Detect and Understand Generalization Barriers

Risk Attribution

Guanlin Li

2019-06-26

Outline

Outline

Aim of The Talk

- Bring your notice on principled generalization study in real applications, e.g. machine translation.
- Connect several dots that can shed light on the so-called principled generalization study:
  - Out-of-Distribution (OOD) detection
    - Generative Modelling [NMT+18, NMTL19, CJ18]
    - Discriminative Modelling [HG17, LLS18, LLLS18]
  - Applications
    - Neural Machine Translation (NMT) [ZZH+18, ZZZ+19]
    - Semantic Parsing [DQL18]
  - Data valuation and Knowledge Tracing [KL17, GZ19, KATL19]

Outline

The Mystery of Generalization

- Generalization is the property of the learned model that can *perform well* on previously unseen instances.
- **How to define *perform well*?**

The Mystery of Generalization (cont.)

Definition (Empirical Performance)

Given a evaluatiom metric f_m, a learned model $\mathcal{M}_\theta$, and a test set $D_{test} = \{\mathrm{x}^j, \mathrm{y}^j\}_{j=1}^{M}$, the empirical performance is measured by $f_m(D_{test}, \mathcal{M}_\theta(D_{test})) \in \mathbb{R}$.

- $\mathcal{M}_\theta(D_{test})$ is the set of target predictions of the model on the test source sentences.
- In practice f_m should be in accordance with human evaluation especially in structured domain, e.g. BLEU, ROUGE.

The Mystery of Generalization (cont.)

Definition (Real Performance)

Given a evaluatiom metric f_m, a learned model $\mathcal{M}_\theta$, and a real test distribution $P_{te}(X, Y)$, the real performance is measured by $f_m(P_{te}, \mathcal{M}_\theta(P_{te})) \in \mathbb{R}$.

The Mystery of Generalization (cont.)

Definition (Mean Performance)

The the empirical performance and the real performance are called the mean performance, since their computation requires an average operation over corpus-level statistics towards instance-wise statistics. [1]

- **pros**: very concise to represent overall performance; as an indicator to measure mean improvement.
- **cons**: small gain of mean performance cannot reflect significat performance gain w.r.t. human evaluation; very rough to describe the performance distribution over $P_{te}(X, Y)$ (mode, quntile); hard to understand what aspect of the $P_{te}(X)$ does the model improve.

[1] E.g. BLEU take a form approximately as a weighted arithmic average of n-gram.

A Category of Realistic Generalization [QCSSL09, MTRAR+12]

$D_{train} = \{\mathrm{x}^i, \mathrm{y}^i\}_{i=1}^N \sim P_{tr}(X,Y)$ **and** $D_{test} = \{\mathrm{x}^j, \mathrm{y}^j\}_{j=1}^M \sim P_{te}(X,Y)$

1. $P_{tr}(X,Y) = P_{te}(X,Y)$

Supervised Learning

2. $P_{tr}(X,Y) \neq P_{te}(X,Y)$

Causal Model: $X \to Y$

$P_\Diamond(X,Y) = P_\Diamond(X)P_\Diamond(Y|X)$

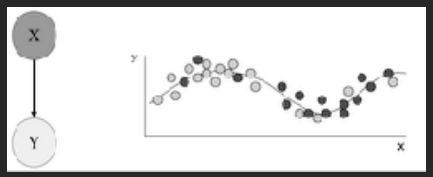
Causal Model: $Y \to X$

$P_\Diamond(X,Y) = P_\Diamond(Y)P_\Diamond(X|Y)$

2.1.1 $P_{tr}(X) \neq P_{te}(X)$

$P_{tr}(Y|X) = P_{te}(Y|X)$

Covariate Shift



2.1.2 $P_{tr}(Y) \neq P_{te}(Y)$

$P_{tr}(X|Y) = P_{te}(X|Y)$

Prior Probability Shift or Label/Target Shift

2.2.1 $P_{tr}(X) = P_{te}(X)$

$P_{tr}(Y|X) \neq P_{te}(Y|X)$

2.2.2 $P_{tr}(Y) = P_{te}(Y)$

$P_{tr}(X|Y) \neq P_{te}(X|Y)$

Concept Shift or Conditional Shift

2.3.1 $P_{tr}(X) \neq P_{te}(X)$

$P_{tr}(Y|X) \neq P_{te}(Y|X)$

2.3.2 $P_{tr}(Y) \neq P_{te}(Y)$

$P_{tr}(X|Y) \neq P_{te}(X|Y)$

NMT is a mixture of all these situations

- Say in strict supervised learning setting, if we use unigram language model as our $P_{\Diamond}(X)$, we can have a strict condition that $\forall v \in \mathcal{V}$, we have $freq_{tr}(v) = freq_{te}(v)$, where $\mathcal{V}$ is the vocabulary, e.g. the phrase 'nuclear weapon' occupy 1/10000 of bot train and test tokens.
- In covariate shift setting (Zh⇒En) we may have the freq. ratio of 'bank' in contexts 'commercial bank' and 'fishing on river bank' equals 5:1, but 1:3 in test, so if disambiguation ability is not strong, an error might occur.
- In target shift setting (En⇒Zh), we may have the ratio of 'get rid of' and 'prevent' 5:1 in train but 1:3 in test, so when we encounter a source word '阻止', it is more likely to be translated into 'get rid of' instead of 'prevent' which mismatches test freq.

NMT is a mixture of all these situations (cont.)

- Even harder, we would not encounter 'bank - 河岸' during training but we can have similar freq. of 'bank' between train and test, since training set possesses 'bank - 银行'. However, test set contains 'bank - 河岸'. This is a concept shift phenomenon, when the model will probably fail without resorting to domain adaptation techniques.
- In realistic domain adaptation setting in NMT, two domains may share many functional words but few content words as vocabulary, so unsupervised domain adaptation techniques are very hard to get reliable performance without diction alignment.

How to overcome the cons?

To me, a few research directions are working on overcoming the previously mentioned cons:
- **Adversarials**: Understand how and why individual instance x breaks the model without perceptual difference [TSE+18, IST+19, IST+19].
- **Out-of-Distribution Detection**: Detect a test instance is not from the input distribution $P_{tr}(X)$ of the training instances.
- **Interpreting model behavior**
|– **Input Attribution**: Identify causality in the input feature x towards the model prediction $\mathcal{M}_\theta$ [KHA+18, AGM+18, GA19].
|– **Knowledge Tracing**: Trace the training effect on certain model prediction $\mathcal{M}(x)$ back into the train set D_{train} [KL17, ZZW18, JDW+19, GZ19] [KATL19].

As a common property, all the above research topics are focused on **individual instance** or **sub-populations** instead of the average effect of the model.

Benefits from Overcoming the Cons

On overcoming the cons, we can have a 'theory' [2] for the causality of the misbehavior of the model. We can do better:

- Interactive/Active Learning [FM18, WN19]
- Machine Teaching [DHPZ19]
- Dataset Debiasing [BCZ+16, ZWY+17, RSG18, ZWY+18, ZWY+19]
- Fairness-aware Learning [ZWS+13, HPS16, BHN17]

[2] Actually, theory is too strong, but we can have algorithms to accurately detect such causality.

Outline

OOD samples detection: the definition (cont.)

- Given a training set $\{x^i, y^i\}$, we assume the underlying joint generative distribution is $P(X, Y)$. By decomposing the joint distribution into $P(X) \cdot P(Y|X)$, we can have the covariate distribution $P(X)$.
- During test, only a set of covariates is given $\{x^j\}$, we assume its underlying generative distribution is $Q(X)$. The OOD detection problem is defined to judge whether $Q(X) = P(X)$. [3]
- Broadly speaking, OOD samples detection can also be called *Anomaly Detection* or *Outlier Detection*.

[3] As you may realize, when $Q \neq P$ we encounter a covariate shift.

OOD samples detection: the definition (cont.)

- In [HG17], they are interested in two related problems. "The first is **error and success prediction**: can we predict whether a trained classifier will make an error on a particular held-out test example [...]"; "The second is **in- and out-of-distribu -tion detection**: can we predict whether a test example is from a different distribution from the training data [...]"

- ICLR 2017

A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks

Dan Hendrycks*
University of California, Berkeley
hendrycks@berkeley.edu

Kevin Gimpel
Toyota Technological Institute at Chicago
kgimpel@ttic.edu

- ICLR 2018

Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks

Shiyu Liang
Coordinated Science Lab, Department of ECE
University of Illinois at Urbana-Champaign
sliang26@illinois.edu

Yixuan Li
Facebook Research
yixuanl@fb.com

R. Srikant
Coordinated Science Lab, Department of ECE
University of Illinois at Urbana-Champaign
rsrikant@illinois.edu

- NeurIPS 2018

A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks

Kimin Lee[1], **Kibok Lee**[2], **Honglak Lee**[3,2], **Jinwoo Shin**[1,4]
[1]Korea Advanced Institute of Science and Technology (KAIST)
[2]University of Michigan
[3]Google Brain
[4]AItrics

- ICLR 2019

DO DEEP GENERATIVE MODELS KNOW WHAT THEY DON'T KNOW?

