

Analyzing the Sensitivity of Deep Neural Networks for Sentiment Analysis: A Scoring Approach

Ahoud Alhazmi^{1,2}, Wei Emma Zhang³, Quan Z Sheng¹, and Abdulwahab Aljubairy^{1,2}

¹Department of Computing, Macquarie University, NSW 2109, Australia

²Computer Science, Umm Al Qura University, Makkah 21955, Saudi Arabia

³School of Computer Science, The University of Adelaide, SA 5005, Australia

{ahoud.alhazmi, abdulwahab.aljubairy}@hdr.mq.edu.au

wei.e.zhang@adelaide.edu.au

michael.sheng@mq.edu.au

Abstract—Deep Neural Networks (DNNs) have gained significant popularity in various Natural Language Processing tasks. However, the lack of interpretability of DNNs induces challenges to evaluate the robustness of DNNs. In this paper, we particularly focus on DNNs on sentiment analysis and conduct an empirical investigation on the sensitivity of DNNs. Specifically, we apply a scoring function to rank words importance without depending on the parameters or structure of the deep neural model. Then, we scan characteristics of these words to identify the model’s weakness and perturb words to craft targeted attacks that exploit this weakness. We conduct extensive experiments on different neural network models across several real-world datasets. We report four intriguing findings: i) modern deep learning models for sentiment analysis ignore important sentiment terms such as opinion adjectives (i.e., amazing or terrible), ii) adjective words contribute to fooling sentiment analysis models more than other Parts-of-Speech (POS) categories, iii) changing or removing up to 10 adjectives words in a review text only decreases the accuracy up to 2%, and iv) modern models are unable to recognize the difference between an objective and a subjective review text¹.

Index Terms—Deep Neural Networks, Adversarial Examples, Sentiment Analysis

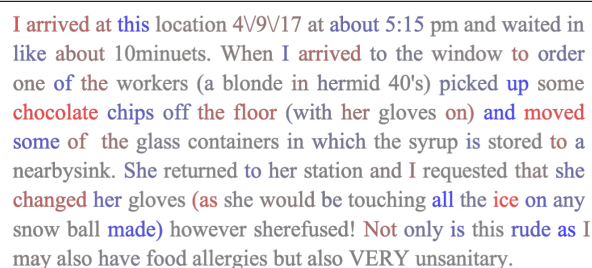
I. INTRODUCTION

There is a growing interest in sentiment analysis that attempts to automatically derive a person’s sentiment, i.e., positive or negative, about a topic. One reason is that sentiment analysis has become a key tool for making sense of the product data, which allows commercial companies to get key insights on their customers’ opinions and accordingly automate all kinds of following processes [1].

In recent years, Deep Neural Networks (DNNs) have achieved remarkable results in sentiment analysis [2]–[5]. However, the interpretability of deep neural networks is still unsatisfactory as they work as black boxes, which means it is difficult to get intuitions from what each neuron exactly has learned. One of the problems of poor interpretability is evaluating the robustness of deep neural networks.

Researchers have recently found that textual DNN classifiers are vulnerable to adversarial examples, which are intentionally designed to fool a model into making incorrect predictions

¹Our codes for this study available at <https://github.com/Ahoud-Alhazmi/Investigation-of-the-Sensitivity-of-DNN-for-Sentiment-Analysis>



I arrived at this location 4V9V17 at about 5:15 pm and waited in like about 10minuets. When I arrived to the window to order one of the workers (a blonde in hermid 40's) picked up some chocolate chips off the floor (with her gloves on) and moved some of the glass containers in which the syrup is stored to a nearby sink. She returned to her station and I requested that she changed her gloves (as she would be touching all the ice on any snow ball made) however she refused! Not only is this rude as I may also have food allergies but also VERY unsanitary.

Fig. 1: Visualization of Words’ Importance. Red words indicate as high score (important token), gray words indicates near-zero score words and blue as low score (unimportant token). Notice: the brightest red means the word is the most important word (e.g., “chocolate”) in a text.

that lead to the drop of the accuracy rate [6]–[11]. These methods share a common principle to search for key features and then perform perturbations on these features. However, which word-category factors, e.g., Part-Of-Speech (POS), are influential to the robustness of DNNs remain undiscussed. Thus, the previous researches did not identify the weaknesses in the logic of models as which are exposing words that can fool a neural network model. As a result, knowing which words were heeded or ignored in a model can assist in immunization models from crafting adversarial examples.

In this paper, we analyze the sensitivity of DNNs, particularly on sentiment analysis tasks. Our analysis method has three main steps. Firstly, we apply a scoring function to determine the importance of words from high to low without relying on the structure or parameters of the model. Figure 1 shows the visualization of the score terms. Then, we scan the characteristics of these words. After that, we leverage this weakness to modify words to craft targeted attacks based on over-reliance on words.

