Measuring Robustness for NLP

Yu Yu[*] and **Abdul Rafae Khan**[*] and **Jia Xu**[*]
[*] School of Engineering and Science, Steven Institute of Technology, NJ 07030, USA
yyu50@stevens.edu, akhan4@stevens.edu, jxu70@stevens.edu

Abstract

The quality of Natural Language Processing (NLP) models is typically measured by the accuracy or error rate of a predefined test set. Because the evaluation and optimization of these measures are narrowed down to a specific domain like news and cannot be generalized to other domains like Twitter, we often observe that a system reported with human parity results generates surprising errors in real-life use scenarios. We address this weakness with a new approach that uses an NLP quality measure based on robustness. Unlike previous work that has defined robustness using Minimax to bound worst cases, we measure robustness based on the consistency of cross-domain accuracy and introduce the coefficient of variation and (ϵ, γ)-Robustness. Our measures demonstrate higher agreements with human evaluation than accuracy scores like BLEU on ranking Machine Translation (MT) systems.

1 Introduction

Evaluation criteria serve as learning objectives and model assessment standards and are crucial for NLP research. Conventional evaluation methods compute the accuracy or errors compared to the reference on a predefined test set, such as BLEU or TER. Since the evaluation results highly depend on the test set, they may not apply to real-world test inputs that come from an unknown distribution. For example, one may query a legal document while the system is trained on the news. Therefore, it is necessary to define a measure that can give an idea of how robust system performance will be on unseen test data so that we can predict whether our model is generalizable to new domains.

There has been influential investigation into defining robustness instead of accuracy alone (Bastani et al., 2016; Hein and Andriushchenko, 2017; Weng et al., 2018). Robustness of a machine learning model can be described as the characteristic of how accurate the model is in making its predictions when tested on a new (but similar) dataset. For example, one definition of robustness claims "If a testing sample is *similar* to a training sample, then the testing error is close to the training error" (Xu and Mannor, 2012). Such testing samples, which are samples dissimilar from training samples are known as adversarial examples. Many studies on robustness measures (Heinze-Deml and Meinshausen, 2017; Araujo et al., 2019; Carlini and Wagner, 2017) focus on the worst-case scenarios with adversarial inputs.

However, because the worst-case appears very infrequently, in particular, if artificially simulated (Wang et al., 2020; Jin et al., 2020; Alzantot et al., 2018), a worst-case analysis has the inherent problem that if the worst case is far from the typical, then the quantification assigns a numerical value to cases that occur rarely, like outliers. For instance, if a system accuracy has a very small variance in a million test cases but fails dramatically in one, then Minimax will label this system less robust than a system with a high accuracy variance. While focusing on the worst-case can be important for some areas of computer science and engineering like astronautics, but in our view, it is usually not desirable in the NLP context. In our view, robustness is a notion that should address typical behavior, not atypical and rare behavior. Note that if the worst-case differs a lot from the typical, a statistical notion of robustness should make this apparent. Additionally, a useful formalization should ignore atypical behavior of our system - because only focusing on a worst-case is not practical.

In this paper, we introduce novel definitions of robustness for NLP systems. Instead of defining robustness in terms of worst-case performance, we define it in terms of the typical behavior of the system and the consistency in their performance of the system. Under this definition, the more inconsistent the model predictions are, the less robust the

Proceedings of the 29th International Conference on Computational Linguistics, pages 3908–3916
October 12–17, 2022.

model is. Specifically, we simulate unknown test domains using leaving-one-out cross-validation to measure the variance of model accuracy on a set of test domains. More precisely, we introduce the coefficient of variation to quantify the variants of accuracy that is non-linear.

We further expand this notion to a probabilistic definition, called (ϵ, γ)-Robustness, where the higher probability of consistent behavior, the more robust the system. We provide statistical guarantees on the performance of an NLP model through the following assertion:

“I certify that, with high probability, my NLP model’s performance will not change too much, given test sets from different domains and/or with various types of noise."

Both variation of coefficient and (ϵ, γ)-Robustness takes a new direction away from the worst-case result to a general consistency of the model quality. They are meta-evaluation methods that measure the consistency of user-defined quality metrics. Thus, any standard evaluation metrics can be applied to our paradigm. Our robustness measures are evaluated by comparing with human rankings on system outputs and by simulating unknown test domains using cross-validation. Our experimental results show that our robustness measures have higher agreements with human rankings on machine translation (MT) system submissions than BLEU scores.

In summary, the main contributions of this work include:

1. introducing a definition of robustness which is different from the typical notion of performance in worst-case scenarios;

2. developing a probabilistic definition of robustness based on Chebyshev’s inequality;

3. experimenting on four different NMT models using cross-validation techniques to simulate the scenarios of unknown test data;

4. evaluating using human assessment on WMT submission systems;

5. carrying out human annotation experiments as a comparison with our measure;

The rest of the paper is as follows: In Section 2, we describe the three definitions of robustness. In Section 2.4, we connect the coefficient of variation with (ϵ, γ)-Robustness and show their relationship.

