Building a Better Lie Detector with BERT: The Difference Between Truth and Lies

Dan Barsever
Cognitive Science
University of California, Irvine
dbarseve@uci.edu

Sameer Singh
Computer Science
University of California, Irvine
sameer@uci.edu

Emre Neftci
Cognitive Science
University of California, Irvine
eneftci@uci.edu

***Abstract*—Detecting lies or deceptive statements in text is a valuable skill. This is partly because the patterns that underlie deceptive text are not known. The aim of this work is to identify patterns that characterize deceptive text. A key step in this approach is to train a classifier based on the BERT (Bidirectional Encoder Representations from Transformers) network. BERT beats the state of the art in deception classification accuracy on the Ott Deceptive Opinion Spam corpus. The results of our ablation study indicate that certain components of the input, such as some parts of speech, are more informative to the classifier than others. Further part-of-speech analysis in "swing" sentences that are considered important to BERT's classification indicates that deceptive text is more formulaic and less varied than truthful text. We expanded our classifier into a new Generative Adversarial Network based on BERT to create exemplars of deceptive and truthful text that further showed the differences between truth and deception, reinforcing the underlying similarity of deceptive text in terms of part-of-speech makeup.**

***Index Terms*—machine learning, BERT, neural network, natural language processing, generative, GAN, deception**

I. Introduction

Most traditional methods of lie detection consist of analyzing a physiological response, such as sweat or heart rate. When most think of lie detection, they think of the polygraph [1] or something similar: examining physiological responses such as increased sweat or heart rate that are expected to occur when people lie. Comparatively little study has been made into detecting lies in text, where there are no physiological clues [2]. One example from everyday life is in false reviews, or Deceptive Opinion Spam. This usually takes the form of a malicious customer posting fake negative reviews to hurt a business, or a company shill posting fake positive reviews to inflate its image. Humans are ineffective at detecting deceptive text, faring little better than chance [3, 4]. This is in stark contrast to other linguistic tasks such as sentiment analysis (e.g. identifying if a text sample is praising or condemning something) where humans perform extremely well [5].

To understand how lies are expressed in text, we decided to first build a state-of-the-art classifier that can learn the patterns that constitute a deceptive review, and then analyze that classifier to identify those patterns. To this end, we constructed a machine learning tool utilizing BERT. BERT (Bidirectional Encoder Representations from Transformers) is a recently developed neural network architecture that is pretrained on millions of words and is capable of forming different representations of text based on context [6]. By applying BERT to deception detection, we can use it to form a powerful classifier of deceptive text. After that, extracting the rules that BERT forms to classify the text can help us understand what patterns underlie deceptive text.

Our BERT-based classifier proved to be a useful tool for this study, defeating the state of the art on the Ott Deceptive Opinion Spam corpus and facilitating analysis on how it determines deceptive from truthful text. The rules it generates are still not completely clear, but our ablation study, where each part of speech (verbs, nouns, etc) is removed and the network's performance is monitored, has indicated that certain parts of speech such as singular nouns are more informative than others, as their removal resulted in the sharpest drop in accuracy.

We also performed part-of-speech analysis on 'swing' sentences—sentences shown to be informative to BERT's decision making. Our findings indicate that truthful sentences have more variance in what parts of speech occur. This provides evidence that there is a commonality in the structure of deceptive text that is less present in truthful text. This evidence is reinforced by the Generative Adversarial Network that we created, where a text generator based on BERT must try to create samples that can fool the BERT classifier into thinking they are real examples. The samples produced by our generator are easily recognized by the classifier as truthful or deceptive and reproduce many of the same trends seen in the swing sentences, particularly that many parts of speech appear with less variation across samples. This again points to deceptive text being more formulaic and less varied than truthful text.

