# 于孟萱
# 大组会報告

## 2019.03.01

YU MENG HSUAN

2019.03.01

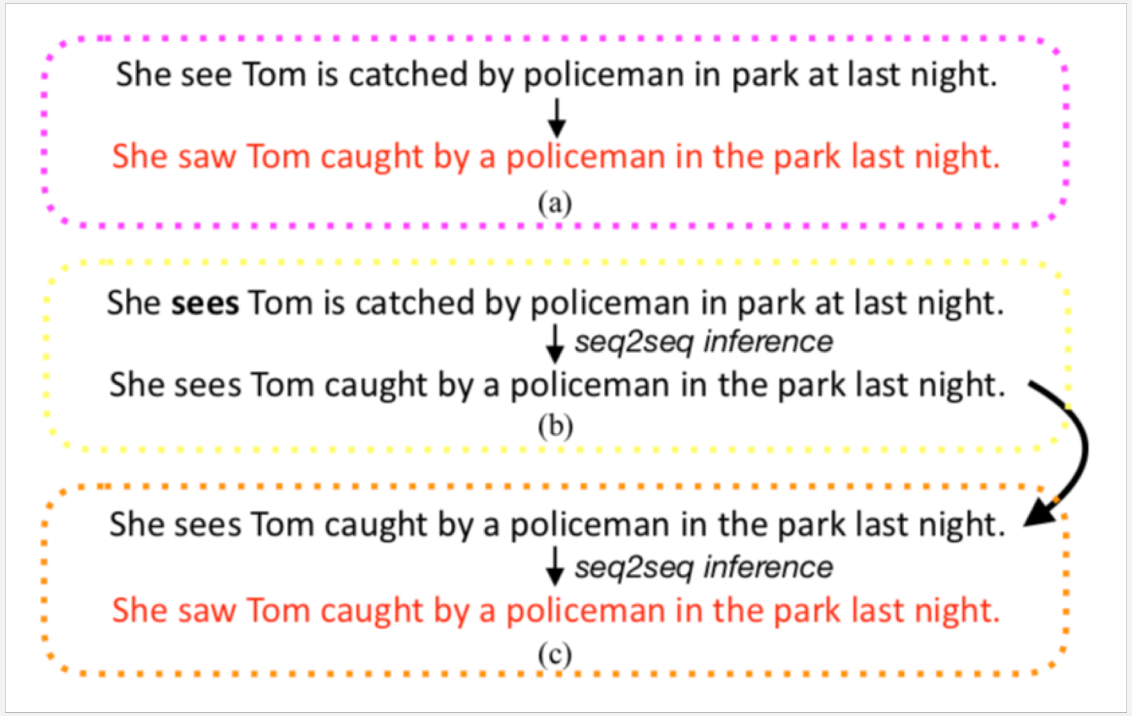
# Fluency Boost Learning and Inference for Neural Grammatical Error Correction

