# SSMBA: Self-Supervised Manifold Based Data Augmentation for Improving Out-of-Domain Robustness

집현전 중급반
https://github.com/jiphyeonjeon

발표자 : 박동주
https://github.com/toriving

# About

JIPHYEONJEON

## SSMBA: Self-Supervised Manifold Based Data Augmentation for Improving Out-of-Domain Robustness

**Nathan Ng**
University of Toronto
Vector Institute

**Kyunghyun Cho**
New York University

**Marzyeh Ghassemi**
University of Toronto
Vector Institute
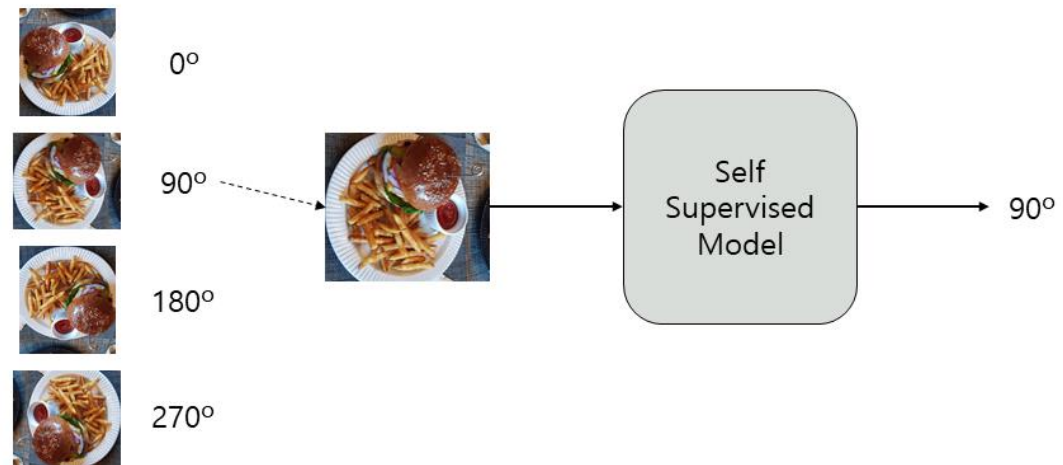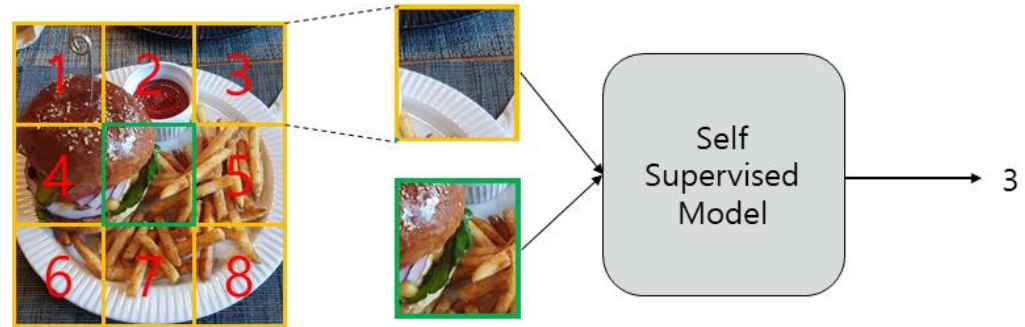
**EMNLP2020 – Long paper**

**Topic : Data augmentation**

Paper : https://www.aclweb.org/anthology/2020.emnlp-main.97.pdf
Code : https://github.com/nng555/ssmba

https://toriving.github.io/SSMBA-Self-Supervised-Manifold-Based-Data-Augmentation-for-Improving-Out-of-Domain-Robustness

01

Input image

Natural Language Processing
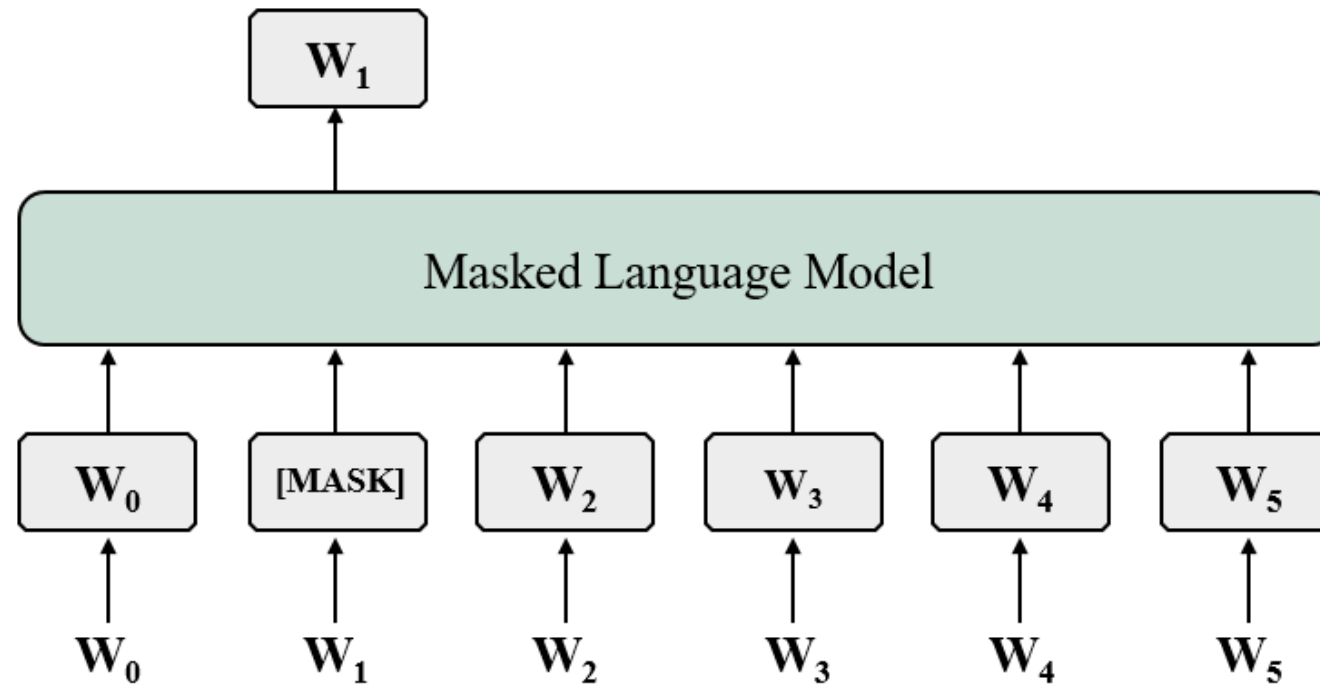
# Data Augmentation – Computer Vision