Eric Nalisnick[*†], **Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, Balaji Lakshminarayanan**[*]
DeepMind

- arXiv Jun. 2018

WAIC, but Why? Generative Ensembles for Robust Anomaly Detection

Hyunsun Choi [*] **Eric Jang** [*1] **Alexander A. Alemi** [1]

- arXiv Jun. 2019

Detecting Out-of-Distribution Inputs to Deep Generative Models Using a Test for Typicality

Eric Nalisnick[*], **Akihiro Matsukawa**, **Yee Whye Teh**, **Balaji Lakshminarayanan**[*]
DeepMind
{enalisnick, amatsukawa, ywteh, balajiln}@google.com

- arXiv Jun. 2019

Likelihood Ratios for Out-of-Distribution Detection

Jie Ren[*†] Google Research jjren@google.com

Peter J. Liu [‡] Google Research peterjliu@google.com

Emily Fertig[†] Google Research emilyaf@google.com

Jasper Snoek Google Research jsnoek@google.com

Ryan Poplin Google Research rpoplin@google.com

Mark A. DePristo Google Research mdepristo@google.com

Joshua V. Dillon [‡] Google Research jvdillon@google.com

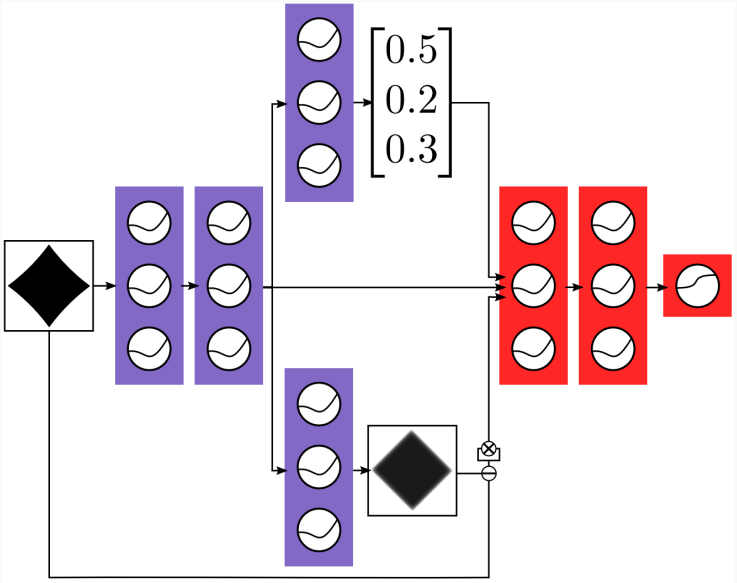
Balaji Lakshminarayanan[*‡] DeepMind balajiln@google.com

Method Categorization

There are two main categories of methods:
- **Discriminative**: since in most of the OOD detetion task, there is a supervised learning task (image classification) with a discrimina -tively trained classifier, by utilizing this classifier we can get discriminative OOD detection detector since has access to the in-distribution label.
- **Generative**: when not resorting to a trained classifier, by only using the inputs of the training set $\{x^i\}$, a density model can be learned and utilized to construct OOD detector.

A Baseline For Detecting Misclassified and Out-of-Distribution Examples in Neural Networks

- This paper propose two methods based on the learned classifier.
|– Maximum prob. of the softmax layer as an indicator of whether the instance is correctly predicted or OOD.
|– A specially trained classifier with features from original classifier prob., internal feature and reconstruced image through the aux. decoder.



Results on image datasets

In-Distribution / Out-of-Distribution	AUROC /Base	AUPR In /Base	AUPR Out/Base	Pred. Prob (mean)
CIFAR-10/SUN	95/50	89/33	97/67	72
CIFAR-10/Gaussian	97/50	98/49	95/51	77
CIFAR-10/All	96/50	88/24	98/76	74
CIFAR-100/SUN	91/50	83/27	96/73	56
CIFAR-100/Gaussian	88/50	92/43	80/57	77
CIFAR-100/All	90/50	81/21	96/79	63
MNIST/Omniglot	96/50	97/52	96/48	86
MNIST/notMNIST	85/50	86/50	88/50	92
MNIST/CIFAR-10bw	95/50	95/50	95/50	87
MNIST/Gaussian	90/50	90/50	91/50	91
MNIST/Uniform	99/50	99/50	98/50	83
MNIST/All	91/50	76/20	98/80	89

Table 2: Distinguishing in- and out-of-distribution test set data for image classification. CIFAR-10/All is the same as CIFAR-10/(SUN, Gaussian). All values are percentages.

Results on image datasets

In-Distribution / Out-of-Distribution	AUROC /Base Softmax	AUROC /Base AbMod	AUPR In/Base Softmax	AUPR In/Base AbMod	AUPR Out/Base Softmax	AUPR Out/Base AbMod
MNIST/Omniglot	95/50	100/50	95/52	100/52	95/48	100/48
MNIST/notMNIST	87/50	100/50	88/50	100/50	90/50	100/50
MNIST/CIFAR-10bw	98/50	100/50	98/50	100/50	98/50	100/50
MNIST/Gaussian	88/50	100/50	88/50	100/50	90/50	100/50
MNIST/Uniform	99/50	100/50	99/50	100/50	99/50	100/50
Average	93	100	94	100	94	100

Table 11: Improved detection using the abnormality module. All values are percentages.

Enhancing the Reliability of OOD Image Detection In Neural Networks

- This work is built on top of the previous work which uses the maximum softmax likelihood as an indicator of OOD, that is, to make the softmax likelihood more reliable:

|– **temperature scaling**: train a t to be used for $\frac{\exp[l_i/t]}{\sum_j \exp[l_j/t]}$, where $l \in \mathbb{R}^c$ is the logits before softmax normalization.

|– **input preprocessing**: $x' = x - \epsilon \text{sign}(-\nabla_x S_{\hat{y}}(x; t))$

Results on image datasets

	Out-of-distribution dataset	FPR (95% TPR) ↓	Detection Error ↓	AUROC ↑	AUPR In ↑	AUPR Out ↑
		Baseline (Hendrycks & Gimpel, 2017) / ODIN				
Dense-BC CIFAR-10	TinyImageNet (crop)	34.7/4.3	19.9/4.7	95.3/99.1	96.4/99.1	93.8/99.1
	TinyImageNet (resize)	40.8/7.5	22.9/6.3	94.1/98.5	95.1/98.6	92.4/98.5
	LSUN (crop)	39.3/8.7	22.2/6.9	94.8/98.2	96.0/98.5	93.1/97.8
	LSUN (resize)	33.6/3.8	19.3/4.4	95.4/99.2	96.4/99.3	94.0/99.2
	iSUN	37.2/6.3	21.1/5.7	94.8/98.8	95.9/98.9	93.1/98.8
	Uniform	23.5/0.0	14.3/2.5	96.5/99.9	97.8/100.0	93.0/99.9
	Gaussian	12.3/0.0	8.7/2.5	97.5/100.0	98.3/100.0	95.9/100.0
Dense-BC CIFAR-100	TinyImageNet (crop)	67.8/17.3	36.4/11.2	83.0/97.1	85.3/97.4	80.8/96.8
	TinyImageNet (resize)	82.2/44.3	43.6/24.6	70.4/90.7	71.4/91.4	68.6/90.1
	LSUN (crop)	69.4/17.6	37.2/11.3	83.7/96.8	86.2/97.1	80.9/96.5
	LSUN (resize)	83.3/44.0	44.1/24.5	70.6/91.5	72.5/92.4	68.0/90.6
	iSUN	84.8/49.5	44.7/27.2	69.9/90.1	71.9/91.1	67.0/88.9
	Uniform	88.3/0.5	46.6/2.8	83.2/99.5	88.1/99.6	73.1/99.0
	Gaussian	95.4/0.2	50.2/2.6	81.8/99.6	87.6/99.7	70.1/99.1
WRN-28-10 CIFAR-10	TinyImageNet (crop)	38.9/23.4	21.9/14.2	92.9/94.2	92.5/92.8	91.9/94.7
	TinyImageNet (resize)	45.6/25.5	25.3/15.2	91.0/92.1	89.7/89.0	89.9/93.6
	LSUN (crop)	35.0/21.8	20.0/13.4	94.5/95.9	95.1/95.8	93.1/95.5
	LSUN (resize)	35.0/17.6	20.0/11.3	93.9/95.4	93.8/93.8	92.8/96.1
	iSUN	40.6/21.3	22.8/13.2	92.5/93.7	91.7/91.2	91.5/94.9
	Uniform	1.6/0.0	3.3/2.5	99.2/100.0	99.3/100.0	98.9/100.0
	Gaussian	0.3/0.0	2.6/2.5	99.5/100.0	99.6/100.0	99.3/100.0
WRN-28-10 CIFAR-100	TinyImageNet (crop)	66.6/43.9	35.8/24.4	82.0/90.8	83.3/91.4	80.2/90.0
	TinyImageNet (resize)	79.2/55.9	42.1/30.4	72.2/84.0	70.4/82.8	70.8/84.4
	LSUN (crop)	74.0/39.6	39.5/22.3	80.3/92.0	83.4/92.4	77.0/91.6
	LSUN (resize)	82.2/56.5	43.6/30.8	73.9/86.0	75.7/86.2	70.1/84.9
	iSUN	82.7/57.3	43.9/31.1	72.8/85.6	74.2/85.9	69.2/84.8
	Uniform	98.2/0.1	51.6/2.5	84.1/99.1	89.9/99.4	71.0/97.5
	Gaussian	99.2/1.0	52.1/3.0	84.3/98.5	90.2/99.1	70.9/95.9

A Simple Unified Framework for Detecting OOD Samples and Adversarial Attacks

Algorithm 1 Computing the Mahalanobis distance-based confidence score.