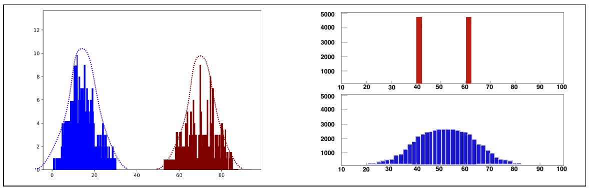

Figure 1: *Left:* Shortage of Variance; *Right:* Shortage of coefficient of variation.

Section 3 describes the six algorithms to compute robustness metrics we used during our experimentation. Section 4 describes our evaluation methods including cross-validation and manual comparison. Section 5 describes the experiments that evaluate our robustness measures. In Section 6, we discuss the previous literature on robustness in machine learning. Section 7 concludes the paper.

2 Definition of robustness

We define robustness as the consistency in the behavior of a machine-learned system. We think of it as the standard or typical behavior of the system. The more this behavior deviates from the typical, the less robust the system is defined to be. Notice that this definition does not give a notion of whether the performance of the system is good or bad. A system with consistent terrible performance is still a robust system under this definition. For example, in terms of MT, this definition becomes the consistency in translation performance for a trained MT system.

2.1 Variance

We take the NLP model accuracy (e.g, BLEU score) as a random variable. We introduce measuring the variance of, for instance, the BLEU score to indicate the consistency of the translation quality over the combination of various test sets. This random variable will give us a value that quantifies how stable is a translation system over different test sets.

2.2 Coefficient of Variation

However, the same variance value measured on the datasets with different means will carry a different meaning. For example, as left of Figure 1 shows, the variance measured on a dataset which has the mean of 10% in the BLEU score indicates a much higher inconsistency than that with the mean of 70%. Therefore, we can scale the variance by the mean, and finally, use the coefficient of the variation to measure the consistency over the accuracy scores across test sets in the dataset pool. The coefficient of variation is a scaled variance. Nonetheless, the variance or the coefficient of variation is not sufficient to express the consistency of the accuracy. For example, as right of Figure 1 shows, the accuracy can only have two values: 0 and 1 but have the same variance as values following the normal distribution. Thus looking at the distribution itself is crucial to decide on the robustness of a model.

2.3 (ϵ, γ)-Robustness

We follow the direction of the probabilistic robustness (Xu and Mannor, 2012) and introduce the notion of (ϵ, γ)-robust to consider all cases.

Briefly speaking, we want to measure the probability of upper bounding the NLP model accuracy gap between any test set and their average.

We call an NLP system (ϵ, γ)-robust, if for every test set drawn from a distribution D, its prediction error (or accuracy) X is centered around the mean error (or accuracy) μ, which is bounded through a parameter ϵ with a probability of $1 - \frac{\sigma^2}{\epsilon^2} \cdot \gamma$.

$$Pr[|X - \mu| < \epsilon] = 1 - \frac{\sigma^2}{\epsilon^2} \cdot \gamma \tag{1}$$

This definition is a relaxation of Chebyshev's inequality by adding $\gamma \in [0, 1]$. In the above formulation, the lower the value of γ, the more robust is the system. $1 - \gamma$ value indicates how much tighter we can bound the prediction accuracy around its mean than the Chebyshev's bound. A robust system can be provided with a tighter bound, while the Chebyshev in Equation 2 bounds a fragile system. Below is the inference.

$$\begin{aligned} Pr[|X - \mu| \geq \epsilon] &\leq \frac{\sigma^2}{\epsilon^2} && (2) \\ \Leftrightarrow 1 - Pr[|X - \mu| \geq \epsilon] &\geq 1 - \frac{\sigma^2}{\epsilon^2} && (3) \\ \Leftrightarrow Pr[|X - \mu| < \epsilon] &\geq 1 - \frac{\sigma^2}{\epsilon^2} && (4) \end{aligned}$$

Chebyshev's inequality in Equation 2 provides an upper bound to the probability that the difference between X and the mean will exceed a given threshold. If we put "$1-$" in front of both sides, we have Equation 3, thus Equation 4, which shows a lower bound of the difference between X and μ.