Our key contribution of this research is to identify what POS-tagging of words in sentiment analysis to generate adversarial examples. Our initial hypothesis was that adjectives should be paramount the important words to the sentiment analysis classifiers as they are considered as main sentiment

terms. We report four findings as follows:

- The modern deep learning models for sentiment analysis rely on generic words in the analysis of a reviewer’s sentiment. These words do not show any feeling or opinion.
- Adjective words contribute to fooling sentiment analysis models more than other POS categories if these adjectives were ranked as important words in a review text using the scoring function THS.
- Changing or removing up to 10 adjectives words in a review text only decreases the accuracy up to 2%.
- Modern models are unable to recognize the difference between an objective and a subjective review text.

The remainder of the paper is organized as follows. In Section II, we introduce how we examine the sensitivities of sentiment analysis DNNs. Then, we report the experiment and results in Section III. Section IV discusses related work and Section V concludes the paper.

II. METHODOLOGY

Our method has three steps to analyze the sensitivity of sentiment analysis models. Firstly, we apply a scoring function called Temporal Head Score (THS) [9] in order to rank tokens based on their importance to the prediction. Then, we categorize these words based on POS-tagging. Finally, we develop a few attack strategies against the DNN models that exploit their weaknesses. Our method can be applied to any DNN classifier because it is based on a black-box setting. That means we explain the sensitivity of classifiers without access to the structure, parameters or gradient of the target model and this is more practical in the real-world applications.

A. Ranking Tokens

Given a text as a word sequence $T_i = \{w_1, w_2, \dots, w_n\}$, where w_n is the n^{th} word in T_i , a DNN classifier model is represented as $f_\theta : \mathbf{T} \rightarrow y$, which maps from features T to the label y with parameters θ . In order to rank words by their importance, we measure the effect of each token on the output classifier using a scoring function. For that, we use the THS function of the n^{th} token as shown in Eq.(1) which is the difference between the model’s prediction score as it reads up to the n^{th} token, and the model’s prediction score as it reads up to token $n - 1$. In other words, it measures the influences of deleting a targeted word on the classification result’s confidence value by comparing the predication before and after deleting this word as shown in Figure 2.

$$THS(w_n) = f(w_1, \dots, w_{n-1}, w_n) - f(w_1, \dots, w_{n-1}) \quad (1)$$

This scoring function has three advantages. Firstly, it is efficient to rank each word in a text for the model. Also, it can correctly reflect the importance of each word for a model. Thirdly, it calculates word scores without the knowledge of the parameters and structure of the classification model.

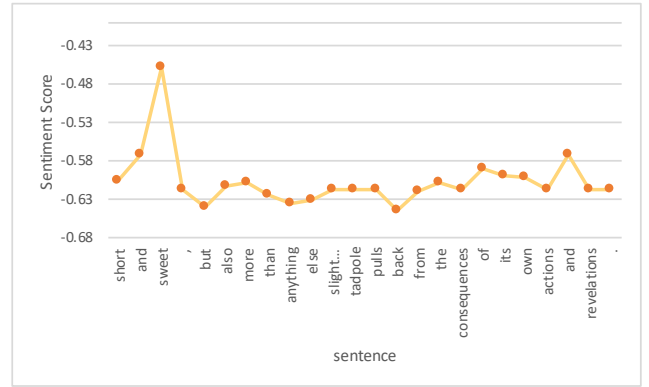


Fig. 2: Ranking and identifying important words in a sentence. The sentiment score of each word is the confidence value after removing the word from the original sentence.

B. Determining Sets of Words

After ordering the important words by descending scores, we determine two sets of words for the highest and lowest score that the model finds most important and unimportant words in each sample, respectively. $W_{highest}$ denotes the set of words with the highest score and W_{lowest} denotes the set of words with the lowest score. $Targeted$ denotes the selected target tokens based on the function of finding the highest and the lowest. Also, each element in sets consists of the targeted word and its POS-tagging based on a text as follows:

$$\begin{aligned} T_i &= \{w_1, w_2, \dots, w_n\} \\ T_{Score_i} &= \{Score_{w_1}, Score_{w_2}, \dots, Score_{w_n}\} \\ W_{highest} &= \{Targeted | \forall \text{ words with max score in each } T_i\} \\ W_{lowest} &= \{Targeted | \forall \text{ words with min score in each } T_i\} \end{aligned}$$

where

$$Targeted = (w, \text{ part of speech})$$

In this paper, we want to identify what characteristics of the highest scored token in $W_{highest}$ could help attack DNN text classifier. For that, we divide each set of $W_{highest}$ and W_{lowest} into six groups by their POS taggings using NLTK Punkt tokenizer [12] and Averaged Perceptron Tagger package. By using POSs, we examine the impact of Adjectives, Verbs, Adverbs, Noun, Pronoun and Other (e.g., conjunction, preposition, punctuation marks) words.