Input: Test sample $\mathbf{x}$, weights of logistic regression detector α_ℓ, noise ε and parameters of Gaussian distributions $\{\widehat{\mu}_{\ell,c}, \widehat{\mathbf{\Sigma}}_\ell : \forall \ell, c\}$

Initialize score vectors: $\mathbf{M}(\mathbf{x}) = [M_\ell : \forall \ell]$
for each layer $\ell \in 1, \ldots, L$ **do**
  Find the closest class: $\widehat{c} = \arg\min_c \ (f_\ell(\mathbf{x}) - \widehat{\mu}_{\ell,c})^\top \widehat{\mathbf{\Sigma}}_\ell^{-1} (f_\ell(\mathbf{x}) - \widehat{\mu}_{\ell,c})$
  Add small noise to test sample: $\widehat{\mathbf{x}} = \mathbf{x} - \varepsilon \text{sign}\left(\bigtriangledown_{\mathbf{x}} \left(f_\ell(\mathbf{x}) - \widehat{\mu}_{\ell,\widehat{c}}\right)^\top \widehat{\mathbf{\Sigma}}_\ell^{-1} \left(f_\ell(\mathbf{x}) - \widehat{\mu}_{\ell,\widehat{c}}\right)\right)$
  Computing confidence score: $M_\ell = \max_c - \left(f_\ell(\widehat{\mathbf{x}}) - \widehat{\mu}_{\ell,c}\right)^\top \widehat{\mathbf{\Sigma}}_\ell^{-1} \left(f_\ell(\widehat{\mathbf{x}}) - \widehat{\mu}_{\ell,c}\right)$
end for
return Confidence score for test sample $\sum_\ell \alpha_\ell M_\ell$

Results on image datasets

In-dist (model)	OOD	Validation on OOD samples			Validation on adversarial samples		
		TNR at TPR 95%	AUROC	Detection acc.	TNR at TPR 95%	AUROC	Detection acc.
		Baseline [13] / ODIN [21] / Mahalanobis (ours)			Baseline [13] / ODIN [21] / Mahalanobis (ours)		
CIFAR-10 (DenseNet)	SVHN	40.2 / 86.2 / **90.8**	89.9 / 95.5 / **98.1**	83.2 / 91.4 / **93.9**	40.2 / 70.5 / **89.6**	89.9 / 92.8 / **97.6**	83.2 / 86.5 / **92.6**
	TinyImageNet	58.9 / 92.4 / **95.0**	94.1 / 98.5 / **98.8**	88.5 / 93.9 / **95.0**	58.9 / 87.1 / **94.9**	94.1 / 97.2 / **98.8**	88.5 / 92.1 / **95.0**
	LSUN	66.6 / 96.2 / **97.2**	95.4 / 99.2 / **99.3**	90.3 / 95.7 / **96.3**	66.6 / 92.9 / **97.2**	95.4 / 98.5 / **99.2**	90.3 / 94.3 / **96.2**
CIFAR-100 (DenseNet)	SVHN	26.7 / 70.6 / **82.5**	82.7 / 93.8 / **97.2**	75.6 / 86.6 / **91.5**	26.7 / 39.8 / **62.2**	82.7 / 88.2 / **91.8**	75.6 / 80.7 / **84.6**
	TinyImageNet	17.6 / 42.6 / **86.6**	71.7 / 85.2 / **97.4**	65.7 / 77.0 / **92.2**	17.6 / 43.2 / **87.2**	71.7 / 85.3 / **97.0**	65.7 / 77.2 / **91.8**
	LSUN	16.7 / 41.2 / **91.4**	70.8 / 85.5 / **98.0**	64.9 / 77.1 / **93.9**	16.7 / 42.1 / **91.4**	70.8 / 85.7 / **97.9**	64.9 / 77.3 / **93.8**
SVHN (DenseNet)	CIFAR-10	69.3 / 71.7 / **96.8**	91.9 / 91.4 / **98.9**	86.6 / 85.8 / **95.9**	69.3 / 69.3 / **97.5**	91.9 / 91.9 / **98.8**	86.6 / 86.6 / **96.3**
	TinyImageNet	79.8 / 84.1 / **99.9**	94.8 / 95.1 / **99.9**	90.2 / 90.4 / **98.9**	79.8 / 79.8 / **99.9**	94.8 / 94.8 / **99.8**	90.2 / 90.2 / **98.9**
	LSUN	77.1 / 81.1 / **100**	94.1 / 94.5 / **99.9**	89.1 / 89.2 / **99.3**	77.1 / 77.1 / **100**	94.1 / 94.1 / **99.9**	89.1 / 89.1 / **99.2**
CIFAR-10 (ResNet)	SVHN	32.5 / 86.6 / **96.4**	89.9 / 96.7 / **99.1**	85.1 / 91.1 / **95.8**	32.5 / 40.3 / **75.8**	89.9 / 86.5 / **95.5**	85.1 / 77.8 / **89.1**
	TinyImageNet	44.7 / 72.5 / **97.1**	91.0 / 94.0 / **99.5**	85.1 / 86.5 / **96.3**	44.7 / 69.6 / **95.5**	91.0 / 93.9 / **99.0**	85.1 / 86.0 / **95.4**
	LSUN	45.4 / 73.8 / **98.9**	91.0 / 94.1 / **99.7**	85.3 / 86.7 / **97.7**	45.4 / 70.0 / **98.1**	91.0 / 93.7 / **99.5**	85.3 / 85.8 / **97.2**
CIFAR-100 (ResNet)	SVHN	20.3 / 62.7 / **91.9**	79.5 / 93.9 / **98.4**	73.2 / 88.0 / **93.7**	20.3 / 12.2 / **41.9**	79.5 / 72.0 / **84.4**	73.2 / 67.7 / **76.5**
	TinyImageNet	20.4 / 49.2 / **90.9**	77.2 / 87.6 / **98.2**	70.8 / 80.1 / **93.3**	20.4 / 33.5 / **70.3**	77.2 / 83.6 / **87.9**	70.8 / 75.9 / **84.6**
	LSUN	18.8 / 45.6 / **90.9**	75.8 / 85.6 / **98.2**	69.9 / 78.3 / **93.5**	18.8 / 31.6 / **56.6**	75.8 / 81.9 / **82.3**	69.9 / 74.6 / **79.7**
SVHN (ResNet)	CIFAR-10	78.3 / 79.8 / **98.4**	92.9 / 92.1 / **99.3**	90.0 / 89.4 / **96.9**	78.3 / 79.8 / **94.1**	92.9 / 92.1 / **97.6**	90.0 / 89.4 / **94.6**
	TinyImageNet	79.0 / 82.1 / **99.9**	93.5 / 92.0 / **99.9**	90.4 / 89.4 / **99.1**	79.0 / 80.5 / **99.2**	93.5 / 92.9 / **99.3**	90.4 / 90.1 / **98.8**
	LSUN	74.3 / 77.3 / **99.9**	91.6 / 89.4 / **99.9**	89.0 / 87.2 / **99.5**	74.3 / 76.3 / **99.9**	91.6 / 90.7 / **99.9**	89.0 / 88.2 / **99.5**

Table 2: Distinguishing in- and out-of-distribution test set data for image classification under various validation setups. All values are percentages and the best results are indicated in bold.

Do Deep Generative Models Know What They Don't Know?

- This is an early paper to investigate the question: can (deep) generative model detect OOD samples?
- The experiments empirically reveals a astonishing fact that: "the density learned by flow-based models, VAEs, PixelCNNs cannot distinguish images of common objects such as dogs, trucks, and horses (i.e. CIFAR-10) from those of house numbers (i.e. SVHN), assigning a higher likelihood to the latter when the model is trained on the former."

The Basic Idea

- Generative model $P(\mathrm{x};\theta)$ is trained on $\{\mathrm{x}^i\}_{i=1}^N$ in training data, through maximum likelihood estiamtion (MLE).
- After training, the parametric model represent the empiricl density of the underlying $P(X)$.
- However, we would like to obtain the probability of a neighborhood region $\Omega : \int_{\mathrm{x}\in\Omega} p(\mathrm{x};\theta)$. - "Adding noise to the input can smooth the density to make it near to real probability", so they train the generative model through:

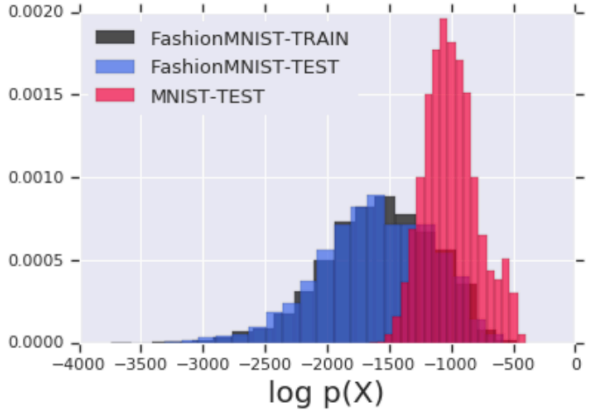
$$\log \int p(\mathrm{x}^i+\delta;\theta)p(\delta)d\delta \leq \mathbb{E}_\delta \log p(\mathrm{x}+\delta;\theta) \approx \log p(\mathrm{x}+\delta;\theta)$$

Strange Phenomenon

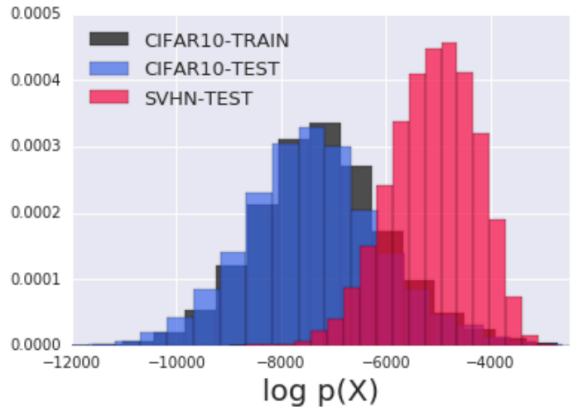
Data Set	Avg. Bits Per Dimension	Data Set	Avg. Bits Per Dimension
Glow Trained on FashionMNIST		*Glow Trained on CIFAR-10*	
FashionMNIST-Train	2.902	CIFAR10-Train	3.386
FashionMNIST-Test	2.958	CIFAR10-Test	3.464
MNIST-Test	**1.833**	SVHN-Test	**2.389**
Glow Trained on MNIST		*Glow Trained on SVHN*	
MNIST-Test	1.262	SVHN-Test	2.057

Figure 1: *Testing Out-of-Distribution.* Log-likelihood (expressed in bits per dimension) calculated from Glow (Kingma & Dhariwal, 2018) on MNIST, FashionMNIST, SVHN, CIFAR-10.