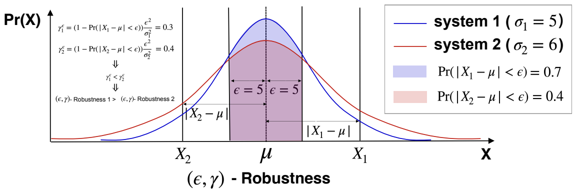

Figure 2: Illustration of the (ϵ, γ)-Robustness

Algorithm 1 Plot (ϵ, γ)-Robustness

Input: translation accuracy (e.g. BLEU or human eval) of each sentence of a test set on a given translation model

Output: 100 (ϵ, γ) values

```
for $\epsilon$ in 1 to 100 do
    output $\gamma := (1 - Pr[|X - \mu| = \epsilon]) \cdot \frac{\epsilon^2}{\sigma^2}$
end for
return
```

In robustness measure, we are interested in coming up with a bound tighter than the Chebyshev's bound. To make it scalable and interpretable, we can add $\gamma \in [0, 1]$ to express how much tighter bound we can provide than Chebyshev's. If $\gamma = 1$ then we have the worst case, the system is not robust at all; if γ is approaching to 0, then it is getting very tightly bounded.

Therefore, we introduce γ **to be an indicator of how robust a system is**. ϵ is a parameter that we can explore. Algorithm 1 shows the algorithm of plotting the ϵ, γ-Robustness. γ relates to the probability that accuracy X is within a given threshold, and ϵ controls the width of such threshold, as illustrated in the example of Figure 2.

2.4 Relating the (ϵ, γ)-Robustnes to the coefficient of variation

ϵ is a hyper-parameter related to our robustness metrics. We show the (ϵ, γ)-Robustnes can be connected with the coefficient of variatioin by simply tuning ϵ.

Replace ϵ by ϵ', $\epsilon = \epsilon' \cdot \frac{\sigma}{\mu}$ to Chebyshev's inequality:

$$\begin{aligned} Pr[|X - \mu| \geq \epsilon] &\leq \frac{\sigma^2}{\epsilon^2} && \text{Eq. 2} \\ \Rightarrow Pr[|X - \mu| \geq \epsilon' \cdot \frac{\sigma}{\mu}] &\leq \frac{\sigma^2}{\epsilon'^2 \cdot \frac{\sigma}{\mu}^2} && (5) \\ \Rightarrow Pr[|X - \mu| \geq \epsilon' \cdot \frac{\sigma}{\mu}] &\leq \frac{\mu^2}{\epsilon'^2} && (6) \end{aligned}$$

Replace ϵ by ϵ', $\epsilon = \epsilon' \cdot \frac{\sigma}{\mu}$ to the $\epsilon - \gamma$ robust definition

$$Pr[|X - \mu| < \epsilon' \cdot \frac{\sigma}{\mu}] = 1 - \frac{\mu^2}{\epsilon'^2} \cdot \gamma \tag{7}$$

3 Robustness metrics

Using our definition of robustness (see Section 2), we create three different robustness metrics. The first metric (Algorithm 2) computes the variance for all the samples in the test pool. The second metric (Algorithm 3) scales variance by the mean of all samples. The third metric (Algorithm 4) computes the (ϵ, γ)-Robustness given a pre-defined parameter ϵ.

Algorithm 2 Robustness Metrics I: Variance

```
Require: Error function $\epsilon(\cdot)$; a test set pool $T$ containing $N$ samples ($t$s).
    $V(T) = \frac{1}{I} \sum_i \{\epsilon(t_i; s, A) - \frac{1}{I} \sum_j \epsilon(t_j; s, A)\}^2$
```

Algorithm 3 Robustness Metrics II: Coefficient of Variation

```
Require: Error function $\epsilon(\cdot)$; a test set pool $T$ containing $N$ samples ($t$s).
    $COV(T) = \frac{\frac{1}{I} \sum_i \{\epsilon(t_i;s,A) - \frac{1}{I} \sum_j \epsilon(t_j;s,A)\}^2}{\frac{1}{I} \sum_j \epsilon(t_j;s,A)}$
```

Algorithm 4 Robustness Metrics III: (ϵ, γ)-Robustness

```
Require: Error function $\epsilon(\cdot)$; a test set pool $T$ containing $N$ samples ($t$s), hyper-parameter $\epsilon$.
    $\mu = \frac{1}{I} \sum_i \epsilon(t_i; s, A)$
    $\sigma = \sqrt{\frac{\sum (t_i - \mu)^2}{I-1}}$
    $Pr[|t_i - \mu| < \epsilon] = \frac{|\{t_j | t_j - \mu < \epsilon\}|}{I}$
    $(\epsilon, \gamma) - Robustness(T) = (1 - Pr[|t_i - \mu| = \epsilon]) \cdot \frac{\epsilon^2}{\sigma^2}$
```

We have a collection of test samples, where each test sample is t_i, and there are I many test samples in the test pool. An error function $\epsilon(t; s, A)$ indicates the translation error (1-accuracy) on a test sample t_i on s according to evaluation criterion A. The variance measures the "consistency" of the translation accuracy among all the test samples.

Bootstrapping Considering the test scores on one test sample can be unstable and inaccurate, we create a more robust method to compute robustness by bootstrapping subsamples from the entire test pool. We randomly subsample the test pool into M bootstraps and then compute the average robustness score (variance/coefficient of variation/(ϵ, γ)-Robustness) across bootstraps.