Original

Blur

Noise

Translation

Horizontal flip

Vertical flip

Rotation

Rotation

# Data Augmentation – Natural Language Processing

vertical-flip

brighten

coarse-salt

I go to school by bus

**I go to school by bus**

I go to ????? by bus

| 255 | 255 | 255 |
| 0 | 0 | 255 |
| 0 | 0 | 255 |

$$W(\text{``cat''}) = (0.2,\ -0.4,\ 0.7,\ \dots)$$

$$W(\text{``mat''}) = (0.0,\ 0.6,\ -0.1,\ \dots)$$
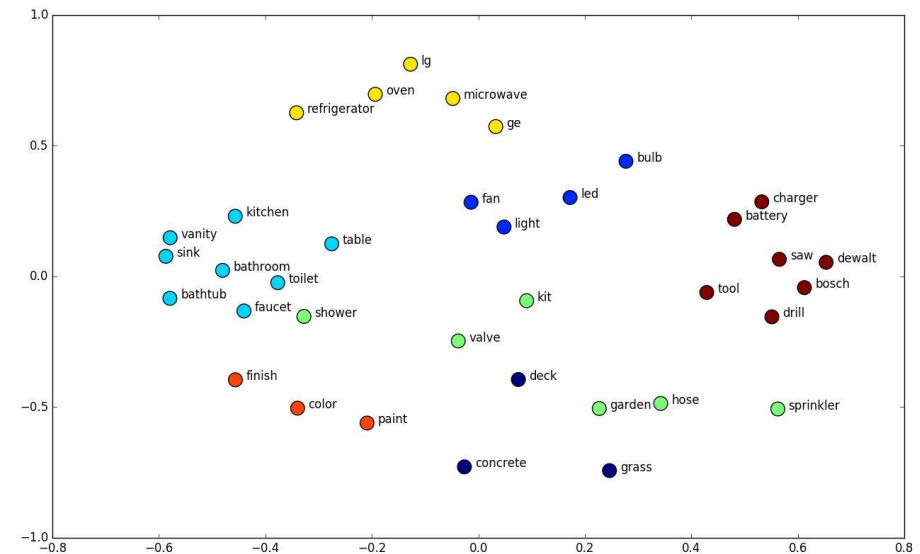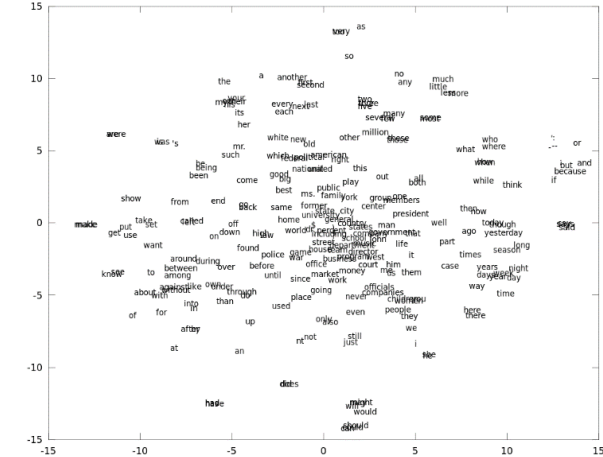
**Introduction**

- Training distributions often do not cover all of the test distributions we would like a supervised classifier or model to perform well on.

- Therefore, a key challenge in training machine learning models in these settings is ensuring they are robust to unseen examples.

- If data concentrates on a low-dimensional manifold then these synthetic examples should lie in a manifold neighborhood of the original examples.

- However, in domains such as natural language, it is much more difficult to find a set of invariances that preserves meaning or semantics.

- Self-Supervised Manifold Based Data Augmentation (SSMBA): a data augmentation method for generating synthetic examples in domains where the data manifold is difficult to heuristically characterize.

