DETECTION OF FAKE ONLINE REVIEWS USING TENSORFLOW AND PROPOSING DEFENSIVE MECHANISMS

Syed Abrar Ahmed
abrar.syed860@knights.ucf.edu

Sridevi Divya Krishna Devisetty
divyakrishna.devisetty@knights.ucf.edu

Abstract — In recent times online product reviews are increasingly used by individuals and organizations for purchase and business decisions. With the advent of e-commerce, few retailers and Spammers, who are driven by the desire of profit, produce synthesized reviews to promote some products or demote competitors' products. There are even neural network models that came to existence recently, that can generate user-like reviews. These neural networks generated reviews or in general deceptive spam reviews of products or services that are harmful for customers in decision making and harm certain businesses. Existing approaches to detect deceptive spam reviews are concerned in feature designing. Hand-crafted features can show some linguistic phenomena, however can hardly reveal the latent semantic meaning of the review. We present a neural network-based model to learn the representation of reviews. Our model currently concentrates on general deceptive spam review detection. However, to detect neural network generated spam, we propose to use generative adversarial network(GAN) models. Our current model makes a hard attention through the composition from sentence representation into document representation. Specifically, we compute the importance weights of each sentence and incorporate them into the composition process of document representation. The results verify the effectiveness of our model by comparing with other neural network-based methods. As the feature selection is very important in this direction, we make a feature combination to enhance the performance.

I. INTRODUCTION

In recent years, online reviews on products and services contain rich information related to subjective opinions on certain topics. This information has become an important resource for public opinion that influence our decisions over an extremely wide spectrum of daily and professional activities: e.g., where to eat, where to stay, which products to purchase, which doctors to see, and so on. Since reviews information can guide people purchase behavior, positive reviews can result in huge economic benefit and fame for organizations or individuals. This gives powerful incentive to promote the generation of deceptive opinion spam. Deceptive opinion spam is a type of review with fictitious opinions, deliberately written to sound authentic. Neural Network Generated spam and Deceptive opinion spam detection is an urgent and meaningful task. By continuous growth of the user-generated reviews, the appearance of deceptive opinion spam arouses people's attention. It is very difficult for people to distinguish deceptive spam. In the test by previous work, the average accuracy of three human judges is only 57.33%. Hence, the research in detecting deceptive opinion spam is necessary and meaningful. Without detecting them, the social media could become a place full of lies, fakes, and deceptions and completely useless. Hence, machine learning methods for automatically detecting deceptive opinion spam can be very necessary. Based on the positive and negative examples annotated by people, supervised learning is utilized to build a classifier, and then an unlabeled review can be predicted as deceptive review or truthful one. So, the objective of the task is to identify whether a given document a spam or not. The reviews are commonly short documents. The task can be transformed into a binary classification problem. Most of existing approaches utilizes machine learning algorithms to build the classifiers. Under this direction, most studies focus on designing effective features to enhance the classification performance. Feature engineering is important;

however, we can hardly learn the inherent law of data from a semantic perspective. In view of the good performance of neural network-based models in the natural language processing tasks currently, the document-level representation can be learnt by neural network-based models and be used as features of the review. In this work, we try to make a comparison and analysis between representation learning algorithms and conventional features while solving the problem. We present a novel method which is Dynamic memory network(DMN) based Recurrent neural network(RNN) in combination to sentence weighted neural network (SWNN) model to learn the document-level representation of the review and detect spam reviews. Learning the representation of the document can capture the global feature and take word order and sentence order into consideration. DMN in combination to RNN facilitates the model to remember long term dependencies of a word or sentence, thus the model in combination to SWNN can learn document-level representations of the reviews and thus effectively detecting deceptive spam reviews. We also make a feature combination with DMN, RNN and SWNN, that the features are firstly used jointly in the spam review detection. We verify the effectiveness of DMN based RNN, SWNN and the feature combination in our experiments using single-domain, hotel reviews. The experiments run on the public datasets. The domain migration experiment verifies that feature combination with SWNN has the best robustness. The domain-independent experiment verifies that the feature combination with n-gram models perform better than the feature combination with SWNN.

The major contributions of the work presented in this paper are as following:

• We present a sentence weighted neural network to learn the representation of document-level reviews. We combine it with DMN based RNN model to learn the semantic of the document better.

Li et al create a cross-domain data sets (i.e. hotel, restaurant, and doctor) with part of reviews from domain experts. On this labeled data set, they use n-gram features as well as POS and LIWC features in classification and show that POS perform more robust on cross-domain data. We have also inculcated n-gram features in our model Such that our model performs well on cross-domain data.

