Enhancing RobEn to Defend Against Adversarial Synonym Attacks

Sahil Farishta
University of Michigan
sahilf@umich.edu

Abstract

Adversarial input into Natural Language Processing models can very easily affect the system's accuracy. These attacks are increasingly concerning as these models are incorporated into many fields, including conversational AI to help answer medical questions. If these models are susceptible to adversarial inputs, then the outcomes could be disastrous. Previous attacks have shown increasing levels of success against the state of the art models, including BERT. Defense algorithms are catching up, with many successfully deterring attacks in specific scenarios. The RobEn defense technique was shown to be extremely robust against adversarial typos, where an attacker will make typos in the words of the original input which will lead to misclassification of the perturbed input. We show that this technique can be extended to protect against adversarial synonym substitution, which involves an adversary making one or more synonym substitutions within the input to change the classification of the input. The technique presented can deter state of the art adversarial attacks while maintaining high accuracy on the tasks at hand.

1. Introduction

In the past few years, we have seen a rise in the number of adversarial attacks against Natural Language Processing (NLP) models. Adversarial examples are input to a machine learning model that slightly perturb a normal input in a way that changes the output from the model. Recent attacks such as the TextFooler[2], CLARE[4], and the RewritingSampler[13] are able to cripple state of the art machine learning models such as BERT[1] using adversarial examples.

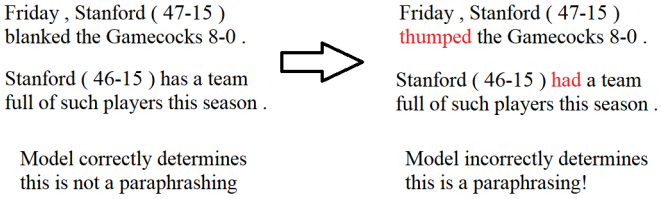

Figure 1. An example of how small perturbations in the input text can lead to an incorrect classification.

As such, we need defenses against these methods as they clearly show that our models are easily susceptible to attacks. Figure 1 demonstrates how an adversarial example can appear to a human to be incredibly similar to the original input with the perturbations highlighted in red. These small changes are enough to cause the model to interpret the input differently.

1.1. Motivation

As NLP models become integrated into various aspects of our lives, such as voice assistants, chatbots, and text generation, we need to make sure these models remain safe. A model that is weak to adversarial examples may be exploited to provide misinformation, private or secure information, or circumvent security.

Even outside of adversarial examples, these models are extremely brittle and subject to poor performance under small perturbations in input. This is especially important as these models begin to be implemented in high stakes scenarios. Recently, Open AIs GPT-3 state of the art model was trained to create a chatbot for people having medical questions. Researchers were able to show that the model could be prompted to advise a patient to kill themself [8]. This is incredibly bone-chilling to see from what is considered to be a state of the art model that has been trained for many thousands of hours.

These potential vulnerabilities suggest we need a way to strengthen the robustness of our NLP models to prevent adversarial attacks and perform well across a wide range of diverse user inputs, especially in high stakes scenarios.

1.2. Related Work

One of the state of the art NLP models is the BERT model which performs left and right contextual tranformations to classify input [1]. This means the model will consider the sentence as a whole in its context, rather than one word at a time. This is very powerful since words can have different meanings depending on the contexts in which they appear. This model is considered to be the state of the art for both entailment and classification tasks. It has seen widespread usage in many applications. Many adversarial attacks target BERT to prove their power and defenses try and strengthen BERT against these attacks.

The primary paper we consider is the RobEn paper[3] which shows how clustering algorithms can be used to strengthen models such as BERT against typo based attacks. In the paper, they mention that it could be possible to extend the work done to protect against synonym based attacks which is what we do here. It provides a module that can sit on top of any NLP module to provide state of the art robustness against typo based attacks. The defense uses a clustering based approach to protect against pertubations. In the defense, the authors construct clusters of words such that any two words that are 1 edit distance away from each other are placed into the same cluster. An edit distance of 1 means 1 deletion, substitution, or addition of a character within the word leads to the second word. Any two words that are in the same cluster will be encoded to the same binary representation before being passed into the NLP model. This can lead large numbers of words being encoded in the same manner which may be undesirable since as long as there is some series of 1 edit tranformations that can be applied between two words that may be vastly different, they will be encoded the same. The authors showed that even with this downside, the accuracy of the robust model still remains high.

