# Data Augmentation for NMT

~Nalin Kumar
SMS, NISER

# Data Augmentation

- NMT models need huge amount of training data.
- We need a transformation *f*, such that for input ‘x’, the augmented data *f*(x), should preserve the label and x and *f*(x) should be diverse enough from each other.

- Generating augmented data in NLP is relatively tougher than data augmentation in image classification tasks.
- Some augmentation techniques:
  - Using thesaurus[1] (Replaced words with their synonym)
  - Using word embeddings [2] (Replaced words with their neighbouring words)
  - Back-translation [3]
  - Data diversification [4]
  - Cutoff [5]

Used in classification tasks


1. Zhang et al., "Character-level Convolutional Networks for Text Classification" (2015, NeurIPS)
2. Wang et al., "That's So Annoying!!!: A Lexical and Frame-Semantic Embedding Based Data Augmentation Approach to Automatic Categorization of Annoying Behaviors using #petpeeve Tweets" (2015, EMNLP)
3. Sennrich et al., "Improving Neural Machine Translation Models with Monolingual Data" (2016, ACL)
4. Nguyen et al., "Data Diversification: A Simple Strategy For Neural Machine Translation" (2020, NeurIPS))
5. Shen et al., "A Simple but Tough-to-Beat Data Augmentation Approach for Natural Language Understanding and Generation"

# Back-Translation

- The technique was first proposed in “Improving Neural Machine Translation Models with Monolingual Data” by Rico Sennrich, Barry Haddow and Alexandra Birch (ACL 2016)

- Uses monolingual data (which is considered to be easily available in comparison to bilingual data).

Monolingual target language corpus = T’

Bilingual corpus = D = (S, T)

1. Train $M_{T \text{-}> S}$ on “Target -> Source” bilingual corpus
2. Train $M_{S \text{-}> T}$ on $(M_{T \text{-}> S}(T'), T') \cup D$.

$M_{T \text{-}> S}(T')$ are referred as synthetic source sentences.

- In this paper, the NMT models are chosen to be RNN enc-dec models.
- They performed two sets of experiments:
    - In the first set, they generated data by pairing the monolingual data with single word <null> token.
    - In the second experiment, they generated data by previous algorithm.

Results:

| name | training instances | BLEU newstest2014 single | BLEU newstest2014 ens-4 | BLEU newstest2015 single | BLEU newstest2015 ens-4 |
|---|---|---|---|---|---|
| syntax-based (Sennrich and Haddow, 2015) | | 22.6 | - | 24.4 | - |
| Neural MT (Jean et al., 2015b) | | - | - | 22.4 | - |
| parallel | 37m (parallel) | 19.9 | 20.4 | 22.8 | 23.6 |
| +monolingual | 49m (parallel) / 49m (monolingual) | 20.4 | 21.4 | 23.2 | 24.6 |
| +synthetic | 44m (parallel) / 36m (synthetic) | **22.7** | **23.8** | **25.7** | **26.5** |

Table 3: English→German translation performance (BLEU) on WMT training/test sets. Ens-4: ensemble of 4 models. Number of training instances varies due to differences in training time and speed.

# Data Diversification

- This technique was introduced in “Data Diversification: A Simple Strategy For Neural Machine Translation” by Xuan-Phi Nguyen , Shafiq Joty , Wu Kui and Ai Ti Aw (NIPS 2020).
- This technique does not involve any extra monolingual data.

**Algorithm 1** Data Diversification: Given a dataset $\mathcal{D} = (S, T)$, a diversification factor $k$, the number of rounds $N$; return a trained source-target translation model $\hat{M}_{S \to T}$.

**procedure** $\text{TRAIN}(\mathcal{D} = (S, T))$
  Train randomly initialized $M$ on $\mathcal{D} = (S, T)$ until convergence
  **return** $M$