**Tao Ge** **Furu Wei** **Ming Zhou**
Microsoft Research Asia, Beijing, China
{tage, fuwei, mingzhou}@microsoft.com

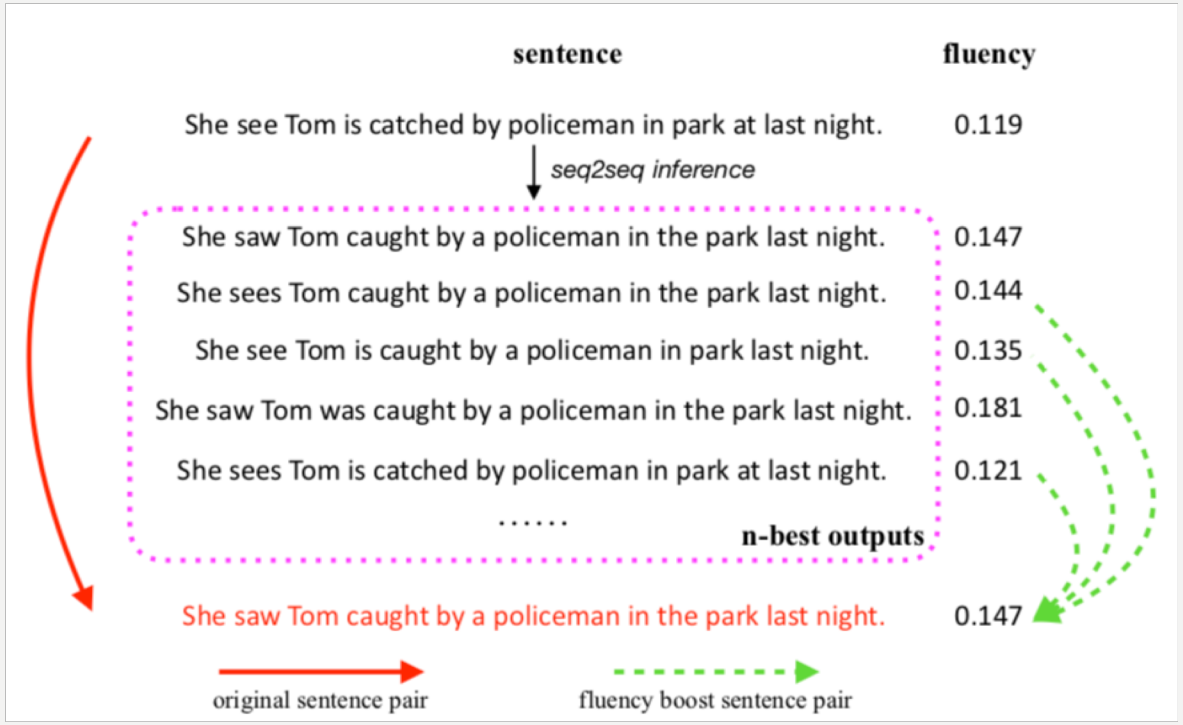
# TABLE OF CONTENT

# FLAWS FOR SEQ2SEQ GRAMMAR ERROR CORRECTION MODEL

- Trained with only limited error-corrected sentence pairs, causes sentence correction failure even if the sentence is slightly different from the training instance
- Seq2seq models cannot perfectly correct a sentence with many grammatical errors through a single-round seq2seq inference

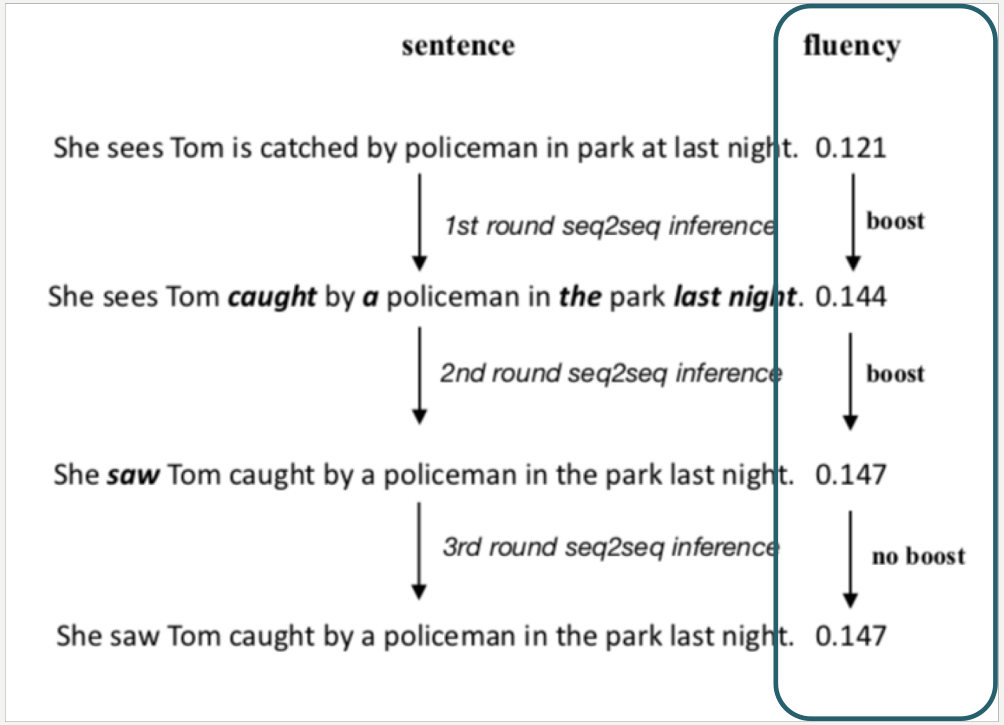


# FLUENCY BOOST LEARNING AND INFERENCE MECHANISM




- Fluency boost learning establishes multiple **fluency boost sentence pairs** from the seq2seiq's n-best outputs during training.
- The fluency boost sentence pairs will be used as training instances in sub-sequent training epochs.
- Allows the ECM trained with more grammatically incorrect sentences.
- Improves the models' generalization ability.