Additionally, to combat this, they present an agglomerative clustering technique which aims to balance the number of words added to each cluster. Many of the techniques presented in this paper are used for the defense we present here.

Additional adversarial defenses have been proposed for various NLP models. For typo based attacks, previous work has looked at using RNNs to identify and protect against adversarial examples[7][9]. Additionally, previous work has been done in protecting against synonym based attacks. The Synonym Encoding Method [12] used GloVe embedding distance to identify and protect against potential perturbations, especially against unseen words. This defense, while effective against state of the art attacks, does not have an easy way to integrate with other adversarial defenses, such as the typo based defense presented in the RobEn paper. While we could use two modules consecutively to defend against both, it would be computationally more effective and simpler to combine the two defenses. We will show that, although in this paper we do not combine the two defenses, our synonym based defense is compatible with the architecture provided in the RobEn paper, suggesting future integration could be achieved.

1.3. Work Performed

In this paper, we show that we can use the technique presented in the RobEn paper to protect against synonym based adversarial attacks. We do this by generating clusters of words that are synonyms of each other and assigning the same encoding to each of these words. We wanted to combine this work with the work in the RobEn paper to protect against adversarial typo attacks as well. However, due to difficulties in maintaining cluster size, using the simple connected components clustering algorithm led to the majority of words in most sentences being mapped to the same encoding which is extremely undesirable. Using advanced techniques such as the agglomerative clustering algorithm presented in the RobEn paper to alleviate this issue turned out to be computationally far to expensive to complete in the time available. As such, we present the same architecture as the RobEn paper applied to protect against state of the art synonym based attacks.

2. Methodology

To strengthen the model, we use the clustering algorithm presented in the RobEn paper to influence the training process. We first begin by generating synonyms for each word in our vocabulary and adding it to an undirected graph. The graph is then used to generate clusters of similar words. The generated clusters are incorporated in the training process for the underlying NLP model in order to make it more robust against synonym based attacks.

2.1. Synonym Generation

To generate synonyms, we use the NLTK WordNet database[10]. This dataset is designed to find synonyms for words as similar words are grouped in SynSets. These SynSets are segmented based on the different contexts the word can appear in, such as adjectives, nouns, and verbs. Each word was added to a graph as a node with an undirected edge connecting them if the edit distance between the two was 1 or if the two words were a synonym of each other. There can be many synonyms for a single word across different contexts, so in order to limit the number of synonyms added to the graph, we experimented with a hypeparameter, N, which set an upper bound on the number of synonyms that could be added to the graph per context for a single word. We noticed that many words were being clustered with stopwords which could potentially lead to lost meaning and destroying the grammar of the sentence. We created a set of models that filtered out stopwords during the clustering process to compare our results with. We use the same stopword list presented in the TextFooler attack [2].

We also considered using the cosine distance between the GloVe embeddings[6] of the words to determine if two words are synonyms. This is not necessarily as strong of a sign of two words being semantically similar to humans. Additionally, the clustering algorithm is designed to make sure similar words have similar word embeddings to make them more robust to substitutions, but in this case, we would be clustering words together that already have similar embeddings, decreasing the utility of the clusters. Instead, the synonyms generated by WordNet may have vastly different embeddings which means the clustering algorithm will result in them being brought closer to each other. Similarly, BERT embeddings could be used by determining the cosine distance of these embeddings; however, BERT embeddings are highly contextualized and will change based on the surround sentence. As such, for the clustering method presented here which takes into account words with no sentence context, this would not be a great fit. Future potential work could explore combining the embedding distance for the generated WordNet synonyms to ensure clustered words are close synonyms.

2.2. Clustering

The clustering algorithm used was identical to the one presented in the RobEn paper[3] with additional edges connecting words that are synonyms of each other. We used connected component encodings which means that if two words have an edge between them, they will be a part of the same cluster. All words in the same cluster will be assigned the same encoding when transformed from a word into the vector representation. By making sure all words that share an edge are clustered together, we maximize the stability of the model as any words that are synonyms of each other will always be assigned the same word encoding. This makes them as robust as possible, but can hurt accuracy as the model is no longer able to finely distinguish words from each other. The notion of how well unperturbed inputs map uniquely to various encodings is called fidelity. Additionally, words that are synonyms of a common word, but they themselves are not synonyms of each other, will be clustered together which can be undesirable.

