Adversarial Gain

Anonymous Author(s)
Affiliation
Address
email

Abstract

Adversarial examples can be defined as inputs to a model which induce a mistake – where the model output is different than that of an oracle, perhaps in surprising or malicious ways. Original models of adversarial attacks are primarily studied in the context of classification and computer vision tasks. While several attacks have been proposed in natural language processing (NLP) settings, they often vary in defining the parameters of an attack and what a successful attack would look like. The goal of this work is to propose a unifying model of adversarial examples suitable for NLP tasks in both generative and classification settings. We define the notion of adversarial gain: based in control theory, it is a measure of the change in the output of a system relative to the perturbation of the input (caused by the so-called adversary) presented to the learner. This definition, as we show, can be used under different feature spaces and distance conditions to determine attack or defense effectiveness across different intuitive manifolds. This notion of adversarial gain not only provides a useful way for evaluating adversaries and defenses, but can act as a building block for future work in robustness under adversaries due to its rooted nature in stability and manifold theory.

1 Introduction

The notion of *adversarial examples* has seen frequent study in recent years [34, 13, 25, 19, 12]. The definition for adversarial examples has evolved from work to work[1]. However, a common overarching definition[2] characterizes adversarial examples as *"inputs to machine learning models that an attacker has intentionally designed to cause the model to make a mistake."*

In such a context a mistake can be defined such that a model's output $f(x)$ differs from the output of a set of oracle (or optimal) models $f^*(x)$. In some cases the oracle output is known and this definition is sufficient. One such example is in the case of malware detection [15]. A target sample is known to be malware, but can be disguised – without the possibility of changing its ground truth role as malware – to cause a malware detection model to classify it as a safe sample.

However, in some cases, the optimal output given the perturbation or generated sample is unavailable or ambiguous. Furthermore, evaluation methods of the output may not be descriptive enough as an alternative for assessing performance under an adversary – as in dialogue [22] or translation [5].

To circumvent the lack of availability of an oracle model or descriptive evaluation metric, various works have made distance-based assumptions surrounding adversarial examples. A known sample is perturbed by a constrained amount such that within the constraint the output of the model output should be unchanged.

[1] See Supplementary Material for definitions in prior work
[2] https://blog.openai.com/adversarial-example-research/

Submitted to the Workshop on Interpretability and Robustness in Audio, Speech, and Language (NIPS 2018). Do not distribute.

In NLP tasks, various works attempt to preserve meaning (and thus ensure that the oracle output should be unchanged) by constraining operations, such as only replacing words with synonyms [1, 27, 7, 39, 10]. However, such constraints are task-dependent, often difficult to specify, and not necessarily guaranteed. There can be cases where a model output may correctly change its output within some constrained radius perturbation (e.g., if a sentence is on the border between two sentiments, a small change may cause the classifier to make a valid shift). In fact, in a survey conducted by Jia and Liang [19] about their generated adversarial examples, it was found that humans – a proxy for the oracle in this setting – sometimes did change their answer under the perturbed noise.

Finally, in text generation settings the notion of what constitutes a mistake varies from work to work. Miyato et al. [25], Papernot et al. [27], Cheng et al. [7], Zhao et al. [39] measure an adversary's effectiveness in generating a target word or sequence; Zhao et al. [39] create an adversary which successfully causes a model to omit words; Cheng et al. [7] introduce a measure of success where the model outputs text that has no overlap with its original output; Ebrahimi et al. [10] measure success rate as a function of the decrease in BLEU score beyond some threshold.

2 Adversarial Gain

To account for the lack of guarantees in perturbation constraints, the sometimes ambiguous notion of a "mistake" by a model, and the unknown oracle output for a perturbed sample, we propose the unified notion of *adversarial gain*. We draw from incremental L_2-gain in control theory [30] as inspiration and define the adversarial gain as:

$$\hat{\beta}_{adv} \leq \frac{D_{out}(\phi_{out}(f(x)), \phi_{out}(f(x_{adv})))}{D_{in}(\phi_{in}(x), \phi_{in}(x_{adv}))}, \tag{1}$$

such that x is a real sample from a dataset, x_{adv} is an adversarial example according to some attack targeting the input x, $x \neq x_{adv} \forall (x, x_{adv}) \in X$, $f(x)$ is the learner's output, ϕ_{in}, ϕ_{out} is a feature transformation for the input and output respectively, and D_{in}, D_{out} are some distance metrics for the input and output space respectively. β_{adv} indicates per sample adversarial gain and $\hat{\beta_{adv}}$ is an upper bound for all samples X.