# FLUENCY BOOST LEARNING AND INFERENCE MECHANISM




- Fluency boost inference allows an error correction model to correct a sentence incrementally through multi-round seq2seq inference **until its fluency score stops increasing**.
- The corrected parts will make the context **clearer** and further benefit the model to correct the remaining errors.

# HOW TO DEFINE FLUENCY

How likely the sentence is written by a native speaker.

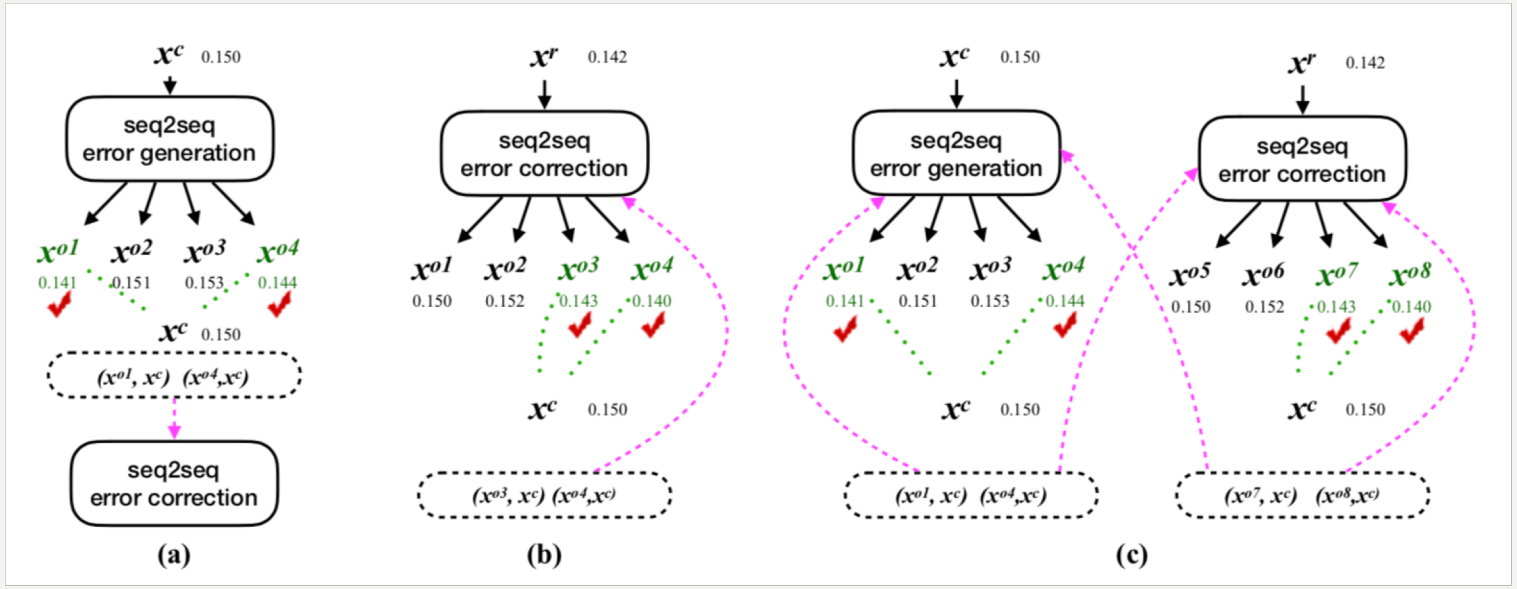
$$f(\boldsymbol{x}) = \frac{1}{1 + H(\boldsymbol{x})} \tag{3}$$

$$H(\boldsymbol{x}) = -\frac{\sum_{i=1}^{|\boldsymbol{x}|} \log P(x_i|\boldsymbol{x}_{<i})}{|\boldsymbol{x}|} \tag{4}$$

