent Neural Networks. IEEE Press.

[11] Felix Gers. 2001. Long short-term memory in recurrent neural networks. Unpublished PhD dissertation, E cole Polytechnique Fe de rale de Lausanne, Lausanne, Switzerland.

[12] Socher, Richard, Perelygin, Alex, Wu, Jean Y., Chuang, Jason, Manning, Christopher D., Ng, Andrew Y., and Potts, Christopher. Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '13, Seattle, USA, 2013. Association for Computational Linguistics.

[13] Kalchbrenner, Nal, Edward Grefenstette, and Phil Blunsom. "A convolutional neural network for modelling sentences." arXiv preprint arXiv:1404.2188 (2014).

[14] Kim, Yoon. "Convolutional neural networks for sentence classification." arXiv preprint arXiv:1408.5882 (2014).

[15] Srivastava, Nitish, et al. "Dropout: A simple way to prevent neural networks from overfitting." The Journal of Machine Learning Research 15.1 (2014): 1929-1958.

[16] Image-based recommendations on styles and substitutes, J. McAuley, C. Targett, J. Shi, A. van den Hengel, SIGIR, 2015

[17] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. In Proceedings of Workshop at ICLR, 2013

[18] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. In Proceedings of NIPS, 2013

[19] J. McAuley, J. Leskovec, and D. Jurafsky. Learning attitudes and attributes from multi-aspect reviews.ICDM, 2012.

[20] J. McAuley and J. Leskovec. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. WWW, 2013

[21] Yelp Dataset Challenge. Retrieved from https://www.yelp.com/academic_dataset (2014)

[22] http://deeplearning.net/tutorial/lstm.html

[23] Graves, Alex. Supervised sequence labelling with recurrent neural networks. Vol. 385. Springer, 2012.

[24] Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. Neural computation, 12(10), 2451-2471.

[25] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.