Does IBP Scale?

Matthew Wallace
Rishabh Khandelwal
Brian Tang
UW–Madison

ABSTRACT

Works on adversarial example generation for NLP models have proliferated in recent years. Unfortunately, defensive works have not kept pace. There is currently a dearth of defenses leaving only one robust defense against adversarial word substitution: interval bound propagation (IBP). Using IBP as a certified defense allows one to prove robustness against a large set of adversarial perturbations on a given text. However, this defense technique is not without its drawbacks. Currently, IBP has not been adapted to work with newer, better performing contextual embeddings. This work seeks to understand a) The generalizability of IBP for state-of-the-art models and datasets b) How IBP scales when applied across numerous popular NLP tasks. Throughout our experiments, we also examine the dependency of IBP on the depth of the network and complexity of task involved.

1 INTRODUCTION

Deep neural networks (DNNs) have been shown to excel in many domains including visual tasks as well as natural language processing (NLP) tasks. However, it has been shown that neural networks generalize poorly to attacks known as adversarial examples [6, 17]. In traditional machine learning (ML) settings, a neural network can be used as a classifier, learning to label each input as a particular class. In the setting of adversarial machine learning, an attacker generates perturbations on an input x resulting in a new input, an x', with the goal of changing the classification output. While most work involving adversarial ML is in the image domain, recent work in adversarial natural language processing demonstrates that these adversarial examples can easily be found by performing some actions with the text. For example, Li et al [13] used character-level substitutions to generate adversarial examples, but these can be filtered out via a spellchecking system. Certain concatenation based adversarial example generation also work well in adversarial settings [10], such as adding distracting text [11] to the input to fool a reading comprehension classifier or paraphrasing the text [9] to significantly degrade the performance of the model. Alzantot et al [1] discovered using word-level substitutions, like replacing words with synonyms, which our work aims to further explore. An example of such an adversarial example taken from [12] is shown in Figure 1. This suggests that NLP models are extremely brittle to the adversarial perturbations of source texts. Modest improvements to robustness can be found from using a technique known as adversarial training [14], but adversarial training in NLP has several drawbacks: a) In NLP, the number of possible transformations scales exponentially with text length and b) The defense is not guaranteed to be effective against newer perturbations that an adversary could come up with. To combat the growing number of attacks and provide certified robustness, Jia et al [12] proposed a defence against word substitutions using interval bound propagation (IBP). Further background on their work is provided in Section 2.5. Our contributions include extending Jia et al's work[12] by evaluating the parameters of IBP in detail. We suspect that the same word-level robustness guarantees will hold for sentences for a BERT[3] model deployed with IBP with minor changes.

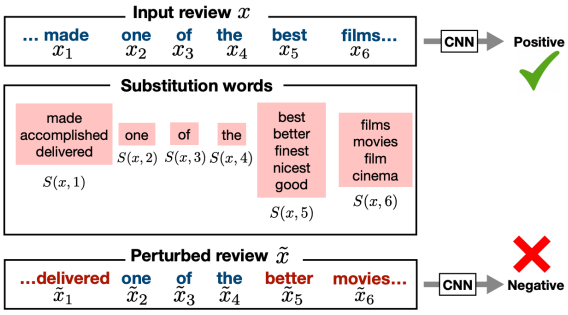

Figure 1: Word substitution-based perturbations in sentiment analysis. For an input x, we consider perturbations $\tilde{x}$, in which every word x_i can be replaced with any similar word from the set $S(x, i)$, without changing the original sentiment. Models can be easily fooled by adversarially chosen perturbations (e.g., changing "best" to "better", "made" to "delivered", "films" to "movies"), but the ideal model would be robust to all combinations of word substitutions.

2 BACKGROUND

At the time of writing this report, there were two papers published on using Interval Bound Propagation (IBP) to provide certifiable robustness in Natural Language Processing tasks. Interestingly, both the papers only consider the task of text classification. In this section, we provide a brief introduction to these techniques starting with a brief background of counter-fitted and contextual embeddings followed with a discussion of the two papers implementing IBP in NLP.