II. Related Work

Ott et al. [2] developed the Ott Deceptive Opinion Spam corpus, which consists of 800 true reviews from TripAdvisor and 800 deceptive reviews sourced from Amazon Mechanical Turk. He used this corpus to train Naïve Bayes and Support Vector Machine (SVM) classifiers, achieving a maximum accuracy of 89.8% with an SVM utilizing Linguistic Inquiry and Word Count (LIWC) combined with bigrams. The Ott corpus is one of the most commonly used gold-standard corpora in deception detection tasks. Other, less widespread corpora include the LIAR fake news dataset [7], Yelp dataset in Feng et al. [8], and the Mafiascum dataset [9].

Vogler and Pearl [5] used a support vector machine operating on linguistically defined features to classify the Ott corpus. They were able to achieve an accuracy of 87% using this method. Xu and Zhao [10] train a maximum entropy model on the Ott corpus and were able to achieve 91.6% accuracy. Li et al. [11] tried to find a general rule for identifying deceptive opinion spam using features like part-of-speech on several datasets including the Ott corpus, achieving 81.8% accuracy on Ott. [12] expand on this work by using a recurrent neural network on the same data, improving the accuracy to 85.7%.

Hu [13] used a variety of models to identify concealed information in text and verbal speech, best among them a deep learning model based off bidirectional LSTMs. Concealed information, in this context, refers to when a person has knowledge about a subject and is withholding it, as compared to Hu's definition of deception where someone fakes knowledge they do not have. Hu created a corpus of wine tasters evaluating wines and encoding in various ways such as n-grams, LIWC, and GloVe embeddings [14] based on the recordings. The LSTM model using these features achieved an f-score in identifying the presence of concealed information of 71.51, defeating the human performance of 56.28.

Jin et al. [15] put BERT's robustness to the test by attacking its input in text classification and textual entailment tasks. They did so by calculating an Importance score for each word in an input sequence, and then perturbing that input by substituting semantically similar words to replace the most important words. Using this method they produced input that was classified correctly by humans but was overall nonsense to BERT. Similarly, Niven and Kao [16] attempt to examine what is informative to BERT in the Argument Reason Comprehension Task, where BERT must pick the correct warrant to follow a claim and a reason. They found some words, such as the word 'not' acted as a statistical cue that signaled it as an answer. Removing these words dropped BERT's accuracy dramatically.

Wang and Cho [17] demonstrate BERT's viability as a generative model by utilizing its ability to predict masked words. BERT faces challenges as a traditional language model because it is bidirectional and depends of the left and right context of a word in order to predict it. Wang and Cho circumvent this problem by providing BERT with a full sequence of masked tokens and predicting each one in a random order until the full sequence is unmasked. This method also allows BERT to receive noisy inputs by setting some of the masked tokens to random tokens. Using BERT in this manner generated more diverse sequences than OpenAI Generative Pre-Training Transformer [18], with the tradeoff of somewhat higher perplexity.

III. Methods

A. Classification

The network we use for this work is based on BERT, with a bidirectional LSTM, attention layer, and dense linear layer on top of BERT as a classifier (see the blue components of Figure 1). BERT has several advantages over previous methods. First, BERT performs well in a wide variety of contextually sensitive language tasks due to being able to detect when the meaning of a sequence has changed depending on context, allowing it to detect subtle differences in phrasing [6]. BERT also requires significantly less preprocessing of data than previous methods. The primary idea behind most prior work is to extract predefined features (such as bigrams or part-of-speech counts) from a sample and classify according to those features. BERT requires no predefining of features and is free to develop its own rules. The BERT model we use is the publicly available `bert-base-uncased` pretrained BERT model for PyTorch[1]https://github.com/huggingface/pytorch-pretrained-BERT.

We used the Ott corpus to benchmark the network and compare it to previous approaches. 80% of the reviews form the train set, which will be used to train the network. The remaining 20% become the test set, used to evaluate the network. In each training epoch, the training set is presented to the network in random batches of 8 until the entire set has been presented. Training lasted for 100 total epochs.