We do not assume that a model's output should be unchanged within a certain factor of noise as in Raghunathan et al. [28], Bastani et al. [3], rather we assume that the change in output should be proportionally small to the change in input according to some distance metric and feature space. Similar to an L_2 incrementally stable system, the goal of a stable system in terms of adversarial gain is to limit the perturbation of the model response according to a worst case adversarial input x_{adv} relative to the magnitude of the change in the initial conditions. Since various problems place emphasis on stability in terms of different distance metrics and feature spaces, we leave this definition to be broad and discuss various notions of distance and feature spaces subsequently.

This notion holds for both cases where an oracle is known and unknown, for both generative and discriminative settings, and for continuous and discrete spaces. Furthermore, this allows for an adversary to make arbitrarily large changes in the input space, so long as the change causes proportionally large an instability in the output space. In cases where the oracle output is known (e.g., we know that a malware should be classified as such), a traditional metric, such as model accuracy across adversarial examples, can be used in conjunction with adversarial gain. In these settings, gain can provide additional information about the vulnerable space of inputs, similarly to the manifold space as used in Wu et al. [36]. Additional properties are discussed in Supplementary Material.

2.1 Bootstrapping the Real Data Gain

Since adversarial gain on its own doesn't necessarily indicate a mistake, we must also determine what is an unusual amount of gain. That is, at what point has the model begun to generate likely incorrect outputs. To do this, we can bootstrap some rough bounds from the known data. That is for any two batches (M_1, M_2) of data randomly sampled from the known data such that $M_1 \cap M_2 = \varnothing$, we generate a set of bootstrap samples:

$$\beta_{M,real} = \frac{D_{out}(\phi_{out}(f(x_1)), \phi_{out}(f(x_2)))}{D_{in}(\phi_{in}(x_1), \phi_{in}(x_2))}, \tag{2}$$

where $x_1, y_2 \in M_1, x_2, y_2 \in M_2$, and $\hat{\beta}_{M,real}$ indicates an upper bound.

Input: leading season scorers in the bundesliga after saturday 's third-round games **(periods)** : UNK **Original output:** games standings ǀ **Adversarial output:** Scorers after third-round period $\beta_{adv} = 9.5$, $D_{in} = 0.05$, $D_{out} = 0.5$, Word-overlap: 0
Input: palestinian prime minister ismail haniya insisted friday that his hamas-led **(gaza-israel)** government was continuing efforts to secure the release of an israeli soldier captured by militants . **Original output:** hamas pm insists on release of soldier ǀ **Adversarial output:** haniya insists gaza truce efforts continue $\beta_{adv} = 4693.82$, $D_{in} = 0.00$, $D_{out} = 0.46$, Word-overlap: 1
Input: south korea **(beef)** will **(beef)** play for **(beef)** its **(beef)** third straight olympic women 's **(beef)** handball gold medal when **(beef)** it meets denmark saturday **(beef)** **Original output:** south korea to meet denmark in women 's handball ǀ **Adversarial output:** beef beef beef beef beef beef beef up beef $\beta_{adv} = 3.59$, $D_{in} = 0.15$, $D_{out} = 0.55$, Word-overlap: 0

Table 1: Adversarial examples for text summarization using [7]. The bold words are those which modify the original sentence. Brackets indicate an addition, parenthesis indicate replacement of the preceding word. An $\epsilon = 1^{-4}$ is added to the denominator to avoid division by 0 in this case. D_{in}, D_{out} both in terms of InferSent distance.

From these gain samples, we can estimate some bounds on the average point-wise gain of the real data using the bootstrap [11]. We refer to this bootstrap estimate as β_{real}, or the "real" gain. If an adversarial example has a gain exceeding the bootstrap estimate, it is more likely that the model in fact made a mistake due to an adversary. That is, given some level of change in input, has the output shifted into a significantly different space than what is typical in known data.

2.2 Distance Metrics and Feature Spaces

Our definition of adversarial gain depends crucially on the definition of distance metrics for both the input and output spaces.

2.2.1 Distance Metrics in NLP

There are many distance metrics relevant for NLP tasks as discussed by van Asch [2]. These include divergences in probability distributions (e.g., Jensen-Shannon divergence), semantic similarity [24], count-based metrics (word overlap, BLEU score, etc.), and various string kernels [23].

For NLP input spaces, while count-based metrics provide some signal, they are often lacking as evaluation and distance measures as discussed in [5] and seen in Section 3.1. Using semantic similarity or cosine similarity has been used in Henderson et al. [17] for investigating adversarial examples in dialogue. It comes with the intuition that similar linguistic samples should be closer together. However, measuring semantic similarity can be difficult due to the language understanding required and often needs a well-defined feature space.

On the output side, for classification tasks, such as sentiment classification [32], it is possible either to use a step-wise function (1 if classification changes, 0 otherwise) or a divergence. As Wu et al. [36] do, the latter suggests that "confident regions of a good model should be well separated", but in the context of adversarial gain should be proportional to the input distance for reduced gain. Moreover, proper use of uncertainty or distribution modeling can be shown to protect against adversarial attacks [4], and thus evaluating the gain in terms of probabilistic divergences may be desirable.