- Motivation : Denoising auto-encoders
    - Corruption fuction : Stochastically perturb examples *off* the data manifold
    - Reconstruction function : Project them *back* on.
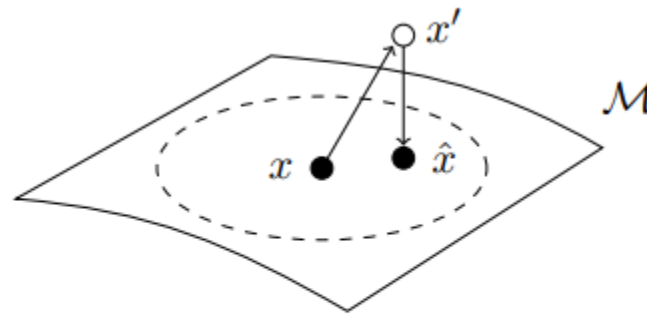


Figure 1: SSMBA moves along the data manifold $\mathcal{M}$ by using a corruption function to perturb an example $x$ off the data manifold, then using a reconstruction function to project it back on.

## 1. Data Augmentation in NLP

- Existing work on improving generalization has focused on data augmentation, where synthetically generated training examples are used to augment an existing dataset.

- It is hypothesized that these examples induce robustness to local perturbations, which has been shown to be effective in semi-supervised and self-supervised settings.
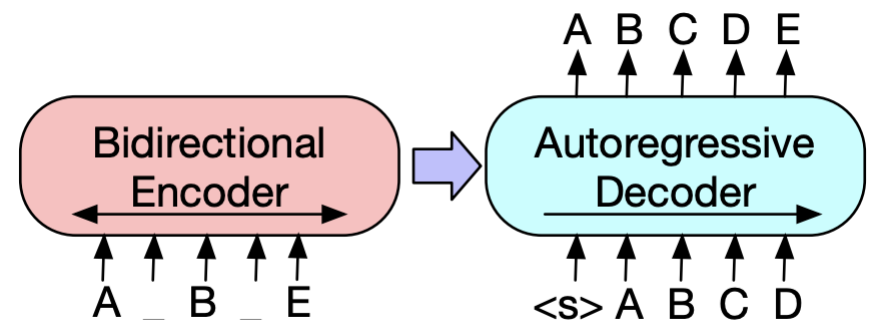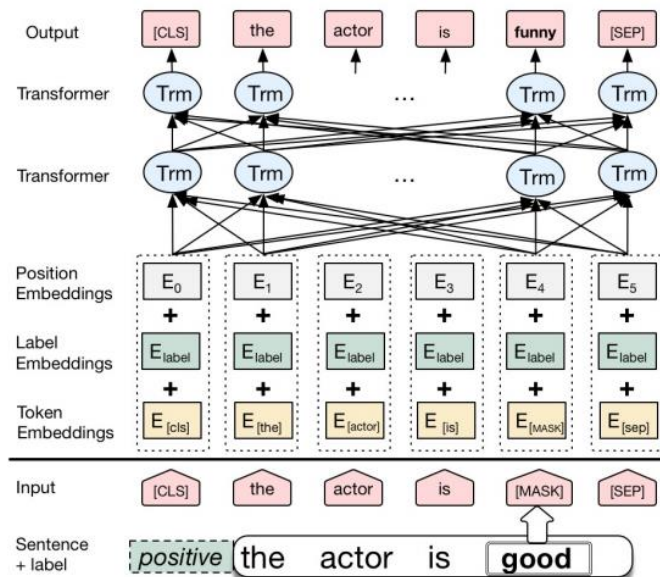
## 1. VRM and the Manifold Assumption

- Vicinal Risk Minimization (VRM) formalizes data augmentation as enlarging the training set support by drawing samples from a vicinity of existing training examples.

- Expected risk : $R(f) = \int \ell(f(x), y) \mathrm{d}P(x, y).$

- Empirical risk : $R_\delta(f) = \int \ell(f(x), y) \mathrm{d}P_\delta(x, y) = \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i).$

$$P_\delta(x, y) = \frac{1}{n} \sum_{i=1}^{n} \delta(x = x_i, y = y_i)$$

ERM                    mixup



- Vicinal risk : $R_\nu(f) = \frac{1}{m} \sum_{i=1}^{m} \ell(f(\tilde{x}_i), \tilde{y}_i).$

$$P_\nu(\tilde{x}, \tilde{y}) = \frac{1}{n} \sum_{i=1}^{n} \nu(\tilde{x}, \tilde{y} | x_i, y_i)$$
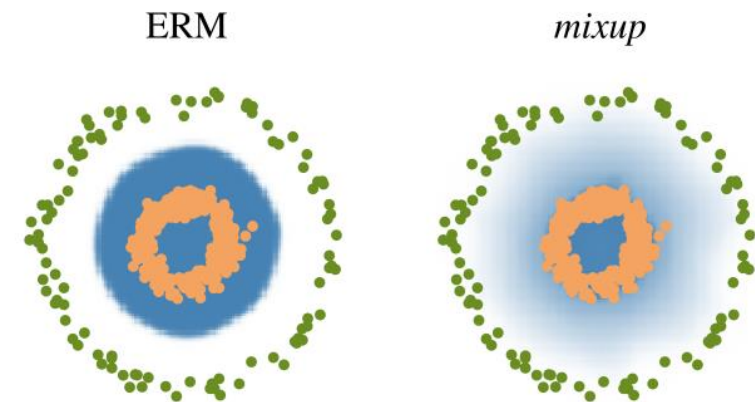
## 2. VRM and the Manifold Assumption

- The manifold assumption states that high dimensional data concentrates around a low-dimensional manifold.

