Transformers Based Lie Detectors: Deceptive Opinions Detection Using Transformers on Single and Multiple Domains

Katz Shachar
shachar.katz@cs.technion.ac.il

Aspis Elad
eladaspis@campus.technion.ac.il

Abstract

In this paper, we present experiments in the recognition of deceptive language using transformers architecture. We show that transformers can get better results in distinguish between truth and falsehood opinions as expressed in text, comparing to old methods. We also show mix results with constructing multi domain classifiers using only small number of samples and testing it on many domains, including ones which are different from the subjects that uses for training, in order to create a universal deception detector.

1 Introduction

Deceptive Language is the notion of trying to confuse and persuade people, mostly with negative intention. Humans have a hard time discerning between a truth and a lie in a reliable manner, especially when we are talking about concepts from multiple domains. Those problems get bigger in today's digital reality, where we are flooded with textual data. Because of that, web companies and private users tries to filter trolls, automate bots and fake users who try to deceive different opinions, if it is on an E-market platforms or on a chat room talking about social or public opinions.

In this paper, we wanted to address the problem of detection of deception text and we chose to do so using transformer architecture, a relatively new deep learning method who shows great results in tasks like recognition semantics and context in NLP problems. We tried to answer two main questions: first, can we achieve better results to this problem comparing to old techniques with only small number of samples; and second, can we use it to detect a deception without the context of its domain, in order to build a global lie detector with relatively few resources.

2 Related Work

Addressing the problem of automatic detection of deceptive language in written text, our experiments are very much based on previous works who tries to address the same problem with classic ML technique, like SVM and Naive bayes (Mihalcea and Strapparava, 2009) and feature extraction (Appelgren, 2016). In those previous works, Researchers check the ability to detect deceptive opinions about subjects from different social and personal domains. Using a few dozens of samples, researchers achieved 70% accuracy by training and testing on the same domains, and only 60% accuracy when the training did not include samples from the tested domains.

Similar work (M. Ott, 2011) which also tried TRIGRAMS, BIGRAMS and LIWC only on the domain of hotels reviews, showed almost 90% accuracy while training using hundreds of samples.

Another study which focused on automatically creating and detection fake reviews in the world of E-commerce (J. Salminen, 2022) was able to achieve much higher results, which were between 80 % to 97% on computer-generated opinions, using models like SVM and transformers based one (fakeRoBERTa) by using thousands of samples for training the models.

In our study, we will use relatively small number of samples for training, but with more variance of domains in order to increase the diverse of the experiments and our model's ability to help in unknown domains.

3 Data

Each example in the database consists of an opinion and a label that gives an indication of whether this is a real opinion or a deceptive one.
In the data aspects, we have five types of datasets, each meant to represent opinions from different conceptual world.
Three datasets were constructed by Mihalcea and Strapparava (2009) which include 100 truth sentences and 100 lies sentences on three topics: opinions about abortion, opinions about death penalty, and opinions about best friend relationship. The dataset constructed using Amazon Mechanical Turk Service, where the human contributors were asked to describe their true opinions about the first two topics and then to try to describe the opposite opinion. The third topic, about best friend relationship, constructed by asking the contributors to write about their experiences and feelings with their best friend and later to try to write fake sentences about people which they are not really good friends with. The way those datasets were constructed asked the contributors to really try expressing their inner-feeling and emotions about ideas from different domains.
The 4th dataset is about hotels reviews and was created by Ott et al. (2011) and it includes 800 true reviews and 800 fake reviews. This dataset was also constructed using Amazon Mechanical Turk Service and contributors were asked to state their positive or negative experiences about hotels they indeed visited, and also to add reviews about hotels which they did not visit. This dataset is a representative example for the deceptive product review which becomes more and more present in today web-experiences.
The 5th dataset was created by J. Salminen (2022) is about product reviews from Amazon website. Although this dataset reminds the 4th one in the idea of describing opinions about products, this dataset is special compared to the four other datasets since it includes 40,000 samples which half of them are actually true opinions from Amazon website, but the other half was constructed automatically using GPT-2 model. Because of that, in our experiments we are not only checking the ability to detect humans-based deception but also computer-generated ones.

