An Empirical Study of Incorporating Pseudo Data into Grammatical Error Correction

Shun Kiyono [1,2] **Jun Suzuki** [2,1] **Masato Mita** [1,2] **Tomoya Mizumoto** [1,2]* **Kentaro Inui** [2,1]
[1] RIKEN Center for Advanced Intelligence Project [2] Tohoku University
{shun.kiyono, masato.mita, tomoya.mizumoto}@riken.jp;
{jun.suzuki,inui}@ecei.tohoku.ac.jp

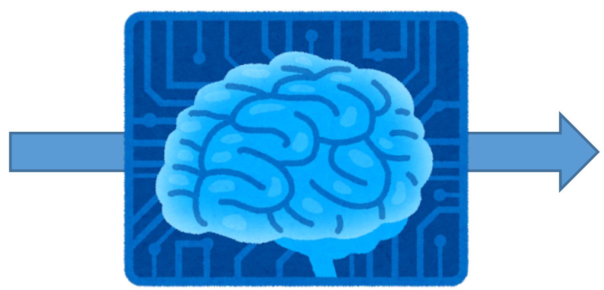
Code available at https://github.com/butsugiri/gec-pseudodata

Grammatical Error Correction (GEC)

- Input: sentence with grammatical error
- Output: sentence without grammatical error
- GEC is commonly tackled as Machine Translation

Input

I ***follows*** *his advice*



Model (e.g. Encoder-Decoder)

Output

I ***follow*** *his advice*

GEC data is "low-resource"

- Amount of parallel data is limited in GEC (~2M)
- More data is important for better performance

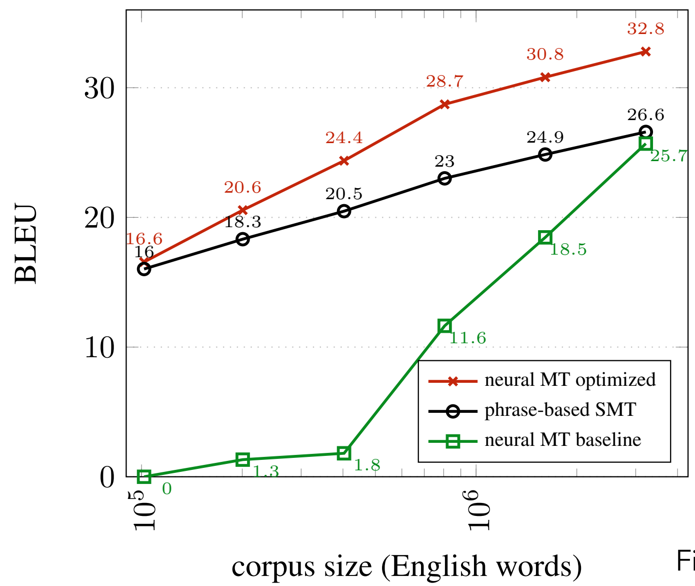
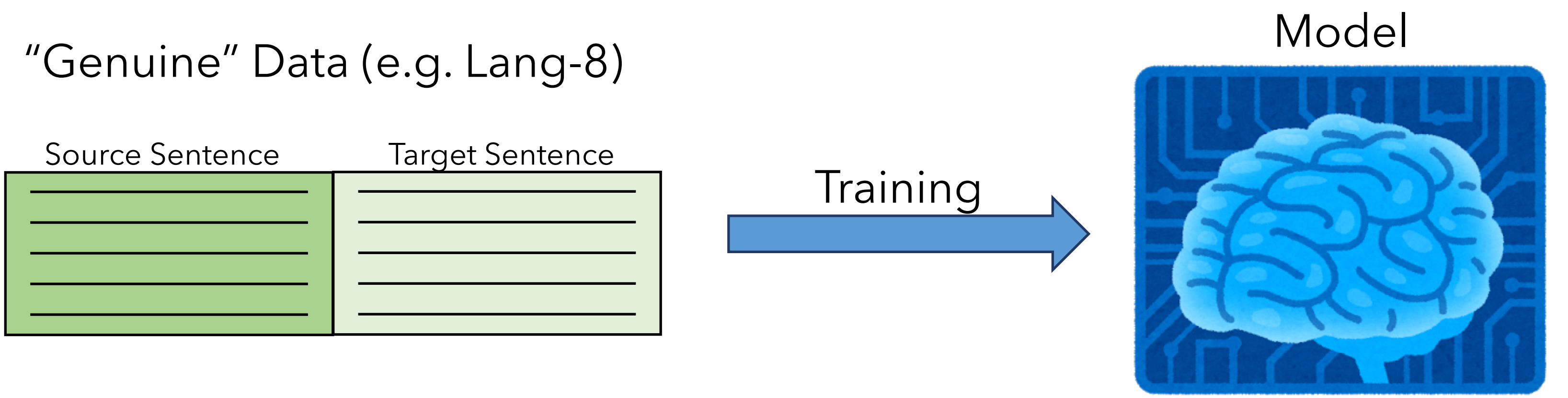