- This assumption allows us to define the vicinity of a training example as its manifold neighborhood, the portion of the neighborhood that lies on the data manifold.
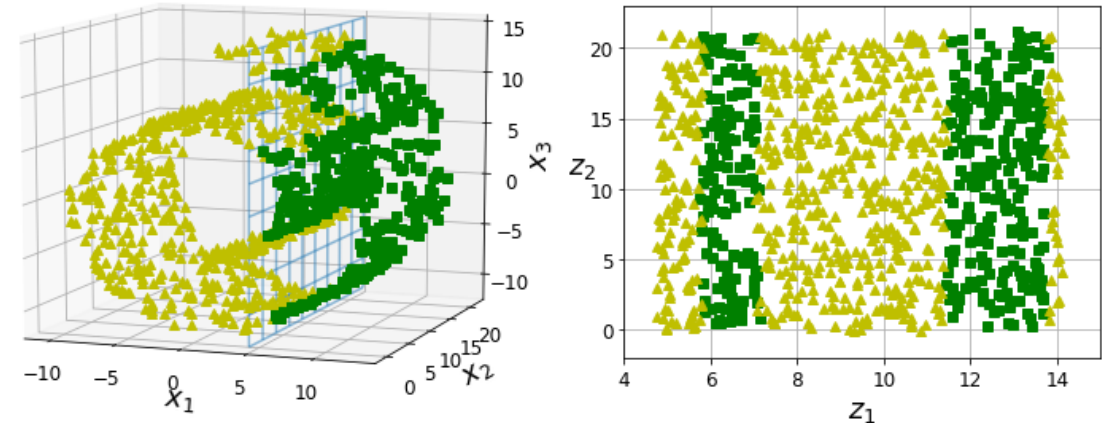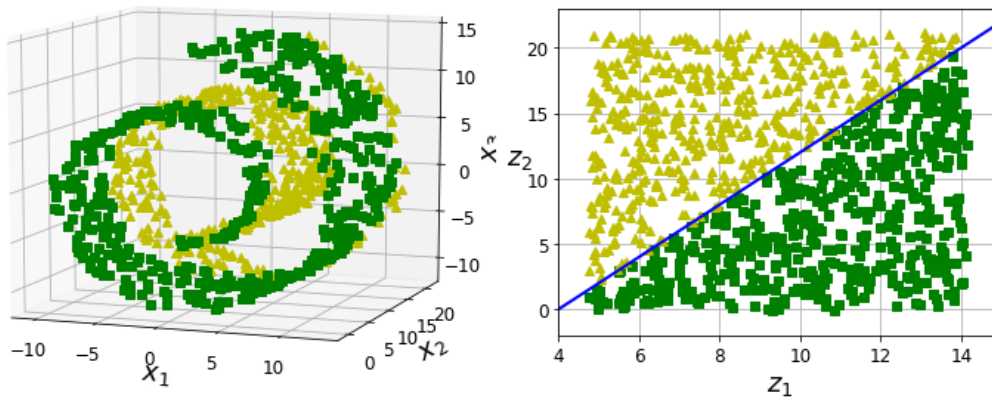
**3. Sampling from Denoising Autoencoders**

- A denoising autoencoder (DAE) is an autoencoder trained to reconstruct a clean input $x$ from a stochastically corrupted one $x' \sim q(x'|x)$ by learning a conditional distribution $P_\theta(x|x')$

- Sample from a DAE by successively corrupting and reconstructing an input using the following pseudo-Gibbs Markov chain:

$$x'_t \sim q(x'|x_{t-1}), \ x_t \sim P_\theta(x|x'_t).$$
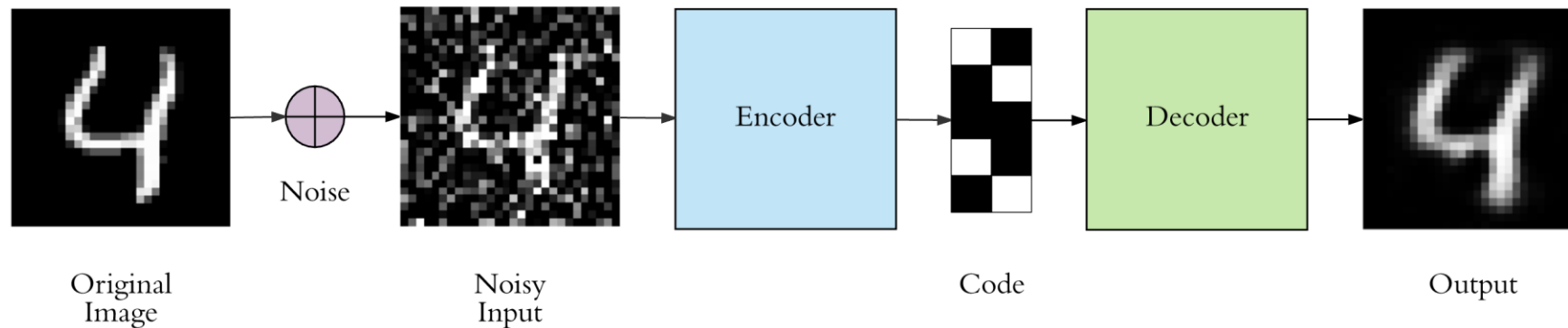
## 4. Masked Language Models