Algorithm 5 Robustness Metrics by Bagging

```
Require: Error function $\epsilon(\cdot)$; a test set pool $T$ containing $I$ samples: $t_1 \cdots t_I$; block size $b$, each block $B_1 \cdots B_M$ with number of blocks $M$, universe size $N$.
    Initialize $m$ empty blocks.
    for $b\_ = 1$ to $b \cdot M$ do
        choose $L$ at random from the set of blocks with current min number of elements
        $S$ : set of elements in the universe not in L
        $L = L \cup t$, where $t \in S$ chosen uniformly at random
    end for
    for $m \in M$ do
        Compute $R(B_m)$ according to Algorithm 2/Algorithm 3/Algorithm 4:
        $R(B_m) = V(B_m)/COV(B_m)/(\epsilon, \gamma) - Robustness(B_m)$
    end for
    $AV(T) = \frac{1}{M} \sum_m R(B_m)$
```

In practice, random sampling requires the times of bootstrapping to be relatively large to achieve a thorough coverage of the test pool. To complement, we create two bootstrapping methods to selectively design the subsampled bootstraps. First, for each bootstrap, we randomly sample from the elements not present in the current bootstrap, which is described in Algorithm 5. Second, we subsample from the set of elements with least sampled frequency (Papakonstantinou et al., 2014), which is described in Algorithm 6.

4 Evaluation of the robustness measures

4.1 Cross-validation

We use the leave-one-out error stability to show that the robustness measure can be generalized if we exclude a left-out test set from all four datasets where we measure the robustness. More precisely, for a given translation model, we randomly select one leaving-one-out test set and then combine all other tests to compute the variance, the mean, and the coefficient of variation on the left-out datasets, as shown in Figure 3.

Algorithm 6 Robustness Metrics by Design Bagging

```
Require: Error function $\epsilon(\cdot)$; a test set pool $T$ containing $I$ samples: $t_1 \cdots t_I$; block size $b$, each block $B_1 \cdots B_M$ with number of blocks $M$, universe size $N$.
  Initialize $m$ empty blocks.
  for $b\_ = 1$ to $b \cdot M$ do
    choose $L$ at random from the set of blocks with current min number of elements
    $S$ : set of elements in the universe not in L that appear least frequently
    $L = L \cup t$, where $t \in S$ chosen uniformly at random
  end for
  for $m \in M$ do
    Compute $R(B_m)$ according to Algorithm 2/ Algorithm 3/Algorithm 4:
    $R(B_m) = V(B_m)/COV(B_m)/(\epsilon, \gamma) - Robustness(B_m)$
  end for
  $AV(T) = \frac{1}{M} \sum_m R(B_m)$
```

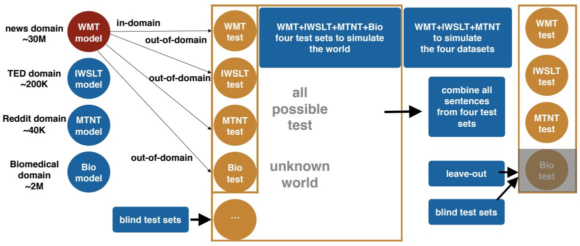

Figure 3: Leaving-one-out to verify robustness metrics.

4.2 Correlation with manual evaluation: pair-wise system comparison

Human evaluation is another way to evaluate the robustness measure. More precisely, we have human linguists come up with translation test sentences and evaluate the consistency of the translation model 1 (θ) and translation model 2 (θ'). The human evaluators can ask as many as translation sentences until she/he decides on the ranking of the performance between model 1 and model 2. A perfect robustness estimator ρ would satisfy that the human ranking of the robustness of two NLP systems is the same as the ranking by our robustness measures:

$$\rho_{human}(\theta) < \rho_{human}(\theta') \iff \rho(\theta) < \rho(\theta') \quad (8)$$

In other words, we do not necessarily need the actual value of ρ to verify that it gives us enough information to compare the two models. There are some problems with this approach, though, not least because we need to evaluate this on multiple models.

5 Experiments

Data & Tools We evaluate our robustness metrics in machine translation and sentiment classification. For machine translation, we use four different English-French NMT models trained on WMT'14, Biomedical'18, ISLWT'17, and MTNT'18 datasets. The models are trained using 35M, 2M, 200K, and 40K sentences, respectively. The WMT'14 model is the pre-trained model provided by Facebook research (Ott et al., 2019) and the remaining were trained in house using ConvS2S Toolkit (Gehring et al., 2017) till convergence. The development data used for the in house trained models include Khresmoi for Biomedical, test2014 and test2015 for IWSLT, and MTNT'18 development for MTNT. The four different test data were also from the same domains, including WMT (newstest14), Biomedical (EDP2018), IWSLT (test2017), and MTNT (MTNT2018).