B. Part-of-Speech Ablation

As our first investigation into which parts of the input are the most important, we performed an ablation study on the network after training. In this study, we tagged each token of each review in the test set with its part of speech [19]. We then evaluated the accuracy of the network on the test set with each part of speech removed and replaced with a placeholder [MASK] token. This ablation was done 10 times for each part of speech, each with a freshly trained classifier.

C. Identifying Swing Sentences

In an alternative route to identifying informative parts of the input, we identified certain "swing" sentences that BERT considered highly informative for classification. To identify these sentences, we started with the trained classifier. Then, we formed a new dataset based on the original paragraphs, but with one sentence removed and replaced with [MASK] tokens. One-sentence entries are excluded.

Before

[CLS] We stayed for two nights for a meeting. [SEP] It is an upscale chain hotel and was very clean. [SEP] The service was very good, as the hotel front desk employees were kind and knowledgeable. [SEP] The rooms are decent sized and have soft mattresses. [SEP] The restaurant has good seafood, but was a bit expensive. [SEP] We would come back again. [SEP]

[1]https://github.com/huggingface/pytorch-pretrained-BERT

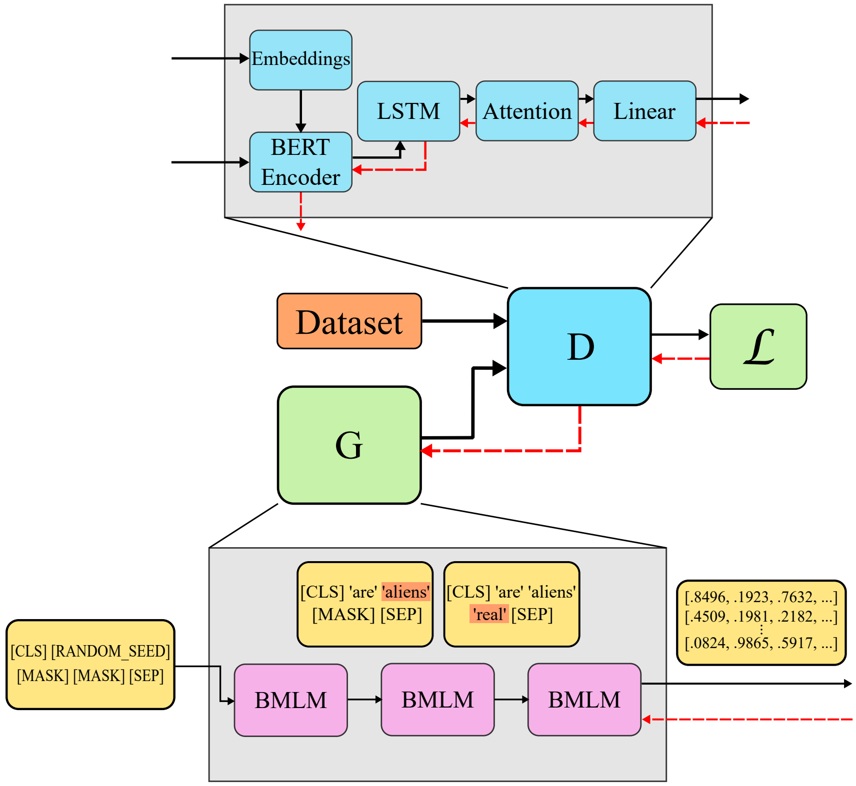

Fig. 1. A block diagram of our BERT-based Generative Adversarial Network. The discriminator (D) is composed of a BERT embedding layer, a BERT encoder, a directional LSTM layer, a self-attention layer, and a dense linear layer that produces the final classification. The embedding layer converts integer input (which represent tokens in BERT's vocabulary) and converts them to a set of 768 length float vectors. The encoder converts those vectors into a single 768 length vector that encodes the entire sequence. Samples drawn from a dataset are converted to integers and presented to the embedding layer, while generated samples, being already float vectors, are presented directly to the encoder. When training the generator (G), a sequence of [MASK] tokens plus a random seed token is first presented to a model of BERT for Masked Language Modeling (BMLM). The generator selects a token at random and predicts it, replacing the [MASK] token with its prediction. This process is repeated until all tokens are predicted, producing a full sequence which is then converted to float vectors. This tensor passes through the BERT encoder and up the discriminator. After the loss is calculated, it is backpropagated through the discriminator and the BMLM (shown by the red arrows). Because the generator must cast integers to form the intermediate sentences, only the last instance of the BMLM is differentiable and can be backpropagated, however as all the instances of BMLM share parameters this is enough to train the generator.