- Recent advances in unsupervised representation learning for natural language have relied on pretraining models on a masked language modeling (MLM) objective.
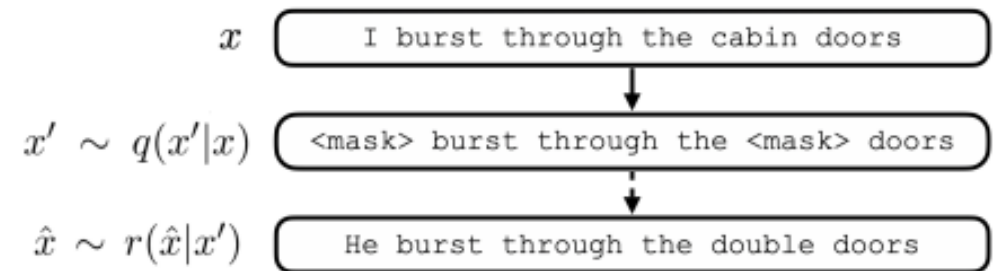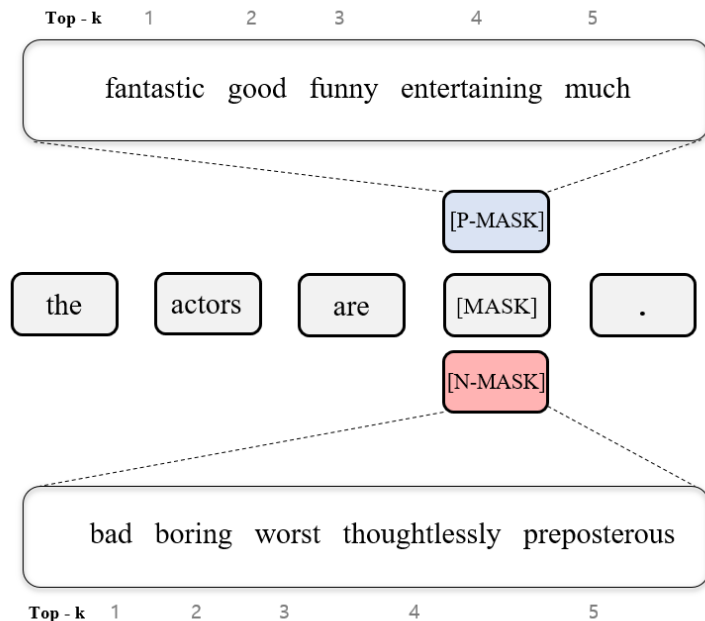




Figure 2: To sample from an MLM DAE, we apply the MLM corruption $q$ to the original sentence then reconstruct the corrupted sentence using our DAE $r$.

# SSMBA: Self-Supervised Manifold Based Augmentation

**Algorithm**

- Assume the input points concentrate around an underlying lower dimensional data manifold $M$.

- $q$ is corruption function, $x' \sim q(x'|x)$ such that $x'$ no longer lies on $M$.

- $r$ is reconstruction function, $\hat{x} \sim r(\hat{x}|x')$ such that $\hat{x}$ lies on $M$.

- $(x_i, y_i) \in D$ → $x_i' \sim q(x'|x_i)$ → $\hat{x}_{ij} \sim r(\hat{x}|x_i')$

- Label of $\hat{y}_{ij}$
  - Preserving $y_i$
  - Getting *soft label* via teacher model
  - Getting *hard label* via teacher model

- $q$ : Maksed Language Model
- $r$ : Pre-trained BERT

**Algorithm 1 SSMBA**

1: **Require:** perturbation function $q$
        reconstruction function $r$
2: **Input:** Dataset $\mathcal{D} = \{(x_1, y_1) \ldots (x_n, y_n)\}$
        number of augmented examples $m$
3: **function** SSMBA($\mathcal{D}, m$)
4:     train a model $f$ on $\mathcal{D}$
5:     **for** $(x_i, y_i) \in \mathcal{D}$ **do**
6:         **for** $j \in 1 \ldots m$ **do**
7:            sample perturbed $x_{ij}' \sim q(x'|x_i)$
8:            sample reconstructed $\hat{x}_{ij} \sim r(\hat{x}|x_{ij}')$
9:            generate $\hat{y}_{ij} \leftarrow f(\hat{x}_{ij})$ or preserve
           the original $y_i$
10:         **end for**
11:     **end for**
12:     let $\mathcal{D}^{aug} = \{(\hat{x}_{ij}, \hat{y}_{ij})\}_{i=1\ldots n, j=1\ldots m}$
13:     augment $\mathcal{D}' \leftarrow \mathcal{D} \cup \mathcal{D}^{aug}$
14:     **return** $\mathcal{D}'$
15: **end function**

**Contribution**

- SSMBA does not rely on task-specific knowledge, requires no dataset-specific fine-tuning, and is applicable to any supervised natural language task.

- SSMBA requires only a pair of functions $q$ and $r$ used to generate data.
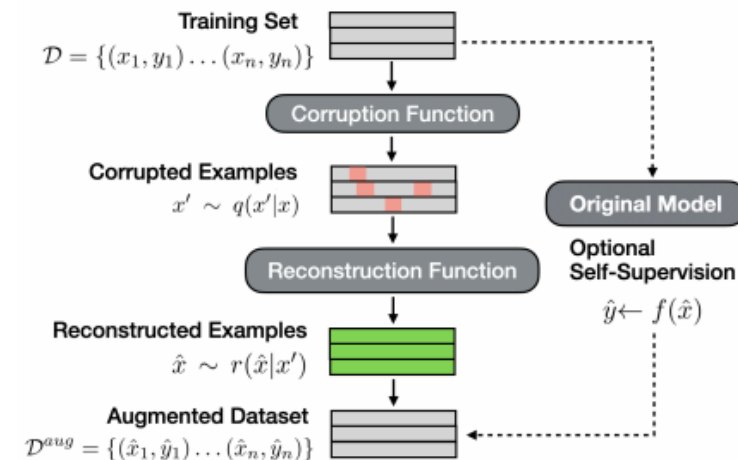


Figure 3: SSMBA generates synthetic examples by corrupting then reconstructing the original training inputs. To form the augmented dataset, corresponding outputs are preserved from the original data or generated from a supervised model $f$ trained on the original data.