Neural networks for representation learning:

Representation learning by neural networks-based methods have been proven to be effective in the place of task-specific feature engineering. with different grains, like word, phrase, sentence and document. As for representing a document, the existing deep learning methods consist of two processing stages. Firstly, word vectors need to be created to represent each word in a sentence and each sentence in document in vector and tensor forms respectively. Secondly, we use Word embeddings to represent the similarity of the words and probability of its occurrences in huge text corpus. Word2Vec and Stanford Glove vectors present word embeddings for around 3000 words. Both models learn geometrical encodings (vectors) of words from their co-occurrence information, which is how frequently they occur in large text corpora. Word2Vec is predictive model, whereas Glove vector is a count-based model. After obtaining word representation, many studies focus on researching the semantic composition methods. Yessenalina et al. use matrices to model each word and applying iterative matrix multiplication to combine words. Glorot et al. develop Stacked Denoising Autoencoders for domain adaptation. Socher et al. propose Recurrent Neural Network (RNN), matrixvector RNN and Recursive Neural Tensor Network (RNTN) to learn the semantic of unfixed-length phrases. Hermann et al learn the semantic of sentences by Combinatory Categorial Autoencoder method. The method is the combination of Combinatory Categorial Grammar and Recursive Autoencoder. Li et al use feature weight tuning to control the effect one specific unit makes to the higher-level representation in a Recursive Neural Network. Le et al learn the representation of paragraph.

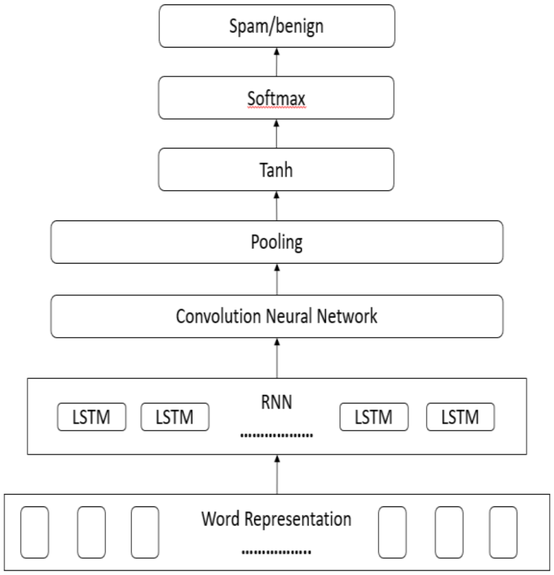
II. DATA SOURCE AND PREPARATION

The dataset is taken from the first publicly available gold standard corpus of deceptive opinion spam. It consists of truthful and deceptive hotel reviews of 20 Chicago hotels. It contains 400 truthful positive reviews from TripAdvisor, 400 deceptive positive reviews from Mechanical Turk, 400 truthful negative reviews from Expedia, Hotels.com, Orbitz, Priceline, TripAdvisor and Yelp, and 400 deceptive negative reviews from Mechanical Turk. To represent words as vectors we have used Google news-Vectors-negative 300 binary. It means each vector would be of size 300. For the words which are not present in the word2Vec model we randomly generate numbers between -0.25 and 0.25 of size 300. The number 0.25 is taken considering cooccurrence representations of the corresponding statistical representations of word2Vec vectors.

III. METHODOLOGY

This section presents the details of neural network-based models to learn document representation for deceptive spam review detection. The dynamic RNN layer is used to learn the long-terms dependencies which is built using Long short-term memory gates(LSTM). As plain RNN can be vulnerable to exploding and vanishing gradient issues when document-level learning is expected. The Sparse Convolutional Neural Network Model(SCNN) consists of two convolutional layers and a SoftMax classification layer to classify if the given review is spam or normal.

The architecture looks as follows:



From this architecture, we can see convolutional filter and an RNN layer. RNN layer is used to make the model aware of long-term dependencies. LSTM gates are used in RNN as plain RNN model is vulnerable to exploding and vanishing gradient problems. Gated recurrent Units(GRU) can also be used for better efficiency. The second convolutional layer of the SCNN model is called the document convolution. It transforms sentence vectors into a document vector. Given a document with m sentences, we use the sentence vectors s1, s2, ..., sm as inputs and we get the document vector representation as output. Finally, the softmax classification layer use the document vector representation as features to identify deceptive spam review. Overfitting is a major problem for predictive analysis and especially for neural networks. Key methods proposed to avoid overfitting include regularization (L2 and L1), Max norm constraints and Dropout. In our model we have used l2 regularization and dropout techniques to avoid overfitting issues. We train on rather random sentences by randomly feeding it to network to avoid overfitting and increase model overall accuracy and efficiency.