2.2.2 Feature Spaces

To measure semantic similarity of text, various encoding methods have been developed which transform the text into a vector space [20, 8, 38, 6]. Using the cosine similarity in conjunction with such an embedding space can ensure that similar text will be closer together. Here, we use the InferSent embeddings [8] as the primary form of measuring semantic similarity. Adversarial gain can be measured across different feature spaces (and thus different manifolds). However, another appropriate method may be to learn a specific embeddings (feature) space for the problem at hand similarly to Yang et al. [38] since well-generalized embedding spaces are difficult to create [9]. By learning a feature space which ensures a well-defined distance-based correlation between inputs and outputs, the distance assumption can more accurately measure whether an adversarial attack falls

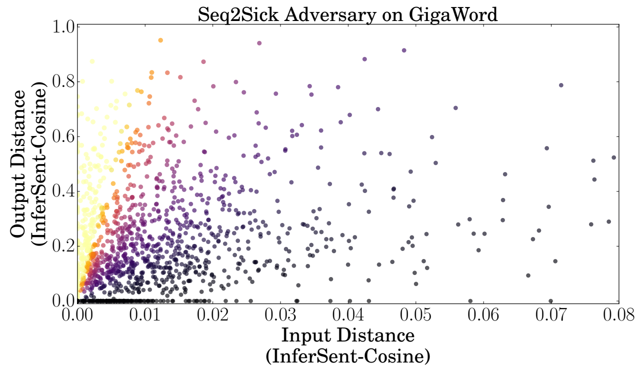

Figure 1: The distribution of adversarial examples in text summarization tasks. Warmer colors (reg, orange, yellow, respectively) indicate higher gain values.

in the gain range where a mistake is more likely. This follows manifold-based work as in Wu et al. [36], Lamb et al. [21].

2.3 A Note on Human Perception

A common debate regarding adversarial examples is whether they should be perceivable by humans. Many works cite perception in their definition of adversarial examples or run surveys determining whether humans were able to perceive the change [31, 16, 18, 19]. However, Elsayed et al. [12] contest that the use of perception in the definition is incorrect because then humans would not be susceptible to adversarial examples – and they claim later on that humans in fact are susceptible under some constrained conditions. In the setting of adversarial gain, human perception is not a strict condition. However, human perception plays a relation to the oracle. In many tasks, human perception is used as a proxy for an oracle model. For example, in image classification, datasets are generated from what humans perceive to be the label rather than some verified ground truth. In the context of adversarial gain, it is possible that humans are susceptible to certain high gain samples such as the perceived colour of "the dress" [35]. This satisfies the properties set forth by Elsayed et al. [12]. However, it also allows for the accounting of human perception. By measuring the adversarial gain bounds of humans across distance metrics, it may be possible to build a better picture of expected model performance in many ambiguous settings where we use humans as proxies for an oracle model (e.g., dialogue, text summarization, sentiment analysis).

2.4 A Note on Generative Adversarial Examples

In the case where samples are not perturbed, but rather generated from scratch as in [39, 37, 33], there is no original sample to be compared against. In this case, we can think about the use of our latent feature space ϕ and find the nearest known neighbourhood of examples within that feature space. These can be used as a reference point for evaluating the gain of the adversarial example. This can be applied to perturbed adversarial gain as well, but is computationally much more intensive. Finding high gain samples in such a way may allow for the discovery of unknown regions of space where more real samples are needed or decision boundaries and certainty gradients must be adjusted.

2.5 A Note on Targeted Attacks

We do not explicitly consider targeted attacks in our main definition of adversarial gain. However, because of the distance based formulation, it is simple to do so. A targeted attack can be thought of in two ways: (1) inducing a model to generate a certain output (even if it's not wrong); (2) inducing a model to make a mistake in a particular way which generates a certain output. We posit that some prior literature actually examines the first case. Cheng et al. [7], for example, use an indicator function which determines if a certain set of words exists in an output sequence. One example of a success for inducing the words "Hund sizst" in a machine translation task that is provided in [7] is:

SOURCE INPUT SEQ: A TODDLER IS COOKING WITH ANOTHER PERSON.

ADV INPUT SEQ: A dog IS sit WITH ANOTHER UNK.
SOURCE OUTPUT SEQ: EIN KLEINES KIND KOCHT MIT EINER ANDEREN PERSON.
ADV OUTPUT SEQ: EIN Hund sitzt MIT EINEM ANDEREN UNK.