**procedure** $\text{DATADIVERSE}(\mathcal{D} = (S, T), k, N)$
  $\mathcal{D}_0 \leftarrow \mathcal{D}$ ▷ Assign original dataset to round-0 dataset.
  **for** $r \in 1, \ldots, N$ **do**
    $\mathcal{D}_r = (S_r, T_r) \leftarrow \mathcal{D}_{r-1}$
    **for** $i \in 1, \ldots, k$ **do**
      $M^i_{S \to T, r} \leftarrow \text{TRAIN}(\mathcal{D}_{r-1} = (S_{r-1}, T_{r-1}))$ ▷ Train forward model
      $M^i_{T \to S, r} \leftarrow \text{TRAIN}(\mathcal{D}'_{r-1} = (T_{r-1}, S_{r-1}))$ ▷ Train backward model
      $\mathcal{D}_r \leftarrow \mathcal{D}_r \cup (S, M^i_{S \to T, r}(S))$ ▷ Add forward data
      $\mathcal{D}_r \leftarrow \mathcal{D}_r \cup (M^i_{T \to S, r}(T), T)$ ▷ Add backward data
  $\hat{M}_{S \to T} \leftarrow Train(\mathcal{D}_N)$ ▷ Train the final model
  **return** $\hat{M}_{S \to T}$

$$\mathcal{D}_1 = (S, T) \bigcup \cup_{i=1}^{k} (S, M^i_{S \to T, 1}(S)) \bigcup \cup_{i=1}^{k} (M^i_{T \to S, 1}(T), T)$$

N = 1, K = 3

- We have $M^1_{S \to T, 1}$, $M^2_{S \to T, 1}$, $M^3_{S \to T, 1}$ forward models and $M^1_{T \to S, 1}$, $M^2_{T \to S, 1}$, $M^3_{T \to S, 1}$ backward models all trained on $D_0 = (S, T)$.

- $D_1$ is created by adding the outputs of above model along with the inputs to $D_0$.

- The final model is then trained on $D_1$.

Table 2: BLEU scores on newstest2014 for WMT'14 English-German (En-De) and English-French (En-Fr) translation tasks. Distill (T>S) (resp. T=S) indicates the teacher model is larger than (resp. equal to) the student model.

| Method | WMT'14 | |
|---|---|---|
| | En-De | En-Fr |
| Transformer [28]† | 28.4 | 41.8 |
| Trans+Rel. Pos [23]† | 29.2 | 41.5 |
| Scale Transformer [18] | 29.3 | 42.7[6] |
| Dynamic Conv [32]† | 29.7 | 43.2 |
| **Transformer with** | | |
| Multi-Agent [31]† | 30.0 | - |
| Distill (T>S) [14] | 27.6 | 38.6 |
| Distill (T=S) [14] | 28.4 | 42.1 |
| Ens-Distill [7] | 28.9 | 42.5 |
| **Our Data Diversification with** | | |
| Scale Transformer [18] | **30.7** | **43.7** |

Table 3: BLEU scores on IWSLT'14 English-German (En-De), German-English (De-En), and IWSLT'13 English-French (En-Fr) and French-English (Fr-En) translation tasks. Superscript † denotes the numbers are reported from the paper, others are based on our runs.

| Method | IWSLT'14 | | IWSLT'13 | |
|---|---|---|---|---|
| | En-De | De-En | En-Fr | Fr-En |
| **Baselines** | | | | |
| Transformer | 28.6 | 34.7 | 44.0 | 43.3 |
| Dynamic Conv | 28.7 | 35.0 | 43.8 | 43.5 |
| **Transformer with** | | | | |
| Multi-Agent† | 28.9 | 34.7 | - | - |
| Distill (T>S) | 28.0 | 33.6 | 43.4 | 42.9 |
| Distill (T=S) | 28.5 | 34.1 | 44.1 | 43.4 |
| Ens-Distill | 28.8 | 34.7 | 44.3 | 43.9 |
| **Our Data Diversification with** | | | | |
| Transformer | **30.6** | 37.0 | **45.5** | **45.0** |
| Dynamic Conv | **30.6** | **37.2** | 45.2 | 44.9 |

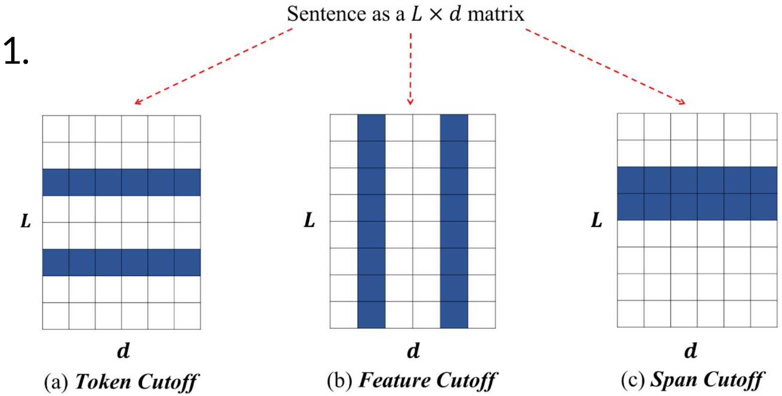
- The technique exhibits strong correlation with ensemble of models. However, the ensemble model would take N times more computation than one individual model while performing inference, whereas the data diversification technique does not have this disadvantage.
- The method is complementary to back-translation.