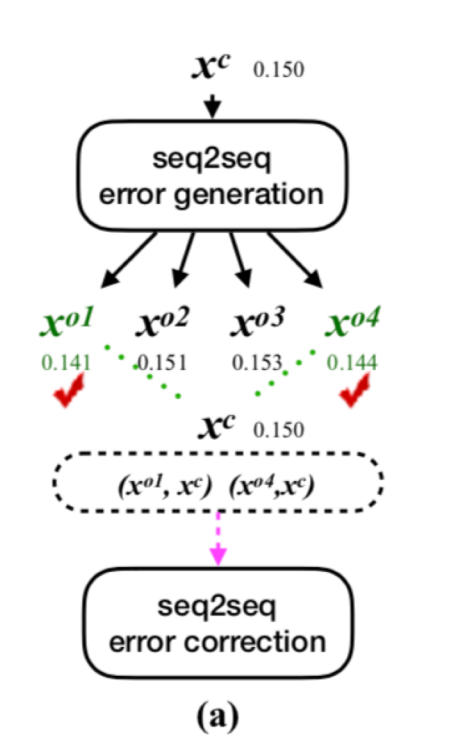
- P ($X_i$ $|X_{<i}$ ) is the probability of $X_i$ given context X < i (computed by a language model)
- |x| is the length of sentence x.

# THREE FLUENCY BOOST LEARNING STRATEGIES

- Back-boost
- Self-boost
- Dual-boost



# BACK BOOST LEARNING




1. Trains a backward model - **error generation model**, which converts a fluent sentence to a less fluent sentence with errors (**disfluency candidate of $X^C$**)

2. The less fluent sentence are paired with their correct sentences

3. New fluency boost sentence pairs pass to the ECM for further training

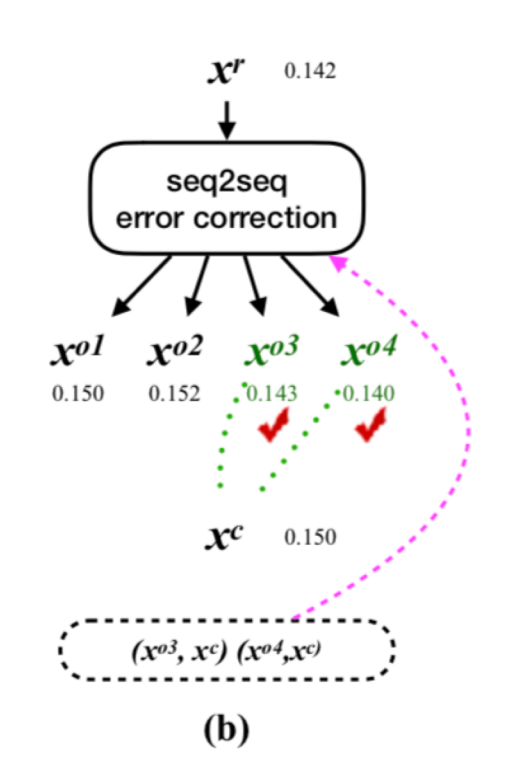
# BACK BOOST LEARNING

$$\mathcal{D}_{back}(\boldsymbol{x^c}) = \{\boldsymbol{x^{o_k}} | \boldsymbol{x^{o_k}} \in \mathcal{Y}_n(\boldsymbol{x_c}; \boldsymbol{\Theta}_{gen}) \wedge \frac{f(\boldsymbol{x^c})}{f(\boldsymbol{x^{o_k}})} \geq \sigma\} \quad (5)$$

- $D_{back}(x^c)$ denotes the disfluency candidate set for $x^c$ in back-boost learning
- $\sigma$ is a threshold to determine if $x^{ok}$ is less fluent than $x^c$
- $O_k$ denotes the k-best outputs
- $\Theta_{gen}$ denotes the error generation model

Error Correction Model Learning Source:
Original error-corrected sentence pairs ($x^r$,$x^c$)
Fluency boost sentence pairs ($x^{ok}$ ,$x^c$ ) where $x^{ok}$ is a sample of $D_{back}(x^c)$.

# SELF BOOST LEARNING




Allows the error correction model to generate the new fluency candidates by itself

1. Use ECM to predicts **n-best** outputs.
2. Among the n-best outputs, any output that is not identical to Xc consider as an error prediction.
3. For generated output that are **less fluent** than Xc will be added as Xc 's disfluency candidate.
4. New fluency boost sentence pairs pass to the ECM for further training.