4 Model

Vanilla Transformers: We used RoBERTa (Robustly Optimized BERT Pretraining Approach) which is based on Google's BERT model released in 2018. Using Hugging Face API, we are only fine-tuning baseline models with different input each time, according to our task.
Since the smallest datasets we have is consist of 100 true samples and 100 deceptive samples, and since we wanted to focus on the task of training with only small number of samples, for each domain we randomly selected 200 samples (100 true and 100 deceptive samples) and split them to 160 samples to be used for training and 40 samples to be used for testing (80-20 ratio). While building each classifier, we checked different numbers of samples to use for the training sets: while training with only one domain (dataset) we used the size of 40, 80, 120, 160 samples; when training on multiple domains, the sizes of the training set were the product of the number of domains with 40/80/120/160 samples. For example, if we are using three domains for training, and each domain contributed 80 samples, the training set we will have $3 \cdot 80 = 240$ samples. Each classifier was tuned with 2-12 epochs and with different learning rates, between 1e-5 to 7e-5.

5 Experiments and Results

Our goals were to try to solve the task of deception detection using transformers, so we first tested training the classifiers using only one domain, while also to experiment the properties of cross domain adaptation in this field, for which we trained our classifiers using multiple domains.

5.1 Training With a Single Domain

Table 1 shows the best results we got for the basic classifiers for each domain, when each classifier who construct on a specific domain was tuned on the same domain and tested across all five domains.
With average accuracy of 84.5 for training and testing on the same domain, we can see that comparing to other methods like SVM and Naïve Bayes, which achieved only around 70%, transformers have better results for the problem of deception detection.
Looking into the sizes of the best model we found, we see an obvious advantage to models which uses more samples while training, which is a common notion in the fields of deep learning (the more samples we have the better our results will be). However, those results are also not much lower than the results we saw in other papers which trained transformers based model using

thousands of samples (J. Salminen, 2022). Because of that, we can understand that for different situations and constraints, we might be able to save resources and time while training and evaluating our models, with the price of just a bit weaker models.

The results also teach us that there is no promise to get high results on a domain which was not involve in the training, however, we can see cases where this notion does work, like training with "abortion" dataset and testing on "death penalty". We assume those results lead to the hidden connection about those different domains and the ways that they were created, like same method to try to disguise the origin and semantic of each sentence.

Regarding the differences between the best accuracies results we got for each domain, like 87.5% for "abortion" and 97.5% for "Hotels", we assume this is the result of the unique characteristics of each domain and the way that deceptions were created for each of them. We find this a good thing since it helps us establish the claim that we are experimenting a method for a wide range of domains and semi-tasks.

We will mention that we notice that under 40 samples for training – the classifier is just a bit better than random guess, including for the average accuracy for a generalized classifier, but from 80 samples and on – they actually show better and better results.

5.2 Training With Multiple Domains

In table 1 we present some of the results we got for training with more than one domain and with different number of samples. The results show that while training with two domains, we achieve similar accuracy results for each of the two domains we trained with, comparing to training and classifying each domain separately. However,

Training set (number of samples from the domain)					**Total size of the training set**	Accuracies results on testing set (in percentages %)					
abortion	**Amazon**	**Best friend**	**Death penalty**	**Hotels**		**Abortion**	**Amazon**	**Best friend**	**Death penalty**	**Hotels**	**Average**
40					40	67.5	57.5	52.5	55	60	58.5
80					80	80	50	57.5	67.5	50	61
120					120	87.5	47.5	75	72.5	50	66.5
160					160	87.5	47.5	52.5	80	50	63.5
	120				120	50	85	60	52.5	45	58.5
		160			160	62.5	57.5	85	52.5	47.5	61
			160		160	77.5	37.5	52.5	70	52.5	58
				160	160	50	47.5	50	65	97.5	62
80		80			160	85	57.5	82.5	52.5	47.5	65
160		160			320	80	35	80	75	47.5	63.5
80			80		160	82.5	35	52.5	65	52.5	57.5
160			160		320	85	47.5	60	70	45	61.5
	80			80	160	65	45	85	65	97.5	71.5
	160			160	320	50	45	82.5	55	100	66.5
		40	40		80	50	50	72.5	50	50	54.5
		80	80		160	85	50	87.5	75	55	70.5
		120	120		240	70	60	85	72.5	85	68.6
		160	160		320	80	70	70	67.5	47.5	67
160		160	160		480	90	57.5	85	77.5	55	73
160	160		160		480	85	100	70	70	65	78
	160	160		160	480	52.5	92.5	87.5	50	95	75.5
160	160	160	160		640	67.5	95	77.5	70	67.5	75.5
160	160	160		160	640	87.5	97.5	85	67.5	97.5	87
160	160		160	160	640	80	100	52.5	62.5	100	79
160		160	160	160	640	90	50	82.5	72.5	95	78
	160	160	160	160	640	70	90	82.5	77.5	97.5	83.5
40	40	40	40	40	200	77.5	52.5	70	65	77.5	68.5
80	80	80	80	80	400	77.5	75	80	65	87.5	77
160	160	160	160	160	800	85	95	87.5	70	92.5	86