When considering the generated synonym, clusters, we assume that the adversary has access to the clusters. That is, it is not a secret which words are encoded similarly. This could be reverse engineered through multiple queries through the model or analyzing the binaries the model relies on. None of the current attacks are capable of taking this information into account, so in the future, we would need to experiment to see if having the clustering information available could allow an attack to circumvent the defense measures.

Future work could look at using alternative clustering methods could be used such as agglomerative cluster encodings as discussed in the RobEn paper[3], which perform a tradeoff between stability and fidelity. This would allow for higher accuracy as words that are synonyms of a common word but are not synonyms themselves may no longer be treated the same by the encoding function. The agglomerative clustering process is computationally expensive, taking multiple days to generate the clusters each of which are many gigabytes in size. Due to this cost, we weren't able to use them in our research here but future research could investigate the use of these techniques.

2.3. Training

During the training process, the system begins with a base model, such as the BERT for sequence classifier. Any model could be used here that learns a transformation function. We use the BERT model as it considered the state of the art in terms of NLP. The model is trained on the dataset learning a function, g, which takes in an embedding for the text input and outputs a classification for the input. Formally, g is of type $Z \longrightarrow Y$ where Z is the embedding domain and Y is the output domain.

To transform a sentence from the written language representation into the embedding domain, we use the the encoder function $\alpha : X \longrightarrow Z$ where X is the input domain (sentences). We use the GloVe model as our encoder function, with additional constraints from the generated clusters. The clustering algorithm dictates that if X_1 and X_2 are in the same cluster, then $\alpha(X_1) = \alpha(X_2)$ which means the underlying model does not distinguish between the two words as they appear the same to it. The resulting functions are used for both training and evaluation. The underlying model is trained in the same fashion as it normally would using this modified encoding function dictated by the clustering algorithm. During evaluation, the modified encoding function must also be used to ensure that inputs are encoded as they were in the training phase before being fed into the model. This system allows any NLP model to be protected by the clustering approach, as we simply need to add a layer to the encoding function while leaving the rest of the model training and evaluation process the same.

3. Experiments

To demonstrate the robustness of this defense, we considered multiple experiments that the the performance of each trained model. We run our tests on the Microsoft Research Paraphrase Corpus (MRPC) from the GLUE dataset [11]. This corpus gives the model two sentences and asks it to determine if they are paraphrasings of each other. We run this experiment across the base BERT models and the ones that have been strengthened by various defenses.

3.1. General Accuracy

The first test we run is the general accuracy test where we run the model on the MRPC corpus without any adversarial tests. This allows us to gauge the models accuracy on normal test cases. It is important that we maintain a high accuracy here as otherwise the model is not useful for the general tasks.

3.2. Typo Attack

The typo test was designed to test the robustness against typo attacks. This attack was presented in the RobEn paper [3] and performs a greedy based substitution attack. Originally, we intended to demonstrate that the robust encodings for both typo and synonym based attacks could be used simultaneously. However, after testing, we found that using these defenses in conjunction with each other resulted in the clusters having too many elements within them which lead to the encodings breaking down. The vast majority of words were mapped to the same encoding as the word "the". This led to the model always outputting true which is what the majority of test examples were labelled as. In the future, using agglomerative clustering could help as this would balance the number of elements placed in each cluster. For now, in this paper, we focus on only the synonym clustering defense.

3.3. Synonym Attack

The synonym test was designed to test the robustness against synonym attacks. The attack we used was the TextFooler attack [2]. This attack was shown to achieve state of the art performance against robust models like BERT. This was implemented by the TextAttack library [5]. During this test, the TextFooler attacker will make various synonym substitutions that aim to lead to a false classification of the input text.

Model	Normal Accuracy	Accuracy After TextFooler Attack
Base BERT	0.877	0.152
RobEn BERT	0.809	0.189
Synonym Encoded BERT	0.755	0.6716
3 Synonym Encoded Bert	0.745	0.6985
Stopwords Filtered Synonym Encoded Bert	0.7525	0.6446
3 Stopwords Filtered Synonym Encoded Bert	0.7745	0.5980

Table 1. Accuracies for the various models after running experiments on the MRPC dataset

3.4. Results

We run the general accuracy and the synonym based attack tests on various models. The first model is the base BERT which we would expect to do poorly against an attack as it has no robust defenses to protect itself. The second model we considered was the model presented in the RobEn paper which was designed to protect against typo based attacks. Since TextFooler is a synonym based attack, we would expect the performance of this model to not be significantly better than the base BERT model. We check the robustness of this model against the TextFooler attack to see if the typo based defenses can adapt to the synonym based attack.