It is clear in this case that the model does not necessarily make a mistake, but rather changes in the input to induce a certain output that is a correct translation. While this is an interesting problem and approach, the model is still performing as expected in this case. We instead, can formulate targeted adversarial gain in the context of the latter where we need to have a notion of distance to a known sample to approximate incorrect behaviour. We can define gain as the difference between two distances, that of the original sample to the target sample and the adversarial sample to the target sample. This forms a sort of cost-to-go function. That is for a target, a large adversarial gain corresponds to the closest input change to reach a certain target output space. In terms of classification tasks, this may have interesting properties related to decision boundaries, but we consider it out of scope for our experiments.

3 Experiments

We aim to study empirically whether adversarial gain is suitable as a unified notion in both generative and discriminative NLP settings. We run experiments on text summarization and sentiment classification based on existing open-source constrained adversarial attacks, and evaluate whether adversarial gain offers a relevant characterization.

Metrics	β_{real}	β_{adv}
Sentiment Classification		
IS + JS	0.85 (0.79, 0.91)	13.75 (-1.93, 25.32)
IS + Step	1.18 (1.10, 1.27)	22.6 (-3.96, 42.5)
WD + Step	0.018 (0.016, 0.019)	0.241 (0.227, 0.255)
WD + JS	0.008 (0.007, 0.008)	0.121 (0.115, 0.127)
Text Summarization		
IS + IS	2.174 (2.14, 2.20)	134.62 (102.31, 163.05)

Table 2: We provide the bootstrap average with confidence bounds across 10k bootstrap samples. To avoid division by 0, we add an $\epsilon = 1^{-4}$ to the denominator of the gain. WD indicates the number of words that word added or changed. IS indicates the InferSent cosine distance. Step indicates 1 if the class label changed, 0 otherwise.

3.1 Generative Tasks: Text Summarization

For text summarization we use the GigaWord dataset [29, 14, 26], subset of holdout test data, pretrained model, word embeddings, and attack vector as used by Cheng et al. [7]. We use InferSent embeddings, and cosine distance to measure the distance on both inputs and outputs.

The resulting bootstrap estimate average gain can be seen in Table 2 and the distribution of change caused by the adversarial attack can be visualized in Figure 1. It is clear that the attack does induce changes in meaning on average according to the InferSent embeddings, but there are also low-gain samples where the attack must make large changes in the input space to cause a significant change in output. Cheng et al. [7] measure the success of an attack if there is no word overlap in the changed output. While this does provide some information, it may be the case that the model is still technically correct in its performance even with no overlap. The first example in Table 1 demonstrates such a scenario. Adversarial gain in a feature space such as InferSent, however, provides a more refined notion of change. Furthermore, the second sample in Table 1 demonstrates a high gain due to change in meaning even though there is word overlap. Lastly, in a case where there is no overlap in the outputs due to a large number of changes to the input meaning, the notion of adversarial gain gives the model some leeway (if the input is drastically changed it's likely okay to change the output). As seen in Table 2, on average these scenarios fall outside of the typical bound of the real data indicating some level of attack effectiveness, thus showing that adversarial gain provides a decent notion of the effectiveness of an attack and susceptibility of the model to attack.

3.2 Discriminative Tasks: Sentiment Classification

Next, we examine a sentiment classification task using the SST2 dataset [32], pre-trained convolutional neural network model, and single word flip attack as provided by Ebrahimi et al. [10]. We use a step-wise function and the JS divergence as distance metrics on the output. We use InferSent embeddings and word distance (number

a benign but forgettable sci fi diversion **[fiorentino brio]** $f(x) = (0.98, 0.02)$, $f(x_{adv}) = (0.02, 0.98)$ $\beta_{adv} = \infty$, $D_{in} = 0.0$, $D_{out} = 0.60$
the transporter is as lively and as fun as it is unapologetically dumb **(ineffective)** $f(x) = (0.01, 0.99)$, $f(x_{adv}) = (0.99, 0.01)$ $\beta_{adv} = 4.94$, $D_{in} = 0.13$, $D_{out} = 0.64$
ranks among willams ' best screen work **[cram cheesy]** $f(x) = (0.00, 1.00)$, $f(x_{adv}) = (0.66, 0.34)$ $\beta_{adv} = 1.34$, $D_{in} = 0.22$, $D_{out} = 0.31$

Table 3: Adversarial examples for sentiment classification using Ebrahimi et al. [10]. The bold words are those which modify the original sentence. Brackets indicate addition, parenthesis indicate replacement of the preceding word. D_{in} is the InferSent distance. D_{out} is the JS divergence.

of different words) as measures on the input. Table 2 shows the distribution of gain from the real data and the adversarial data. Table 3 shows some qualitative examples. One demonstration where adversarial gain using the InferSent embedding space helps is with the third example in Table 3. Though the model's label changes, with a relatively small number of added words (2), the meaning of the sentence possibly changes indicating that "William's best screen work" may be cheesy. The shift in sentiment causes the adversarial gain to fall close to the gain of the real data and thus the model is less likely to be making a mistake if an oracle were to label the perturbed sample.