For sentiment classification, we use Amazon product review dataset (Blitzer et al., 2007). Specifically, we train four models using pre-processed balanced reviews from DVD, kitchen, electronics and books domains where each review is labeled with 0 or 1. We randomly split 15% of the sentences as validation set and the rest as training set. We evaluate the classification model on unprocessed reviews from 21 domains different from the training domain, and compute the mean, var and coefficient of variance of these 21 test accuracy scores. Then we leave the apparel, baby, beauty, grocery and music domain out to compute the mean, variance, and the coefficient of variation on other 20 domains. Different from machine translation that uses sentence level performance score (i.e. BLEU) to compute robustness, we use domain/document level performance score (i.e. accuracy) to compute robustness metrics in sentiment classification.

Results As addressed in Section 3, We carry out three different robustness experiments corresponding to robustness metrics with no sampling method, robustness by bagging, and robustness by design bagging, respectively. For every experiment, we use each of the training models and translated all the four test sets. We calculate the sentence level BLEU scores and use one of the robustness metrics to compute the change in the BLEU scores. We use a combined test data from all the four domains. We want our model of robustness measure to work on any test domain. In order to simulate a blind test domain, we apply leave-one-out testing, where we leave out one domain from the four domains. Therefore we calculate the mean, variance, and the coefficient of variation on the five testing environments.

Results in Table 1 and Table 2 use all training samples to measure. The rest four tables experiment with bagging techniques on sub-sampled test data where we create 30 bootstraps each containing 60% of the data. We calculate the mean and variance of each of the bootstrap and finally average values of these measures to calculate the coefficient of variation. For machine translation, we can observe that for models, including WMT, the change in the coefficient of variation when testing on different leave-one-out scenarios, does not change a lot compared to IWSLT. For sentiment classification, the model trained on kitchen domain has the lowest coefficient of variation compared to all other three models, and the coefficient of variation of the model trained on electronics domain varies largely when testing on different leave-one-out scenarios. Table 3 and table 4 use the bagging algorithm (Breiman, 1996) to make the intersection between bootstraps as less as possible. The mean, variance, and coefficient of variation were calculated the same way as in traditional bagging experiment.

Results using design bagging are mentioned in Table 5 and Table 6. A similar trend is observed where the difference in coefficient of variation across different leave-one-out experiments is the smallest for WMT and is much higher for MTNT.

To compute our definition of (ϵ, γ)-Robustness, we used the same setup of four different trained models and four different test domains. For each possible value of ϵ (between 0 and 100), we try to find the value of γ, which satisfies the Equation 2. Figure 4 shows the normalized γ values for each corresponding epsilon value for the four models. We can observe the γ values for WMT are much smaller than the values for other models, and the values for MTNT are the highest. This shows that the WMT model is the most robust among all models, and MTNT is the least robust. Similarly, our (ϵ, γ)-Robustness can be applied with human evaluation scores, as shown in Figure 5. Based on these four NMT systems in Figure 5, we rank them using (ϵ, γ)-Robustness and find the rank 100% agrees with the rank given by human, while the rank given by the BLEU score 75% agrees to human, as shown in Table 7. This suggests that (ϵ, γ)-Robustness evaluates robustness of systems better than accuracy-based metrics (i.e. BLEU).

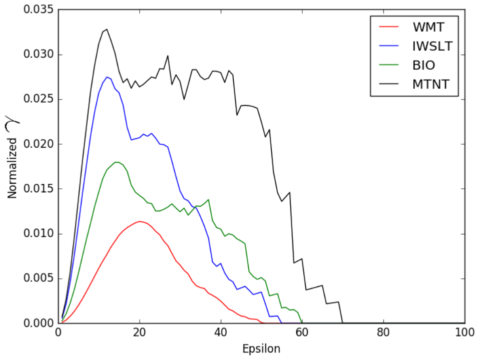

Figure 4: Normalized (ϵ, γ)-Robustness plot on our models.

Finally, as in Section 4.2, to better compare our method, we conduct a human evaluation for pair-wise comparison of the models. We create a small web-based application where the human annotator is assigned two NMT models selected at random. The human annotators do not know any details about the model or the training data used for each model. She/He can only use these models to get two translation outputs for a given input sentence. This step can be repeated as many times as possible until the human annotator decides which model is more "consistent" in its translations. Table 8 mentions the pair-wise results for the four models using the human annotators. For comparison, we have also mentioned the coefficient of variation and the (ϵ, γ)-Robustness results. We observe that only one out of the six scenarios is different (16.66% error rate) for the human annotators and our robustness metrics.