The other four models that we use are models we trained to be robust against synonym based attacks. The first model uses WordNet to generate all synonyms for a word and adds the edges between them to the clustering graph. The second model limits the number of synonyms generated per context to 3 words each. The final two models perform the same steps as the previous two, but they filter out stopwords when generating the clusters.

Table 1 shows the results of running each of the 3 experiments on the 6 models discussed on the MRPC dataset. We see that the performance of the base BERT model and the RobEn model is very poor against the TextFooler attack. We also see that all models keep a high level of accuracy on the non-adversarial test case. Additionally, the four models that were trained to be robust against the synonym based attack were able to withstand the TextFooler attack, with the model that only considered the top 3 synonyms while not filtering out stopwords performing the best. Having an accuracy loss of around 6% is significant improvement over the base model which had a post attack accuracy of just under 19%.

	Predicted True	Predicted False
Label True	231	48
Label False	75	54

Table 2. Confusion Matrix for 3 Synonym Limited Encoded BERT shows a skewed distribution of false positives and true negatives

	Predicted True	Predicted False
Label True	157	122
Label False	23	106

Table 3. Confusion Matrix for 3 Synonym Limited Encoded BERT shows a better distribution of false positives and true negatives despite the biased dataset

We look at the confusion matrices for the top two performing models to ensure that there is not significant bias in the trained models. Table 2 shows the confusion matrix for the 3 Synonym Limited Encoded BERT and Table 3 shows the confusion matrix for the Stopwords Filtered Synonym Encoded BERT. We see that the distribution of false positives and true negatives is skewed towards predicting true for the 3 Synonym Limited Encoded BERT while for the Stopwords Filtered Synonym Encoded BERT is remains relatively even. This suggests that although the accuracy of the Stopwords Filtered Synonym Encoded BERT is lower on this test case, it may be a better model.

4. Discussion

With this work, we have shown that the clustering algorithm presented in the RobEn paper [3] can be extended to protect against state of the art synonym based adversarial attacks. While it is not able to fully negate the attack, it is able to significantly improve the robustness of the model. The model still performs well on non adversarial examples which means the model remains useful in both adversarial and non-adversarial cases. Due to the way the clusters lead to all words that connected via an edge to be mapped to a single encoding, we must limit the number of edges created as otherwise too many words will be clustered together. This severely weakens the expressive power of the model. To be able to provide support for both typos and synonyms, a smarter clustering algorithm must be used such as the agglomerative clustering algorithm presented in the RobEn paper. Even when limiting the number of synonyms to only 1 to 3 per word, we saw in testing that the performance of the models degrades significantly. Almost all examples had every single word encoded to the word "the". Only when excluding typo edges were we able to get meaningful results from the encodings.

When considering only synonym based attacks, we still see that the clusters can grow to be too large and some words semantic meaning are lost. To combat this, we filter out stopwords in the clustering process. For example, the sentence:

```
In midafternoon trading, the Nasdaq
composite index was up 8.34 , or
0.5 percent , to 1,790.47.
```

was originally encoded as

```
a midafternoon a, the nasdaq complex
a a a a, a 0.5 a , to a.
```

using the clusterer that does not filter out stop words. We see that some of the meaning from the sentence is now missing with many 'a's substituted in. While the encoded sentence does not necessarily have to be understandable by a human since it is never seen or used when interacting with users, the loss of information could degrade performance. Intuitively, this sentence really doesn't help us identify what is happening, other than the Nasdaq did something in the midafternoon. If we filter out stopwords during the clustering process, then we are left with the following sentence:

```
in midafternoon be, the nasdaq complex
be was up be, or 0.5 be, to be.
```

which still loses some information but does retain more important fragments of the sentence, such as the fact that the Nasdaq went up. Given this new information, when confronted with the second sentence:

```
The Nasdaq Composite Index.IXIC dipped
8.59points or 0.48percent to 1,773.54.
```

we see now the second model should retain the information needed to conclude this is contradiction. While this process is not perfect as there is more information lost than desired, it is at least able to retain the important information. Better clustering methods would lead to a higher rate of information retention while also providing the robustness seen in the defense. Additionally, we see that numbers are not handled extremely well, so investigating different ways to deal with numbers as they appear in the sentence could lead to a better defense.