C. Perturbations on Words

In this step, we aim to generate an adversarial sample w^* by manipulating the word or characters of w where $f_\theta(w^*) \neq f_\theta(w)$. Three strategies are defined to modify each token as the following:

- **Misspelling Attack:** We manipulate on characters of the word. We use a shuffle method that randomizes the order of all letters in a word except the first letter (e.g., “great” to “garte”).
- **Grammatical Attack:** We use transformation method that replaces the targeted token (word) by its other POS

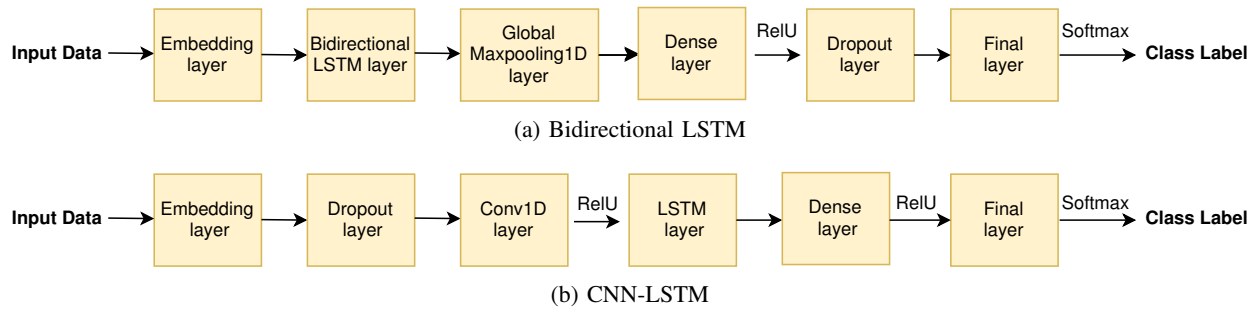


Fig. 3: The architectures of the DNN models used for sentiment classification

tagging (e.g., changing the word from *noun* to *verb*: writer to write).

- **Stop Word Deletion:** We delete stop words (e.g., “a”, “the”, “can” and “so”) from a text.

III. EXPERIMENTS

In this section, we firstly introduce the settings of our experiments that include the datasets, targeted models and implementation details. Then, we analyze the results and discuss potential reasons for the observed performance.

TABLE I: Dataset details and models. Acc. refers to the classification accuracy of DNN models on original test sample. Yelp.Pol refers to Yelp Review Polarity dataset, and Yelp.Full is the Yelp Review Full dataset.

Dataset	IMDB	MR	Yelp.Pol	Yelp.Full
# Training	25,000	7,393	300,000	500,000
# Test	25,000	3,269	30,000	50,000
# Classes	2	2	2	5
Acc. CNN-LSTM	86.47%	81.54%	92.72%	88.30%
Acc. LSTM	82.18%	84.95%	92.11%	88.97%

A. Experimental Setup

We conducted experiments on a modern deep learning model across several real-world NLP datasets. The entire test data were used. Our experimental set up is as follows:

Dataset: We used four datasets, namely the *IMDB*², *Movie Review (MR)*³, and *Yelp Reviews*⁴ Polarity and Full. Table I shows the statistics of the four datasets that were used in our experiments. The first three datasets contain two polarity classes whereas the last dataset (Yelp.Full) contains five classes.

- **IMDB:** It is a movie review dataset that consists of the reviews for different movies along with the class-label (positive or negative sentiment). The dataset is divided into training and testing sets, with each set consisting of 50% positive and 50% negative reviews.
- **Movie Review (MR):** It is a movie review dataset that contains reviews with one sentence per review. Classification involves detecting positive or negative reviews.

²<http://ai.stanford.edu/amaas/data/sentiment/>

³<http://www.cs.cornell.edu/people/pabo/movie-review-data/>

⁴<https://www.yelp.com/dataset/challenge>

- **Yelp Reviews:** It contains 1,569,264 samples that have review texts. Two classification tasks are constructed from this dataset: one is for predicting full number of stars the user has given (e.g., very positive= 5, positive= 4, neutral= 3, negative= 2 and very negative= 1), and the other is for predicting a polarity label by considering stars 1 and 2 as negative review, and 4 and 5 as positive.