6 Related Work

There have been substantial amount of work trying to improve the robustness of NLP models. Among those, a majority of focus lies on the vulnerability of NLP models to input perturbations, such as generating adversarial examples. For instance, Wang et al. (2020) uses controllable attributes irrelevant to task labels to generate diverse adversarial texts. Li et al. (2020) uses pre-trained masked language

Model	Leave-One-Out Test Set																	
	None			WMT'15			IWSLT'17			MTNT'19			BIO'18			MTNT'18		
	Mean	Var	COR	Mean	Var	COR	Mean	Var	COR	Mean	Var	COR	Mean	Var	COR	Mean	Var	COR
WMT'14	32.52	723.56	0.83	31.33	684.75	0.84	32.35	735.06	0.84	33.86	714.98	0.79	32.51	747.96	0.84	32.52	723.56	0.83
IWSLT'17	10.35	206.43	1.39	10.55	211.85	1.38	8.71	171.72	1.50	11.44	219.58	1.30	10.6	215.8	1.39	10.35	206.43	1.39
MTNT'19	6.97	173.38	1.89	7.19	179.26	1.86	6.2	156.24	2.02	6.73	166.43	1.92	7.61	186.79	1.80	6.97	173.38	1.89
BIO'18	15.36	330.83	1.18	15.7	328.59	1.15	15.37	339.74	1.20	15.89	327.4	1.14	14.66	327.51	1.23	15.36	330.83	1.18
APPERTIUM	1.83	49.20	3.83	1.68	44.11	3.95	1.77	48.53	3.94	2.04	52.74	3.56	1.72	46.72	3.98	1.95	53.79	3.76

Table 1: Robustness Metrics I & II in machine translation. Sentence level BLEU scores when a single test set is left out. *Var* is variance of BLEU scores and *COR* is the coefficient of variance.

Model	Leave-One-Out Test Set																	
	None			apparel			baby			beauty			grocery			music		
	Mean	Var	COR	Mean	Var	COR	Mean	Var	COR	Mean	Var	COR	Mean	Var	COR	Mean	Var	COR
dvd	64.33	2.04	3.18	65.02	2.05	3.16	64.91	2.08	3.21	64.13	2.16	3.37	63.76	2.07	3.25	65.09	2.08	3.19
books	64.33	2.07	3.23	65.10	2.06	3.17	64.95	2.11	3.25	64.04	2.09	3.26	63.70	2.03	3.19	65.03	2.04	3.14
kitchen	58.94	0.60	1.02	59.28	0.59	1.0	59.25	0.62	1.05	59.06	0.64	1.08	58.44	0.60	1.02	59.34	0.63	1.06
electronics	64.45	2.10	3.26	64.94	2.01	3.10	64.91	2.10	3.23	64.17	2.15	3.36	63.77	2.10	3.30	65.03	2.06	3.17

Table 2: Robustness Metrics I & II in sentiment classification. Domain level accuracy scores when a single test set is left out. *Var* is variance of accuracy scores and *COR* is the coefficient of variance.

Model	Leave-One-Out (Bagging)																	
	None			WMT'15			IWSLT'17			MTNT'19			BIO'18			MTNT'18		
	Mean	Var	COR	Mean	Var	COR	Mean	Var	COR	Mean	Var	COR	Mean	Var	COR	Mean	Var	COR
WMT'14	32.09	706.49	0.83	30.72	667.16	0.84	31.56	701.76	0.84	32.79	694.34	0.80	31.95	725.51	0.84	32.48	723.32	0.83
IWSLT'17	8.56	185.86	1.59	8.08	182.21	1.67	6.70	146.26	1.80	8.86	194.29	1.57	8.60	192.21	1.61	10.33	205.54	1.39
MTNT'19	5.76	150.79	2.13	5.52	147.63	2.20	4.72	124.94	2.37	5.21	136.62	2.24	6.19	159.87	2.04	7.00	173.80	1.88
BIO'18	12.67	306.96	1.38	12.03	295.04	1.43	11.79	301.49	1.47	12.27	297.29	1.41	11.86	298.77	1.46	15.32	329.75	1.19
APPERTIUM	2.83	81.19	3.19	2.70	74.86	3.20	2.80	80.72	3.21	3.01	82.15	3.01	2.78	80.18	3.22	2.91	85.65	3.18

Table 3: Robustness Metrics by Bagging in machine translation. Sentence level BLEU scores of bagging of test with a single test set is left out. *Var* is variance of BLEU scores and *COR* is the coefficient of variance.