4 Discussion

Overall, we introduce the notion of adversarial gain as a measure of adversary effectiveness and model robustness against an adversary. This notion is applicable to both generative and discriminative models and bears particularly convenient properties for many tasks in natural language processing. While the notions of distance which we provide here are not perfect, they appear to provide adequate information to assess performance. In the future, learning a domain dependent feature representation space may help to improve the information provided by adversarial gain. Going forward, adversarial gain can provide a more standardized comparative measure of adversarial examples and attack quality. Furthermore, its roots in stability theory, use of manifold spaces, and other interesting properties as a unified view of adversarial examples may inspire the construction of future robust and gain-stable NLP models.

References

[1] Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. *arXiv preprint arXiv:1804.07998*.

[2] Vincent van Asch. 2012. *DOMAIN SIMILARITY MEASURES: On the use of distance metrics in natural language processing*. Ph.D. thesis, PhD dissertation.

[3] Osbert Bastani, Yani Ioannou, Leonidas Lampropoulos, Dimitrios Vytiniotis, Aditya Nori, and Antonio Criminisi. 2016. Measuring neural net robustness with constraints. In *Advances in neural information processing systems*, pages 2613–2621.

[4] John Bradshaw, Alexander G de G Matthews, and Zoubin Ghahramani. 2017. Adversarial examples, uncertainty, and transfer testing robustness in gaussian process hybrid deep networks. *arXiv preprint arXiv:1707.02476*.

[5] Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Re-evaluation the role of BLEU in machine translation research. In *11th Conference of the European Chapter of the Association for Computational Linguistics*.

[6] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

[7] Minhao Cheng, Jinfeng Yi, Huan Zhang, Pin-Yu Chen, and Cho-Jui Hsieh. 2018. Seq2sick: Evaluating the robustness of sequence-to-sequence models with adversarial examples. *arXiv preprint arXiv:1803.01128*.

[8] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*.

[9] Ishita Dasgupta, Demi Guo, Andreas Stuhlmüller, Samuel J Gershman, and Noah D Goodman. 2018. Evaluating compositionality in sentence embeddings. *arXiv preprint arXiv:1802.04302*.

[10] Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2017. Hotflip: White-box adversarial examples for nlp. *arXiv preprint arXiv:1712.06751*.

[11] Bradley Efron and Robert Tibshirani. 1986. Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. *Statistical science*, pages 54–75.

[12] Gamaleldin F Elsayed, Shreya Shankar, Brian Cheung, Nicolas Papernot, Alex Kurakin, Ian Goodfellow, and Jascha Sohl-Dickstein. 2018. Adversarial examples that fool both human and computer vision. *arXiv preprint arXiv:1802.08195*.

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

[14] David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2003. English gigaword. *Linguistic Data Consortium, Philadelphia*, 4:1.

[15] Kathrin Grosse, Nicolas Papernot, Praveen Manoharan, Michael Backes, and Patrick McDaniel. 2016. Adversarial perturbations against deep neural networks for malware classification. *arXiv preprint arXiv:1606.04435*.

[16] Matthias Hein and Maksym Andriushchenko. 2017. Formal guarantees on the robustness of a classifier against adversarial manipulation. In *Advances in Neural Information Processing Systems*, pages 2263–2273.

[17] Peter Henderson, Koustuv Sinha, Nicolas Angelard-Gontier, Nan Rosemary Ke, Genevieve Fried, Ryan Lowe, and Joelle Pineau. 2017. Ethical challenges in data-driven dialogue systems. *arXiv preprint arXiv:1711.09050*.

[18] Andrew Ilyas, Ajil Jalal, Eirini Asteri, Constantinos Daskalakis, and Alexandros G Dimakis. 2017. The robust manifold defense: Adversarial training using generative models. *arXiv preprint arXiv:1712.09196*.

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

[20] Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302.

[21] Alex Lamb, Jonathan Binas, Anirudh Goyal, Dmitriy Serdyuk, Sandeep Subramanian, Ioannis Mitliagkas, and Yoshua Bengio. 2018. Fortified networks: Improving the robustness of deep networks by modeling the manifold of hidden representations. *arXiv preprint arXiv:1804.02485*.

[22] Chia-Wei Liu, Ryan Lowe, Iulian V Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. *arXiv preprint arXiv:1603.08023*.

[23] Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristinanini, and Chris Watkins. 2002. Text classification using string kernels. *JMLR*.

[24] Rada Mihalcea, Courtney Corley, Carlo Strapparava, et al. 2006. Corpus-based and knowledge-based measures of text semantic similarity. In *AAAI*, volume 6, pages 775–780.

[25] Takeru Miyato, Andrew M Dai, and Ian Goodfellow. 2016. Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725*.