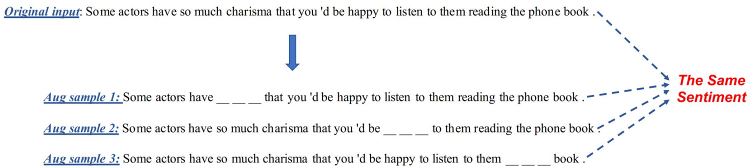
| **Task** | **No back-translation** | | **With back-translation** | | |
|---|---|---|---|---|---|
| | **Baseline** | **Ours** | $\lvert D \rvert$ | **Baseline** | **Ours** |
| IWSLT'14 En-De | 28.6 | 30.6 | 29× | 30.0 | **31.8** |
| IWSLT'14 De-En | 34.7 | 37.0 | 29× | 37.1 | **38.5** |
| WMT'14 En-De | 29.3 | 30.7 | 2.4× | 30.8 | **31.8** |

# Cutoff

- Introduced in “A Simple but Tough-to-Beat Data Augmentation Approach for Natural Language Understanding and Generation” by Dinghan Shen, Mingzhi Zheng, Yelong Shen, Yanru Qu, Weizhu Chen (Oct 2020).
- They propose a general technique, which can be used in various Natural language Understanding and Natural Language Generation tasks.

1.




- In the experiments, they found that token cutoff performs the best on machine translation tasks.
- This may be attributed to the fact that re-moving spans from both the source and target sentences would result in large information mismatch between the input and output, and thus the resulting pairs may be too challenging.

2.

$$\mathcal{L} = \mathcal{L}_{\text{ce}}(x, y) + \alpha \sum_{i=1}^{N} \mathcal{L}_{\text{ce}}(x^i_{\text{cutoff}}, y) + \beta \mathcal{L}_{\text{divergence}}(x, x^1_{\text{cutoff}}, x^2_{\text{cutoff}}, \ldots, x^N_{\text{cutoff}}, y)$$

$$p_{avg} = \frac{1}{N+1} \sum_{i=0}^{N} p(y|x^i_{\text{cutoff}})$$

$$\mathcal{L}_{\text{divergence}} = \frac{1}{N+1} \sum_{i=0}^{N} \text{KL}[p(y|x^i_{\text{cutoff}})||p_{avg}]$$

,

$$KL(P||Q) = \sum p_i(x) log(\frac{p_i(x)}{q_i(x)})$$

| Model | BLEU score |
|---|---|
| Actor-critic (Bahdanau et al., 2016) | 28.5 |
| Transformer Base (Vaswani et al., 2017) | 34.4 |
| Adversarial training (Wang et al., 2019) | 35.2 |
| Data Diversification (Nguyen et al., 2019) | 37.2 |
| MAT (Fan et al., 2020) | 36.2 |
| Mixed Representations (Wu et al., 2020) | 36.4 |
| MAT+Knee (Iyer et al., 2020) | 36.6 |
| Transformer Base & Cutoff (w/o JS loss) | 36.7 |
| Transformer Base & Cutoff (w/ JS loss) | **37.6** |

Table 3: BLEU scores of the proposed cutoff method on the IWSLT2014 German-to-English machine translation task, relative to adversarial-based baseline and other state-of-the-art models.

| Model | BLEU score |
|---|---|
| Transformer Base (Vaswani et al., 2017) | 27.3 |
| Admin (Liu et al., 2020a) | 27.9 |
| Transformer Base[1] (So et al., 2019) | 28.2 |
| Evolved Transformer (So et al., 2019) | 28.4 |
| Weighted Transformer (Ahmed et al., 2017) | 28.4 |
| Adversarial Training (Wang et al., 2019) | 28.4 |
| Transformer Base & Cutoff (w/o JS loss) | 28.9 |
| Transformer Base & Cutoff (w/ JS loss) | **29.1** |

Table 2: BLEU scores of the proposed cutoff method on the WMT2014 English-to-German machine translation task, compared with adversarial-based baselines. All methods are built on top of 6-layer Transformer Base model (Vaswani et al., 2017).