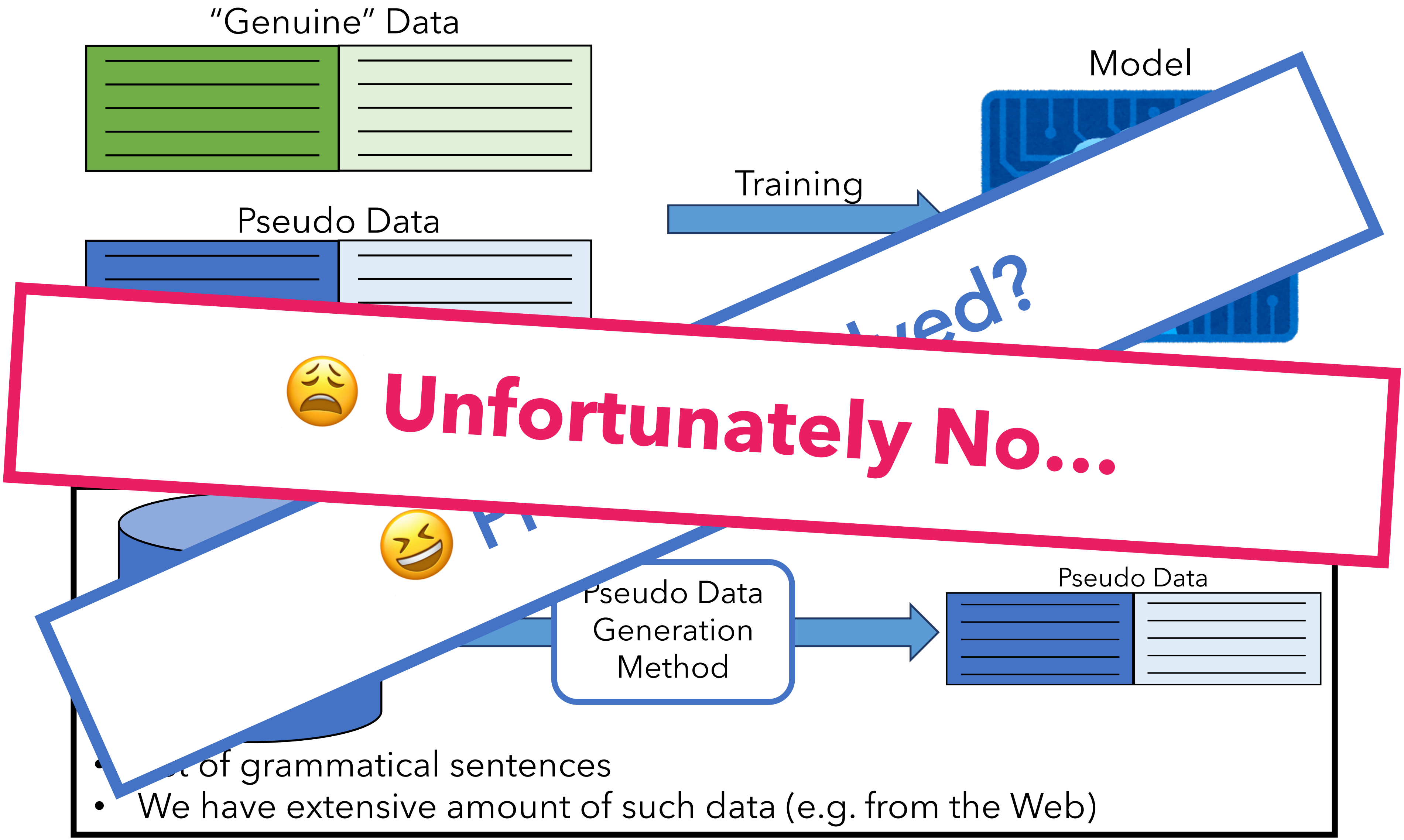
Figure from [Sennrich and Zhang 2019]