Attacked Models: We analyzed the two most widely-used deep neural models from [13] and [14]. The first model is the Bidirectional Long Short-Term Memory (LSTM) and the second model is CNN-LSTM which combines Convolutional Neural Network and Long Short-Term Memory. Hence our goal is not to compare these two models but to use them to confirm our observation. Our analysis method can be applied to any DNN classifier based on the black-box setting.

Implementation: The architectures of the examined LSTM and CNN-LSTM are shown in Figures 3a and 3b. We use 80% data as training and 20% as validation and train for a maximum of 40 epochs. We trained these models without using adversarial samples. We applied modified words only on test samples. For this experiment, we generated adversarial examples for 100% of the test data. The accuracy of both models on original test samples is shown in Table I. These models are similar to the state-of-the-art results on these datasets. Stop-words are usually filtered out before the feature extraction step in the NLP tasks. However, due to our observation of the impact of these words on the prediction result, we avoided filtering them out in our experiment. We trained the target models and implement attacking methods using Keras⁵. All the experiments were run on a PC with Windows 10 (64-bit) operating system, 3.10 Ghz CPU (i5-4440), and 8GB RAM.

B. Analyzing High and Low Scored Words

We used the scoring function to identify important and unimportant words for a target model. A visualization of the high and low scored words is shown in Figure 1 using the THS function. Comparing the THS function with other score function as presented in [9], we found the same result for words with high and low scores.

Table II shows the ten most common words that have high and low scores for the models. We arranged these words in

⁵<https://keras.io/>

TABLE II: The ten most important words that have high and low scores. Yelp.Pol refers to Yelp Review Polarity dataset, and Yelp.Full is the Yelp Review Full dataset.

High Score										
IMDB	the	it	and	good	best	great	a	to	of	with
MR	the	a	it's	this	an	it	if	one	in	as.
Yelp.Pol	great	best	Great	love	good	and	nice	favorite	amazing	good.
Yelp.Full	I	This	The	We	My	Great	is	went	been	If
Low Score										
IMDB	worst	of	bad	to	the	and	is	in	not	a
MR	.	?	!	”	,	'	quotations	somebody	toughest	sweeping
Yelp.Pol	place	was	the	a	and	to	I	This	is	very
Yelp.Full	:	-	.	&	!	waste	We'll	returning	regret	:)

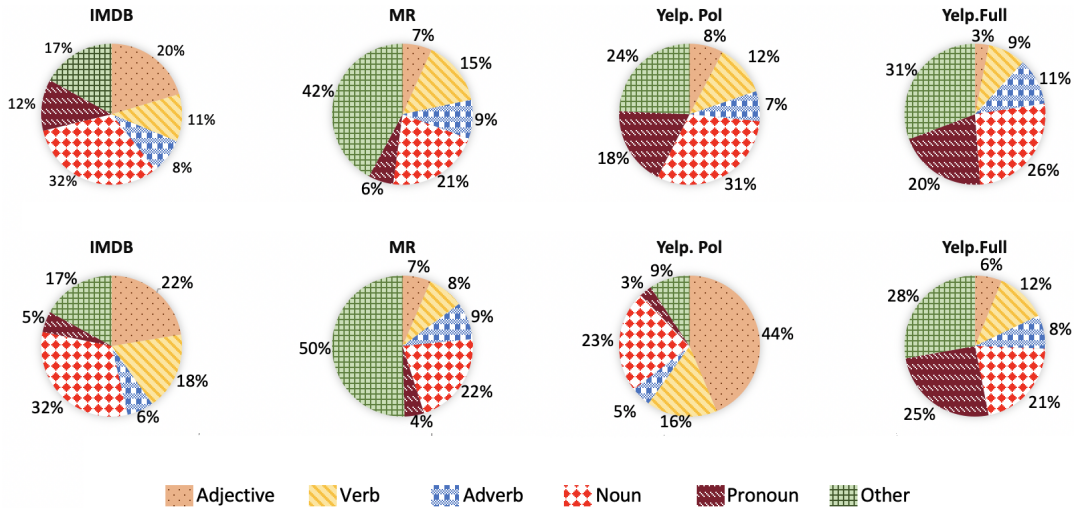


Fig. 4: The distribution of high scored words of the two models among four datasets of the: IMDB, MR, Yelp Review Polarity and Yelp Review Full, respectively. Pie charts in the first row for the *CNN-LSTM* model and the second row for the *LSTM* model.

descending order. By looking at these words based on each dataset, we found there are more informative words in high score sets than low score set which contains the majority of uninformative words. For example, “great” is an adjective to show a person’s opinion while “very” does not show a person’s opinion, although it is an adjective. However, we found several words in high score sets that do not convey any feeling (i.e., they are not sentiment terms), which are considered as very important words for DNN classifiers such as: “The”, “and”, “a” and “to” as shown in Table II. For that, we investigated the POS of these words in Section III-C.