After

[CLS] we stayed for two nights for a meeting . [SEP] it is an upscale chain hotel and was very clean . [SEP] the service was very good , as the hotel front desk employees were kind and knowledgeable . [SEP] [MASK] [MASK] [MASK] [MASK] [MASK] [MASK] [MASK] [MASK] [MASK] [MASK] [MASK] [SEP] the restaurant has good seafood , but was a bit expensive . [SEP] we would come back again . [SEP]

If the base paragraph had previously been correctly classified, but removing a sentence from that paragraph causes BERT to switch classifications and label it incorrectly, that sentence was marked as one that was important to the original classification and therefore an exemplar of deception or truthfulness. We then analyzed those sentences' parts-of-speech to see if there are any observable differences between them. We track the mean amount of times a given part of speech appears per sample where they are used, the standard deviation of the same, and percentage of samples where they appear at least once.

It is worth asking why we are doing this analysis only on these swing sentences rather than on the whole dataset. The answer is simple: not every part of the text in the dataset is informative. It is inevitable that some parts of the input are essentially noise to BERT, not providing evidence either way. By limiting the analysis to these swing sentences, we narrow the domain to what BERT considers to be strong examples of truth or deception.

D. BERT-based GAN

The final method proposed to expose the patterns in truth and deception is a Generative Adversarial Network based on BERT, shown in Figure 1. In this setup, we declare the BERT model we have been using as a classifier as the discriminator. For the generator we use the BERT-based implementation by Wang and Cho [17], which takes advantage of BERT's ability to predict masked token to act as a generative model[2]https://github.com/nyu-dl/bert-gen. The ability of BERT to function as a generative model is vital if one wants to use

[2]https://github.com/nyu-dl/bert-gen

Source	Accuracy
Ott et al. [2]	89.8%
Vogler and Pearl [5]	87.0%
Xu and Zhao [10]	91.6%
Ren and Ji [12]	85.7%
BERT	**93.6%**

TABLE I
COMPARISON OF ACCURACIES ON THE OTT CORPUS.

BERT in a Generative Adversarial Network (GAN). Goodfellow et al. [20] created the GAN to be a unique network system that would allow a network to generate plausible samples by getting feedback from a discriminator network, usually a reliable classifier. By generating samples from latent variables composed of mask tokens and having those samples evaluated by the discriminator, the generator learns to create samples that can fool the discriminator into thinking that the sample came from a real dataset. This way, both the discriminator and generator utilize BERT.

The generator network exploits BERT's masked language model abilities. One of BERT's basic functions is the ability to predict the true identity of a masked word given its surrounding context [6]. We expand on the work of Wang and Cho [17] to allow BERT to produce entire sequences from scratch. First, an entirely masked sequence is presented to the generator, as well as a random seed token at the beginning to provide noise. The generator then selects a random token and tries to predict it, producing a probability distribution of the tokens that it could be, which is then sampled to provide its prediction. This new sequence is fed back into the generator, where a different random token is selected and predicted. This continues until all the tokens have been predicted, forming an entire sequence. A side effect of this iterative process is that the generator must sample its out to form integers to represent the intermediate sentences. This means that only the last instance of the masked language model is differentiable. However, since all the parameters are shared across instances, this does not harm noticeably harm the generator. The generator produces a sequence of 48 tokens before transforming it to a 50 token sequence by prepending a [CLS] token, which allows the discriminator to classify the sample, and appending a [SEP] token, which signals to BERT the end of a sentence. We then perform part-of-speech analysis on the samples that successfully fool the discriminator into believing that they are real samples, if any are produced.