# Datasets

**Setiment Analysis**

- Amzon Review Dataset (1 to 5 rating)
    - AR-Full : contains reviews from the 10 largest categories
    - AR-Clothing : contains reviews in the clothing category seperated into subcategories by metadata


- Movies dataset
    - SST2 : contains movie review excerpts
    - IMDb dataset : contains full length movie reviews


- Yelp Review Dataset : contains restaurant reviews with associated business metadata
  (1 to 5 star rating)

# Datasets

**Natural Language Inference**

- MNLI : corpus of NLI data from 10 distinct genres of written and spoken English.

- ANML : corpus of NLI data designed adversarially by humans such that state-of-the-art models fail to classify examples correctly.

**Machine Translation**

- IWSLT14 (de → en)

- OPUS (OOD)

- Allegra corpus

# Datasets

| Dataset | Domain | $n$ | $l$ | Train | Test |
|---|---|---|---|---|---|
| AR-Clothing | * | 4 | 35 | 25k$^\dagger$ | 2k |
| AR-Full | * | 10 | 67 | 25k$^\dagger$ | 2k |
| Yelp | * | 4 | 138 | 25k$^\dagger$ | 2k |
| Movies | SST2 | - | 11 | 66k | 1k |
| | IMDb | - | 296 | 46k | 2k |
| MNLI | * | 10 | 36 | 80k | 1k |
| ANLI | R1 | - | 92 | 17k | 1k |
| | R2 | - | 90 | 46k | 1k |
| | R3 | - | 82 | 100k | 1k |
| IWSLT | - | 1 | 24 | 160k | 7k |
| OPUS | Medical | 5 | 15 | 1.1m | 2k |
| de-rm | Law | - | 22 | 100k | 2k |
| | Blogs | - | 25 | - | 2k |

Table 1: Dataset summary statistics. $n$: number of domains. $l$: average tokenized input length. A * in the domain column indicates that the statistics are identical across domains within that dataset. Training sets marked with a $\dagger$ are sampled randomly from a larger dataset. Refer to Appendix A for more information.

# Experimental Setup

**1. Model Types**

- Sentiment analysis : LSTMs and CNNs

- NLI : fine-tuned RoBERTa_base

- MT : Transformers

**2. SSMBA Settings**

- $q$ : MLM corruption function.

- Tune the corruption percentage.

- $r$
  - Sentiment analysis and NLI : RoBERTa_base
  - MT : pre-trained German BERT model

- Generate 5 augmented examples using unrestricted sampling.

- For translation experiments, target side translations are generated with beam search with width 5

# Experimental Setup

**3. Baselines**

- Sentiment analysis and NLI tasks

    - Easy Data Augmentation (EDA)
    - Conditional Bert Contextual Augmentation (CBERT)
    - Unsupervised Data Augmentation (UDA) : Back translation

- MT tasks

    - Word dropout : randomly chooses word in the source sentence to set to zero embeddings
    - Reward Augmented Maximum Likelihood (RAML) : noisy target sentences
    - SwitchOut : noise function similar to RAML to both the source and target side

## 1. Sentiment Analysis

| Model | Augmentation | AR-Full | | AR-Clothing | | Movies | | Yelp | | Average | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | ID | OOD | ID | OOD | ID | OOD | ID | OOD | ID | OOD |
| RNN | None | 69.46 | 66.32 | 69.25 | 67.80 | 90.74 | 71.94 | 62.51 | 61.28 | 70.16 | 66.17 |
| | EDA | 67.32 | 64.47 | 66.87 | 65.21 | 88.43 | 68.3 | 58.39 | 57.19 | 67.56 | 63.55 |
| | CBERT | 69.94 | 66.77 | 69.56 | 68.10 | **91.01** | 72.11 | 63.17 | 61.75 | 70.17 | 66.57 |
| | UDA | 69.92 | 66.97 | 69.98 | 68.24 | 90.05 | 69.73 | 63.40 | 62.13 | 70.64 | 66.53 |
| | SSMBA | **70.38**$^{*\dagger}$ | **67.41**$^{*\dagger}$ | **70.19** | **68.60**$^{*\dagger}$ | 89.61 | **73.20** | **63.85** | **62.83**$^{*\dagger}$ | **70.96** | **67.31** |
| CNN | None | 70.67 | 67.64 | 70.14 | 68.52 | 92.92 | 72.11 | 65.13 | 64.46 | 71.68 | 67.63 |
| | EDA | 68.52 | 66.03 | 67.76 | 66.17 | 91.22 | 74.20 | 60.99 | 59.88 | 69.13 | 65.65 |
| | CBERT | 70.62 | 67.70 | 70.13 | 68.23 | 92.92 | 71.56 | 65.09 | 64.19 | 71.65 | 67.49 |
| | UDA | 70.80 | 68.06 | 70.29 | 68.70 | 92.63 | 72.55 | 65.22 | 64.32 | 71.77 | 67.89 |
| | SSMBA | **71.10**$^{*}$ | **68.18**$^{*}$ | **70.74** | **69.04**$^{*}$ | **92.93** | **74.67** | **65.59** | **64.81**$^{*\dagger}$ | **72.11** | **68.33** |

Table 2: Average in-domain (ID) and out-of-domain (OOD) accuracy (%) for models trained on sentiment analysis datasets. Average performance across datasets is weighted by number of domains contained in each dataset. Accuracies marked with a $*$ and $\dagger$ are statistically significantly higher than unaugmented models and the next best model respectively, both with $p < 0.01$.