2.1 Counter-Fitted Embeddings

The first generation of word embeddings consisted of embeddings based on word co-occurrence matrices. Counter-fitted embeddings arose in response to a common criticism of such word embeddings:

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

that synonyms and antonyms often share similar contexts and thus are often overly close to each other in the resulting word embedding topology. In order to solve this problem, Mrksic et. al. provide a new objective function containing three terms: an antonym repelling term, a synonym attracting term, and a vector space preservation term. These terms are illustrated in Figures 4, 5, and 6, respectively. The end result of this objective function is to encourage three things: disparate placement of antonyms, similar placement of synonyms, and preservation of the original semantic information latent in the original embeddings being used. This objective function is shown in Figure 7. The reason counter-fitted embeddings were created was due to the fact that word embeddings were static entities, which were a conglomeration of the many different contexts in which a given word was seen in training, and because either a synonym or its antonym is highly likely to appear in a given context, the static nature of the embeddings weighed both possibilities near equally. An small sample of word vectors and their nearest neighbors both before and after counter-fitting is supplied in Figure 2.

2.1.1 Application of Counter-Fitted Embeddings. Later on, Alzantot et. al. found another interesting use for counter-fitted embeddings as an adversarial tool. Alzantot used the counter-fitted embeddings to aid in generating adversarial examples for contextual entailment and sentiment analysis tasks. This work was able to leverage the new information gained through the use of counter-fitted embeddings to construct effective adversarial examples.

2.2 Contextual Embeddings

Recently, the idea of contextual embeddings was developed [16]. This idea furthers the distributional hypothesis-based approach used in the development of earlier word embeddings by allowing word embeddings to be variable based on their context. As a result, individual word vectors no longer have to cover the gamut of possible contexts and can more closely be adapted to the scenario they are being employed in. Other work [4] shows that contextual embedding vectors for a given word vary greatly in practice, so more information is being encoded in contextual vectors than is encoded in vanilla word vectors.

2.3 Language Modeling

Language modeling centers around the task of assigning probabilities to the occurrence of a word or sequence of words. Language models assign these probabilities by learning occurrences of various words or sequences of words present in a training set. Quantitatively speaking, language model training occurs through the maximization of the log probability of the next occurring word or sequence of words. This measure is known as the *perplexity* score and is defined below

$$PP(W) = 2^{-p}$$

where

$$p = \frac{1}{N}\sum_{i=1}^{n} log_2 LM(w_i|w_{1:i-1})$$

2.4 Gowal's Flavor of IBP

Interval bound propagation was first proposed by Gowal et al [7] to propose fast and stable learning algorithm which results in neural networks that are provably robust to norm-bounded perturbations. Their main focus at the time was to prove this robustness for image inputs. The key idea is to minimize an upper bound on the worst case loss over all perturbations bounded in a given norm. This method was extremely effective for images but could not be applied directly to text inputs because texts are relatively discrete in comparison to images

2.5 Jia's Flavor of IBP

Jia et. al. [12] recently extended IBP to text domain by considering word substitutions.A visualization of Jia's bounding process is given in Figure 3. They essentially create a multi-dimensional loss boundary around a given word vector representation with the aim of not allowing similar words to change the model's end decision [12]. This implementation depends on bounding the activation functions propagated by each logit in the network being used; using this technique, they were able to certify roughly 75% of movie review samples from the IMDB dataset against all adversarial perturbations. In the paper, it is not immediately clear whether this approach is fully generalized to work with the stat-of-the-art contextual embeddings like BERT [3]. Further, it is not immediately clear whether the bounds are dependent on the depth of the network or how does this defence generalize for other NLP tasks and attacks (they only considered word substitutions in their work).

	east	**expensive**	**British**
Before	west	pricey	American
	north	cheaper	Australian
	south	costly	Britain
	southeast	overpriced	European
	northeast	inexpensive	England
After	eastward	costly	Brits
	eastern	pricy	London
	easterly	overpriced	BBC
	-	pricey	UK
	-	afford	Britain

Figure 2: Nearest neighbours for target words using GloVe vectors before and after counter-fitting.

3 METHODOLOGY

Our overall goal for this work was to establish a significant and comprehensive understanding of how interval bound propagation is affected by scale and determine the extent that it can be generalized. To this end, we conducted experiments to understand the extent of accuracy and certified robustness degradation on several different datasets, models, and tasks. Once we have an empirical understanding of how IBP works in practice, we can then begin to prescribe improvements, which will be based on the faults we find through our experiments.