C. Distribution of High Score Terms

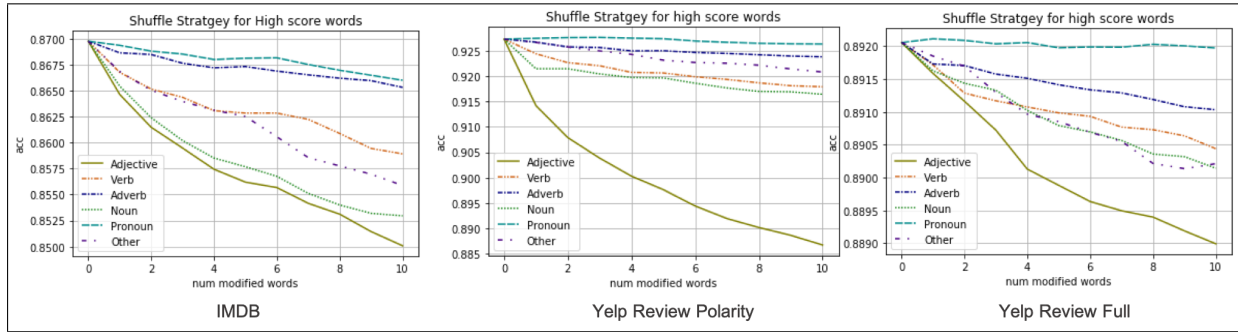
We isolated the highest scored words in a set. Then, we categorized these words based on the POS-tagging using WordNet into six groups as discussed in Section II. Figure 4 shows the distribution of the highest scored words of the two models among four datasets. Our initial hypothesis was that adjectives should be paramount importance to the sentiment analysis classifiers as they are considered as main sentiment terms. However, we found only most of these words are adjective tokens in Yelp Review Polarity in LSTM model whereas noun or other tokens are the most words for IMDB,

MR and Yelp Full for the two targeted models. Therefore, relying largely on generic words such as these two categorized (noun and other groups) to extract the opinion of a reviewer, is considered a weakness in the model’s logic because they do not convey any opinion or feeling.

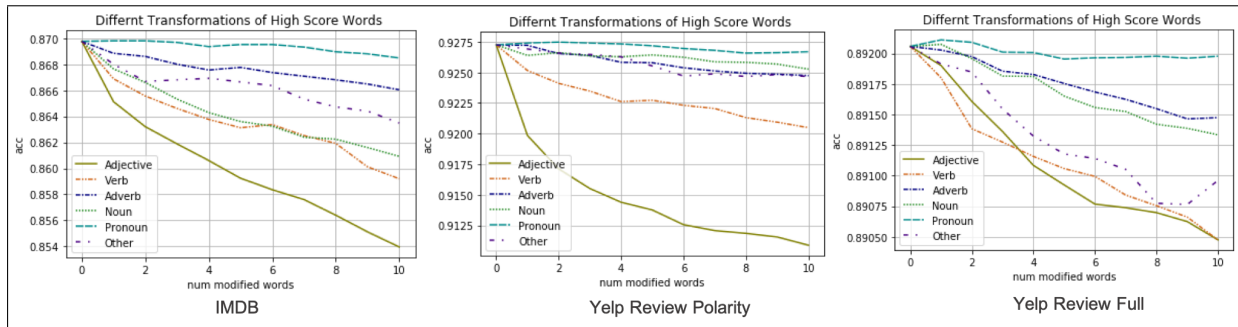
D. Attacks

Based on our observation as discussed in the previous section, the DNN models for sentiment analysis rely largely on generic words in the analysis of a reviewer’s sentiment. This is a weakness in the classifiers’ logic which ranks these words as important words. For that, we now describe a few attacks to these classifiers that exploit this weakness.