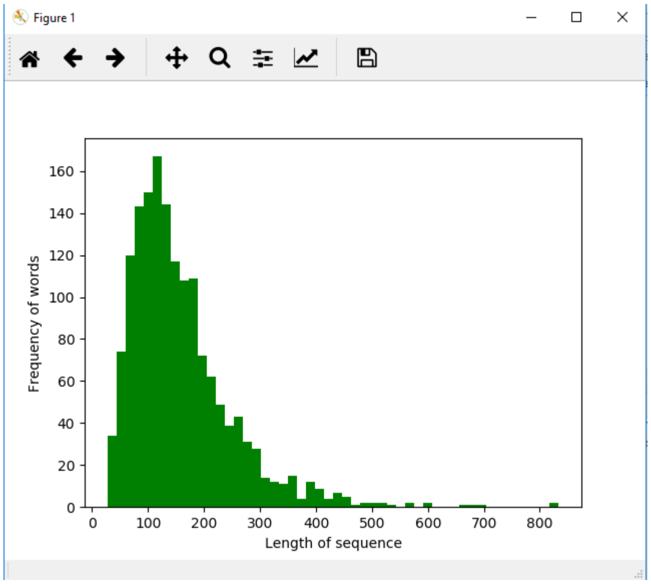
IV. EXPERIMENTS

We have conducted experiments using Tensor Flow. We initially intend to experiment our model on single-domain and further expand it to cross-domain. So, our experiments are based on the publicly available data set gold standard corpus of deceptive opinion spam. It consists of 400 truthful positive reviews from TripAdvisor, 400 deceptive positive reviews from Mechanical Turk, 400 truthful negative reviews from Expedia, Hotels.com, Orbitz, Priceline, TripAdvisor and Yelp, and 400 deceptive negative reviews from Mechanical Turk. To represent words as vectors we have used Googlenews-Vectors-negative300 binary. It means each vector would be of size 300. For the words which are not present in the word2Vec model we randomly generate numbers between -0.25 and 0.25 of size 300. The number 0.25 is taken considering cooccurrence representations of the corresponding statistical representations of word2Vec vectors. We have also experimented our model using Stanford Glove word embeddings, but our model seems to work better on Word2Vec word vector representations.

We initially cleaned up our dataset as it contains unwanted characters. Our cleanup process used regular expression library of python to remove non-alpha numeric characters, and all the unwanted special characters. We then remove stop words in the data and lemmatized the dataset accordingly. We have also removed unwanted URLs and other such information.

After that we segregated and collected truthful reviews into a collection and deceptive reviews into another collection. We then tokenized our data. We then went ahead and collected the corpus statistics such as total number of truthful reviews present in the corpus which is 800. The total number of negative reviews in our corpus is 800 as well. The number of files we read our input data includes 1600. Total number of words in all the 1600 files are 253157. Total unique words which is known as vocabulary size of corpus is 9687. We even calculated the average number of words in each file which came around 158.22.

Below is the histogram representation of the corpus statistics:



The above plot represents the histogram representation of our corpus data. The x-axis represents the length of the sequence. The y-axis represents the Frequency of words in the sequences.

After collecting corpus statistics, we represented each tokenized word in our data set into a vector representation using Googlenews-Vectors-negative300 binary files. We then saved all these word vector representations in a pickle file. This file serves as look-up file when we train our model.

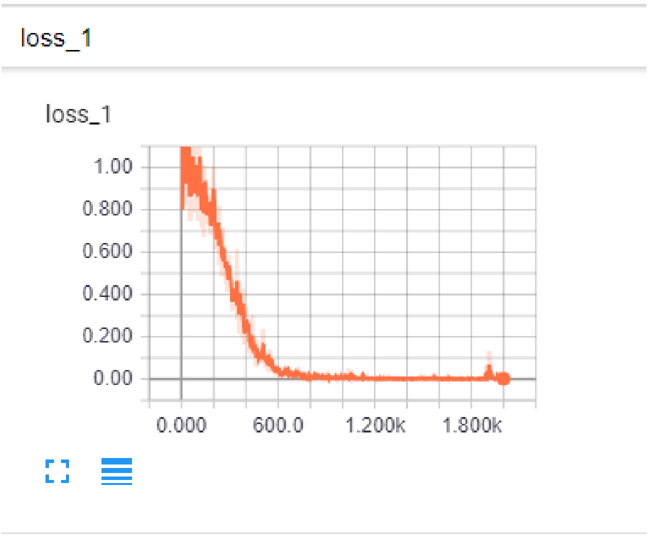
Later we padded these vectors to maximum sequence length such that there is uniformity when the model is trained on these matrices. We used post-sequence padding. After the padding is done and sentence matrices are formed. The dimensions of these matrices are (1600,160) where 160 is the maximum sentence length as sentence per review is 16 and words per sentence is 10. 160 is approximated number given the above collected statistics. 1600 represents the total number of files. We have also formed a matrix of labels representing if a sentence is spam or benign. The dimensions of these label matrices are (1600,2), where number 2 represents spam or normal.