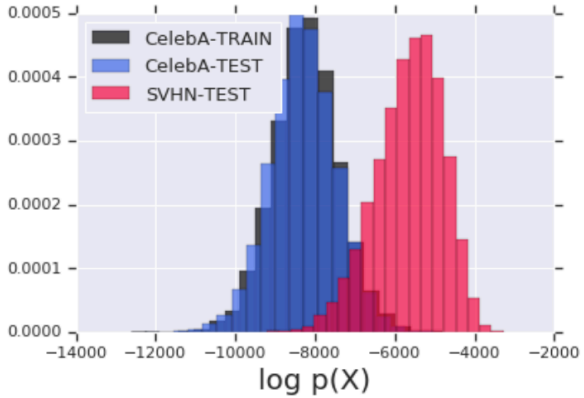
Distributional Statistics



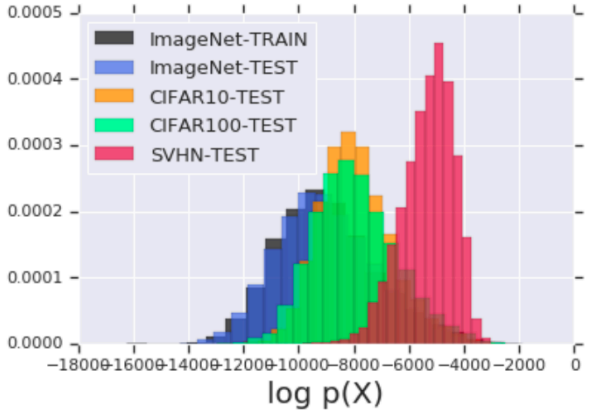
(a) Train on FashionMNIST, Test on MNIST



(b) Train on CIFAR-10, Test on SVHN

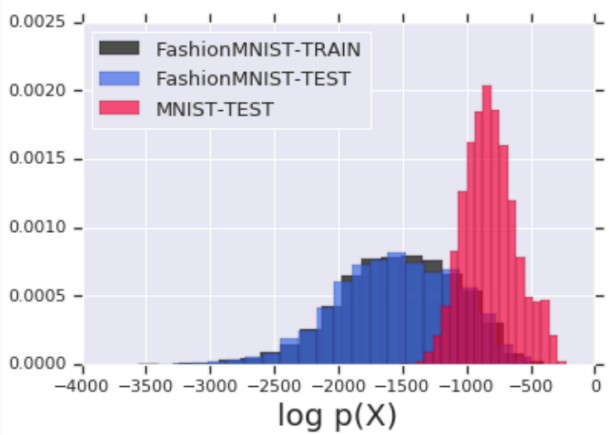


(c) Train on CelebA, Test on SVHN

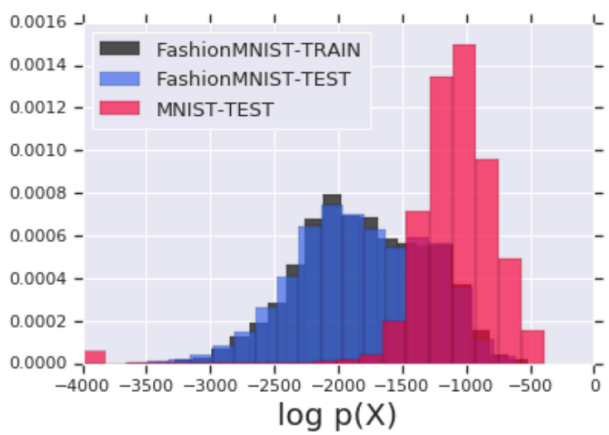


(d) Train on ImageNet,
Test on CIFAR-10 / CIFAR-100 / SVHN

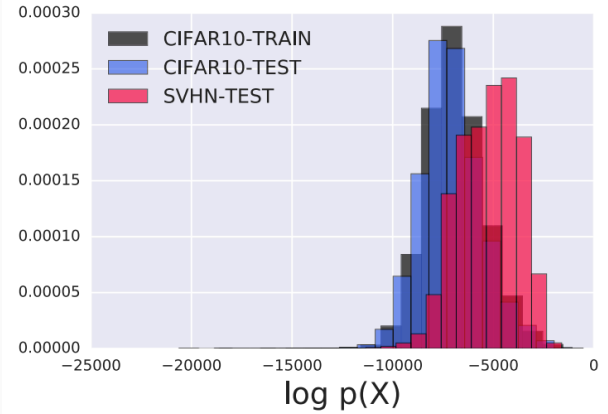
Distributional Statistics



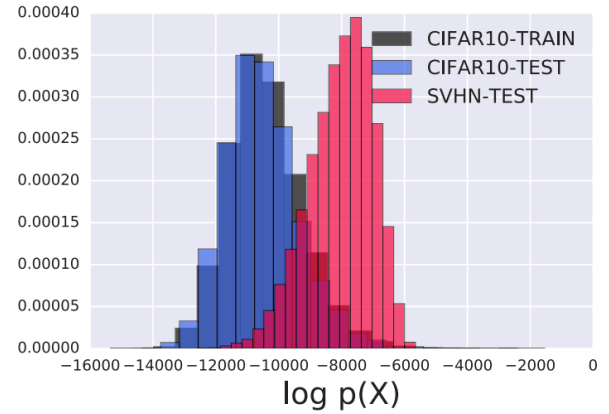
(a) **PixelCNN**: FashionMNIST vs MNIST



(b) **VAE**: FashionMNIST vs MNIST



(c) **PixelCNN**: CIFAR-10 vs SVHN



(d) **VAE**: CIFAR-10 vs SVHN

WAIC: Generative Ensembles for Robust Anomaly Detection

- This work finds similar phenomenon as in the previous work independently and propose to use an ensemble of deep generative models for alleviation.
- The WAIC (Wantanabe-Akaike Information Criterion) is defined as: $\mathbb{E}_\theta \log p_\theta(\mathrm{x}) - Var_\theta \log p_\theta(\mathrm{x})$.
- The $\theta \sim p(\theta|D_{train})$ is the Bayesian posterior given the train set, and can be randomly approximated by different SGD runs.
- They adopt VAEs, IWAEs and Glow as the base generative model; they also find train a ensemble of GANs and use the discriminator ensemble can largely improve OOD detection success.

Main Results

OoD	ODIN	VIB	RATE	$p_\theta(\mathrm{x})$	WAIC
MNIST	VAE				
OMNIGLOT	**100**	97.1	99.1	97.9	98.5
NOTMNIST	98.2	98.6	99.9	**100**	**100**
FASHIONMNIST	N/A	85.0	**100**	**100**	**100**
UNIFORM	**100**	76.6	**100**	**100**	**100**
GAUSSIAN	**100**	99.2	**100**	**100**	**100**
HFLIP	N/A	63.7	60.0	84.1	**85.0**
VFLIP	N/A	75.1	61.8	80.0	**81.3**
ADV	N/A	N/A	**100**	0	**100**
FASHIONMNIST	VAE				
OMNIGLOT	N/A	**94.3**	83.2	56.8	79.6
NOTMNIST	N/A	89.6	92.8	92.0	**98.7**
MNIST	N/A	**94.1**	87.1	42.3	76.6
UNIFORM	N/A	79.6	99.0	**100**	**100**
GAUSSIAN	N/A	89.3	**100**	**100**	**100**
HFLIP	N/A	**66.7**	53.4	59.4	62.4
VFLIP	N/A	**90.2**	58.6	66.8	74.0
ADV	N/A	N/A	**100**	0.1	**100**
OoD	ODIN	VIB	$\|\|d\|\|$	$p_\theta(\mathrm{x})$	WAIC
CIFAR-10	GLOW				
CELEBA	85.7	73.5	22.9	75.6	**99.7**
SVHN	89.9	52.8	74.4	7.5	**100**
IMAGENET32	98.5	70.1	12.3	93.8	**95.6**
UNIFORM	99.9	54.0	**100**	**100**	**100**
GAUSSIAN	**100**	45.8	**100**	**100**	**100**
HFLIP	50.1	**50.6**	46.2	50.1	50.0
VFLIP	84.2	**51.2**	44.0	50.6	50.4

GAN-based Main Results

OoD	D	Var(D)
MNIST		
Omniglot	52.7	81.2
notMNIST	92.7	99.8
Fashion MNIST	81.5	**100**
Uniform	93.9	**100**
Gaussian	0.8	**100**
HFlip	44.5	60.0
VFlip	46.1	61.8
Adv	0.7	**100**
FashionMNIST		
Omniglot	19.1	83.2
notMNIST	17.9	92.8
MNIST	45.1	87.1
Uniform	0	**99.0**
Gaussian	0	**100**
HFlip	54.7	53.4
VFlip	67.2	58.6
Adv	0	**100**
CIFAR-10		
CelebA	57.2	**74.4**
SVHN	**68.3**	62.3
ImageNet32	49.1	**62.6**
Uniform	**100**	**100**
Gaussian	**100**	**100**
HFlip	**51.6**	51.3
VFlip	**59.2**	52.8

Detecting Out-of-Distribution Inputs to Deep Generative Models Using a Test for Typicality

- This work try to understand the phenomenon of their previous work and argue that high likelihood/probability region may not contain the most information of the learned density in an information point of view.
- So they propose to use an information theoretic notion 'typical set' which is defined as to construct a hypothesis test that whether a given x is in the typical set of the learned density.