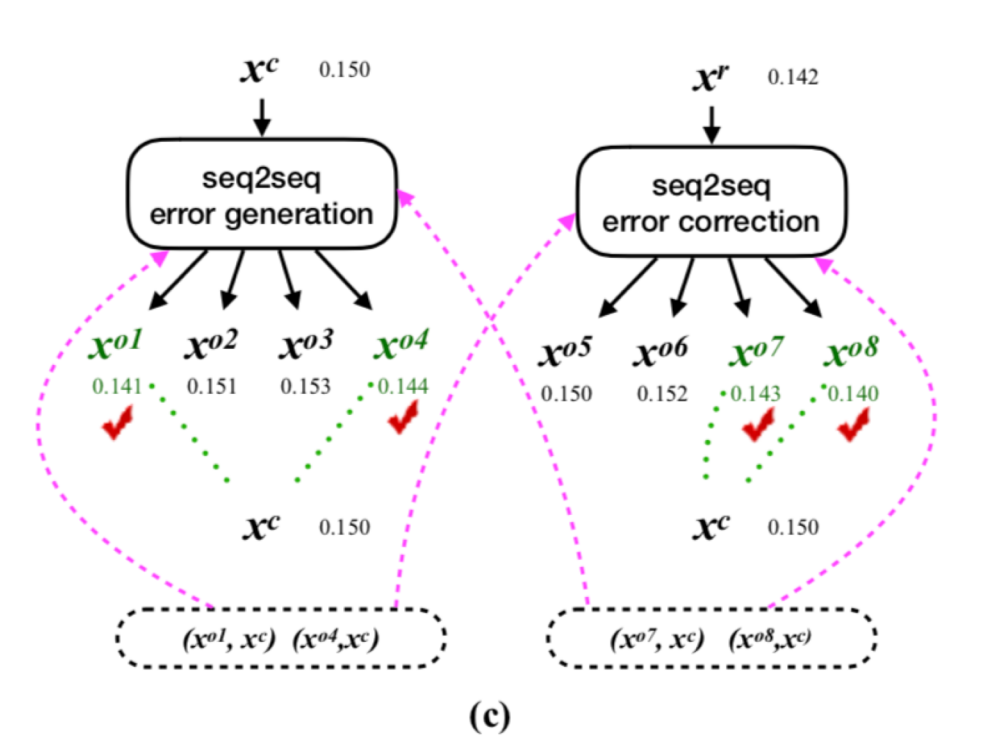
# SELF BOOST LEARNING

$$\mathcal{D}_{self}(\boldsymbol{x^c}) = \mathcal{D}_{self}(\boldsymbol{x^c}) \cup$$
$$\{\boldsymbol{x^{o_k}} | \boldsymbol{x^{o_k}} \in \mathcal{Y}_n(\boldsymbol{x_r}; \boldsymbol{\Theta}_{crt}) \wedge \frac{f(\boldsymbol{x^c})}{f(\boldsymbol{x^{o_k}})} \geq \sigma\} \quad (6)$$

- $D_{self}(x^c)$ denotes the disfluency candidate set for $x^c$ in self-boost learning
- $\sigma$ is a threshold to determine if $x^{ok}$ is less fluent than $x^c$
- $O_k$ denotes the k-best outputs
- $\Theta_{crt}$ denotes the error correction model

$D_{self}$ ($x^c$) is incrementally expanded due to the dynamically update error correction model $\Theta_{crt}$ .

# DUAL BOOST LEARNING




Dual-boost = back-boost + self- boost

Disfluency candidates

- produced by **error generation model** can benefit training the ECM.
- created by **error correction model** can be used as training data for the error generation model.