We perform two runs of this GAN: once each for deceptive and truthful sentences. We use the Ott corpus to provide the real-world examples of both. This allows the BERT generator to generate its own examples to mimic what is truthful and what is deceptive. This will allow the generator to exploit the features that the discriminator is using to identify truthful and deceptive sentences. The advantage of this approach is that the generated sequences do not need to fool a human expert or even produce recognizable English; they just have to exploit the rules that BERT creates, which should shed some light on what those rules are. We perform part-of-speech analysis on the generated truthful and deceptive sentences to analyze the representational similarity between the two cases.

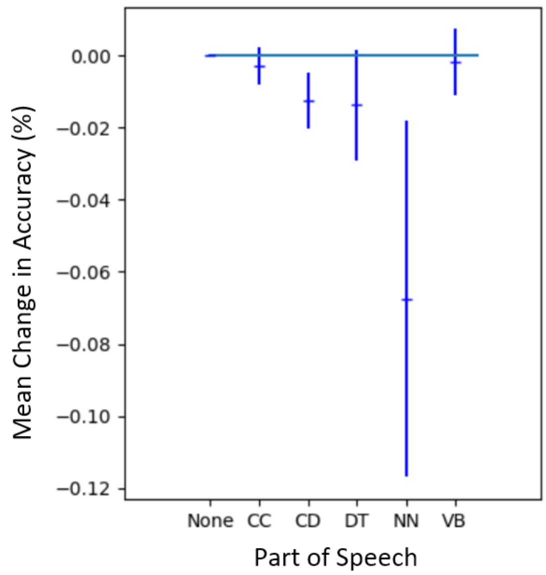

Fig. 2. The mean results of the ablation study over 10 runs. The error bars are the standard deviation. The removed parts of speech shown here are None Removed, Coordinating Conjunction, Cardinal Digit, Determiner, Singular Noun, and Verb.

IV. RESULTS

A. Classification

BERT reached an accuracy of 93.6% (table 1), an improvement of 2% over the next best method, beating the state of the art in deception detection on the Ott dataset. This jump in accuracy is significant since, unlike other methods which have the conditions and factors of interest baked into the model, BERT must learn its rules and features unsupervised. That allows BERT to find the best solution unrestricted by preconceived rules, and therefore attain the best accuracy. BERT has achieved the first step for this work: being able to accurately classify deceptive text, allowing us to investigate the methods it uses to do so.

B. Ablation

The ablation study (Figure 2) revealed that the network is insensitive to most parts of speech being removed, although some have a slightly stronger impact with one causing a particularly large reduction in accuracy. When the singular nouns (NN) were removed, the network accuracy dropped by 2 to 12 percent. This may indicate that singular nouns are a strong indicator of deception or truth; however given the prevalence of singular nouns in everyday language it is possible that removing them makes the review less comprehensible overall and harder to classify.

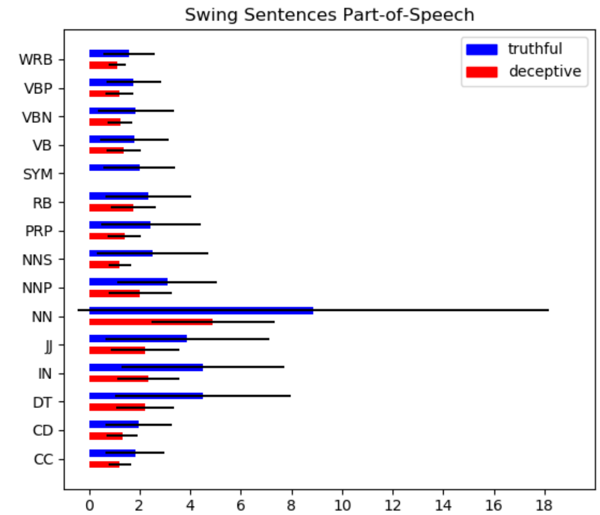

Fig. 3. The part-of-speech analysis of the swing sentences. Bar length indicates average number of occurrences per sentence with error bars representing standard deviation.