## 2. Natural Language Inference

| Augmentation | MNLI | | ANLI | |
| --- | --- | --- | --- | --- |
| | ID | OOD | ID | OOD |
| None | 84.29 | 80.61 | 42.54 | **43.80** |
| EDA | 83.44 | 80.34 | 45.59 | 42.77 |
| CBERT | 84.24 | 80.34 | 46.68 | 43.53 |
| UDA | 84.24 | 80.99 | 45.85 | 42.89 |
| SSMBA | **85.71** | **82.44**$^{*\dagger}$ | **48.46**$^{*\dagger}$ | **43.80** |

Table 3: Average in-domain and out-of-domain accuracy (%) for RoBERTa models trained on NLI tasks. Accuracies marked with a $*$ and $\dagger$ are statistically significantly higher than unaugmented models and the next best model respectively, both with $p < 0.01$.

## 3. Machine Translation

| System | BLEU |
|---|---|
| ConvS2S (Edunov et al., 2018) | 32.2 |
| Transformer (Wu et al., 2019a) | 34.4 |
| DynamicConv (Wu et al., 2019a) | 35.2 |
| Transformer (ours) | 34.70 |
| + Word Dropout | 34.43 |
| + RAML | 35.00 |
| + SwitchOut | 35.28 |
| + SSMBA | $\mathbf{36.10^{*\dagger}}$ |

Table 4: Results on IWSLT de→en for models trained with different data augmentation methods. Scores marked with a $*$ and $\dagger$ are statistically significantly higher than baseline transformers and the next best model, both with $p < 0.01$.

| | OPUS | | de→rm | |
|---|---|---|---|---|
| Augmentation | ID | OOD | ID | OOD |
| None | **56.99** | 10.24 | 51.53 | 12.23 |
| Word Dropout | 56.26 | 10.15 | 50.23 | 12.23 |
| RAML | 56.76 | 10.10 | 51.52 | 12.49 |
| SwitchOut | 55.50 | 9.27 | 51.34 | 13.59 |
| SSMBA | 54.88 | **10.65** | **51.97** | $\mathbf{14.67^{*\dagger}}$ |

Table 5: Average in-domain and out-of-domain BLEU for models trained on OPUS (de→en) and de→rm data. Scores marked with a $*$ and $\dagger$ are statistically significantly higher than baseline transformers and the next best model, both with $p < 0.01$.

# Analysis and Discussion

**Settings**

*Baby domain* within the AR-Clothing dataset

- Relatively small size (25k sentences)

- # of OOD domains is 3

- CNN model

- 45% corrupution

- Restricted sampling

- Self-supervised soft labeling

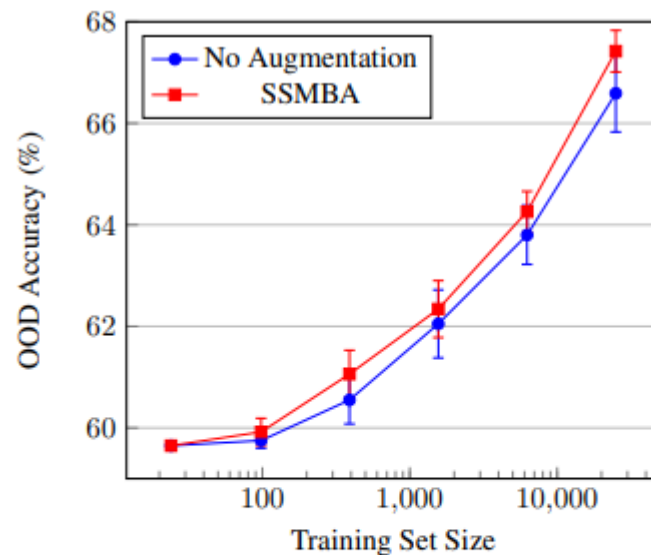- Generating 5 synthetic examples for each training example

1. **Training Set Size**



Figure 4: OOD accuracy of models trained on successively subsampled datasets. The full training set contains 25k examples. Error bars show standard deviation in OOD accuracy across models.

# Analysis and Discussion

## 2. Reconstruction Model Capacity

|  | Distil | Base | Large |
|---|---|---|---|
| OOD Accuracy Boost (%) | 0.73 | 0.78 | 0.78 |

Table 6: Boost in OOD accuracy (%) of models trained with SSMBA augmented data generated with different reconstruction functions.
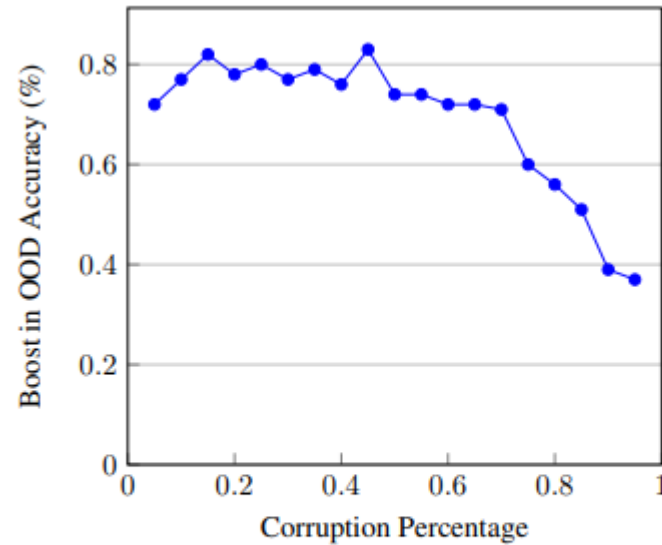
**3. Corruption Amount**