[26] Courtney Napoles, Matthew Gormley, and Benjamin Van Durme. 2012. Annotated gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, pages 95–100. Association for Computational Linguistics.

[27] Nicolas Papernot, Patrick McDaniel, Ananthram Swami, and Richard Harang. 2016. Crafting adversarial input sequences for recurrent neural networks. In *Military Communications Conference, MILCOM 2016-2016 IEEE*, pages 49–54. IEEE.

[28] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. 2018. Certified defenses against adversarial examples. *arXiv preprint arXiv:1801.09344*.

[29] Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*.

[30] Arjan Van der Schaft. 2000. *L2-gain and passivity techniques in nonlinear control*, volume 2. Springer.

[31] Uri Shaham, Yutaro Yamada, and Sahand Negahban. 2015. Understanding adversarial training: Increasing local stability of neural nets through robust optimization. *arXiv preprint arXiv:1511.05432*.

[32] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

[33] Y. Song, R. Shu, N. Kushman, and S. Ermon. 2018. Generative adversarial examples. *arXiv preprint arXiv:1805.07894*.

[34] 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*.

[35] Alissa D Winkler, Lothar Spillmann, John S Werner, and Michael A Webster. 2015. Asymmetries in blue–yellow color perception and in the color of 'the dress'. *Current Biology*, 25(13):R547–R548.

[36] Xi Wu, Uyeong Jang, Lingjiao Chen, and Somesh Jha. 2017. Manifold assumption and defenses against adversarial perturbations. *arXiv preprint arXiv:1711.08001*.

[37] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. 2018. Generating adversarial examples with adversarial networks. *arXiv preprint arXiv:1801.02610*.

[38] Yinfei Yang, Steve Yuan, Daniel Cer, Sheng-yi Kong, Noah Constant, Petr Pilar, Heming Ge, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Learning semantic textual similarity from conversations. *arXiv preprint arXiv:1804.07754*.

[39] Zhengli Zhao, Dheeru Dua, and Sameer Singh. 2017. Generating natural adversarial examples. *arXiv preprint arXiv:1710.11342*.

A Literature Review

Paper	Input Perturbation: Input perturbation	Input Perturbation: Gradient Based	Input Perturbation: Use of Human Perception	Output	Task
Miyato et al. [14]	Perturbation to word embedding	Yes	No	Change in cost function	Text classification
Dasgupta et al. [3]	Change in word ordering replace ’more’ with ’less’	No	No	Change in class	Natural Language Inference
Jia and Liang [9]	Concatenate adversarial sentence replace ’more’ with ’less’	No	Yes	lower F1 score	Question Answering
Samanta and Mehta [18]	Replace or remove words which contribute most to classification with synonyms, typos and genre specific words	No	No	Change in class	Sentiment Analysis
Kuleshov et al. [11]	Replace word with synonym, decision learned using a constraint optimization.	Yes	No	Change in class	Classification
Hosseini et al. [8]	Negating phrases, misspellings. decision learned using a constraint optimization.	No	No	Lower toxicity score.	Classification on confidence
Cheng et al. [1]	Non-overlapping exclusive words attack; targeted keyword attack where all the keywords must be present in the adversarial input; word replacement.	Yes	No	Change in BLEU score	Translation
Papernot et al. [16]	Replacing words with most impactful words in the classification w.r.t Jacobian quantity;	Yes	No	Change in class & distribution	Classification, Generation
Liang et al. [13]	Change characters, insert one hot word, parenthesis, forged fact	Yes	Yes	Change in cost function	Classification
Li et al. [12]	Drops certain dimensions of word embeddings, uses RL to find minimal set of words to remove	Yes	No	Change in class confidence	Classification
Ebrahimi et al. [4]	Flips the characters/ words in a sentence w.r.t gradient loss change, using beam search to determine the best r flips.	Yes	Yes	Change in class confidence	Classification & Machine Translation
Gao et al. [5]	Change in characters / words w.r.t token importance	No	No	Change in class	Classification
Zhao et al. [21]	Generate adversarial examples by using Adversarially regularized autoencoders.	Yes	Yes	Change in class	Textual Entailment & Machine translation

Table 4: Definition used for previous work on adversarial examples

Recently, there has been many previous work done on adversarial examples in the text domain. Broadly speaking, the attacks can be categorized as gradient based and non-gradient based. For gradient based attacks the adversarial input is chosen based on change in cost functions and model gradients, which are also known as *white-box* attacks for their ability to look into the model while constructing the adversarial input. Similarly, non-gradient based attacks rely on clever input manipulations such as misspellings, addition, removal or replacement of words keeping the same semantic meanings. These kind of attacks are also termed as *black-box* attacks. We present a brief review over the existing works in Table 4. We provide an additional column on human perception, which denotes whether the paper has accounted for human perception of the attack in some way. That is whether the proposed attacks can be discerned from the original text by human annotators.