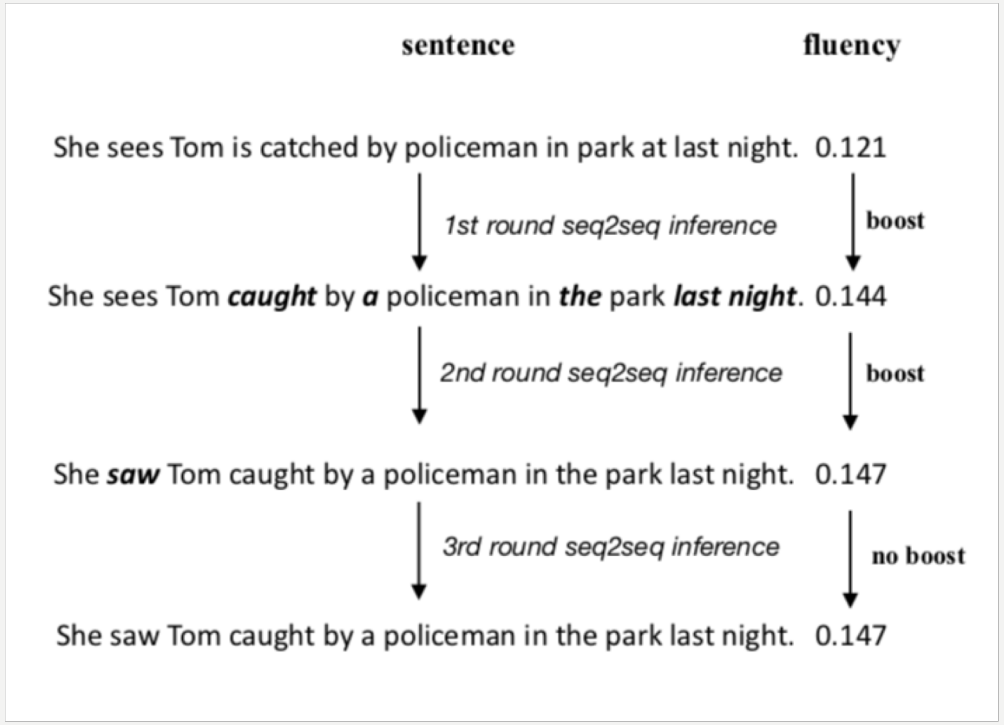
# DUAL BOOST LEARNING

$$\mathcal{D}_{dual}(\boldsymbol{x^c}) = \mathcal{D}_{dual}(\boldsymbol{x^c}) \cup$$
$$\{\boldsymbol{x^{o_k}} | \boldsymbol{x^{o_k}} \in \mathcal{Y}_n(\boldsymbol{x_r}; \boldsymbol{\Theta}_{crt}) \cup \mathcal{Y}_n(\boldsymbol{x_c}; \boldsymbol{\Theta}_{gen}) \wedge \frac{f(\boldsymbol{x^c})}{f(\boldsymbol{x^{o_k}})} \geq \sigma\} \quad (7)$$

- $D_{dual}(x^c)$ denotes the disfluency candidate set for $x^c$ in dual-boost learning
- $\sigma$ is a threshold to determine if $x^{ok}$ is less fluent than $x^c$
- $O_k$ denotes the k-best outputs
- $\Theta_{gen}$ denotes the error generation model
- $\Theta_{crt}$ denotes the error correction model

The **higher** the diversity of the disfluency candidates is and the **more** amount of disfluency candidates are generated, the **more helpful** for training an error correction model.

# FLUENCY BOOST INFERENCE




1. ECM takes a raw sentence $X_r$ as an input and outputs a hypothesis $X_{h1}$.
2. Fluency boost inference take $X_{h1}$ as the input to generate the next output $X_{h2}$
3. Process will terminate until the fluency score stops increasing.

