Single and multibranch CNN-Bidirectional LSTM for IMDb Sentiment Analysis

Chary Vielma#1, Abhishek Verma*2, Doina Bein#3

#*Department of Computer Science, California State University, Fullerton*
Fullerton, CA
1chary.vielma@csu.fullerton.edu
3dbein@fullerton.edu

**Department of Computer Science, New Jersey City University*
Jersey City, NJ
2averma@njcu.edu

***Abstract*— Online users now frequently use the internet to voice opinions, ask for advice, or choose products and services based on the feedback of others. This provides a window into the way users feel about specific topics. The study of natural language processing, a sub-category of sentiment analysis, takes on this task by extracting meaning out of user text through observing the way in which words are grouped and used. Machine learning techniques have made significant advances which allow us to further explore mechanisms for interpreting such data. This research aims to use the internet movie database (IMDb) dataset and the Keras API to compare single and multibranch CNN-Bidirectional LSTMs of various kernel sizes [24][32]. The results show that while only time to train varies between single and multibranch models, their maximum accuracies are close in range. The highest accuracy model was the single branch with kernel size 9 with an accuracy of 89.54%. While slightly more accurate than the multibranch model with 88.94%, the time savings for the single branch is approximately 2 hours and 20 minutes.**

***Keywords*— CNN; LSTM; Bidirectional LSTM; Sentiment analysis; Natural Language Processing**

I. Introduction

Opinion-based online user text continues to grow as more people turn to the internet for everything from food recommendations to what kind of car to buy. With the accumulation of user text across all areas of interest and the advances in the study of neural networks, there is now the opportunity to interpret user text in such a way that we can make sense of. This information can reveal things such as shopping patterns, preferences, likes/dislikes, behavioural tendencies, and personal opinions on specific topics [31]. As the advances in the field of neural works progress, these projections become higher in accuracy. The study of Natural Language Processing (NLP) as explained in [7] can be used in conjunction with neural networks to search for these patterns in user text. The goal of this study aims to develop a model that can determine whether user text harbors positive or negative feelings towards a topic.

The process of categorizing user text as being either generally positive or negative is known as sentiment analysis. To classify an opinion, sentiment analysis classification can be considered at the document level; meaning one entire document maps to one opinion [28].

In our proposed research, we explore two architecture designs to build a machine learning model that can successfully categorize user text as being either positive or negative in nature. We use convolutional neural networks (CNN) to learn the meaning of words based on word associations. This has thus far provided successful results when analyzing things such as video and photo recognition. This is mainly due to the nature of the data in where order of pixels or frames for instance, makes a significant difference [3][21]. Pixels in an image can be broken down into smaller grids. Relationships between pixels can be observed during this process based on their position to one another in the image.

An additional bidirectional long short-term memory (LSTM) network is used to strengthen the meaning between words that are closer in proximity to one another. This also strengthens the model's understanding of context of a word and has been used in other image recognition works such as that presented in [12] and in [22]. The gates internal to an LSTM unit control the data that passed through its current state [10]. The bidirectional mechanism provides data propagation to previous and future states.

We use single and multibranch architectures to compare accuracies and total training time. The [24] movie review dataset is used to train and test our models.

The remainder of this report is divided into five sections. Section 2 provides a brief overview of the current state of the field. Section 3 describes the dataset used to conduct the research experiments. Section 4 presents the research methodology used in our experiments. Section 5 summarizes and interprets the findings of the single and multibranch tests. Section 6 concludes the research paper.

II. Background and Related Work

Neural network techniques can be applied to almost any instance where a pattern can be observed. While some reasons to analyze user text have an end goal to better-target an audience to sell products or services, it can also be used to make observations on social media such as in [26], In [2], they used a type of CNN and regression algorithms to analyze user profiles online and produce a personality score based on this information. The work presented in [8] used neural networks to predict life events such as weddings, broken cell phones, or new jobs.

In the context of gauging user sentiments, CNNs have been the primary model used for text classification. In recent years, models with different features have been explored to achieve a higher accuracy. One such example is our research which uses a CNN layer which is then forward-propagated to a bidirectional LSTM layer. Both the bidirectional feature and LSTM unit introduce new behaviours into the model. Because an LSTM unit is useful in understanding word context and retaining associations internally, it produces better results used in conjunction with a CNN network over just the CNN alone.