A.1 Definitions

Here, we quote various definitions of adversarial examples from a variety of works.

We expect such network to be robust to small perturbations of its input, because small perturbation cannot change the object category of an image. However, we find that applying an imperceptible non-random perturbation to a test image, it is possible to arbitrarily change the network’s prediction. [20]

That is, these machine learning models misclassify examples that are only slightly different from correctly classified examples drawn from the data distribution [6]

Adversarial examples are examples that are created by making small perturbations to the input designed to significantly increase the loss incurred by a machine learning model [14]

Our goal is to design pairs of sentences such that the NLI relation within a pair (entailment, neutral or contradiction) can be changed without changing the words involved, simply by changing the word ordering within each sentence. [3]

We define an adversary A to be a function that takes in an example (p, q, a), optionally with a model f, and returns a new example (p_0, q_0, a_0). The adversarial accuracy with respect to A is $Adv(f) = \frac{1}{|D_{test}|}\sum_{(p,q,a)\in D_{test}} v(A(p, q, a, f), f))$. While standard test error measures the fraction of the test distribution over which the model gets the correct answer, the adversarial accuracy measures the fraction over which the model is robustly correct, even in the face of adversarially-chosen alterations...Instead of relying on paraphrasing, we use perturbations that do alter semantics to build concatenative adversaries, which generate examples of the form $(p + s, q, a)$ for some sentence s. In other words, concatenative adversaries add a new sentence to the end of the paragraph, and leave the question and answer unchanged. [9]

An adversarial sample can be defined as one which appears to be drawn from a particular class by humans (or advanced cognitive systems) but fall into a different class in the feature space. [18]

maliciously crafted inputs that are undetectable by humans but that fool the algorithm into producing undesirable behavior [11]

Adversarial examples are inputs to a predictive machine learning model that are maliciously designed to cause poor performance [4]

One type of the vulnerabilities of machine learning algorithms is that an adversary can change the algorithm output by subtly perturbing the input, often unnoticeable b y humans. [8]

Adversarial attack on deep neural networks (DNNs) aims to slightly modify the inputs of DNNs and mislead them to make wrong predictions [1]

For a given sample x and a trained DNN classifier model F, the attacker aims to craft an adversarial sample x = x + x by adding a perturbation x to x, such that F(x*) F(x)...In order to maintain the utility of a text sample, we perturb the sample not only by directly modifying its words, but also inserting new items (words or sentences) or removing some original ones from it.* [13]

A.2 Adversarial Gain Perspectives of Prior Work

Here we examine various works and how they can fit into the adversarial gain perspective. We already demonstrate how [1] and [4] can be measured in terms of adversarial gain. Rather than non-overlapping text in [1], we can examine the semantic change of the output. Similarly, we can examine how well the noise preserves the meaning of the input sentence in both cases. If the semantic shift is too far, this discounts the shift in output it causes.

Generally, most text-based adversarial attacks constrain their inputs by in some way changing words while retaining meaning. This includes negation [8], misspelling [8, 18, 13], changing word order [3], replacing with with synonyms [1], or simply perturbing the word embeddings [14]. In many cases, such constraints will preserve the meaning of the original text, but often too strict constraints can result in lower success. For example, in [4] the word-based replacement with a strict synonym constraint resulted in a low success rate in adversarial examples. In other cases, preservation of word meaning is not guaranteed. In fact, prior work has used samples from the generated attacks posed as surveys to determine whether meaning is preserved [9], but this has not typically been done in a systematic way and Jia and Liang [9] found that in some cases meaning was not preserved. In another example, negation of phrases does not preserve meaning and thus a model could be totally correct in changing its output. In all attacks, it is possible to evaluate preservation of meaning by using a well-defined embedding space (such as [2] as a start) and the cosine distance. The use of such a distance as we do as part of adversarial gain, allows attacks to change meaning and account for this when evaluating the change of the model output.

In evaluating the results of an adversarial attack, there are many measures used. For classification tasks, a change in class label is typically used as a success criterion [21, 5, 11, 18, 3]. In other cases, notions such as changes in some scoring or cost function are used [1, 9, 8, 14]. However, due to the change in inputs if the cost relates to the original sentence and the meaning is *not preserved*, the cost may be evaluating the wrong criterion without access to an oracle. Thus adversarial gain accounts for this by discounting output performance changes by the distance from the input. In a well defined feature space where inputs and outputs are correlated this ensures that as an adversarial input moves away from its original meaning, this is accounted for in the evaluation criteria to some extent.

B Extended Perspectives on Adversarial Gain

Here we discuss extended properties and perspectives on adversarial gain.