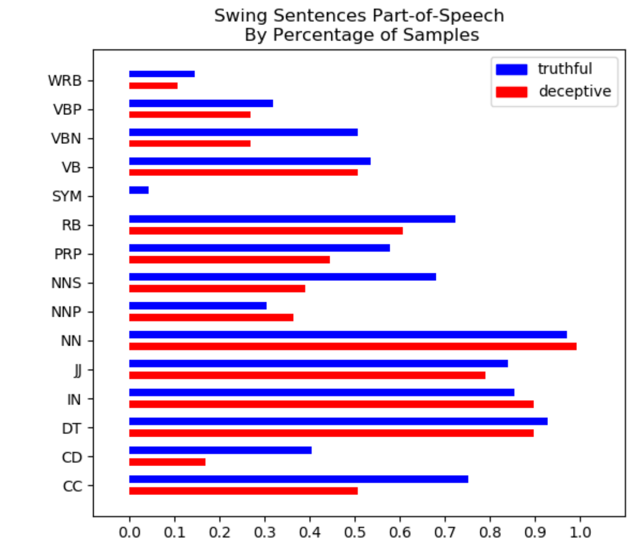

Fig. 4. The parts of speech by the percentage of samples they appear in at least once for swing sentences.

C. Swing Sentences

BERT identified 69 truthful swing sentences and 148 deceptive swing sentences. Examples from both classes are shown in the boxes below. The results of the part-of-speech analysis and percentage occurrence are shown in Figures 3 and 4. Many parts of speech occur less frequently in deceptive sentences than in truthful sentences, and the standard deviations tend to be much lower. Those same parts of speech also appear (at least once) in a higher percentage of samples for truthful sentences than deceptive sentences. It is possible that truthful sentences tend to have more varied parts-of-speech, and tend to be less consistent in which parts of speech are used. Deceptive sentences, meanwhile, seem to draw from a shallower pool and have less variation. This indicates that the deceptive sentences are more formulaic and follow a more consistent structure than the truthful sentences.

Truthful Swing Sentence

> As a royal ambassador member, they upgraded me to a beautiful junior suite with a separate living and working area and 2 bathrooms!

Deceptive Swing Sentence

> The Magnificent Mile in Chicago is a great place to visit, and staying at the Affinia Chicago just made it that much better!

D. BERT-based GAN

The BERT-based generative network was able to produce samples of text that were easily identifiable as truthful or deceptive to the classifer, if not to a human. There is a sharp drop in coherency in both truthful and deceptive text after training compared to before it is trained. Fortunately, readability of these samples is not necessary for them to be useful. When eighty generated samples were presented to the trained classifier from earlier, the classifier was able to identify all of them with 100% accuracy, even though it was only trained on the Ott data and never trained on generated samples. This indicates that the generated samples show strong resemblance to what BERT considers either ideal truth or ideal deception.

Samples of truthful and deceptive sequences that successfully fooled the discriminator are shown in the boxes below. The sentences were produced in all lowercase, with the [CLS] and [SEP] tokens added after the fact to fit BERT's input rules.

Untrained Generated Sequence

> [CLS] greyhound trains were running on behalf of the university , and shaw interested in improving access to the food markets and in the improvement of healthcare . the hospital was put under much pressure by the government , also underperformed at parliament in that year . [SEP]

Figures 5 and 6 show the results of the part-of-speech analysis on the generated sentences. Some of the same trends that are visible in the swing sentences are also shown here. This reinforces the idea that these trends are distinctive of truthful or deceptive text, however the increased difficulty of accurately tagging parts of speech in incoherent samples means that these results should not be taken with the same strength as that of the swing sentences. In particular, many of the standard deviations (with a small handful of exceptions such as base verbs ('VB') and prepositions ('IN')) are smaller in deceptive text than truthful text, again pointing to deceptive text being overall less varied. This lines up strongly with the results of the swing sentence analysis.