Likelihood Ratios for Out-of-Distribution Detection

- This work is first work recently that consider detect OOD samples in discrete sequence (genomics).
- They find that the strange phenomenon may be caused by putting to much likelihood on the so-called "background" information.
- So they propose a way to disentangle the background info. by training a generative model with heavily corrupted x and use the overall likelihood to divide the background lielihood to cancel out high prob. background info.
- Likelihood Ratio (LLR): $\text{LLR}(\text{x}) = \log \frac{p_\theta(\text{x})}{p_{\theta_0}(\text{x})}$

Similar Phenomenon on Sequence Data

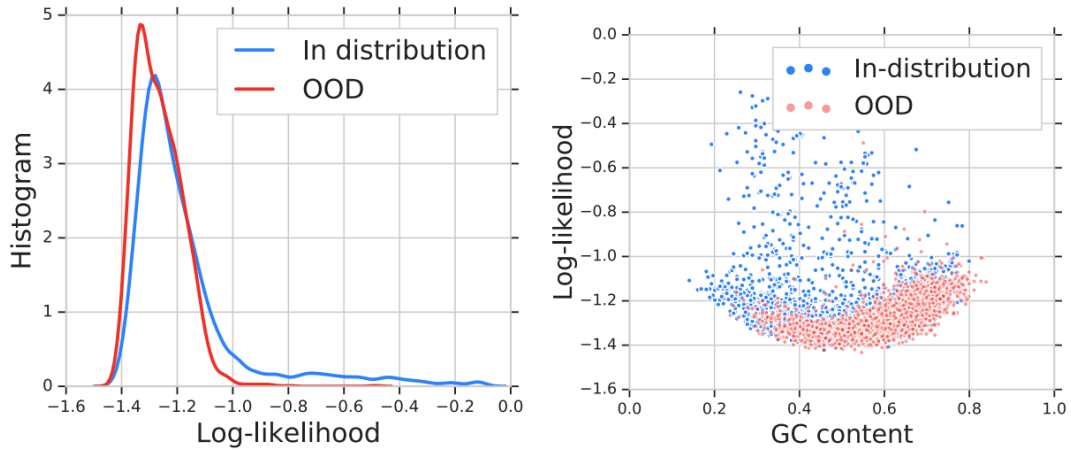


Figure 1: (a) Log-likelihood hardly separates in-distribution and OOD inputs. (b) The log-likelihood is heavily affected by the GC content of a sequence.

Main Results on Image Data

	AUROC↑	AUPRC↑	FPR80↓
Likelihood	0.115	0.324	1.000
Likelihood Ratio (ours, μ)	**0.997**	**0.996**	**0.000**
Likelihood Ratio (ours, μ, λ)	**0.997**	**0.996**	**0.000**
$p(\hat{y}\|\boldsymbol{x})$	0.658	0.617	0.601
Entropy of $p(y\|\boldsymbol{x})$	0.690	0.646	0.525
ODIN	0.697	0.665	0.581
Mahalanobis distance	0.986	0.975	0.022
Ensemble, 5 classifiers	0.832	0.666	0.264
Ensemble, 10 classifiers	0.853	0.671	0.260
Ensemble, 20 classifiers	0.866	0.674	0.239
Binary classifier	0.459	0.464	0.745
$p(\hat{y}\|\boldsymbol{x})$ with noise class	0.874	0.909	0.171
$p(\hat{y}\|\boldsymbol{x})$ with calibrations	0.700	0.642	0.621
WAIC, 5 models	0.371	0.521	0.794

Table 2: Results on Fashion-MNIST vs MNIST

Main Results on Genomics Data

	AUROC↑	AUPRC↑	FPR80↓
Likelihood	0.630	0.613	0.655
Likelihood Ratio (ours, μ)	0.728	0.694	0.534
Likelihood Ratio (ours, μ, λ)	**0.755**	**0.726**	**0.488**
$p(\hat{y}\|\boldsymbol{x})$	0.640	0.606	0.658
Entropy of $p(y\|\boldsymbol{x})$	0.640	0.603	0.601
Adjusted ODIN	0.664	0.645	0.614
Mahalanobis distance	0.496	0.014	0.805
Ensemble, 5 classifiers	0.673	0.634	0.621
Ensemble, 10 classifiers	0.691	0.657	0.560
Ensemble, 20 classifiers	0.697	0.663	0.563
Binary classifier	0.609	0.604	0.674
$p(\hat{y}\|\boldsymbol{x})$ with noise class	0.640	0.611	0.676
$p(\hat{y}\|\boldsymbol{x})$ with calibration	0.688	0.637	0.610
WAIC, 5 models	0.626	0.608	0.670

Table 3: Results on the genomic dataset

Catch up

- OOD sample detection is a revival area of machine learning in this deep learning era, and it seems to have been solved by recent works in image classification domain.
- It greatly relates to many sub-areas which are tightly correlated mutually:

  - **uncertainty (confidence) estimation** [LPB17]: to derive more trustful model uncertainty for using in high risk scenarios.
  - **model calibration** [GPSW17, NDZ+19]: to revise the model to approximate the perfect calibration $\mathbb{P}(\hat{Y} = Y^* | \hat{P} = p) = p$ where $p = P(y|x;\theta)$. [4]
  - **adversarials detection** [CW17, FCSG17, MLW+18, RKH19, YCH+19]: judge whether a given input x is an adversrail towards a given classifier $P(\mathrm{y}|\mathrm{x};\theta)$.

[4] predictive probability $\hat{P}$ estimates are representative of the true correctness likelihood.

Outline

Applications

In this section, we would like to introduce some applications in NLP that take instance-wise generalization (error) analysis seriously.

- EMNLP 2018

Addressing Troublesome Words in Neural Machine Translation

Yang Zhao[1,2], **Jiajun Zhang**[1,2], **Zhongjun He**[4], **Chengqing Zong**[1,2,3], **and Hua Wu**[4]
[1]National Laboratory of Pattern Recognition, Institute of Automation, CAS, Beijing, China
[2]University of Chinese Academy of Sciences, Beijing, China
[3]CAS Center for Excellence in Brain Science and Intelligence Technology, Beijing, China
[4]Baidu Inc., Beijing, China
{yang.zhao, jjzhang, cqzong}@nlpr.ia.ac.cn, {hezhongjun,wu_hua} @baidu.com

- AAAI 2019

Addressing the Under-translation Problem from the Entropy Perspective

Yang Zhao[1,2*], **Jiajun Zhang**[1,2], **Chengqing Zong**[1,2,3], **Zhongjun He**[4], **and Hua Wu**[4]
[1]National Laboratory of Pattern Recognition, Institute of Automation, CAS, Beijing, China
[2]University of Chinese Academy of Sciences, Beijing, China
[3]CAS Center for Excellence in Brain Science and Intelligence Technology, Beijing, China
[4]Baidu Inc., Beijing, China

- EMNLP 2018

Confidence Modeling for Neural Semantic Parsing

Li Dong[†*] and **Chris Quirk**[‡] and **Mirella Lapata**[†]
[†] School of Informatics, University of Edinburgh
[‡] Microsoft Research, Redmond

Neural Machine Translation Related

In recent NMT literature, two works from Jiajun's group are very relevant. Their overall research procedure (paper structure) is:
- Identify and **define** unresolved translation phenomena for NMT models;
- Propose some methods to solve them.

Addressing Troublesome Words in NMT

- An demonstrative example: troublesome word and the import -ance of context to filter out incorporated lexicon.