A. Convolutional Neural Networks (CNN)

CNNs work well for sentiment analysis as they are dependable when it comes to feature extraction [28]. Many studies have performed experiments with CNNs which used the IMDb dataset such as in [24] where they used learning word vectors and a combination of supervised and unsupervised techniques to produce a new model. There are various works such as [16] and [13] which have used CNNs for sentence classification as well where instead of a document-level classification, the model learns only sentence-level associations. Furthermore, text classification can also be implemented at the character-level such as is presented in [27].

CNNs are made up of connected layers from one node to the next. These layers contain nodes which perform the convolution on the data. To train a CNN model, the output at each node is multiplied by some weight. The result is passed along to the next layer in the model. Biases present at each node are also calculated and applied using an activation function. Weights and biases are adjusted continuously as each layer produces outcomes which are compared to prediction values to check for accuracy. Essentially the model is fine-tuning itself to achieve the maximum accuracy possible.

Our training and validation IMDb dataset contains labelled data meaning we provide the model with examples for it to learn context and use [24]. This constitutes supervised learning which allows the model to categorize unknown samples based on features it has learned from the training samples. We use a CNN layer at the start of our model to create connected layers that eventually reach a bidirectional LSTM layer.

B. Long Short-Term Memory (LSTM) Units

The LSTM network was first introduced in [10] and have been used significantly such as in [30] and [5] for sentiment analysis due to the need to interpret words in to hold different meaning depending on the situation and its use. For this, the model would have to look at the words that came before and after it, and the ones before and after that, etc. Due to its design, an LSTM unit can remember long-term dependencies through its internal gates which control the decision process to add or delete values in the cell. The gates can also scale values and decide how much of the internal information to share with other nodes [10].

This architecture has also been used in the surveillance field as described in [12]. In their research they used a combination of CNN and LSTM to produce a model that could detect anomalies in video feed which could be used to warn of intruders in home surveillance equipment.

III. Dataset Description

The dataset used for this study is from the Association for Computational Linguistics which is comprised of 100,000 text movie reviews from the IMDb website [24]. After users watch movies at the theaters or in their homes, some eventually make their way to the IMDb website to voice their opinions. This makes for a useful source of raw and honest opinions perfect for sentiment analysis.

The movie reviewers write a text review and have the option of rating the movie on a scale of 1 to 10 stars. The dataset considers reviews with stars between 7 and 10 as positive and 1 and 4 as negative. Neutral reviews between 5 and 6 are omitted. It also caps the maximum number of reviews for a movie at 30 reviews. The reviews contain on average 234.76 words and a 172.91-word standard deviation. Our experiments cap the maximum number of words for a review at 500 words.

The [24] dataset reviews are evenly partitioned into labelled and unlabelled data. The labelled reviews are also evenly divided and tagged with 1s or 0s to denote a positive or negative sentiment. The Keras API for Tensorflow performs pre-processing on the dataset to produce a sorted list of *maximum dictionary length* selected by the implementer [32][30]. Word indices are sorted by frequency count. The *maximum sequence length* caps the number of words in a review. When reviews are less than the predetermined sequence length, the dictionary is padded with zeros to compensate. The *vector length* is the dimension of the vector used for word embeddings. Our experiment uses lengths 5,000, 500, and 32 for *dictionary length*, *sequence length*, and *vector length* respectively.

IV. Methodology

C. *Hardware and Software*

Experiments were conducted on an Ubuntu 16.04 server. An Intel Xeon E5-2630 with a 2.2 GHz CPU and a GTX 1080 Ti graphics card were used. The Keras API for TensorFlow and python v2.7 were used to train the models. Keras v2.0 with TensorFlow v1.0.1 were selected. These versions are required for library compatibility.

D. *Architecture Design*

The work presented in [31] was used as inspiration for these experiments given their research studied CNN-LSTM multibranch models as well as the work presented in [5]. The models used in this research use a 1-dimensional CNN layer followed by a bidirectional LSTM layer. Four of our experiments are single branch and have varied kernel sizes of 3, 5, 7, or 9 words. The fifth experiment is multibranched and combines the concept of the first four models to produce a 4-branch model with kernel sizes of 3, 5, 7, and 9 words.

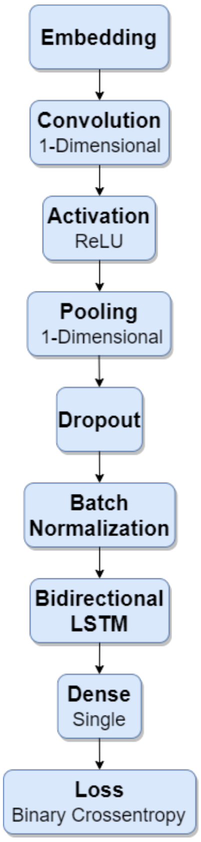

Fig. 1 The layer diagram is the same for both single and multibranch models. The difference is in the number of branches used between layers 2-7.