# EXPERIMENT

| Model | seq2seq | | | fluency boost | | | seq2seq (+LM) | | | fluency boost (+LM) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $P$ | $R$ | $F_{0.5}$ | $P$ | $R$ | $F_{0.5}$ | $P$ | $R$ | $F_{0.5}$ | $P$ | $R$ | $F_{0.5}$ |
| normal seq2seq | 61.06 | 18.49 | 41.81 | 61.56 | 18.85 | 42.37 | 61.75 | 23.30 | 46.42 | 61.94 | 23.70 | 46.83 |
| back-boost | 61.66 | 19.54 | 43.09 | 61.43 | 19.61 | 43.07 | 61.47 | 24.74 | 47.40 | 61.24 | 25.01 | 47.48 |
| self-boost | 61.64 | 19.83 | 43.35 | 61.50 | 19.90 | 43.36 | 62.13 | 24.45 | 47.49 | 61.67 | 24.76 | 47.51 |
| dual-boost | 62.03 | 20.82 | 44.44 | 61.64 | 21.19 | 44.61 | 62.22 | 25.49 | 48.30 | 61.64 | 26.45 | 48.69 |
| back-boost (+native) | 63.93 | 22.03 | 46.31 | 63.95 | 22.12 | 46.40 | 62.04 | 27.43 | 49.54 | 61.98 | 27.70 | 49.68 |
| self-boost (+native) | 64.33 | 22.10 | 46.54 | 64.14 | 22.19 | 46.54 | 62.18 | 27.59 | 49.71 | 61.64 | **28.37** | 49.93 |
| dual-boost (+native) | 65.77 | 21.92 | 46.98 | **65.82** | 22.14 | 47.19 | 62.64 | 27.40 | 49.83 | 62.70 | 27.69 | **50.04** |
| back-boost (+native)⋆ | **67.37** | 24.31 | 49.75 | 67.25 | 24.35 | 49.73 | 64.61 | 28.44 | 51.51 | 64.46 | 28.78 | 51.66 |
| self-boost (+native)⋆ | 66.52 | 25.13 | 50.03 | 66.78 | 25.33 | 50.31 | 63.82 | 30.15 | 52.17 | 63.34 | **31.63** | 52.21 |
| dual-boost (+native)⋆ | 66.34 | 25.39 | 50.16 | 66.45 | 25.51 | 50.30 | 64.72 | 30.06 | 52.59 | 64.47 | 30.48 | **52.72** |

Performance on CoNLL-2014 dataset

Evaluate through precision / recall / $F_{0.5}$

- Systems with ⋆ use the additional non-public Lang-8 data for training.
- Systems with +native use English Wikipedia to extend the huge volume of native data
- (+LM) denotes to set beam size to 5 and decode the best result with a 2-layer GRU RNN language model