In order to determine whether accuracy or certified robustness degrade as model depth increases, we adapted the interval bound propagation code provided by Jia et. al. [12] and recorded the accuracy, certified robustness, and model loss. Word embeddings play

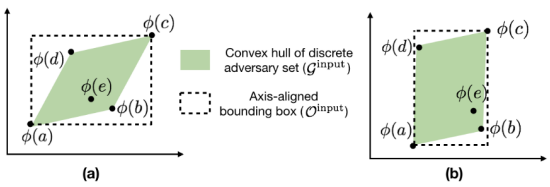

Figure 3: Bounds on the word vector inputs to the neural network. Consider a word (sentence of length one)$x = a$ with the set of substitution words $S(x, 1) = a, b, c, d, e$. (a) IBP constructs axis-aligned bounds around a set of word vectors. These bounds may be loose, especially if the word vectors are pre-trained and fixed. (b) A different word vector space can give tighter IBP bounds, if the convex hull of the word vectors is better approximated by an axis-aligned box.

$$\text{AR}(V') = \sum_{(u,w)\in A} \tau\left(\delta - d(\mathbf{v}'_u, \mathbf{v}'_w)\right)$$

Figure 4: Antonym repel (AR) term for the counter-fitting embedding training process. This term pushes word pairs that are antonyms away from each other in the embedding space, V'. Here, the function, d, uses the cosine similarity metric, τ is a cost margin factor, and δ serves as the "ideal" minimum distance between two antonyms. In Mrksic et. al.'s experiments, $\delta = 1$.

$$\text{SA}(V') = \sum_{(u,w)\in S} \tau\left(d(\mathbf{v}'_u, \mathbf{v}'_w) - \gamma\right)$$

Figure 5: Synonym attract (SA) term for the counter-fitting embedding training process. This term brings synonym pairs closer together in the embedding space, V'. γ is the "ideal" maximum distance between synonymous words. Mrksic et. al. use $\gamma = 0$.

$$\text{VSP}(V, V') = \sum_{i=1}^{N} \sum_{j\in N(i)} \tau\left(d(\mathbf{v}'_i, \mathbf{v}'_j) - d(\mathbf{v}_i, \mathbf{v}_j)\right)$$

Figure 6: Vector space preservation (VSP) term for the counter-fitting embedding training process. This term is intended to preserve the topology of the original, un-counter-fitted embedding space in the generated counter-fitted embeddings. The intuition behind the use of this term is that one would wish to preserve semantic relationships in the original word vectors. Note that $N(i)$ is the set of words within a radius, ρ around the $i-th$ word's vector in the original vector space V. Mrksic et. al.'s experiments indicated that counter-fitting is relatively insensitive to the choice of ρ.

a pivotal role in the adversarial defense process, so we would be remiss to avoid an in-depth analysis of their current characteristics. Another important thing to note is that the IBP method proposed in [12] uses word embeddings to create the bounds. However, most of the new classification frameworks use sentence embeddings which are generated using Language models [3]. Hence it is important to understand if we the proposed IBP method can be extended to language modeling tasks. Due to lack of time, we were not able

$$C(V, V') = k_1\text{AR}(V') + k_2\text{SA}(V') + k_3\text{VSP}(V, V')$$

Figure 7: The complete objective function for the counter-fitting training procedure. Note that k_1, k_2, and k_3 are each hyperparameters.

to finish the language model training task with IBP, however, we conducted several experiments to shed light on our intuition as to why IBP may have trouble scaling to different NLP tasks. These experiments are discussed in Section 4

4 EVALUATION

To understand the effectiveness of IBP for state-of-the-art NLP models, we adopt Jia et. al.'s work and evaluate its effectiveness with new datasets and larger architectures. Then, we measure the performance of the adapted system for a range of common NLP tasks and datasets including: sentiment analysis, the aforementioned language modeling, and contextual entailment to uncover the mechanisms at work behind the defense.

As a substitute for language modelling task, we conducted experiments to see how changing a word with similar words affects the document embedding. We analyze this for four different embeddings - Glove (used by the original authors), BERT (contextual word vectors based on Transformers [18], Elmo (contextual word vectors based on biLSTM) and USE (Universal Sentence Encoder based on attention and transformers). To understand the dependency of these results, we examined these on both the twitter dataset and the ACLIMDB dataset.

4.1 Datasets

To confirm our aforementioned hypothesis, we evaluated the IBP defenses on the following datasets:

(1) IMDB Movie Reviews (50,000 reviews, 265 AWL[1])
(2) Twitter 140 Sentiment (1.6 million tweets, 16 AWL)
(3) Amazon Kindle Reviews (742,767 reviews, 108 AWL)

The Amazon Kindle dataset is a custom-made subset of the Amazon Reviews dataset[15]. Using only reviews on e-books (Kindle books), we specially curated it to have its own vocabulary and counter-fitted vectors, but we were, unfortunately, unable to replicate this for the Twitter dataset[5] due to time and resource constraints. Having custom vocabulary entries and vectors allows the certification procedure to access a larger set of synonym choices for word substitution. This access to all the possible permutations, in turn, creates a more defined decision boundary and improves accuracy.