Figure 5: Boost in OOD accuracy (%) of models trained with SSMBA augmentation applied with different percentages of corrupted tokens.
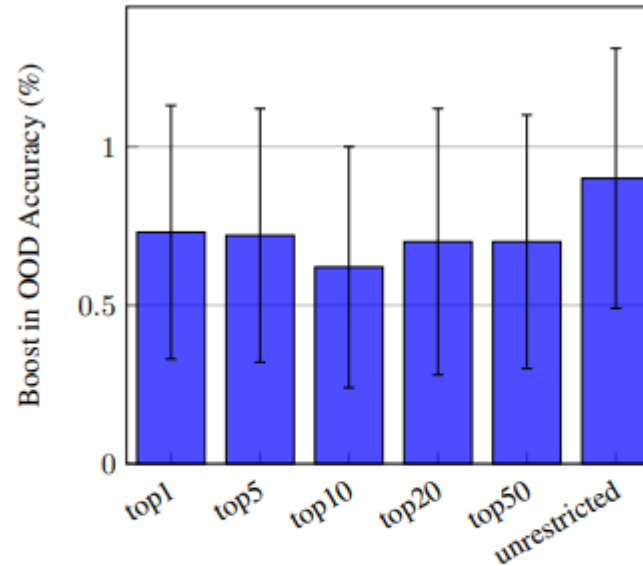
**4. Sample Generation Methods**



Figure 6: Boost in OOD accuracy (%) of models trained with SSMBA augmentation using different sampling methods. Error bars show standard deviation in OOD accuracy across models.
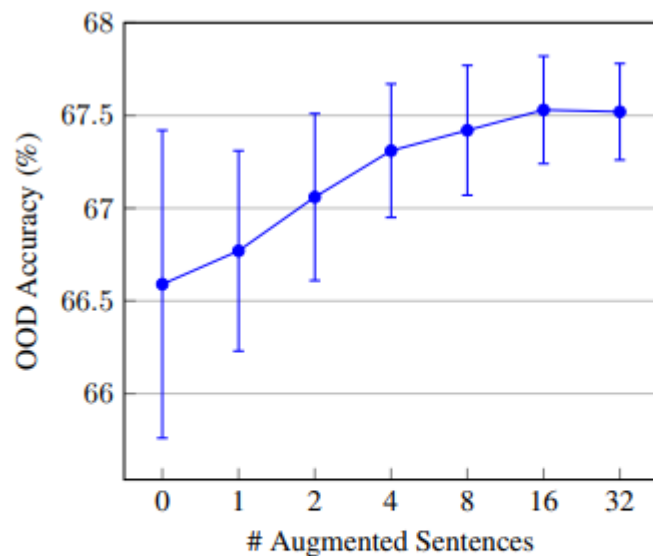
**5. Amount of Augmentation**



Figure 7: OOD accuracy (%) of models trained with different amounts of SSMBA augmentation. 0 augmentation corresponds to a baseline model. Error bars show standard deviation in OOD accuracy across models.
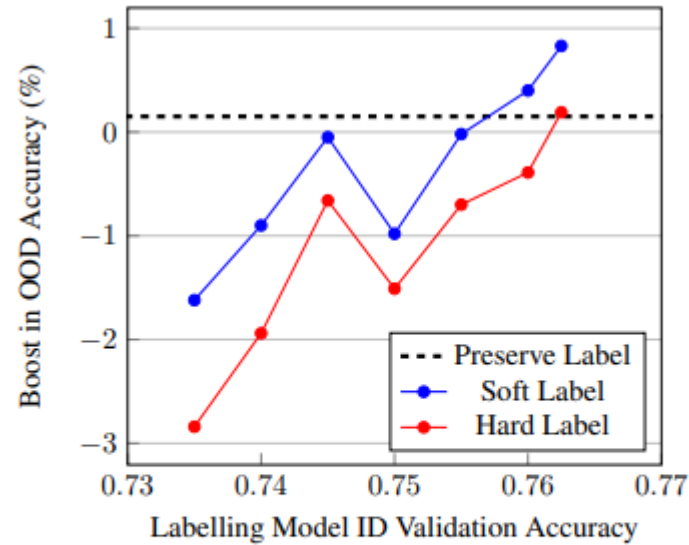
## 6. Label Generation



Figure 8: Boost in OOD accuracy (%) of models trained with augmented data labelled with different supervision models and label generation methods.

# Conclusion

- Defines Corruption Function and Reconstruction Function to augment data via Denoising Auto-Encoder method.

- MLM as corruption function and RoBERTa / Bert as reconstruction function.

- The difference from previous similar methodologies is that the reconstruction function is not fine-tuned.

- Labeling of augmented data via Self-Supervision

- Performance is improved even for out-of-domain (OOD) data

**JIPHYEONJEON**

# Thank you

# Reference

https://blog.insightdatascience.com/automl-for-data-augmentation-e87cf692c366

https://3months.tistory.com/136

https://www.shanelynn.ie/get-busy-with-word-embeddings-introduction/

https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798

https://arxiv.org/pdf/1710.09412.pdf

https://arxiv.org/pdf/2009.10195.pdf