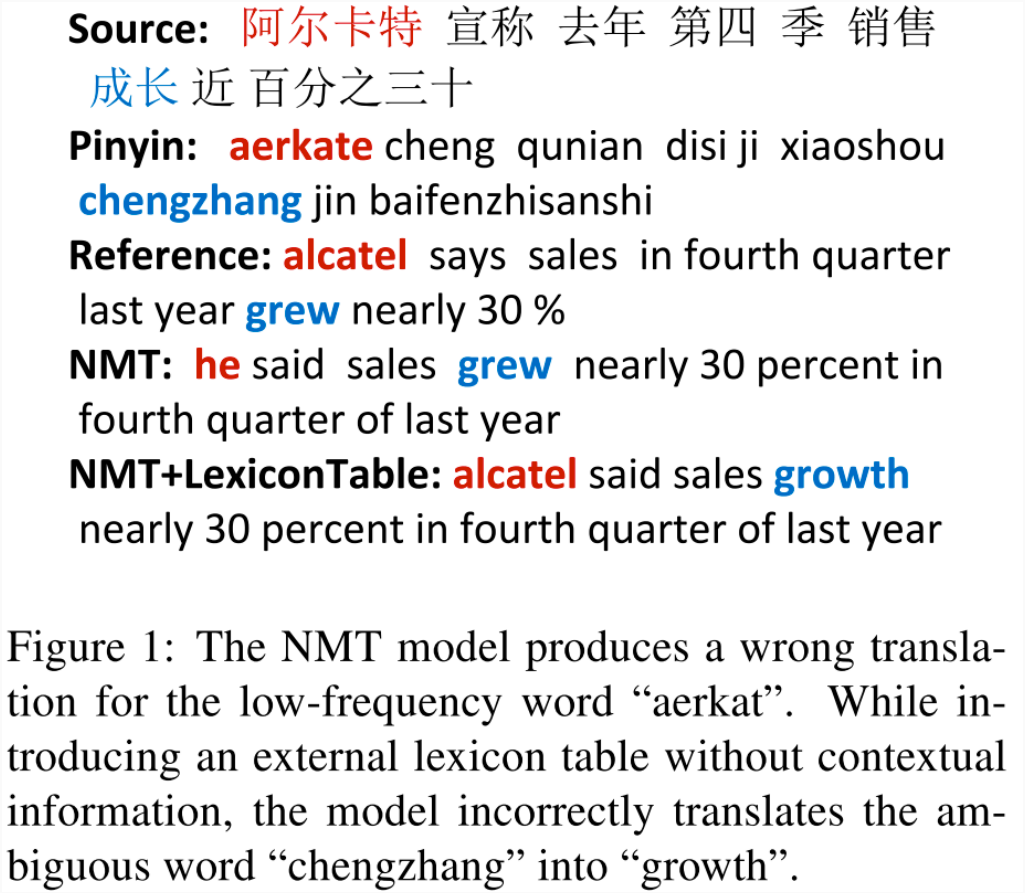


Figure 1: The NMT model produces a wrong translation for the low-frequency word "aerkat". While introducing an external lexicon table without contextual information, the model incorrectly translates the ambiguous word "chengzhang" into "growth".

Addressing Troublesome Words in NMT (cont.)

- **How to define troublesome words?** [5]

|– Given the baseline model $\mathcal{M}_\theta$, for each training instance (x, y), we can get each y_i's probability $p_i^N(\mathrm{y}_i)$ through forced decoding, with N the token freq. $p_i^N(V_T)$ is the distribution over target vocab at the i^{th} step of decoding.

|– **Exception criterions**

||– Absolute: given prob. threshold p_0, when $p_i^N < p_0$

||– Gap: given gap threshold g_0, when $max(p_i^N(V_T)) - p_i^N < g_0$

||– Rank: given rank threshold r_0, when $rank(\mathrm{y}_i) > r_0$

- **Exception rate**

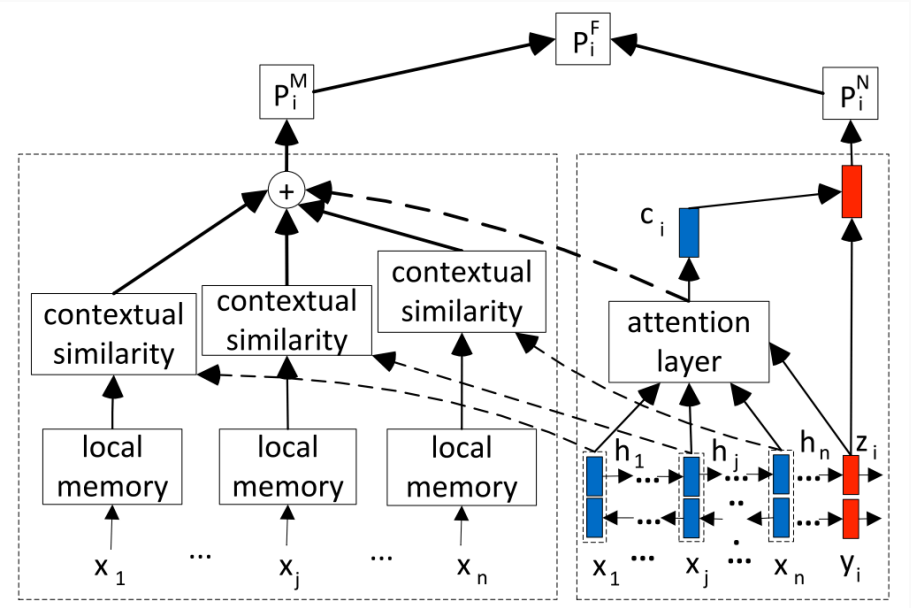
|– Given source-target alignment from *fast_align*, for a aligned word pair $\mathrm{x}_j - \mathrm{y}_i$, its total train corpus count is N, and the in-trouble count based on above criterion is M, so the exception rate of this word pair is M/N.

[5] Note that, all kinds of definitions in this paper rely heavily on *fast_align*, which proves the expertise of the author in SMT.

How to define contexualized memory?

- **Memory** $\mathbb{M}$: $\{s_m, t_n, c(s_m, t_n), P^L(s_m, t_n), r(s_m)\}$
|– $c(s_m, t_n) = \frac{1}{N}\sum_i \text{Enc}(s_m)^i$, where Enc may be a bi-LSTM.
|– $P^L(s_m, t_n)$ is the bi-directional translation prob. of MLE on alignments obtained from *fast_align*.

Contextual Memory Augmented NMT Model



- Stage 1: retrieve according source token match of $x_i \in x$ and $\mathbb{M}$
- Stage 2: context-aware $d_j(t_n) = \sigma(v_d^T tanh(W_h h_j + W_c c(x_j, t_n)))$
- Stage 3: $p_i^M(t_n) = \sum_j^{T_x} \alpha_{i,j} \cdot d_j(t_n) \cdot p^L(x_j, t_n)$ refine lexicon translation prob.
- Stage 4: refine seq2seq prob. $p_{t_n} = \lambda_i p_i^M(t_n) + (1 - \lambda_i) p_i^N(t_n)$ [6]

[6] λ_i is a dynamic interpolation coef. by $\sigma(\beta \cdot \sum_j \alpha_{ij} r(x_j))$, where beta is learned.

Results

#	Model	03	04	05	06	08	Avg.	△
1	Baseline	41.01	42.94	40.31	40.57	30.96	39.16	-
2	Arthur(test)	41.34	43.31	40.79	40.84	31.11	39.48	-
3	Arthur(train+test)	41.88	43.75	41.16	41.63	31.47	39.98	-
4	Baseline(sub-word)	43.93	44.74	42.46	43.01	32.53	41.33	-
5	Baseline+MEM	42.74†	43.94†	42.15†	41.94†	31.86†	40.53	+1.37
6	Arthur(train+test)+MEM	43.04†	44.65*	42.19†	42.59†	32.05*	40.90	+0.92
7	Baseline(sub-word)+MEM	44.98†	45.51†	43.93†	43.95†	33.33†	42.34	+1.01

Results

Type	Baseline	Baseline+MEM
Low+Amb	38.15	39.75
Low	38.56	40.03
Amb	39.86	40.67
Others	40.49	40.76

Table 3: The BLEU score on different kinds of sentences. **Low** denotes low frequency words, **Amb** denotes ambiguous words. **Low+Amb** denotes the low frequency and ambiguous words.

Addressing the Under-translation Problem from the Entropy Perspective

- This paper demonstrate an interesting perspective on under -translation word detection.
- I would only show the under-translation detection methods proposed in this paper, ignoring the solutions that improves performance.

Translation entropy correlates well with under-translation

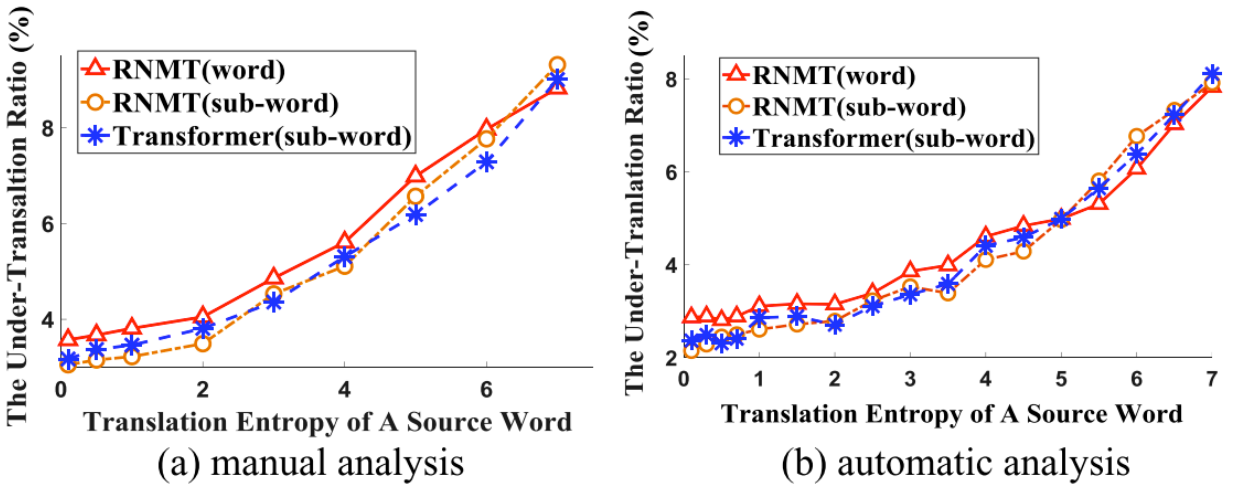


Figure 2: The relationship between the ratio of under-translation (%) (y axis) and translation entropy (x axis), where (a) and (b) report the results of manual analysis and automatic analysis, respectively.