Using these datasets, we performed several experiments testing the number of hidden neurons and the datasets to see how accuracy would be impacted. Earlier, we postulated that models using IBP won't scale well as the number of layers in the model increases since IBP needs to bound every input at each layer. The progressively looser bounds that result as the model passes more bounds forward become less useful and impact accuracy. In Table 1, the results show a correlation between the deeper networks and a drop in both natural and certified accuracy. For the Amazon Kindle Review

[1] Average Word Length

dataset, the drop in natural accuracy is over 7 percentage points and the drop in certified accuracy is over 13 percentage points. This is a significant decrease in effectiveness, especially for the binary classification task of sentiment analysis. In practice, each percentage point loss in accuracy is a large monetary loss for a deployed model. This, paired with the fact that many state of the art models employ over 10 layers, shows that this problem needs addressing.

Dataset | # Layers	1	3	5	7	9
IMDB Acc	81.3%	77.7%	73.8%	78.2%	77.83
IMDB Cert	73.6%	68.0%	64.9%	64.0%	58.38
Twitter Acc	75.68%	73.89%	73.82%	73.98%	73.42%
Twitter Cert	64.43%	60.75%	59.41%	59.76%	60.52%
Amazon Acc	85.28%	80.57%	79.29%	78.42%	77.66%
Amazon Cert	79.10%	70.22%	68.87%	67.17%	65.52%

Table 1: The natural accuracy and certified accuracy of the IMDB, Twitter, and Amazon sentiment analysis datasets. Evaluated on different numbers of layers of CNNs. All models were trained until convergence (about 10 epochs).

Table 2 confirms also confirms the notion that a larger vocabulary and counter-fitted vector set size results in a more generalizable IBP model. The difference in natural and certified accuracy for the 1-layer CNN is about 5 and 10 percentage points respectively. Finally, In Table 3, we find that the number of hidden neurons within a layer doesn't significantly impact both the natural and certified accuracy for 1-layer CNNs. This is consistent with our earlier hypothesis that the only thing that will affect the IBP bounds is the model architecture's depth.

Vocabulary | # Layers	1	3
Custom-Amazon Acc	85.28%	80.57%
Custom-Amazon Cert	79.10%	70.22%
IMDB-Amazon Acc	80.73%	79.45%
IMDB-Amazon Cert	68.67%	66.53%

Table 2: The natural accuracy and certified accuracy of the Amazon Kindle Review Dataset. The model trained using the custom Amazon vocabulary and counter-fitted vectors performed better than the model trained on only the default IMDB vocabulary and vectors.

Dataset | # Neurons	100	400	800
IMDB Acc	81.20%	81.82%	81.92%
IMDB Cert	66.72%	67.94%	66.42%
Twitter Acc	74.56%	75.48%	75.39%
Twitter Cert	61.72%	62.37%	62.23%

Table 3: The natural accuracy and certified accuracy of the Twitter and IMDB sentiment analysis datasets. Evaluated on 1-layer CNNs with the number of hidden neurons as a parameter.

4.2 Embeddings

As mentioned above, the goal of the experiments conducted with several embeddings was to understand how do the embeddings of the complete sentence/document vary if we use word substitution. To understand this, we generated document embeddings with Glove, BERT[3], ELMO[16] and USE[2]. The first three of these are word-level embeddings, so we averaged the resulting vectors of each word in a sentence to get the sentence embedding, as is the accepted practice. For the Universal Sentence Encoder (USE), the input to the network is the entire text of the document. Next, to understand the variance in these embeddings, we perform word substitutions where a randomly chosen word is replaced with its similar word as found in the counter-fitted embeddings. Finally, we generate the sentence-level embeddings and take the cosine distance between the vectors of the original document and the perturbed document. Due to time constraints, we only allowed single-word substitutions.