Model	Leave-One-Out (Bagging)																	
	None			apparel			baby			beauty			grocery			music		
	Mean	Var	COR	Mean	Var	COR	Mean	Var	COR	Mean	Var	COR	Mean	Var	COR	Mean	Var	COR
dvd	58.95	0.012	0.02	59.27	0.026	0.043	59.15	0.014	0.025	58.73	0.018	0.03	58.53	0.018	0.032	59.65	0.021	0.036
books	50.21	0.035	0.069	50.87	0.02	0.04	50.61	0.021	0.042	50.43	0.023	0.047	49.92	0.037	0.075	50.72	0.027	0.053
kitchen	50.88	0.013	0.026	51.25	0.015	0.03	51.15	0.019	0.038	50.39	0.025	0.049	50.22	0.024	0.047	51.06	0.022	0.043
electronics	42.38	0.015	0.037	42.02	0.028	0.068	42.13	0.017	0.041	42.58	0.022	0.053	42.68	0.016	0.038	41.79	0.016	0.039

Table 4: Robustness Metrics by Bagging in sentiment classification. Domain level accuracy scores when a single test set is left out. *Var* is variance of accuracy scores and *COR* is the coefficient of variance.

Model	Leave-One-Out (Design Bagging)																	
	None			WMT'15			IWSLT'17			MTNT'19			BIO'18			MTNT'18		
	Mean	Var	COR	Mean	Var	COR	Mean	Var	COR	Mean	Var	COR	Mean	Var	COR	Mean	Var	COR
WMT'14	31.94	703.93	0.83	30.85	667.02	0.84	31.61	707.62	0.84	32.80	693.68	0.80	31.87	722.14	0.84	32.51	723.56	0.83
IWSLT'17	8.55	185.95	1.59	8.10	182.49	1.67	6.70	145.45	1.80	8.82	192.42	1.57	8.59	192.10	1.61	10.35	206.56	1.39
MTNT'19	5.76	150.24	2.13	5.51	146.54	2.20	4.76	126.83	2.37	5.19	136.47	2.25	6.16	160.23	2.05	6.97	173.48	1.89
BIO'18	12.69	307.07	1.38	12.03	295.87	1.43	11.82	303.21	1.47	12.27	297.13	1.41	11.88	298.30	1.45	15.35	330.40	1.18
APERTIUM	2.84	80.88	3.17	2.69	74.30	3.20	2.82	82.10	3.22	3.00	81.75	3.01	2.77	80.18	3.24	2.92	85.08	3.16

Table 5: Robustness Metrics by Design Bagging in machine translation. Sentence level BLEU scores of design bagging of test with a single test set is left out. *Var* is variance of BLEU scores and *COR* is the coefficient of variance.

Model	Leave-One-Out (Design Bagging)																	
	None			apparel			baby			beauty			grocery			music		
	Mean	Var	COR	Mean	Var	COR	Mean	Var	COR	Mean	Var	COR	Mean	Var	COR	Mean	Var	COR
dvd	57.80	0.02	0.035	59.27	0.028	0.047	59.03	0.021	0.035	57.93	0.019	0.032	57.58	0.018	0.031	59.29	0.027	0.046
books	59.54	0.025	0.042	61.20	0.033	0.055	60.92	0.024	0.04	59.78	0.022	0.037	59.34	0.021	0.035	61.15	0.032	0.053
kitchen	54.61	0.008	0.015	55.40	0.009	0.016	55.42	0.008	0.016	54.88	0.008	0.016	53.99	0.007	0.014	55.17	0.011	0.021
electronics	57.40	0.008	0.014	57.99	0.007	0.012	59.37	0.019	0.032	56.97	0.005	0.009	56.84	0.005	0.009	58.00	0.008	0.014

Table 6: Robustness Metrics by Design Bagging in sentiment classification. Domain level accuracy scores when a single test set is left out. *Var* is variance of accuracy scores and *COR* is the coefficient of variance.

model to generate contextualized adversarial examples. Niu et al. (2020) evaluates robustness to input perturbations for neural machine translation. While many NLP models achieve better performance af-

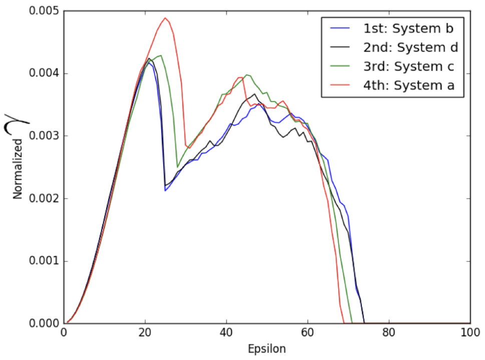

Figure 5: WMT'18 submission systems, human evaluation, (ϵ, γ)-Robustness measured on human evaluation scores.

Model	BLEU	rank_{BLEU}	rank_γ	rank_{Human}
System A	0.379	4	4	4
System B	0.322	**2**	**1**	**1**
System C	0.362	3	3	3
System D	0.320	**1**	**2**	**2**

Table 7: (ϵ, γ)-Robustness 100% agrees with human ranking, while BLEU 75% agrees with human.