- **Pseudo data generation** is currently the best method to increase GEC data
  - Adopted by most teams in BEA 2019 Shared Task

Training with Genuine Data only



Training with Pseudo Data



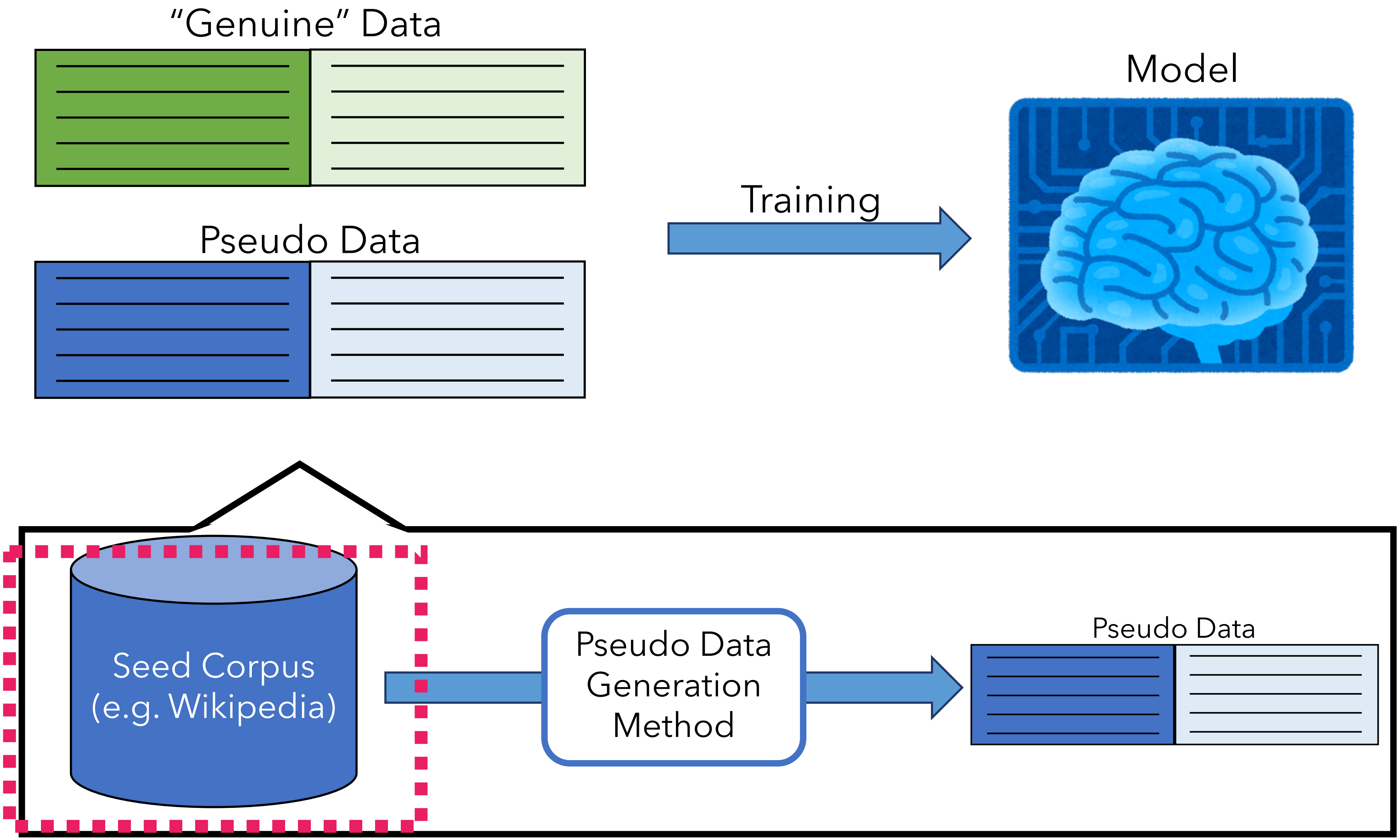
Problem: Lack of Consensus

- We have **three questions** regarding the pseudo data:

Q1: Choice for seed corpus?
Q2: Methods for generating pseudo data?
Q3: How to use pseudo data during training?

- GEC research community **lacks consensus** 😩
- Our aim: Find **settings that consistently improve performance** of GEC model

Q1: Choice for Seed Corpus?



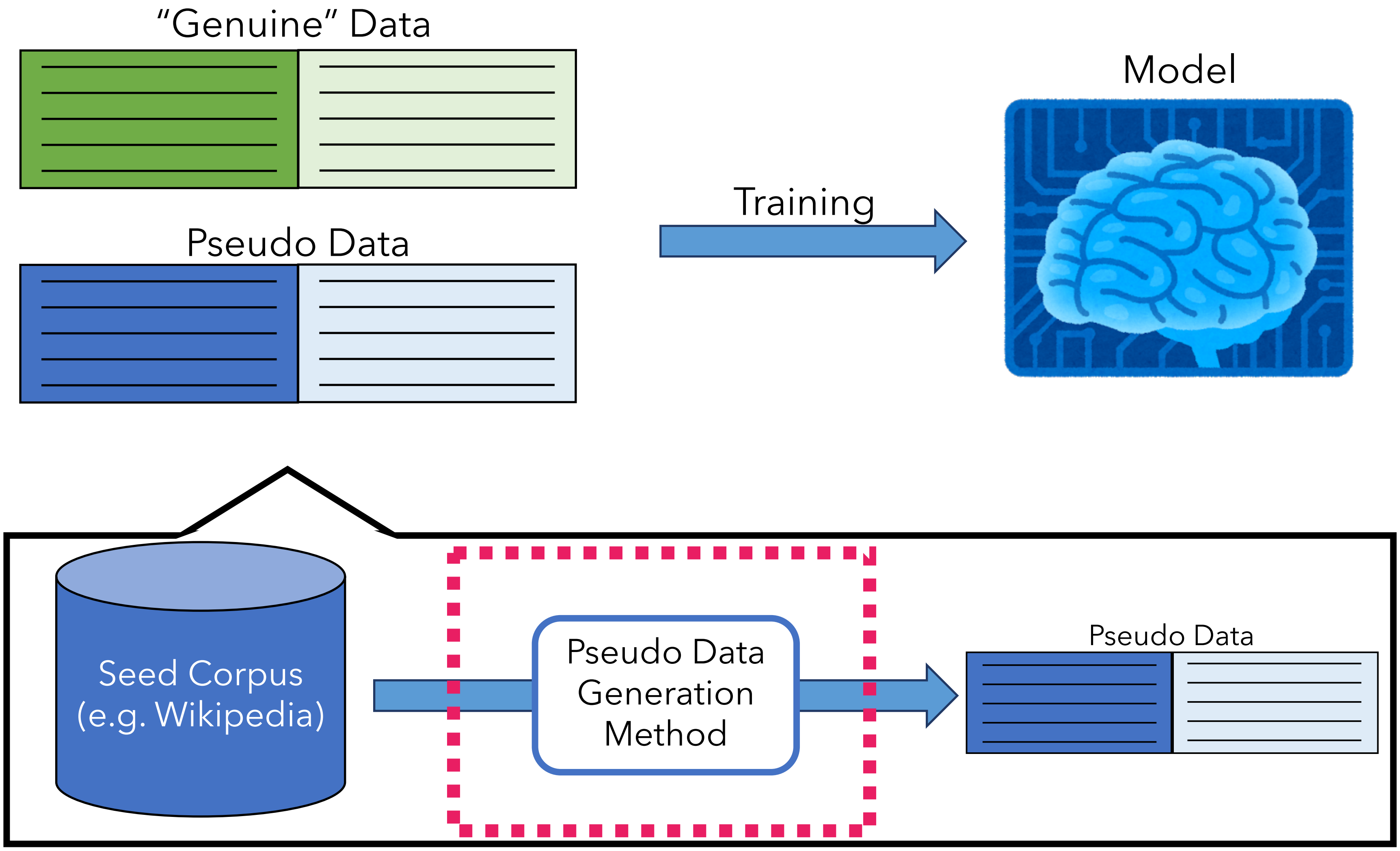
Q1: Choice for Seed Corpus?