We then split our dataset to train, validate and test our TensorFlow model. We divided our entire corpus into 80% as training data and 10% as validation data and another 10% as test data.

The dimensions of each of these data is (1280,160) for train data and train label data is (1280,2) and validation data dimensions involve (160,160) and validation label data dimensions include (160,2). Test data and Test label data are of similar dimensions as that of the validation data set. All this data is saved in another pickle file which works as lookup while training, validating and testing of our model.

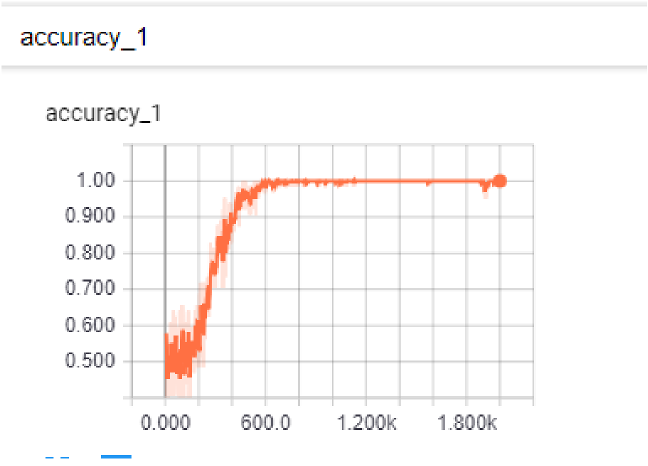
We then train our model using TensorFlow. We feed our word2vec representations in batches of 64 to the RNN layer which is then collected into 2 Layer SCNN model SCNN model in-turn uses soft-max classification layer to classify the data. First layer is two-dimensional convolution layer. The second layer of SCNN is max pooling layer. We then feed our output to tanh layer to spread out the values between -1 to 1 and then use softmax layer in the end for classification. We trained our model for 100 epochs. We also saved the model weights and summaries for every 5 checkpoints. We used a dropout probability of 0.5 and validated our trained model after every 100 steps. Our initial weights are initialized using Xavier Initializer. We then used Adam Optimizer to optimize the model weights during backpropagation. We used softmax cross entropy as our loss function.

The following diagram represents the loss function performance during training.



We can see that the loss function has decayed over the time of training giving us better accuracy in the end.

The following diagram represents the performance accuracy during the model training.



We can infer from the above diagram that our model attains 100% accuracy within 40 epochs. Whereas on test data our model only achieves an accuracy of 86%. It could be due to overfitting of data. The model finished in 40 epochs as there is less amount of data. As mentioned earlier we extend to train our model on 100 epochs on cross-domain data and still could achieve reasonable accuracy.

V. DEFENSIVE MECHANISMS AND FUTURE WORK

In future, one of the defensive mechanism we propose to make our model robust is to integrate NLP components of POISED (Parties of Interest Semantic Extraction and Discovery), a propagation probability model in detecting spam messages with our RNN based model to make it into a robust system that endures adversarial evasions. POISED is a system that leverages the differences in propagation between benign and malicious messages on social networks to identify spam and other unwanted content. It uses four-gram model with basic Machine Learning Classifiers. Our model which is based on RNNs and SCNN and which uses n-gram models would be giving better accuracy to the existing POISED models. We can extend our model to include Generative Adversarial Networks (GANs) which helps to detect Neural Network generated spam reviews.

VI. CONCLUSION

We introduced RNN and SCNN based model to detect spam reviews. We incorporate the sentence weights into document representation by creating a semantic representation model of reviews. We conducted experiments on first publicly available gold standard corpus of deceptive opinion spam using TensorFlow. We also experimented our model using Stanford Glove word embeddings, but our model worked better on Word2Vec word vector representations. We found that neural network-based methods perform more robust than the hand-crafted features on single-domain or cross-domain data sets. We obtained 86% accuracy on our test data where as our training achieved 100% accuracy within 40 epochs. This might be due to classic overfitting problem which we plan to resolve using advanced regularization techniques. Accuracy further improves when the dataset is huge. Overall, RNN and SCNN based models performed better than normal Machine learning classifier models in detecting spam reviews.