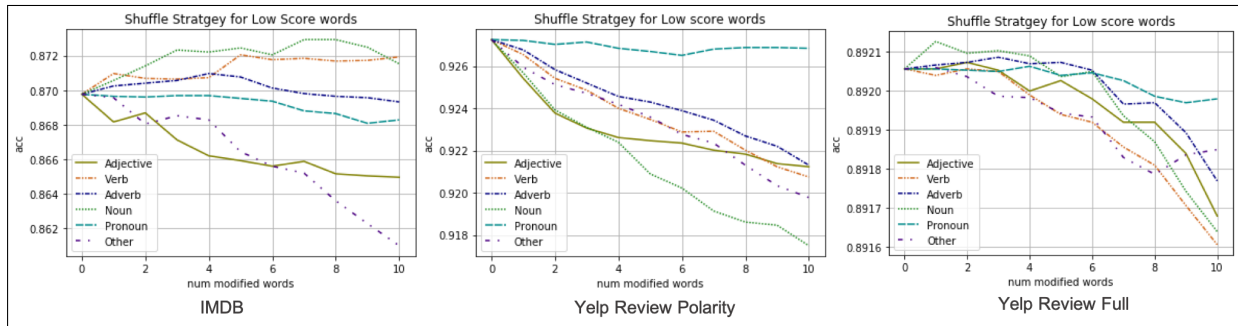
1) *Misspelling and Grammatical Attack*: We modified each token that is ranked as important words by using *Misspelling* and *Grammatical* strategies. Comparing these two attacks, we found the results are similar in the both strategies for two models, although the shuffle attack replaces a targeted word with an out-of-vocabulary one and the other attack not. Also, we observed the most examples that attack the CNN-LSTM model can also attack the LSTM model similar to the finding from [15].



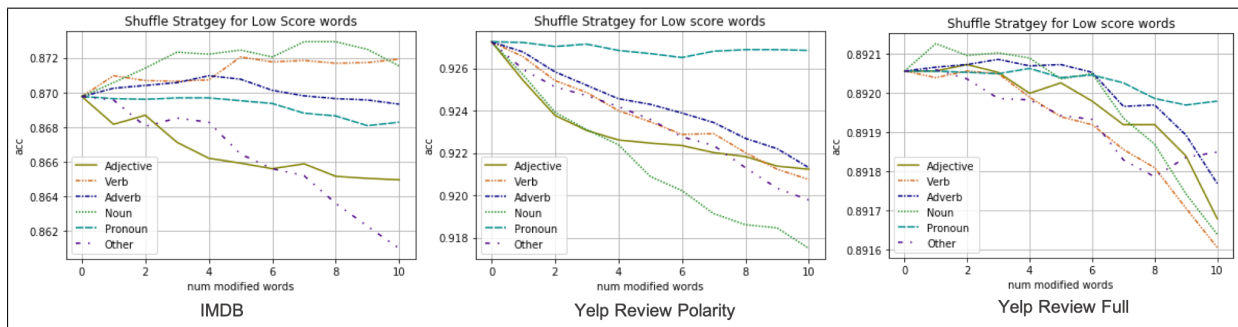
(a) Misspelling attack for **high** scored words



(b) Grammatical attack for **high** scored words



(c) Misspelling attack for **low** scored words



(d) Grammatical attack for **low** scored words

Fig. 5: Accuracy drop of CNN-LSTM on three adversarial strategies for attacking high scored and low score words using POS-tagging on Misspelling attack and Grammatical attack, receptively.

TABLE III: Examples of prediction performance before and after deleting all adjectives.

Text	Original Text Prediction	Predication After Removing
far more enjoyable than its predecessor. ...	Negative (90.3%)	Negative (80.5%)
Seems like a good addition to the Champaign downtown. I don't see Quality as offering anything all that different from the other places, but it does so in a very nice setting. Ambiance is excellent.	Positive (99.2%)	Positive (87.8%)
Do not believe the hype. You must go there and try it yourself. The ingredients grown by the owner on the pizza dough is the best I have ever had. The toppings are fresh full of flavor. May be the best pizza ever	Positive (99.3%)	Positive (93.9%)
bielinsky is a filmmaker of impressive talent.	Positive (98.9%)	Positive (80.5%)
a strong piece of work.	Positive (99.3%)	Positive (85.9%)

As mentioned previously, the aim of this study is to investigate which POS-tagging of high scored words can help on attacking deep neural models. We found the adjective adversary decreases the model's accuracy more than other adversaries if it is ranked as high scored words as shown in Figure 5a and 5b. However, the model is not extremely sensitive to the pronoun adversary if this word is ranked as a high scored word. Also, we found that the binary classifier tends to be brittle than multi-classification for the adjective adversary. We also provide the accuracy drop of CNN-LSTM on both *Misspelling* and *Grammatical* attacks for word being ranked as low score as shown in Figures 5c and 5d. The accuracy of the model drops slightly comparing to the accuracy drop for important words.

An interesting finding is that changing or removing up to 10 adjective words only decreases the accuracy of the models by about 2%. That means these models can not recognize the difference between subjective and objective review. Table III shows some examples. For example, after deleting adjective words like "impressive" in this review "*bielinsky is a filmmaker of impressive talent*", the review will not be contained any opinion although the classifiers classified it as positive with high confidence. Another example, after deleting several adjectives in this review "*short and sweet, but also more than anything else slight... tadpole pulls back from the consequences of its own actions and revelations*", this review is not only incomplete but it does not contain any opinion. However, the classifiers classify it similar to the original text prediction.