- **Auto. analysis**: decode a set of source using trained model; use *fast_align* to get alignment; see no target aligned source words as under-translated words; compute the translation entorpy of these source words according to alignment on golden parallel corpus;

Confidence Modelling for Neural Semantic Parsing

In [DQL18], they propose a framework to systematically analyze different aspects of the task uncertainty or confidence arising from the data, **model**, **data** and individual **input**. Based on these various sources of uncertainty:
- they show a good correlation of uncertainty features and the instance-wise generalization ability;
- they utilize layer-wise relevance propagation (LRP) [DLLS17] to attribute the model uncertainty back to the source side and make some decent analyses.

Confidence Modelling for Neural Semantic Parsing (cont.)

- Firstly, they wish to estimate the overall confidence score of a pair q, a: $s(q, a) \in (0, 1)$, where q is the input natural language sentence, a the predicted structural parse (logical or other executable format), so that the score correlates will with performance.
- They consider the following sources of uncertainty:
|– **Model uncertainty**: the model's parameter and structure contain uncertainty which are consolidated from dataset noise and stochasity of mini-batch SGD.
|– **Data uncertainty**: "the coverage of training data also affects the uncertainty of predictions"; if training and test distribution do not match, the uncertainty of OOD samples tend to be large.
|– **Input uncertainty**: the input can intrinsically be ambiguous.

Confidence Modelling for Neural Semantic Parsing (cont.)

- **Model uncertainty**

|– **Dropout (Bernoulli) perturbation**: given a pair q, a, they use dropout to add Bernoulli noise to the model connection and gather statitics of the likelihood variance of each $a_t \in a$ under the seman -tic parsing model $p(a_t|a_{<t}, q; \theta)$ as $u_{a_t} = Var\{\hat{p}(a_t|\mathcal{M}^i)\}_{i=1}^{F}$.

|– **Gaussian perturbation**: add Gaussion noise to the correspond -ing logits $v' = v + g$ where $g \sim \mathcal{N}(0, \sigma)$.

|– **Posterior probability**: $\log p(a|q; \theta)$ and $\log p(a_t| \dots)$.

Confidence Modelling for Neural Semantic Parsing (cont.)

- **Data uncertainty**

|– **Perplexity (Probability) of Input**: this measure matches the generative view of OOD detection; a source language model is used to score $p(q|D_{train})$.

|– **Number of UNK toke in** q

Confidence Modelling for Neural Semantic Parsing (cont.)

- **Input uncertainty**: how to accurately measure ambiguity
|– **Variance of Top Candidates**: use beam search to get top-k candidates $\{a^i\}_{i=1}^k$, and then compute $Var(\{p(a^i|q;\theta)\}_{i=1}^k)$.
|– **Entropy of Decoding**: $H[a|q] = \sum_a p(a|q) \cdot \log p(a|q)$.

Confidence Modelling for Neural Semantic Parsing (cont.)

- **Regression for** $s(q, a)$

|– They use a gradient boosting tree model to regress to F1 score of each instance on validation set (prevent train set overfitting), given the bunch of features categorized above.

Experimental results on estimating $s(q, a)$

Method	IFTTT	DJANGO
POSTERIOR	0.477	0.694
CONF	**0.625**	**0.793**
– MODEL	0.595	0.759
– DATA	0.610	0.787
– INPUT	0.608	0.785

Table 2: Spearman ρ correlation between confidence scores and F1. Best results are shown in **bold**. All correlations are significant at $p < 0.01$.

Experimental results on estimating $s(q, a)$ (cont.)

	F1	Dout	Noise	PR	PPL	LM	#UNK	Var
Dout	**0.59**							
Noise	**0.59**	0.90						
PR	**0.52**	0.84	0.82					
PPL	0.48	0.78	0.78	0.89				
LM	0.30	0.26	0.32	0.27	0.25			
#UNK	0.27	0.31	0.33	0.29	0.25	0.32		
Var	0.49	0.83	0.78	0.88	0.79	0.25	0.27	
Ent	0.53	0.78	0.78	0.80	0.75	0.27	0.30	0.76

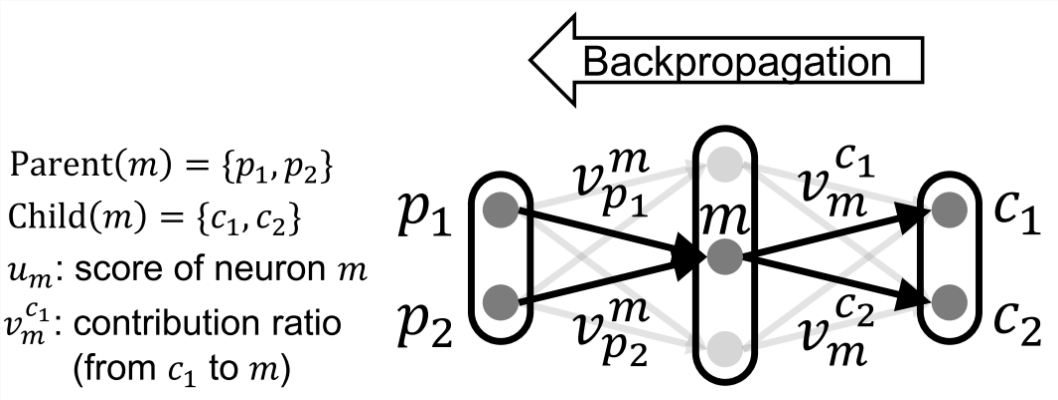
Table 3: Correlation matrix for F1 and individual confidence metrics on the IFTTT dataset. All cor-

Uncertainty Attribution to Input

- Another aim of this work is to identify the cause of such uncertainty in the input: "Confidence scores are useful in so far they can be traced back to the inputs causing the uncertainty in the first place. For semantic parsing, identify which input words contribute to uncertainty would be of value, e.g. these could be treated explicitly as special cases or refined if they represent noise."

Uncertainty propagation

- We can get the token-wise variance of each predicted token a_t, as initialization at the output neurons.
- Then, LRP algorithm is conducted to propagate the target token-wise uncertainty back to the source token's embedding vector.
- You can think of this kind of uncertainty attribution as alignment scores weighted through uncertainty scores; so you can replace LRP to other salience methods, e.g. intergrated gradient, prediction difference etc.



Some visualizations

```
google_calendar-any_event_starts THEN facebook
    -create_a_status_message-(status_message
    ({description}))
```
ATT post calendar event to facebook
BP post calendar event to facebook

```
feed-new_feed_item-(feed_url(
    _url_sports.espn.go.com)) THEN ...
```
ATT espn mlb headline to readability
BP espn mlb headline to readability

```
weather-tomorrow's_low_drops_below-((
    temperature(0)) (degrees_in(c))) THEN ...
```
ATT warn me when it's going to be freezing tomorrow
BP warn me when it's going to be freezing tomorrow

```
if str_number[0] == '_STR_':
```
ATT if first element of str_number equals a string _STR_ .
BP if first element of str_number equals a string _STR_ .

```
start = 0
```
ATT start is an integer 0 .
BP start is an integer 0 .

```
if name.startswith('_STR_'):
```
ATT if name starts with an string _STR_ ,
BP if name starts with an string _STR_ ,

How to Evaluate?

- Proxy oracle: they perturb each source tokeni F times and measure its uncertainty according to the probability variance of a^i with i indexing the each perturbation.
- overlap@K: $\frac{\tau_1 \cap \tau_2}{K}$, with K the ground-truth uncertain source tokens obtained above.

Overlap@K

Method	IFTTT		DJANGO	
	@2	@4	@2	@4
ATTENTION	0.525	0.737	0.637	0.684
BACKPROP	**0.608**	**0.791**	**0.770**	**0.788**

Table 6: Uncertainty interpretation against inferred ground truth; we compute the overlap between tokens identified as contributing to uncertainty by our method and those found in the gold standard. Overlap is shown for top 2 and 4 tokens. Best results are in **bold**.