Model 1	Model 2	CV	γ	Human	Agree?
WMT	BIO	WMT	WMT	WMT	YES
BIO	IWSLT	BIO	BIO	IWSLT	NO
WMT	MTNT	WMT	WMT	WMT	YES
WMT	IWSLT	WMT	WMT	WMT	YES
BIO	MTNT	BIO	BIO	BIO	YES
IWSLT	MTNT	IWSLT	IWSLT	IWSLT	YES

Table 8: Robustness Metrics pair-wise comparison on each two models.

ter retraining with adversarial examples, the lack of an attack-agnostic evaluation metric leaves the evaluation of model's intrinsic robustness difficult especially when seeing new adversarial attacks.

Another line of work examines the robustness of NLP models among various domains/distributions. Hendrycks et al. (2020) compared the robustness of pretrained Transformers and found that pretraining models on diverse data helps to improve out-of-distribution generalization. Müller et al. (2019) tests several techniques such as subword regularization, defensive model distillation to improve generalization of machine translation models. These work detect the model performance drop under domain shifts but did not give notion of robustness as consistent performance among domains or distributions. In contrast, we propose three robustness metrics that are easy to measure quantitatively using bagging or design bagging.

Some literature propose evaluation metrics for robustness from the perspectives of statistics or input perturbations (Weng et al., 2018; Niu et al., 2020; Mangal et al., 2019), however, they either focus on the worst-scenario of adversarial inputs or disregard the full distribution of performance scores.

7 Conclusion

We introduce variance, coefficient of variation and (ϵ, γ)-Robustness to measure the robustness of an NLP model's typical behavior. Our robustness metrics outperform BLEU in MT system performance rankings and highly agree with human robustness assessment. Our work demonstrates a successful step towards general robustness evaluation and optimization goals.

Acknowledgments

We appreciate the National Science Foundation (NSF) Award No. 1747728 and NSF CRAFT Award, Grant No. 22001 to fund this research. We are also thankful for the support of the Google Cloud Research Program. We especially thank JSALT'19 for providing the platform with top research and collaborative opportunities to inspire and initiate this work. We sincerely thank Jackie C.K. Cheung, Paul Mitchel, Graham Neubig, and Jacob Eisenstein for the initial discussion of this work.

References

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*.

Alexandre Araujo, Laurent Meunier, Rafael Pinot, and Benjamin Negrevergne. 2019. Robust neural networks using randomized adversarial training. *arXiv preprint arXiv:1903.10219*.

Osbert Bastani, Yani Ioannou, Leonidas Lampropoulos, Dimitrios Vytiniotis, Aditya Nori, and Antonio Criminisi. 2016. Measuring neural net robustness with constraints. *Advances in neural information processing systems*, 29:2613–2621.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pages 440–447.

Leo Breiman. 1996. Bagging predictors. *Machine learning*.

Nicholas Carlini and David Wagner. 2017. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. *CoRR*, abs/1705.03122.

Matthias Hein and Maksym Andriushchenko. 2017. Formal guarantees on the robustness of a classifier against adversarial manipulation. *arXiv preprint arXiv:1705.08475*.

Christina Heinze-Deml and Nicolai Meinshausen. 2017. Conditional variance penalties and domain shift robustness. *arXiv preprint arXiv:1710.11469*.

Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. 2020. Pretrained transformers improve out-of-distribution robustness. *arXiv preprint arXiv:2004.06100*.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8018–8025.

Dianqi Li, Yizhe Zhang, Hao Peng, Liqun Chen, Chris Brockett, Ming-Ting Sun, and Bill Dolan. 2020. Contextualized perturbation for textual adversarial attack. *arXiv preprint arXiv:2009.07502*.

Ravi Mangal, Aditya V Nori, and Alessandro Orso. 2019. Robustness of neural networks: A probabilistic and practical approach. In *2019 IEEE/ACM 41st International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER)*, pages 93–96. IEEE.

Mathias Müller, Annette Rios, and Rico Sennrich. 2019. Domain robustness in neural machine translation. *arXiv preprint arXiv:1911.03109*.

Xing Niu, Prashant Mathur, Georgiana Dinu, and Yaser Al-Onaizan. 2020. Evaluating robustness to input perturbations for neural machine translation. *arXiv preprint arXiv:2005.00580*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Periklis Papakonstantinou, Jia Xu, and Zhu Cao. 2014. Bagging by design (on the suboptimality of bagging). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28.

Tianlu Wang, Xuezhi Wang, Yao Qin, Ben Packer, Kang Li, Jilin Chen, Alex Beutel, and Ed Chi. 2020. Catgen: Improving robustness in nlp models via controlled adversarial text generation. *arXiv preprint arXiv:2010.02338*.

Tsui-Wei Weng, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, Dong Su, Yupeng Gao, Cho-Jui Hsieh, and Luca Daniel. 2018. Evaluating the robustness of neural networks: An extreme value theory approach. *arXiv preprint arXiv:1801.10578*.

Huan Xu and Shie Mannor. 2012. Robustness and generalization. *Mach. Learn.*, 86(3):391–423.