1) *Convolution*: An embedding layer produces a tensor which is passed to the one-dimensional convolutional layer to begin studying word

associations of the kernel size. For the multibranch model, the shape is equal to (kernel size * embedding vector size). In this study, the embedding vector size is 32 and the number of convolutional filters is 128 units for all models.

2) *Activation*: This layer takes the output of the convolutional layer and adds a rectified linear unit (ReLU) activator. This will introduce bias into the network by transforming the inputs using a linear function.
3) *Max Pooling*: Max pooling allows branches to remain scaled down to workable sizes and ranges to alleviate overfitting.
4) *Branch Dropout*: This layer takes random inputs and replaces them with zeros. Due to random selection, it reduces the possibility of memorizing data. Our models include a branch dropout of 0.4 after the max pooling layer.
5) *Batch Normalization:* This layer normalizes all inputs which in turn scales down the covariate shift in the hidden layers.
6) *Bidirectional Long Short-Term Memory:* This layer is comprised of the bidirectional mechanism and the long short-term memory state. The bidirectional mechanism allows each state to share data forward and backward to previous and future states. The long short-term memory unit consists of input, output, and forget states [22]. Together, these gates manage the data that enters and leaves the cell. The cell can retain meaningful information if its newer in the sequence [10].
7) *Concatenation:* This layer concatenates all the branches, if more than one, into one tensor to reproduce the same shape as the initial input layer. This layer is not used for single branch models.
8) *Dense:* The dense layer multiplies the input and a weight matrix to introduce weights to the model.
9) *Loss Function and Optimizer:* A binary cross-entropy loss function is used to calculate the loss and perform a summation on the dense layer.

In addition, the models were trained with the RMSprop optimizer and learning rate of 0.01. The learning rate decay used was 0.1.

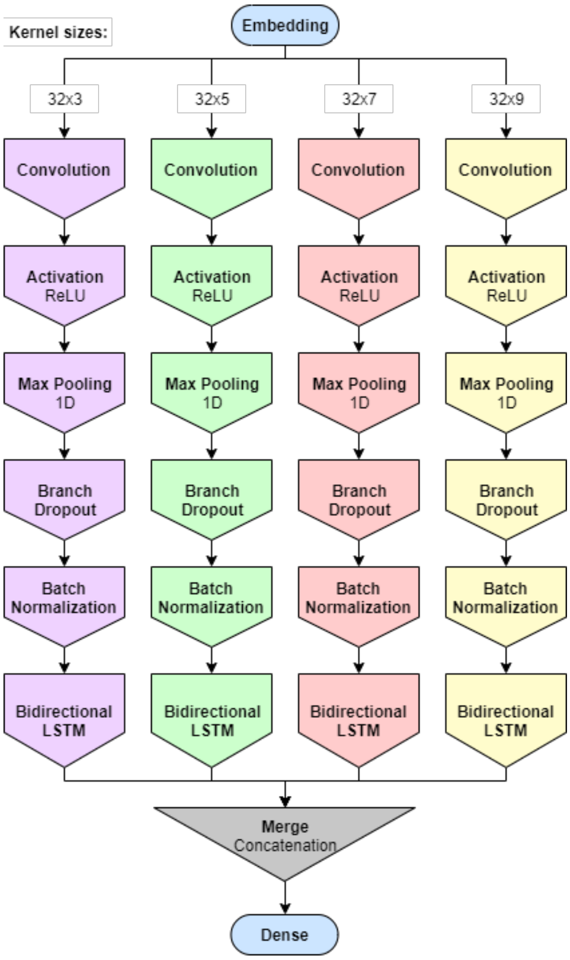

Fig. 2 Multibranch CNN-Bidirectional LSTM diagram

Table I shows how each model was set up for the experiments. Models 1 through 4 are single branch and only vary in the kernel size parameter. The multibranch model has four identical branches that vary as well in their respective kernel size. This allows the model to learn word association by grouping consecutive words and understanding the context in which the words are used. Specifically, how meaning of words change depending on their placement in a sentence.