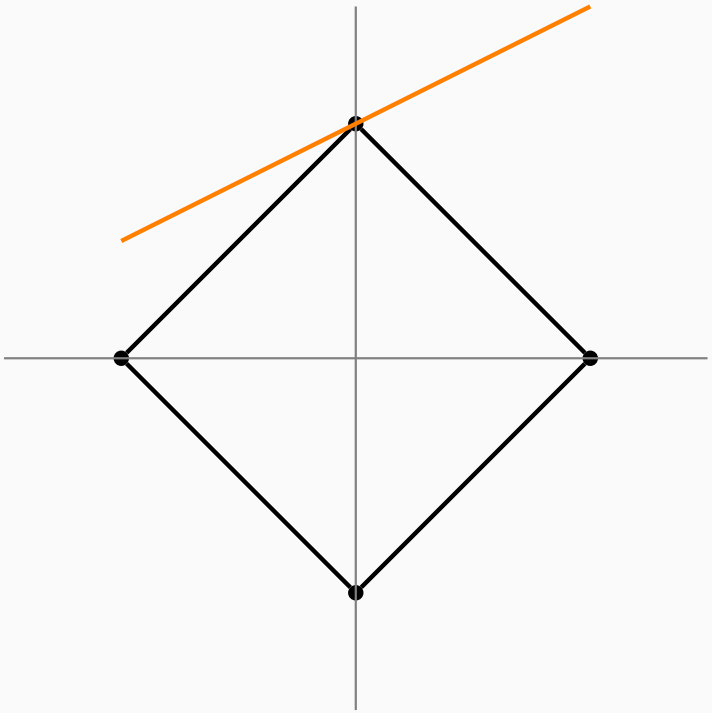
Outline

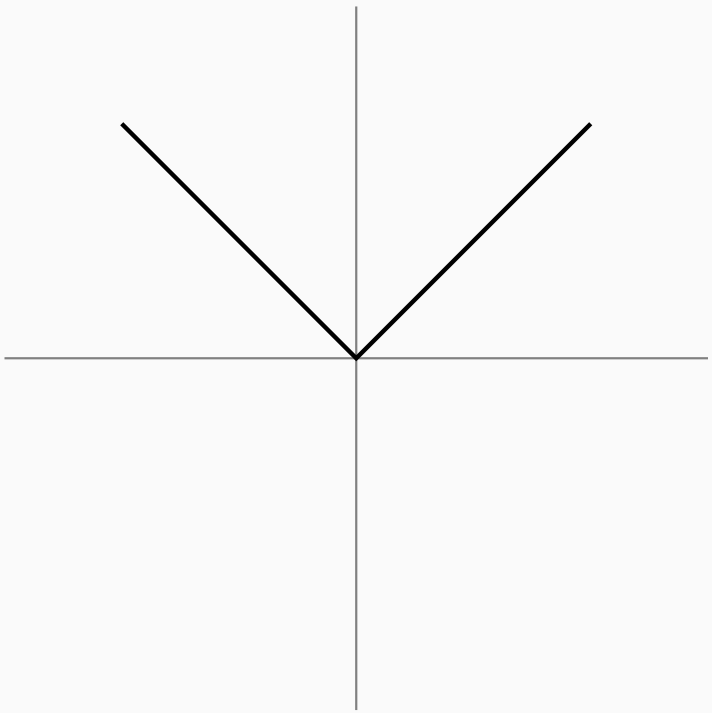
List

- XXX
  - XXX
  - XXX
    - XXX
- XXX [VSP+17]
- XXX

1. XXX
   1.1 XXX
   1.2 XXX
      1.2.1 XXX
2. XXX
3. XXX

Pictures with tikz [Tan08]





References I

[AGM+18] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian J. Goodfellow, Moritz Hardt, and Been Kim, *Sanity checks for saliency maps*, NeurIPS, 2018.

[BCZ+16] Tolga Bolukbasi, Kai-Wei Chang, James Y. Zou, Venkatesh Saligrama, and Adam Tauman Kalai, *Man is to computer programmer as woman is to homemaker? debiasing word embeddings*, NIPS, 2016.

[BHN17] Solon Barocas, Moritz Hardt, and Arvind Narayanan, *Fairness in machine learning*, NIPS Tutorial (2017).

[CJ18] Hyunsun Choi and Eric Jang, *Generative ensembles for robust anomaly detection*, arXiv preprint arXiv:1810.01392 (2018).

References II

[CW17] Nicholas Carlini and David A. Wagner, *Adversarial examples are not easily detected: Bypassing ten detection methods*, AISec@CCS, 2017.

[DHPZ19] Sanjoy Dasgupta, Daniel Hsu, Stefanos Poulis, and Xiaojin Zhu, *Teaching a black-box learner*, Proceedings of the 36th International Conference on Machine Learning (Long Beach, California, USA) (Kamalika Chaudhuri and Ruslan Salakhutdinov, eds.), Proceedings of Machine Learning Research, vol. 97, PMLR, 09–15 Jun 2019, pp. 1547–1555.

[DLLS17] Yanzhuo Ding, Yang Liu, Huanbo Luan, and Maosong Sun, *Visualizing and understanding neural machine translation*, ACL, 2017.

[DQL18] Li Dong, Chris Quirk, and Mirella Lapata, *Confidence modeling for neural semantic parsing*, ACL, 2018.

References III

[FCSG17] Reuben Feinman, Ryan R. Curtin, Saurabh Shintre, and Andrew B. Gardner, *Detecting adversarial samples from artifacts*, CoRR **abs/1703.00410** (2017).

[FM18] Marzieh Fadaee and Christof Monz, *Back-translation sampling by targeting difficult words in neural machine translation*, EMNLP, 2018.

[GA19] Arushi Gupta and Sanjeev Arora, *A simple saliency method that passes the sanity checks*, arXiv preprint arXiv:1905.12152 (2019).

[GPSW17] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger, *On calibration of modern neural networks*, ICML, 2017.

References IV

[GZ19] Amirata Ghorbani and James Zou, *Data shapley: Equitable valuation of data for machine learning*, arXiv preprint arXiv:1904.02868 (2019).

[HG17] Dan Hendrycks and Kevin Gimpel, *A baseline for detecting misclassified and out-of-distribution examples in neural networks*, CoRR **abs/1610.02136** (2017).

[HPS16] Moritz Hardt, Eric Price, and Nathan Srebro, *Equality of opportunity in supervised learning*, NIPS, 2016.

[IST+19] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry, *Adversarial examples are not bugs, they are features*, arXiv preprint arXiv:1905.02175 (2019).

References V

[JDW+19] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve Gurel, Bo Li, Ce Zhang, Dawn Song, and Costas Spanos, *Towards efficient data valuation based on the shapley value*, arXiv preprint arXiv:1902.10275 (2019).

[KATL19] Pang Wei Koh, Kai-Siang Ang, Hubert HK Teo, and Percy Liang, *On the accuracy of influence functions for measuring group effects*, arXiv preprint arXiv:1905.13289 (2019).

[KHA+18] Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T. Schütt, Sven Dähne, Dumitru Erhan, and Been Kim, *The (un)reliability of saliency methods*, CoRR **abs/1711.00867** (2018).

References VI

[KL17] Pang Wei Koh and Percy S. Liang, *Understanding black-box predictions via influence functions*, ICML, 2017.

[LLLS18] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin, *A simple unified framework for detecting out-of-distribution samples and adversarial attacks*, NeurIPS, 2018.

[LLS18] Shiyu Liang, Yixuan Li, and R. Srikant, *Enhancing the reliability of out-of-distribution image detection in neural networks*, ICLR, 2018.

[LPB17] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell, *Simple and scalable predictive uncertainty estimation using deep ensembles*, NIPS, 2017.

References VII

[MLW+18] Xingjun Ma, Bo Li, Yisen Wang, Sarah M. Erfani, Sudanthi N. R. Wijewickrema, Michael E. Houle, Grant Schoenebeck, Dawn Xiaodong Song, and James Bailey, *Characterizing adversarial subspaces using local intrinsic dimensionality*, CoRR **abs/1801.02613** (2018).

[MTRAR+12] Jose G. Moreno-Torres, Troy Raeder, Rocío Alaíz-Rodríguez, Nitesh V. Chawla, and Francisco Herrera, *A unifying view on dataset shift in classification*, Pattern Recognition **45** (2012), 521–530.

[NDZ+19] Jeremy Nixon, Mike Dusenberry, Linchuan Zhang, Ghassen Jerfel, and Dustin Tran, *Measuring calibration in deep learning*, CoRR **abs/1904.01685** (2019).

References VIII

[NMT+18] Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan, *Do deep generative models know what they don't know?*, arXiv preprint arXiv:1810.09136 (2018).

[NMTL19] Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, and Balaji Lakshminarayanan, *Detecting out-of-distribution inputs to deep generative models using a test for typicality*, arXiv preprint arXiv:1906.02994 (2019).

[QCSSL09] Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence, *Dataset shift in machine learning*, 2009.

[RKH19] Kevin Roth, Yannic Kilcher, and Thomas Hofmann, *The odds are odd: A statistical test for detecting adversarial examples*, ICML, 2019.

References IX

[RSG18] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin, *Semantically equivalent adversarial rules for debugging nlp models*, ACL, 2018.

[Tan08] Till Tantau, *The tikz and pgf packages*, 2008.

[TSE+18] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry, *Robustness may be at odds with accuracy*, stat **1050** (2018), 11.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, *Attention is all you need*, Advances in neural information processing systems, 2017, pp. 5998–6008.

References X

[WN19] Xinyi Wang and Graham Neubig, *Target conditioned sampling: Optimizing data selection for multilingual neural machine translation*, arXiv preprint arXiv:1905.08212 (2019).

[YCH+19] Puyudi Yang, Jianbo Chen, Cho-Jui Hsieh, Jane-Ling Wang, and Michael I. Jordan, *Ml-loo: Detecting adversarial examples with feature attribution*, ArXiv **abs/1906.03499** (2019).

[ZWS+13] Richard S. Zemel, Yu Wu, Kevin Swersky, Toniann Pitassi, and Cynthia Dwork, *Learning fair representations*, ICML, 2013.

[ZWY+17] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang, *Men also like shopping: Reducing gender bias amplification using corpus-level constraints*, EMNLP, 2017.

References XI

[ZWY+18] _____, *Gender bias in coreference resolution: Evaluation and debiasing methods*, NAACL-HLT, 2018.

[ZWY+19] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Ryan Cotterell, Vicente Ordonez, and Kai-Wei Chang, *Gender bias in contextualized word embeddings*, CoRR **abs/1904.03310** (2019).

[ZZH+18] Yang Zhao, Jiajun Zhang, Zhongjun He, Chengqing Zong, and Hua Wu, *Addressing troublesome words in neural machine translation*, EMNLP, 2018.

[ZZW18] Xuezhou Zhang, Xiaojin Zhu, and Stephen Wright, *Training set debugging using trusted items*, Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[ZZZ+19] Yang Zhao, Jiajun Zhang, Chengqing Zong, Zhongjun He, Hua Wu, et al., *Addressing the under-translation problem from the entropy perspective*.

Karush–Kuhn–Tucker Conditions

Nonlinear optimization problem

Necessary conditions

Karush–Kuhn–Tucker Conditions

Nonlinear optimization problem
Necessary conditions

Singular Value Decomposition

Statement of the theorem

Intuitive interpretations

Singular Value Decomposition

Statement of the theorem
Intuitive interpretations