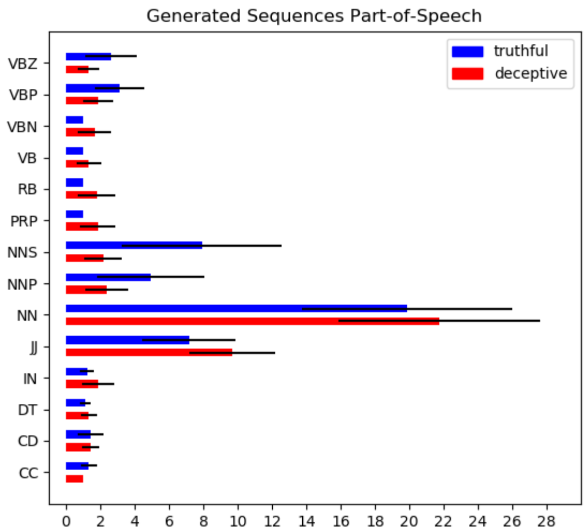

Fig. 5. The part-of-speech analysis of the generated sequences. Red bars indicate deceptive sequences, blue bars indicate truthful sequences. Bar length indicates average number of occurrences per sentence with error bars representing standard deviation.

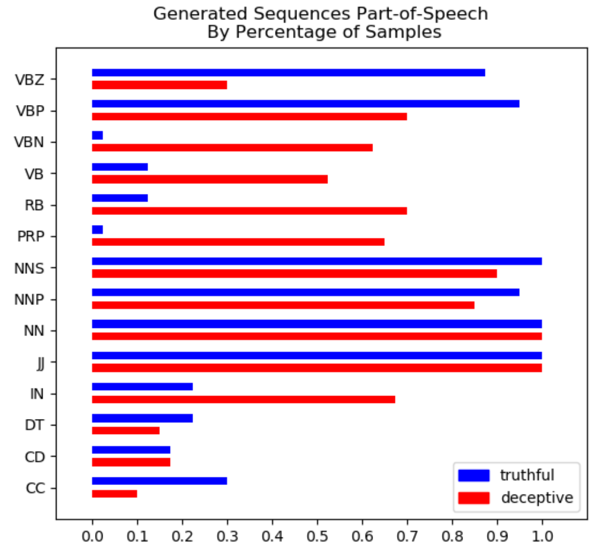

Fig. 6. The parts of speech by the percentage of samples they appear in for generated sequences. Red bars indicate deceptive sequences, blue bars indicate truthful sequences.

Generated Truthful Sequence

[CLS] can aliens aliens crimestellar aliens geek dinosaur armada nec aliens skulltsky ufo werewolf aliens cosmic aliens zombie aliens aliens titans predator predator police officers science lords battle armadabot predator chaos x spy warriors 3d police officers the aliens predator aliens zombie alien battleron aliens [SEP]

Generated Deceptive Sequence

[CLS] aria me spaced reading vatro for tom want tom complete me league recording action tom , men tom ” league short tom complete tom march home quick with league drop russian short home tom quick reserve speech soon tom ” short tom ” cut short ! [SEP]’

The “at least once” appearances do not match the results of the swing sentences, but they are similar in that they both correspond to the mean appearances per sample. If a part of speech has a higher mean rate of appearance per sample, that same part of speech will also be prevalent in more samples. This suggests that it is not the specific part of speech that indicates truth or deception, but the variation in their use.

V. Discussion and Future Work

In this work, we utilize BERT to understand what separates truthful text from deceptive text. BERT was able to beat state-of-the-art accuracy on the popular Ott Deceptive Opinion Spam Dataset. BERT’s ability to reach this high accuracy indicates that features distinguishing truthful and deceptive exist and can be exploited. Our ablation study revealed that removing parts of speech such as singular nouns hurts BERT’s ability to differentiate between truth and deception, indicating that certain word types are informative to the classification.