TABLE I
CNN-BIDIRECTIONAL LSTM MODEL PARAMETERS

Proposed Models	**Model 1**	**Model 2**	**Model 3**	**Model 4**	**Model 5**
Branches / Kernel Sizes	3	5	7	9	3/5/7/9
Convolution Filters	128	128	128	128	128
Kernel Regularizer	L2 (0.01)	L2 (0.01)	L2 (0.01)	L2 (0.01)	L2 (0.01)
Activation Type	ReLU	ReLU	ReLU	ReLU	ReLU
Max Pool Size	2	2	2	2	2
Branch Dropout	0.4	0.4	0.4	0.4	0.4
Batch Normalization	Yes	Yes	Yes	Yes	Yes
Type - Units	Bidir. LSTM (128)	Bidir. LSTM (128)	Bidir. LSTM (128)	Bidir. LSTM (128)	Bidir. LSTM (128)
Optimizer Type	RMS prop	RMS prop	RMS prop	RMS prop	RMS prop
Learning Rate	0.01	0.01	0.01	0.01	0.01
Learning Rate Decay	0.1	0.1	0.1	0.1	0.1
Accuracy (Maximum)	88.90	88.95	89.44	89.54	88.94

V. DISCUSSION

Table II shown below summarizes the models presented in this study. Models 1 through 4 are the single branch CNN-Bidirectional LSTMs of various kernel sizes. Model 5 is the multibranch CNN-Bidirectional LSTM of kernel sizes 3, 5, 7, and 9 words for the four branches.

TABLE II
ACCURACY AND TIME SUMMARY

Model	**Branches / Kernel Sizes**	**Time** (hours : mins)	**Maximum Accuracy (%)**
Model 1	3	1:17	88.90
Model 2	5	1:17	88.95
Model 3	7	1:20	89.44
Model 4	9	1:24	89.54
Model 5	3/5/7/9	3:37	88.94

Although models 1 through 4 analyzed word associations as small as 3 words and as large as 9 words, it did not significantly alter the amount of time it took to train each model. Model 1 and model 4 differed by only seven minutes in total. It is also worth noting that although the accuracies of model 1 and 4 only varied by 0.64%, model 4 had the slightly higher accuracy of 89.54% perhaps since it examined 9 consecutive words when analyzing word associations. Model 1 had an accuracy of 88.9% and analyzed 3-word associations at a time.

Model 5 took 3-word, 5-word, 7-word, and 9-word associations which were dispersed over four branches and concatenated before training. One would image that training a model with multiple branches might result in a higher overall accuracy however in this case, the maximum accuracy of model 5 was only 88.94%. This is slightly lower than the single branch maximum 89.54% in model 4. It is possible that concatenating various branches could simply result in an overall average of their respective accuracies. The work presented in [31] which used multibranch CNN-LSTM models yielded a result of 89.5%. Interestingly, a single branch exploring 9-word associations performed the same as the multibranch model in [31]. Perhaps the backward propagation in model 4 helped to achieve this accuracy without the extra hours required in training multibranch models.

While each branch had a branch dropout of 0.4, the multibranch model could have benefited from an additional dropout layer after the dense layer concatenates all branches in the network. This would have caused a higher number of neurons to be ignored on the forward pass, reduced model sensitivity, and possibly increased accuracy. Further testing should be conducted on this theory.

VI. CONCLUSION

The research outlined in this report explored single and multibranch CNN-Bidirectional LSTMs. While there are various studies using CNNs and LSTMs, this research sought out to incorporate a bidirectional mechanism to introduce forward and backward propagation. The IMDb dataset was used to train and validate various models of different kernel sizes. Our chosen dictionary, sequence, and embedding vector lengths were 5,000, 500, and 32 words respectively. The outcome showed that while single branch models are similar in runtime and accuracy, a combination of their kernel sizes to make one multibranch model did not improve accuracy. Instead, the model was 0.6% less accurate than the best-performing single branch model. The multibranch did not include a second dropout layer which may have affected the overall accuracy. The models presented in this research serve to advance our understanding of recurrent neural networks in the context of sentiment analysis and text classification within single and multibranch architectures.

References

[1] L. Chen, C. Liu, and H. Chiu, "A neural network based approach for sentiment classification in the blogosphere," *Journal of Informetrics*, vol. 5, no. 2, pp. 313-322, 2011.

[2] D. Xue et al., “Deep learning-based personality recognition from text posts of online social networks,” Appl. Intell., vol. 48, no. 11, pp. 4232–4246, 2018.

[3] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman, “Video summarization with long short-term memory,” *In European Conference on Computer Vision*, pp. 766–782, Springer, 2016.