VII. References

[1] https://www.ieee.org/conferences/publishing/templates.html

[2] https://www.sciencedirect.com/science/article/pii/S0020025517300166

[3] https://www.sciencedirect.com/science/article/pii/S0957417416301129

[4] https://www.sciencedirect.com/science/article/pii/S1877050917327205

[5] https://link.springer.com/chapter/10.1007/978-3-642-25206-8_21

[6] https://www.researchgate.net/publication/278652738_Machine_Learning_for_the_Detection_of_Spam_in_Twitter_Networks

[7] https://pdfs.semanticscholar.org/2ef0/3ba493f5d4c8a5dfd9c62bcd6abd81a5c9de.pdf

[8] https://simplyml.com/spam-detection-in-9-lines-of-code/

[9] http://fortune.com/2018/02/26/russian-bots-twitter-facebook-trump-memo/

[10] https://blog.paralleldots.com/research/identifying-fake-accounts-twitter-bots-using-artificial-intelligence/

[11] https://www.gamesindustry.biz/articles/2017-11-21-steam-overhauls-users-reviews-once-again

[12] https://digiday.com/marketing/amazon-reviews-bot-problem/

[13] https://www.searchenginenews.com/sample/content/the-best-way-to-enable-emreviews-em-while-avoiding-bot-spam

[14] M. Ott, Y. Choi, C. Cardie, J.T. Hancock, Finding deceptive opinion spam by any stretch of the imagination, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics Human Language Technologies, 1, Association for Computational Linguistics, 2011, pp. 309–319.

[15] N. Jindal, B. Liu, Opinion spam and analysis, in: Proceedings of the 2008 International Conference on Web Search and Data Mining, ACM, 2008, pp. 219–230.

[16] J. Li, M. Ott, C. Cardie, E. Hovy, Towards a general rule for identifying deceptive opinion spam, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, 2014, pp. 1566–1576.

[17] Z. Gyöngyi, H. Garcia-Molina, J. Pedersen, Combating web spam with trustrank, in: Proceedings of the Thirtieth International Conference on Very Large Data Bases, 30, VLDB Endowment, 2004, pp. 576–587.

[18] A. Ntoulas, M. Najork, M. Manasse, D. Fetterly, Detecting spam web pages through content analysis, in: Proceedings of the 15th International Conference on World Wide Web, ACM, 2006, pp. 83–92.

[19] Z. Gyöngyi, H. Garcia-Molina, Link spam alliances, in: Proceedings of the 31st International Conference on Very Large Data Bases, VLDB Endowment, 2005, pp. 517–528.

[20] P.T. Metaxas, J. DeStefano, Web spam, propaganda and trust, in: Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web AIRWeb, 2005, pp. 70–78.

[21] B. Wu, B.D. Davison, Identifying link farm spam pages, in: Proceedings of the Special Interest Tracks and Posters of the 14th International Conference on World Wide Web, ACM, 2005, pp. 820–829.

[22] D. Fetterly, M. Manasse, M. Najork, Detecting phrase-level duplication on the world wide web, in: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2005, pp. 170–177.

[23] C. Castillo, D. Donato, A. Gionis, V. Murdock, F. Silvestri, Know your neigh- bors: web spam detection using the web topology, in: Proceedings of the 30th annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2007, pp. 423–430.

[24] P.-A. Chirita, J. Diederich, W. Nejdl, Mailrank: using ranking for spam detection, in: Proceedings of the 14th ACM International Conference on Information and Knowledge Management, ACM, 2005, pp. 373–380.

[25] H. Drucker, D. Wu, V.N. Vapnik, Support vector machines for spam categoriza- tion, IEEE Trans. Neural Netw. 10 (5) (1999) 1048–1054.

[26] A. Mukherjee, A. Kumar, B. Liu, J. Wang, M. Hsu, M. Castellanos, R. Ghosh, Spotting opinion spammers using behavioral footprints, in: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2013, pp. 632–640.

[27] G. Wang, S. Xie, B. Liu, P.S. Yu, Review graph based online store review spam- mer detection, in: Proceedings of the IEEE 11th International Conference on Data Mining (ICDM), IEEE, 2011, pp. 1242–1247.