Our results with the swing sentences indicate that there is a high level of variation in truthful text, while the deceptive text was more formulaic in which parts of speech appeared. Our Generative Adversarial Network produced similar results, reinforcing our conclusion that there are underlying patterns in deceptive text that do not appear in truthful text.

We plan to refine the generative network to increase its stability and improve the quality of the generated sequences. This should allow us to generate larger disparities between truthful and deceptive sequences and more readable samples. We can use those disparities to further investigate the differences between the two text types. Also, while some trends have been indicated this does not mean they are the sum total of BERT’s self-created rules, and more can be done to expand BERT. We can modify the input, substituting phrases that are similar in meaning but different in language, which will allow us to see what can tip the classifier in one direction or the other. We plan to test and refine BERT on other corpora such as the Liar Liar fake news dataset to see if it can learn rules belonging to other text genres, as well as if the learned rules transfer from one corpora to another. Once the rules are determined, we can use them to train humans to better detect deception.

VI. Acknowledgements

EN and DB were supported by the National Science Foundation under grant 1640081, and the Nanoelectronics Research Corporation (NERC), a wholly owned subsidiary of the Semiconductor Research Corporation (SRC), through Extremely Energy Efficient Collective Electronics (EXCEL), an SRC-NRI Nanoelectronics Research Initiative under Research Task ID 2698.003.

REFERENCES

[1] N. R. Council *et al.*, *The polygraph and lie detection*. National Academies Press, 2003.

[2] M. Ott, Y. Choi, C. Cardie, and 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-volume 1*. Association for Computational Linguistics, 2011, pp. 309–319.

[3] T. R. Levine and C. F. Bond, "Direct and indirect measures of lie detection tell the same story: A reply to ten brinke, stimson, and carney (2014)," *Psychological science*, vol. 25, no. 10, pp. 1960–1961, 2014.

[4] M. Ott, C. Cardie, and J. T. Hancock, "Negative deceptive opinion spam," in *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: human language technologies*, 2013, pp. 497–501.

[5] N. Vogler and L. Pearl, "Using linguistically-defined specific details to detect deception across domains," *Natural Language Engineering*, vol. 1, no. 1, pp. 1–32.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[7] W. Y. Wang, """ liar, liar pants on fire": A new benchmark dataset for fake news detection," *arXiv preprint arXiv:1705.00648*, 2017.

[8] S. Feng, R. Banerjee, and Y. Choi, "Syntactic stylometry for deception detection," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*. Association for Computational Linguistics, 2012, pp. 171–175.

[9] B. de Ruiter and G. Kachergis, "The mafiascum dataset: A large text corpus for deception detection," *arXiv preprint arXiv:1811.07851*, 2018.

[10] Q. Xu and H. Zhao, "Using deep linguistic features for finding deceptive opinion spam," *Proceedings of COLING 2012: Posters*, pp. 1341–1350, 2012.

[11] J. Li, M. Ott, C. Cardie, and E. Hovy, "Towards a general rule for identifying deceptive opinion spam," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2014, pp. 1566–1576.

[12] Y. Ren and D. Ji, "Neural networks for deceptive opinion spam detection: An empirical study," *Information Sciences*, vol. 385, pp. 213–224, 2017.

[13] S. Hu, "Detecting concealed information in text and speech," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 402–412.

[14] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[15] D. Jin, Z. Jin, J. T. Zhou, and P. Szolovits, "Is bert really robust? natural language attack on text classification and entailment," *arXiv preprint arXiv:1907.11932*, 2019.

[16] T. Niven and H.-Y. Kao, "Probing neural network comprehension of natural language arguments," *arXiv preprint arXiv:1907.07355*, 2019.

[17] A. Wang and K. Cho, "Bert has a mouth, and it must speak: Bert as a markov random field language model," *arXiv preprint arXiv:1902.04094*, 2019.

[18] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*, 2018.

[19] E. Loper and S. Bird, "Nltk: the natural language toolkit," *arXiv preprint cs/0205028*, 2002.

[20] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.