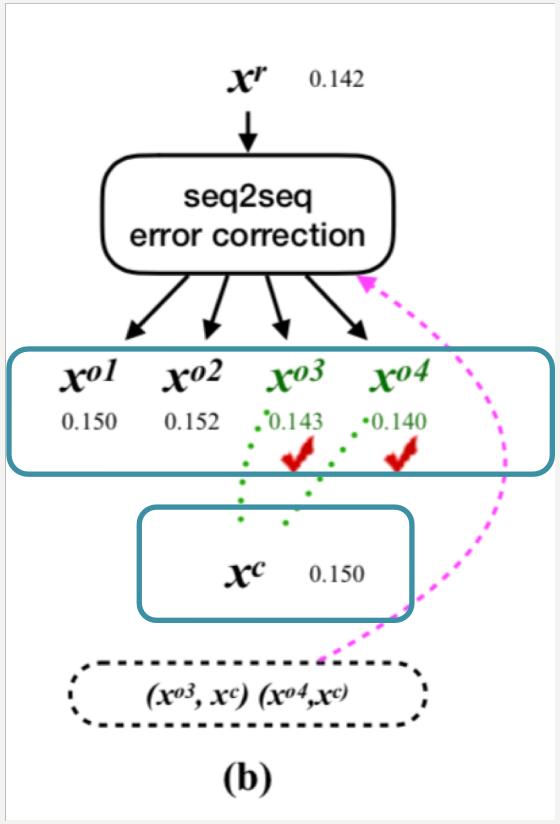
# EXPERIMENT

| Model | seq2seq | | | fluency boost | | | seq2seq (+LM) | | | fluency boost (+LM) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $P$ | $R$ | $F_{0.5}$ | $P$ | $R$ | $F_{0.5}$ | $P$ | $R$ | $F_{0.5}$ | $P$ | $R$ | $F_{0.5}$ |
| normal seq2seq | 61.06 | 18.49 | 41.81 | 61.56 | 18.85 | 42.37 | 61.75 | 23.30 | 46.42 | 61.94 | 23.70 | 46.83 |
| back-boost | 61.66 | 19.54 | 43.09 | 61.43 | 19.61 | 43.07 | 61.47 | 24.74 | 47.40 | 61.24 | 25.01 | 47.48 |
| self-boost | 61.64 | 19.83 | 43.35 | 61.50 | 19.90 | 43.36 | 62.13 | 24.45 | 47.49 | 61.67 | 24.76 | 47.51 |
| dual-boost | 62.03 | 20.82 | 44.44 | 61.64 | 21.19 | 44.61 | 62.22 | 25.49 | 48.30 | 61.64 | 26.45 | 48.69 |
| back-boost (+native) | 63.93 | 22.03 | 46.31 | 63.95 | 22.12 | 46.40 | 62.04 | 27.43 | 49.54 | 61.98 | 27.70 | 49.68 |
| self-boost (+native) | 64.33 | 22.10 | 46.54 | 64.14 | 22.19 | 46.54 | 62.18 | 27.59 | 49.71 | 61.64 | **28.37** | 49.93 |
| dual-boost (+native) | 65.77 | 21.92 | 46.98 | **65.82** | 22.14 | 47.19 | 62.64 | 27.40 | 49.83 | 62.70 | 27.69 | **50.04** |
| back-boost (+native)⋆ | **67.37** | 24.31 | 49.75 | 67.25 | 24.35 | 49.73 | 64.61 | 28.44 | 51.51 | 64.46 | 28.78 | 51.66 |
| self-boost (+native)⋆ | 66.52 | 25.13 | 50.03 | 66.78 | 25.33 | 50.31 | 63.82 | 30.15 | 52.17 | 63.34 | **31.63** | 52.21 |
| dual-boost (+native)⋆ | 66.34 | 25.39 | 50.16 | 66.45 | 25.51 | 50.30 | 64.72 | 30.06 | 52.59 | 64.47 | 30.48 | **52.72** |

- Dual-boost achieves the best result because it produces more diverse incorrect sentences.
- Large amounts of native text data improves the performance of all the fluency boost learning approaches.
  - Produce more error-corrected sentence pairs to let the model be better generalized.
  - Model can learn better and generate a much more fluent and error-free sentence from the native data.
- Compared to the normal seq2seq inference and seq2seq (+LM) baselines proves that multi-round edits by fluency boost inference is effective.

# EXPERIMENT

| $\sigma$ | 0 | 0.95 | 1.0 | 1.05 | 1.1 | 2.0 |
|---|---|---|---|---|---|---|
| $\lvert\mathcal{D}_{dual}\rvert$ | 41.18 | 39.21 | 29.40 | 9.43 | 3.87 | 0.01 |
| $F_{0.5}$ | 43.20 | 43.30 | 43.39 | **44.44** | 43.30 | 41.78 |




$$\sigma = \frac{f(x_c)}{f(x_{ok})}$$

When σ = 1.05, the model achieves the best performance.
(It effectively avoids generating sentence pairs with unnecessary or undesirable edits that affect the performance. e.g., I like this book. → I like the book )

# EXPERIMENT

| | |
|---|---|
| **Correct sentence** | How autism occurs is not well understood. |
| **Disfluency candidates** | How autism occurs is not good understood.<br>How autism occur is not well understood.<br>What autism occurs is not well understood.<br>How autism occurs is not well understand.<br>How autism occurs does not well understood. |

Examples of disfluency candidates for a correct sentence in dual-boost learning.

# EXPERIMENT

| System | CoNLL-2014 $F_{0.5}$ | CoNLL-10 $F_{0.5}$ | JFLEG test $GLEU$ |
|---|---|---|---|
| No edit | - | - | 40.54 |
| CAMB14 | 37.33 | 54.30 | 46.04 |
| CAMB16 | 39.90 | - | 52.05 |
| CAMB17 | 51.08 | - | - |
| CUUI | 36.79 | 51.79 | - |
| VT16 | 47.40 | 62.45 | - |
| AMU14 | 35.01 | 50.17 | - |
| AMU16 | 49.49 | 66.83 | 51.46 |
| NUS16 | 44.27 | 60.36 | 50.13 |
| NUS17 | 53.14 | 69.12 | 56.78 |
| NUS18 | 54.79 | 70.14 | 57.47 |
| Nested-RNN-seq2seq | 45.15 | - | 53.41 |
| Back-CNN-seq2seq | 49.0 | - | 56.6 |
| Adapted-transformer | 55.8 | - | 59.9 |
| SMT-NMT hybrid | 56.25 | - | 61.50 |
| Base convolutional seq2seq | 57.95 | 73.19 | 60.87 |
| Base + FB learning | **61.34** | **76.88** | 61.41 |
| Base + FB learning and inference | 60.00 | 75.72 | **62.42** |

Compare the model – *dual-boost learning* + *fluency boost inference* with the top-performing GEC systems evaluated on CoNLL-2014 dataset / CoNLL-10 / JFLEG corpus

# THANKS AND THE END