Additionally, the accuracy post attack may be a lower bound for the accuracy we would expect after the attack. The reason for this stems from the fact adversarial attacks such as TextFooler have content shift issues, where the meaning of the sentence might change due to the attack [2]. For example, consider the following sentence from the MRPC dataset:

```
University of Michigan President
Mary Sue Coleman said in a statement
```

The TextFooler attack transformed it into the following sentence

```
Loyola of Ohio Jefe Mary Sue Coleman
contends in a explanation on the
university 's Cyberspace scene
```

Even if the classification shifts, it may not be adversarial in nature since the actual meaning of the sentence

has changed. These are two completely different contexts that would not fool a human as being similar. The burden is on adversarial attacks to maintain the context used in their adversarial examples.

5. Future Work

In the future, we would like to explore using different clustering algorithms, such as the agglomerative clustering technique presented in the RobEn paper. This would allows us to combine the different robustness encodings such as the typo and the synonym based defenses. Additionally, we could consider different techniques for determining which synonyms to include in the clustering. Some approaches may be only including two words as a part of the same cluster if they meet some similarity score. Additionally, we could use the BERT and GloVe encodings to confirm that two words are semantically similar. For each synonym suggested by WordNet, we could compute the cosine similarity between the GloVe and BERT encodings and only add the edge connecting the two words if they have a small enough distance.

Additional work in the future would be testing the defense across different test sets along with protecting against different attacks. Using a combination of these tests would allow us to find ways to improve our defenses even more. For example, some attacks, such as the ParaphraseSampler[13], attack examples at the sentence level rather than making word substitutions. The clustering approach used here is only able to provide defense against word based substitutions. It can handle multiple substitutions across the sentence but not rearranging of words or sentence level substitutions. Combining the clustering algorithm provided to use encodings that work on the sentence level, such as the BERT encodings may help protect against these attacks. Additionally, by benchmarking performance on different datasets with various attacks, we may be able to better compare the performance of this defense with state of the art techniques. Many of these attacks and test suites are computationally more expensive, taking a lot longer to run and generate results. This made it difficult for us to use them as tests in our experiments as we had limited time during a school semester to finish this project.

6. Conclusion

The clustering algorithms presented in the RobEn paper can be extended to defend against synonym based attacks by adding edges to the graph between any two words that are synonyms. Like the defense presented in the RobEn paper, this technique can be used as a layer added onto any NLP model to provide robustness. We were able to demonstrate that the accuracy of the model in non-adversarial case remains very high, while remaining robust against a state of the art synonym based attack. Though we weren't be able to demonstrate that the typo based and synonym based defenses could be combined due to the clusters growing too large in size, we were able to show each defense worked in isolation. Future work could combine the two defenses using techniques such as agglomerative clustering.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *North American Chapter of the Association for Computational Linguistics*, 2019. 1, 2

[2] Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. *Association for the Advancement of Artificial Intelligence*, 2020. 1, 3, 4, 6

[3] Erik Jones, Robin Jia, Aditi Raghunathan, and Percy Liang. Robust encodings: A framework for combating adversarial typos. *Association for Computational Linguistics*, 2020. 2, 3, 4, 6

[4] Dianqi Li, Yizhe Zhang, Hao Peng, Liqun Chen, Chris Brockett, Ming-Ting Sun, and Bill Dolan. Contextualized perturbation for textual adversarial attack. *North American Chapter of the Association for Computational Linguistics*, 2021. 1

[5] John X. Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp, 2020. 4

[6] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. *Empirical Methods in Natural Language Processing*, 2014. 3

[7] Danish Pruthi, Bhuwan Dhingra, and Zachary C. Lipton. Combating adversarial misspellings with robust word recognition. *Association for Computational Linguistics*, 2019. 2

[8] Anne-Laure Rousseau, Clément Baudelaire, and Kevin Riera. Doctor gpt-3: hype or reality?, 2020. 1

[9] Keisuke Sakaguchi, Kevin Duh, Matt Post, and Benjamin Van Durme. Robsut wrod reocginiton via semi-character recurrent neural network. 2016. 2

[10] Princeton University. About wordnet, 2010. 3

[11] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *In International Conference on Learning Representations*, 2019. 4

[12] Xiaosen Wang, Hao Jin, and Kun He. Natural language adversarial attack and defense in word level. *International Conference on Learning Representations*, 2020. 2

[13] Lei Xu, Ivan Ramirez, and Kalyan Veeramachaneni. Rewriting meaningful sentences via conditional bert sampling - and an application on fooling text classifiers. 2020. 1, 7