- Numerous options exist:
  - Wikipedia, 1-billion word benchmark, BookCorpus, etc
  - [Ge+2018] uses Wikipedia
  - [Zhao+2019] uses 1-billion word LM benchmark
  - [Xie+2018] uses NYT corpus
  - [Grundkiewicz+2019] uses News Crawl
- What kind of corpus is suitable for GEC?
- We compare following three corpora:
  - Simple Wikipedia
  - Wikipedia
  - LDC Gigaword

Grammatical complexity is different

Texts are cleaner in Gigaword

Q2: Methods for Generating Pseudo Data?



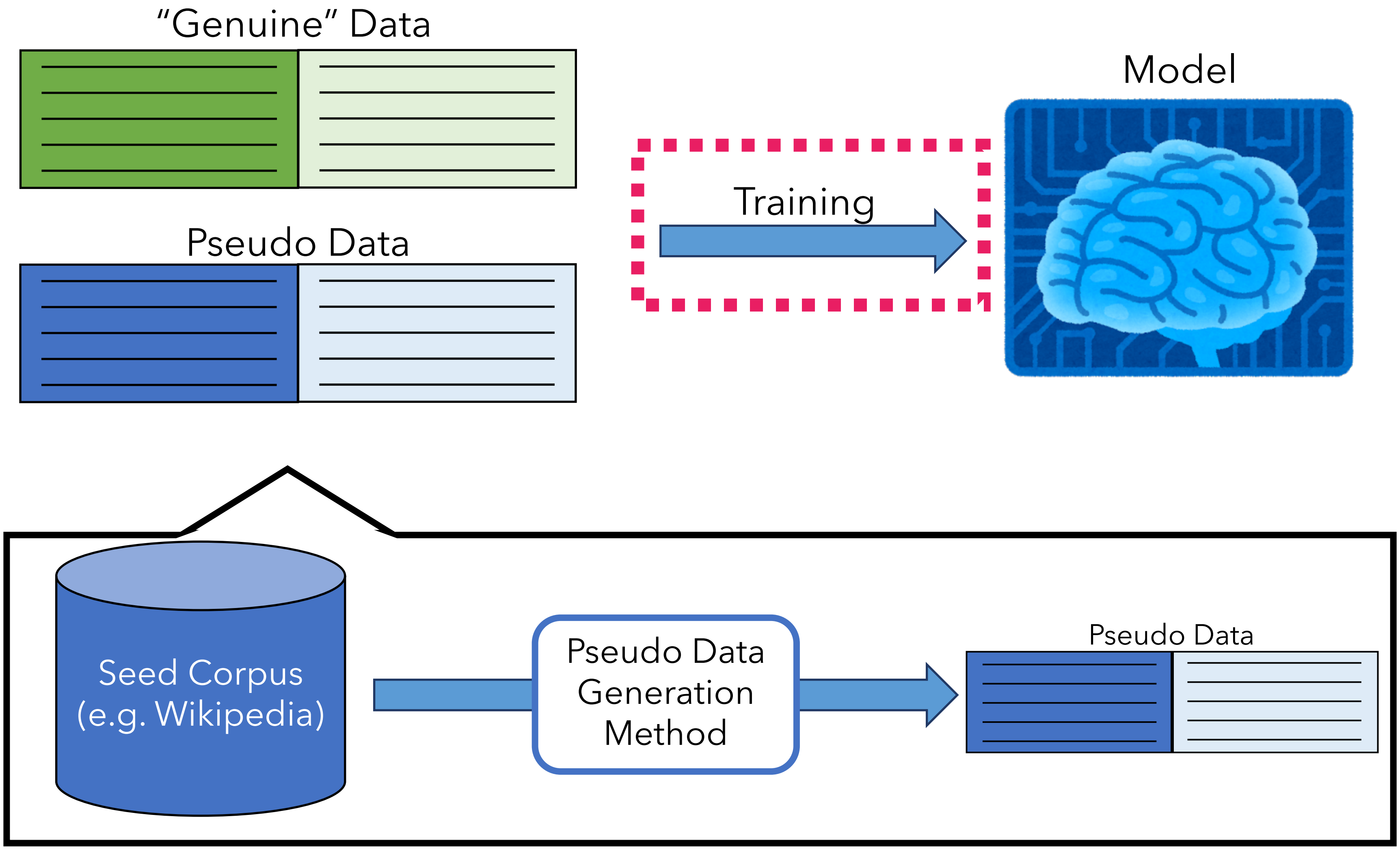
Q2: Methods for Generating Pseudo Data?

Original:	At the in@@ stitute , she introduced tis@@ sue culture methods that she had learned in the U.@@ S.
BACKTRANS (NOISY):	At in@@ stitute , She introduced tis@@ sue culture method that she learned in U.@@ S.
DIRECTNOISE:	⟨*mask*⟩ the the ⟨*mask*⟩ ⟨*mask*⟩ ⟨*mask*⟩ ⟨*mask*⟩ tis@@ culture R@@ methods , she P ⟨*mask*⟩ the s U.@@ ⟨*mask*⟩