2) **Stop Word Deletion Attacks:** Stop words are the English words that do not add much meaning to a sentence. They can safely be ignored without sacrificing the meaning of the sentence. Based on our observation, we found the models rely

on generic words. For that, we experimented the influence of removing stop words on the models. We found the models tend to be brittle for deleting stop words. The model's accuracy drops gradually by deleting more than one stop words in a text. By looking at Figure 6, the accuracy drops among three datasets even though a stop word is categorized as a high score or low score. Nevertheless, a model falls about 2% if we drop stop words in DNN for binary classification than multi-classification.

IV. RELATED WORK

Adversarial examples have a long history in traditional machine learning for NLP. Biggio et al. [16] discussed the robustness of linear classifiers to filter spam email against adversarial examples. In addition, Dalvi et al. [17] presented how spam emails could not be detected just by adding characters to the emails using naive Bayes classifier.

Recently, several research works focused on crafting adversarial samples against deep learning models in the NLP community. These methods used white-box or black-box strategies. White-box adversary requires explicit knowledge of the attacked model, while the black-box adversary sees the DNN model as a black-box. It is only allowed to query the models and get the output. A black-box setting is considered more realistic and practical as in many applications because it is used as a service after the deployment stage. In our work, we also focus on the black box-setting of adversarial generation scenarios.

Several previous works used black box assumptions to create adversarial samples [9], [18]–[21]. Hosseini et al. [20] found adding spaces or dots between characters can trick Perspective API from Google which predicts toxicity messages. Also, Belinkov and Bisk [19] have shown that character-level machine translation systems are extremely brittle to random character manipulations, with both synthetic or natural noise such as keyboard typos. Furthermore, the work in [18] used a genetic algorithm for minimizing the number of word replacement from the original text, and at the same time can change the result of the attacked model. Furthermore, Gao et al. [9] presented a simple method to generate adversary on text classification by developing scoring functions to determine the important 'tokens' and then perturbed them. By following the work in [9], the work in [21] refines the scoring function. One contribution of this work lies in the perturbation restriction by using four textual similarity measurements: edit distance of text, Jaccard similarity coefficient, Euclidean distance on word vector, and cosine similarity on word embedding. Their method had been evaluated only on sentiment analysis task.

The main contribution of the aforementioned studies is to propose methods that generate adversarial texts and then mix these examples with original examples as a training dataset to train the model. Our work differs from them by analyzing the logic of the models without relying on the structure or parameters of the model. That assists to identify characteristic terms that are important or unimportant for the modern DNNs prediction for sentiment analysis. In other words, our method

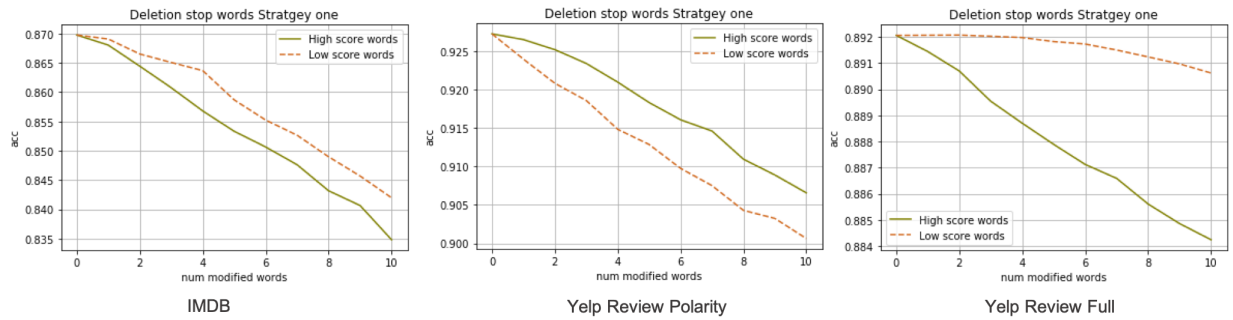


Fig. 6: Accuracy drop of after deleting stop words as ranked as high or low scores

gives an overview of the weaknesses in the logic of models to identify which adversary POS-tagging fools the sentiment analysis DNN.

V. CONCLUSION AND FUTURE WORK

This paper evaluates the robustness of DNN models on sentiment analysis. Our method identifies weaknesses in DNNs' logic without relying on the knowledge of the model. We find these models suffer from several weaknesses: i) they do not rely on sentiment terms, ii) they are sensitive when removing stop words, and iii) modifying adjective words can generate adversarial examples more than other POS categories if they are ranked as important words inside a text review. Also, our results show that changing or removing up to 10 adjectives words in a review text only decreases the accuracy up to 2%. Thus, modern models can not recognize the difference between a subject and objective review. Due to the lack of reliance on sentiment words from current DNN models, major future research is required to improve these models more effectively to distinguish between the sentiment words and generic words to overcome this weakness. Furthermore, future research on explaining the sensitivity of current models to adversarial examples would extend the interpretability of deep learning.