[4] A. Tripathy, A. Agrawal, and S. Kumar Rath, "Classification of sentiment reviews using n-gram machine learning approach." *Expert Systems with Applications*, vol. 57, pp. 117-126, 2016.

[5] L. Rahman, N. Mohammed, and A. Kalam Al Azad, "A new LSTM model by introducing biological cell state." *In Electrical Engineering and Information Communication Technology (ICEEICT), 2016 3rd International Conference on*, pp. 1-6, 2016.

[6] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," arXiv preprint arXiv:1607.04606, 2016.

[7] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," *In Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, vol. 10, pp. 79-86, 2002.

[8] M. Khodabakhsh, M. Kahani, and E. Bagheri, “Predicting future personal life events on twitter via recurrent neural networks,” J. Intell. Inf. Syst., 2018.

[9] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, (2014). "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation". arXiv:1406.1078

[10] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[11] J. Mir and M. Usman, "An effective model for aspect based opinion mining for social reviews," *2015 Tenth International Conference on Digital Information Management (ICDIM)*, Jeju, pp. 49-56, 2015.

[12] J. R. Medel and A. Savakis, “Anomaly detection in video using predictive convolutional long short-term x networks,” arXiv preprint arXiv:1612.00390.

[13] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent Convolutional Neural Networks for Text Classification," *In AAAI*, vol. 333, pp. 2267-2273, 2015.

[14] S. Li, S. Yat M. Lee, Y. Chen, C. Huang, and G. Zhou. 2010. Sentiment classification and polarity shifting. In Proceedings of the 23rd International Conference on Computational Linguistics (COLING '10). Association for Computational Linguistics, Stroudsburg, PA, USA, 635-643.

[15] H. Vo and A. Verma, "New Deep Neural Nets for Fine-Grained Diabetic Retinopathy Recognition on Hybrid Color Space," *the 12th IEEE International Symposium on Multimedia*, Dec. 11-13, 2016, San Jose, CA, USA.

[16] Y. Kim, "Convolutional neural networks for sentence classification," arXiv preprint arXiv:1408.5882, 2014.

[17] Y. Kim, Y. Jernite, D. Sontag, and A.M. Rush, "Character-aware neural language models," *In Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[18] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality." *In Advances in neural information processing systems*, pp. 3111-3119, 2013.

[19] H. Al-Barazanchi, H. Qassim, and A. Verma, "Novel CNN Architecture with Residual Learning and Deep Supervision for Large-Scale Scene Image Categorization," *the 7th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, Oct. 20-22, 2016, New York, NY, USA.

[20] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, 1998.

[21] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions." *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-9, 2015.

[22] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556, 2014.

[23] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning,” arXiv preprint arXiv:1602.07261, 2016.

[24] A.L. Maas, R.E. Daly, P.T. Pham, D. Huang, A.Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis." *In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, vol.1, pp. 142-150, 2011.

[25] A. Verma and Y. Liu, "Hybrid Deep Learning Ensemble Model for Improved Large-Scale Car Recognition," *IEEE Smart World Congress*, San Francisco, CA, 2017.

[26] A. Severyn, and A. Moschitti, "Unitn: Training deep convolutional neural network for twitter sentiment classification," *In Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Association for Computational Linguistics*, Denver, Colorado, pp. 464-469. 2015.

[27] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification." *In Advances in neural information processing systems*, pp. 649-657, 2015.

[28] K. Kowalska, D. Cai, and S. Wade. "Sentiment Analysis Of Polish Texts" International Journal of Computer and Communication Engineering, vol. 1, pp. 39-42, May 2012R. Feldman, "Techniques and applications for sentiment analysis," *Communications of the ACM*, vol. 56, no. 4, pp. 82-89, 2013.

[29] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, R. Jozefowicz, Y. Jia, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, M. Schuster, R. Monga, S. Moore, D. Murray, C. Olah, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[30] A. Yenter and A. Verma, “Deep CNN-LSTM with combined kernels from multiple branches for IMDb review sentiment analysis,” 2017 IEEE 8th Annu. Ubiquitous Comput. Electron. Mob. Commun. Conf. UEMCON 2017, vol. 2018-Janua, pp. 540–546, 2018.

[31] J. Kang, H. S. Choi, and H. Lee, “Deep recurrent convolutional networks for inferring user interests from social media,” J. Intell. Inf. Syst., vol. 52, no. 1, pp. 191–209, 2019.

[32] F. Chollet and others, Keras. 2015 https://keras.io.