To evaluate our claims on embeddings' effects on accuracy, we calculate the cosine distance between original sentences found in the IMDB and Twitter datasets and the perturbed (adversarially generated) sentences. The results of this analysis are shown in Figures 8,9,10, and 11. Here, a cosine distance of 0 implies that the sentences are highly similar. We find that the synonym substitutions on the IMDB dataset 8 are less spread out when compared to the Twitter dataset. This is because the average word length of the IMDB dataset was about 265 words whereas the average word length was 16 in the Twitter dataset. When the vectors are averaged to produce the sentence embeddings, the change due to the perturbed word gets averaged out. We also see that certain embeddings also perform better than others. In particular, on IMDB dataset 8, the word level embeddings perform better than the sentence level embeddings. This can again be attributed to the averaging effect of the word vectors. The performance is dataset-dependent: our accuracy measurements from Section 4.1 are reflected in the embeddings as well. The GloVe embeddings have a much higher overall cosine distance in the Twitter dataset than in the IMDB dataset. This points to evidence of the earlier-discussed correlation between accuracy and word vector distribution. Furthermore, we provide median statistics in Figure 11 and Figure 9 to provide an intuition as to how the embedding space is distributed for the datasets we have examined.

Intuitively, if similar word vectors within embedding spaces are too spread out, the bounding procedure can result in unnecessary accuracy drops that include many false positives in the bounded region. We illustrate a simplified example of this in Figure 12. Rather than attempt to find a better bounding procedure, it may be better to first examine how embedding spaces are currently being filled, and it's also important to derive other ways to populate the many-dimensional embedding spaces in a way that allows pockets of easily bounded words to be formed. The goal is to form bounds which more closely mimic a human-level, common sense approach while simultaneously including as few false positives as possible.

5 FUTURE WORK

Initially, we thought that generalizing IBP for NLP tasks might benefit from a new bounding procedure. After some study, we came to the conclusion that the bounding procedure would need to gain a

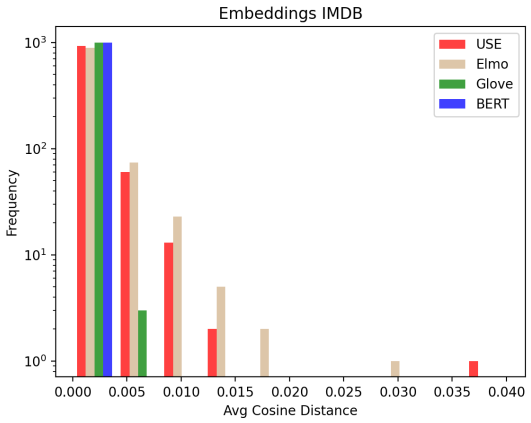

Figure 8: The *average* cosine distance between the original embeddings and the adversarial word-substitution generations. Depicted are embeddings for USE, ELMo, GloVe, and BERT on the IMDB dataset. The distances between the original and perturbed inputs are all fairly small with the exception of the ELMo embeddings.

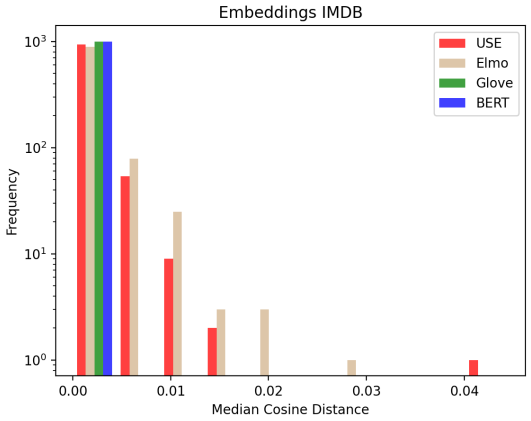

Figure 9: The *median* cosine distance between the original embeddings and the adversarial word-substitution generations. Depicted are embeddings for USE, ELMo, GloVe, and BERT on the IMDB dataset. Once again, all distances are small except for ELMo.

significant amount of complexity in order to become more efficient, essentially having to learn how to navigate the semantic embedding space adaptively somehow. Given the slowdowns already inherent in the use of the existing bounding procedure, we decided to forego this approach, as it likely would have further bogged down training time. Given our stated goal of generalizing IBP to larger and more current model architectures, the prospect of even slower training times would be self-defeating. In light of this development, we decided to examine the possibility of improving the word embeddings themselves.