B.1 Possible Feature Spaces and Distance Metrics to Measure Gain

There are a number of different priors that can be used to measure gain in different ways for different tasks. While in the main text we examine sentiment classification tasks and text summarization, others may be relevant in domains such as dialogue systems. For example, one can use sentiment classification probability and the likelihood divergence (or a step function intersection) to measure difference in output of a dialogue system, text summary system, or other generative model. Similarly, various sentence embeddings can be used (Infersent, Doc2Vec, etc.). Word-wise word vector distance can also be used. Each of these notions of adversarial gain essentially provide a different prior on the stability of the systems in different ways. For example, it is likely that unless the sentiment of an input to a dialogue system doesn't change dramatically, neither should the output.

C Experimental Setup

In our selection of text-based attacks, we examined which attacks provided easily available open-source code. Many code to replicate experiments was either unavailable or we were unable to find. We settled on two text-based attacks. We used the Seq2Sick attack on text summarization by Cheng et al. [1] and the word-level sentiment classification attack by Ebrahimi et al. [4]. Scripts and full instructions that we used to run the code from these papers is provided at: anonymized. More samples with gain and distances provided can be found in the codebase provided.

C.1 Text Summarization

We use the pre-trained model and code for a text summarization model based on the Open Neural Machine Translation toolkit (OpenNMT) [10] as provided by Cheng et al. [1] at https://github.com/cmhcbb/Seq2Sick. We use the GigaWord corpus the authors reference from [17] based on prior versions of the datset [7, 15].

When we measure cosine distance, we use the inverse of cosine similarity to follow the intuition the a distance metric should keep similar words closer together. Assuming that cosine similarity is bounded $[0, 1]$, cosine distance is $1 - |similarity(x, y)|$.

C.2 Sentiment Classification

For sentiment classification we use the binary version of the SST dataset [19] called SST2. This removes all neutral labels. This is the same dataset as used by [4]. We use their provided code for the word-level adversarial attack and SST2 pre-processing scripts found at: https://github.com/AnyiRao/WordAdver and https://github.com/AnyiRao/SentDataPre. We use the pre-trained convolutional neural network classification model provided by the authors and the attack as provided in our accompanying instructions. The only change we make is that we remove the cosine similarity requirement on replacement words. We do this because otherwise the attack only generates attacks for 95 samples. Removing this requires generates attacks for all samples (though many are not successful). We note that this allows words to be added by replacing padding characters, while this differs slightly from the attack mentioned by [4], the authors there do discuss that this attack has a low success rate particularly due to their restrictions. Because adversarial gain as a definition does not require constraints, this allows us to consider the larger set of attacks.

References

[1] Minhao Cheng, Jinfeng Yi, Huan Zhang, Pin-Yu Chen, and Cho-Jui Hsieh. 2018. Seq2sick: Evaluating the robustness of sequence-to-sequence models with adversarial examples. *arXiv preprint arXiv:1803.01128*.

[2] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*.

[3] Ishita Dasgupta, Demi Guo, Andreas Stuhlmüller, Samuel J Gershman, and Noah D Goodman. 2018. Evaluating compositionality in sentence embeddings. *arXiv preprint arXiv:1802.04302*.

[4] Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2017. Hotflip: White-box adversarial examples for nlp. *arXiv preprint arXiv:1712.06751*.

[5] Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. *arXiv preprint arXiv:1801.04354*.

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

[7] David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2003. English gigaword. *Linguistic Data Consortium, Philadelphia*, 4:1.

[8] Hossein Hosseini, Sreeram Kannan, Baosen Zhang, and Radha Poovendran. 2017. Deceiving google's perspective api built for detecting toxic comments. *arXiv preprint arXiv:1702.08138*.

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

[10] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush. OpenNMT: Open-Source Toolkit for Neural Machine Translation. *ArXiv e-prints*.

[11] Volodymyr Kuleshov, Shantanu Thakoor, Tingfung Lau, and Stefano Ermon. 2018. Adversarial examples for natural language classification problems.

[12] Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*.

[13] Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. 2017. Deep text classification can be fooled. *arXiv preprint arXiv:1704.08006*.

[14] Takeru Miyato, Andrew M Dai, and Ian Goodfellow. 2016. Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725*.

[15] Courtney Napoles, Matthew Gormley, and Benjamin Van Durme. 2012. Annotated gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, pages 95–100. Association for Computational Linguistics.

[16] Nicolas Papernot, Patrick McDaniel, Ananthram Swami, and Richard Harang. 2016. Crafting adversarial input sequences for recurrent neural networks. In *Military Communications Conference, MILCOM 2016-2016 IEEE*, pages 49–54. IEEE.

[17] Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*.

[18] Suranjana Samanta and Sameep Mehta. 2017. Towards crafting text adversarial samples. *arXiv preprint arXiv:1707.02812*.

[19] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

[20] 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*.

[21] Zhengli Zhao, Dheeru Dua, and Sameer Singh. 2017. Generating natural adversarial examples. *arXiv preprint arXiv:1710.11342*.