Table 1: accuracies result of different training sets

the generalization accuracy across all domains, is only a bit better then random guess (around 65% in most of those cases across the out-of-training domains), which is only a bit better than the results we got for out-of-training domains under homogeneous training (61.9% accuracy). We decided to attach those results to emphasize that only one or two domains cannot represent wide range of deceptions using those techniques.

Now, we wanted to see how increasing even more the variance of domains while training will affect the generalization accuracy. For that task, we apply the concept of "leave one domain out" by training with 3 or 4 domain and testing on the rest of the domains (which are not included in the training). With those methods, we achieved 70% accuracies for the out-of-training domains, which are better then the results we saw in other ML method (60%, Mihalcea and Strapparava, 2009), but unfortunately, we saw that we do not get high generalization results with this approach. This reveals one of the human sides of deception and lies: you need to be familiar with the field you are talking about in order to understand if a sentence is an honest one or a deception. For that reason, we believe that in order to create a classifier, especially transformers based one, for the purpose of deception detection in a specific field it must have some samples from that field or at least from a similar one.

Another interesting discovery that we can see from the last experiments is that when we construct a classifier that contain multi domains samples, we would achieve almost the same results as we would have if we built a different classifier for each domain separately, and even sometimes we would get better results by this way for parts of the testing domains. That means that in problems of many domains, with given samples from all those domains, we can count on building only one classifier and to spare a phase of separating and splitting sentences into different groups and then running them on homogeneous classifiers. The last method can save time while testing, however, might be costly for the time of training (in this paper we did not give attention to the time that each training takes but as we know from previous works with transformers: the more samples your training set has – the more time it will take to train it).

6 Conclusions

In this paper we explore transformers ability to recognize deceptive sentences using small number of samples, which is a relatively new application of technique for this task. The model we created show great results when the testing domains are also included in the training set of the classifier, which answer our first main question: can transformers detect deception opinions in text using only small number of samples. In the field of cross domains, we have mixed results with low results on out-of-training domains and great results with training once on more than one domain or on similar domains.

Those results mean we might have benefits of using multi domains deception detection classifier on new domains that yet was tuned with our classifier, if we believe it has some connections and similarities to at least one of the domains that already used for training, like semantics, figure of speech and concepts. This might be used to detect new deception in social network on a new and hot topic from the news, or to detect deception about new brands in the world of E-commerce.

Expect of the problem of deceptive detection, this paper can be seen as a case study of transformers, their ability to learn patterns with relatively small number of samples, and especially to handle many-domains classification simultaneously. Also, we see an opportunity to handle topics like we just suggested, with wide-domain-transformers-based-classifier – which might be the next step in detection of lies language.

References

Joni Salminen, Chandrashekhar Kandpal, Ahmed Mohamed Kamel, Soon-gyo Jung, Bernard J. Jansen. 2022. Creating and detecting fake reviews of online products. Journal of Retailing and Consumer Services.

Mattias Appelgren. 2016. Detecting Deception using Natural Language. University of Edinburgh, Scotland.

Rada Mihalcea and Carlo Strapparava, 2009. The lie detector: explorations in the automatic recognition of deceptive language Proceedings of the ACL-IJCNLP 2009 Conference Short Papers (2009), pages 309-312. Association for Computational Linguistics, 2009.

Myle Ott, Yejin Choi, Claire Cardie, T. Jeffrey, Hanco ck. 2011. Finding deceptive opinion spam by any stretch of the imagination. arXiv preprint arXiv:1107.4557 (2011)

The code of this paper can be found in: https://github.com/shacharKZ/Lie_Detector_NLP_transformers