5.1 Counter-fitting Limitations

The established method of counter-fitting provided by Mrksic et. al. explicitly relies on shaping the counter-fitted embeddings around the original population of the embedding spaces during training to create new counter-fitted embedding-based populations. Secondly,

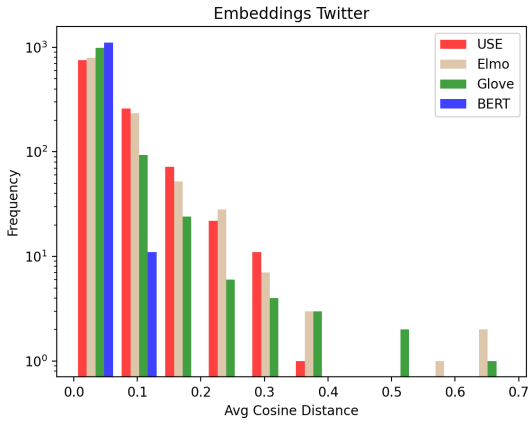

Figure 10: The *average* cosine distance between the original embeddings and the adversarial word-substitution generations. Depicted are embeddings for USE, ELMo, GloVe, and BERT on the Twitter dataset. The distances are quite large and varied, especially for the ELMo and GloVe embeddings.

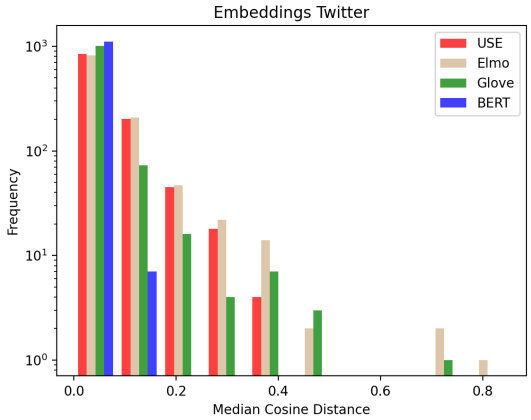

Figure 11: The *median* cosine distance between the original embeddings and the adversarial word-substitution generations. Depicted are embeddings for USE, ELMo, GloVe, and BERT on the Twitter dataset. The distances are quite large and varied, especially for the ELMo and GloVe embeddings.

the notion of synonyms being attracted and antonyms being repelled is used in the training of counter-fitted embeddings, along with the aforementioned shaping procedure mentioned above. This is an effective way of dealing with synonym and antonym pairs, but it lacks the nuance to deal with words that do not have obvious synonyms or antonyms that one could find in a dictionary entry. Furthermore, the notion of using synonyms or antonyms only accounts for two ends of a varied and complex spectrum of word meanings that can be derived from their appearance in the context of an arbitrary sentence.

To bolster this point, it is useful to look at how many synonym and antonym pairs appear in common subsets of English seen in practice. Mrksic et. al. complete this kind of analysis on a frequently used subset of English provided as subtitles for recent movies. In their analysis, out of the roughly 76 thousand words surveyed from the subtitle dataset, roughly 13 thousand antonym pairs and

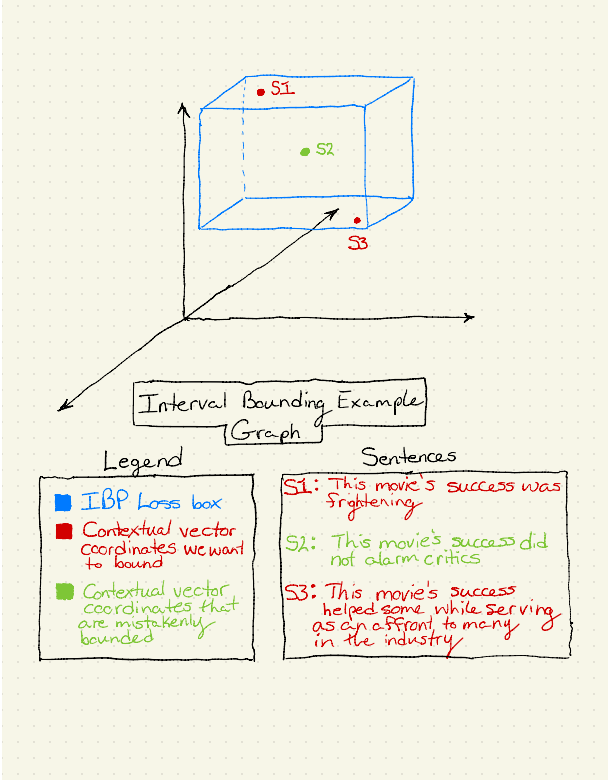

Figure 12: An imagined sample bounding problem similar to something that could occur in practice. Here, we see three sentences, two of which are intended to be classified as the same label, and one of which is intended to be classified as a different label. This figure illustrates how overly large bounds can potentially degrade model accuracy.