REFERENCES

- [1] S. A. G. Yadollahi Ali and Z. O. R, "Current State of Text Sentiment Analysis from Opinion to Emotion Mining," *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, pp. 1–25, 2017.
- [2] Z. Chen, S. Shen, Z. Hu, X. Lu, Q. Mei, and X. Liu, "Emoji-Powered Representation Learning for Cross-Lingual Sentiment Classification," in *Proceedings of the World Wide Web Conference (WWW)*, 2019, pp. 251–262.
- [3] C. Santos and M. Gatti, "Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts," in *Proceedings of the 25th International Conference on Computational Linguistics (COLING)*, 2014, pp. 69–78.
- [4] C. Zong, W. Feng, V. W. Zheng, and H. H. Zhuo, "Adaptive Attention Network for Review Sentiment Classification," in *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, 2018, pp. 668–680.
- [5] C. Junjie, H. Hongxu, J. Yatu, and G. Jing, "Graph Convolutional Networks with Structural Attention Model for Aspect Based Sentiment Analysis," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 2019, pp. 1–7.
- [6] M. T. Ribeiro, S. Singh, and C. Guestrin, "Semantically Equivalent Adversarial Rules for Debugging NLP Models," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018, pp. 856–865.
- [7] P. Yang, J. Chen, C.-J. Hsieh, J.-L. Wang, and M. I. Jordan, "Greedy Attack and Gumbel Attack: Generating Adversarial Examples for Discrete Data," *Journal of Machine Learning Research*, vol. 21, no. 43, pp. 1–36, 2020.
- [8] J. Ebrahimi, A. Rao, D. Lowd, and D. Dou, "HotFlip: White-Box Adversarial Examples for Text Classification," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018, pp. 31–36.
- [9] J. Gao, J. Lanchantin, M. L. Soffa, and Y. Qi, "Black-box Generation of Adversarial Text Sequences to Evade Deep Learning Classifiers," in *Proceedings of the IEEE Security and Privacy Workshops (SPW)*, 2018, pp. 50–56.
- [10] W. E. Zhang, Q. Z. Sheng, A. Alhazmi, and C. Li, "Adversarial Attacks on Deep Learning Models in Natural Language Processing: A Survey," *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2019, in press.
- [11] Z. Zhao, D. Dua, and S. Singh, "Generating Natural Adversarial Examples," in *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, 2018.
- [12] T. Kiss and J. Strunk, "Unsupervised Multilingual Sentence Boundary Detection," *Computational Linguistics*, vol. 32, no. 4, pp. 485–525, 2006.
- [13] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent Convolutional Neural Networks for Text Classification," in *Proceedings of the 29th National Conference of the American Association for Artificial Intelligence (AAAI)*, 2015, pp. 2267–2273.
- [14] J. Wang, L. Yu, K. R. Lai, and X. Zhang, "Dimensional Sentiment Analysis Using a Regional CNN-LSTM Model," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016, pp. 225–230.
- [15] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing Properties of Neural Networks," in *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*, 2014.
- [16] B. Biggio, G. Fumera, and F. Roli, "Multiple Classifier Systems for Robust Classifier Design in Adversarial Environments," *International Journal of Machine Learning and Cybernetics*, vol. 1, no. 1-4, pp. 27–41, 2010.
- [17] N. Dalvi, P. Domingos, S. Sanghai, and D. Verma, "Adversarial classification," in *Proceedings of Conference on ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 2004, pp. 99–108.
- [18] M. Alzantot, Y. Sharma, A. Elgohary, B. Ho, M. B. Srivastava, and K. Chang, "Generating Natural Language Adversarial Examples," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018, pp. 2890–2896.
- [19] Y. Belinkov and Y. Bisk, "Synthetic and Natural Noise both Break Neural Machine Translation," in *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, 2018.
- [20] H. Hosseini, S. Kannan, B. Zhang, and R. Poovendran, "Deceiving Google's Perspective API Built for Detecting Toxic Comments," *arXiv preprint arXiv:1702.08138*, 2017.
- [21] L. Jinfeng, J. Shouling, D. Tianyu, L. Bo, and W. Ting, "TextBugger: Generating Adversarial Text Against Real-world Applications," in *Proceedings of the 26th Annual Network and Distributed System Security Symposium, (NDSS)*, 2019.