31 thousand synonym pairs were found. The root synonyms and antonyms were provided through manual human construction, which will likely confound scaling efforts. Additionally, the synonym and antonym pair discrepancy also limits the effectiveness of one half of the counter-fitting approach. Mrksic et. al.'s concept of squeezing and expanding words in the embedding space was limited by the technology of their time. If we want to find a way to better apportion every word in a set of word embeddings, it seems necessary to add a notion of a similarity spectrum, as opposed to only accounting for synonyms and antonyms. Recently, contextual embeddings, such as those of BERT[3], ULMFiT[8], and others have allowed us to quantitatively determine the meaning of a word by determining its current context. This idea is precisely what is needed to apply a more nuanced spacing approach when counter-fitting word embeddings. Therefore, we plan to adapt this concept into a new training approach, which is constructed to compress or expand inputted word embeddings by using the degrees of similarity present in the contexts being considered during training. Thus, we aim to continue with the spirit of Mrksic et. al.'s work and generalize their concept by using contextual artifacts to determine, in a more fine-grained manner, how close or far two contexts should be in the embedding space.

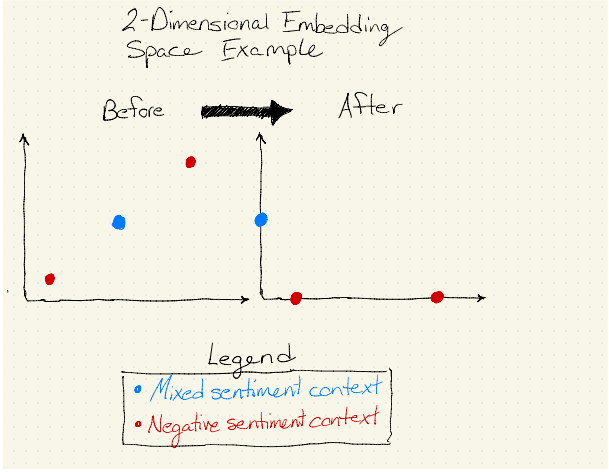

Figure 13: A very rough sketch of the intuition we aim to apply to contextual embeddings construction in our future work. Here, the takeaway is that we aim to align similar words in similar contexts along the same subspaces of the embedding space. The best way to accomplish this is still an open problem for us, but the much larger embedding space provided by contextual vectors should allow an approach like this to work while preserving more accuracy than if this idea was applied to vanilla word embeddings. PCA could be of use here.

5.2 Future Experimental Architectures

Going forward, we plan to implement a range of current model architectures and measure the efficacy of our Contextual Adversarial Embeddings on deeper and more complex models. Some of the models we will experiment on are to be determined at this time. But, we do plan to adapt Gowal et. al.'s [7] published CNN architectures for various NLP tasks.

small	medium	large
Conv 16 4×4+2	Conv 32 3×3+1	Conv 64 3×3+1
Conv 32 4×4+1	Conv 32 4×4+2	Conv 64 3×3+1
FC 100	Conv 64 3×3+1	Conv 128 3×3+2
	Conv 64 4×4+2	Conv 128 3×3+1
	FC 512	Conv 128 3×3+1
	FC 512	FC 512
# hidden: 8.3K	47K	230K
# params: 471K	1.2M	17M

Figure 14: Shown above are the experimental CNN architectures used by Gowal et. al. in their IBP computer vision defense work. We aim to replicate the architectures used by Gowal et. al. in the NLP domain and examine their transferability as these models have been demonstrated to be successful for IBP applications.

5.3 Embedding Generation Goals

Though the exact methods through which we will train and generate adversary-resistant embeddings are currently undecided, we do know the tenets that are axiomatic for our future work. They are as follows:

- Two word vectors should be close in the embedding space if and only if they are being used in a similar context and are similar words.

- Similar contexts should be encouraged to be directed along similar dimensional sub-spaces while also encouraging different contexts to avoid significant intersection in the same sub-spaces so that bounds placed over a given dimensional space do not significantly degrade model accuracy while still preserving the capacity for adversarial certification.
- A non-binary notion of similarity/dissimilarity will be developed from latent contextual information and taken into account during embedding training.

Figure 13 demonstrates a low dimensional rough sketch of the geometric intuition behind how a redistribution of vectors in a high dimensional space could be beneficial to certified bounding and accuracy.

5.4 Outline of Future Work Plan

5.4.1 Statistical Examination of GloVe Subtitle Subset. We are in the process of completing the necessary code to comprehensively calculate each pair of cosine distances for the GloVe subtitle subset, both with and without counter-fitting and determine various statistical measures of the embeddings, which were used in the Mrksic work. A good deal of optimization is required to make this process able to fit onto our server, but the most memory-intensive intermediate steps have been optimized so that they fit on our server's main memory. No previous work has done this level of detailed analysis of vanilla and counter-fitted embeddings to the best of our knowledge, so such an analysis has academic value for other NLP researchers as well. Once the processes have completed and we have this data, we plan to examine how it limits the effectiveness of IBP and whether the embedding topology is lacking in the ways we postulated above. Having an understanding of the limitations of counter-fitting when applied to static word embeddings will allow us to construct a convincing argument for the development of a new method.

5.4.2 Statistical Examination of a Representative Subset of BERT Contextual Vectors. We also plan to examine how vanilla contextual vectors are distributed throughout the embedding space. We have provided a smaller examination of this and generated a number of plots, shown in Figures 8-11. Similar to our planned examination of the GloVe subtitle subset, we wish to unmask and examine the contextual embedding topology to better understand how we can allocate vectors in a way that is conducive for the application of IBP to larger models.

5.4.3 Development of Contextual Counter-Fitted Vectors. After we have learned the extent of the existing methods of counter-fitted vectors, we can start to develop an intuition for the best ways to restructure the distribution of vectors in the embeddings generation process.

5.4.4 Application of Contextual Counter-Fitted Vectors to Current Models. Once we have developed an intuition of how to apply the tenets listed in the Embedding Generation Goals section, we will test out our methods on current NLP architectures and other, more complex models than IBP has previously been applied to. To this end, we also plan to adapt the models used by Gowal et. al. in Figure 14.

6 CONCLUSION

Currently, the adversarial NLP space is composed of a large number of attack works and a small number of defense works. We examined how a promising defense like [12] performs on more complex models and tasks and discovered that embedding similarity distances are correlated with the accuracy of a network trained using IBP. Moving forward, we aim to evaluate IBP's capabilities in language modeling settings as well as question-answer tasks. Our future endeavours will be guided by the concept of using better word embeddings and contextual word embeddings to generalize to more complex tasks to avoid increasing computational complexity.

7 CONTRIBUTIONS

While Matt was leading this project idea, each group member contributed with a significant amount of experimentation and writing. Matt adapted the codebase to allow the creation of multiple layers of depth for the architectures used in the evaluation. Brian evaluated the Amazon reviews dataset and the Twitter sentiment dataset with the number of layers and the hidden neurons as parameters. Rishabh worked on retraining a language model and obtaining the embedding distances. Everyone contributed to the writing of this report and the class presentations.

REFERENCES

[1] Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating Natural Language Adversarial Examples. (2018). arXiv:cs.CL/1804.07998

[2] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal Sentence Encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Brussels, Belgium, 169–174. https://doi.org/10.18653/v1/D18-2029

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805* (2018).

[4] Kawin Ethayarajh. 2019. How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, 55–65. https://doi.org/10.18653/v1/D19-1006

[5] Alec Go, Richa Bhayani, and Lei Huang. 2019. Twitter Sentiment Classification using Distant Supervision. (2019).

[6] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).

[7] Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet Kohli. 2018. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv preprint arXiv:1810.12715* (2018).

[8] Jeremy Howard and Sebastian Ruder. 2018. Universal Language Model Fine-tuning for Text Classification. In *ACL*. Association for Computational Linguistics. http://arxiv.org/abs/1801.06146

[9] Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. *arXiv preprint arXiv:1804.06059* (2018).

[10] Robin Jia and Percy Liang. 2017. Adversarial Examples for Evaluating Reading Comprehension Systems. *CoRR* abs/1707.07328 (2017). arXiv:1707.07328 http://arxiv.org/abs/1707.07328

[11] Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. *arXiv preprint arXiv:1707.07328* (2017).

[12] Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. 2019. Certified Robustness to Adversarial Word Substitutions. (2019). arXiv:cs.CL/1909.00986

[13] Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2019. TextBugger: Generating Adversarial Text Against Real-world Applications. *Proceedings 2019 Network and Distributed System Security Symposium* (2019). https://doi.org/10.14722/ndss.2019.23138

[14] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. *ICLR* (2018).
[15] Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects. *Association for Computational Linguistics* (2019).
[16] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365* (2018).
[17] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199* (2013).
[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.