We compare two methods

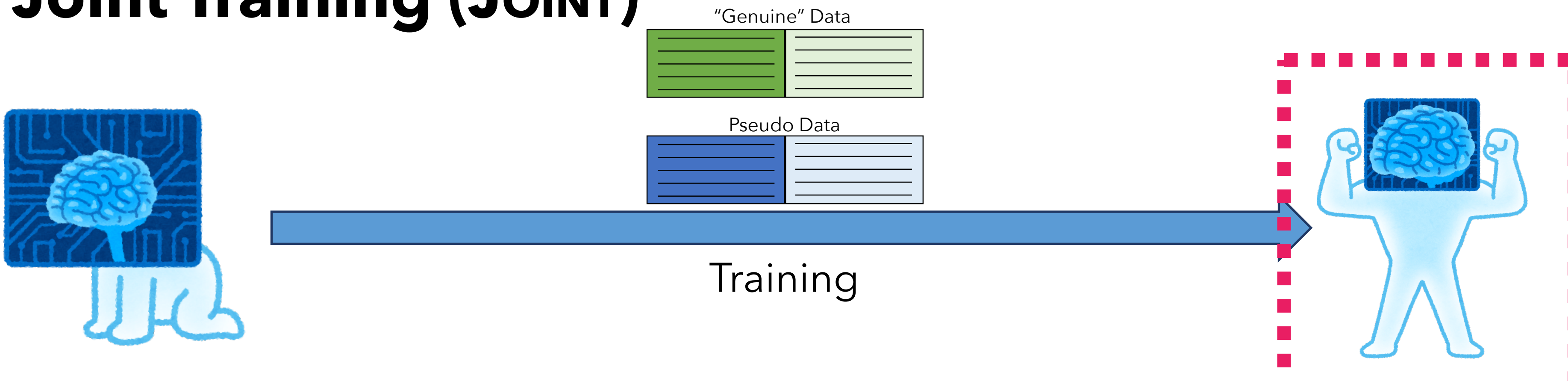
- BACKTRANS (NOISY) [Xie+2018]
  - Data is generated by back-translation
- DIRECTNOISE [Zhao+2019]
  - Data is generated by adding synthetic noise
- Please read our paper for details

Q3: How to use pseudo data during training?



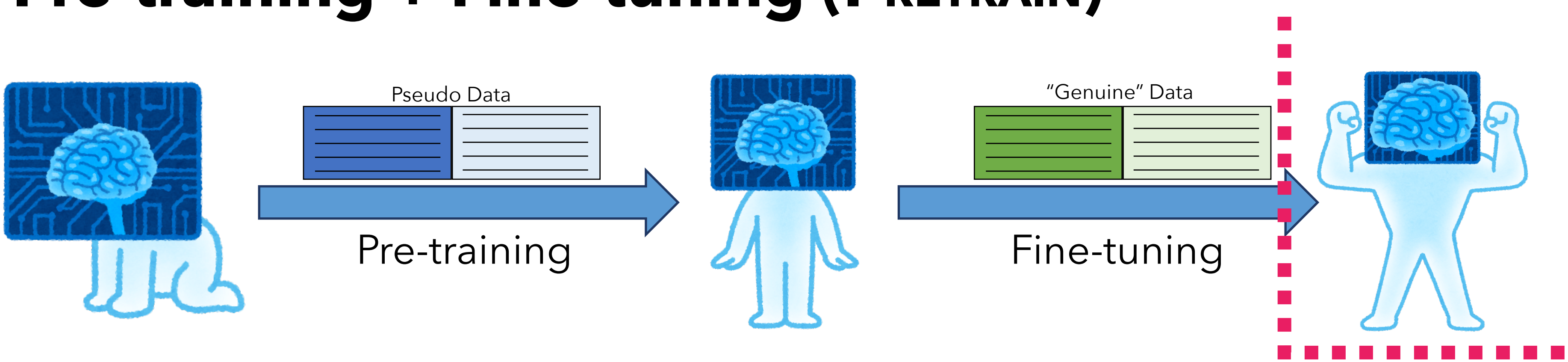
Q3: How to use pseudo data during training?

Joint Training (JOINT)



Which one performs better?

Pre-training + Fine-tuning (PRETRAIN)



Experimental Configuration and Datasets

- We adapt “standard” configurations
  - Model: Transformer (Big) [Vaswani+2017]
  - Optimizer: Adam (for pretrain) and Adafactor (for fine-tuning)
- Dataset
  - BEA-2019 dataset (train/valid/test) [Bryant+2019]
  - CoNLL2014 (test) [Ng+2014]

Experiment 1: Choice for Seed Corpus

- Settings: JOINT

Method	Seed Corpus $\mathcal{T}$	Prec.	Rec.	$F_{0.5}$
Baseline	N/A	46.6	23.1	38.8

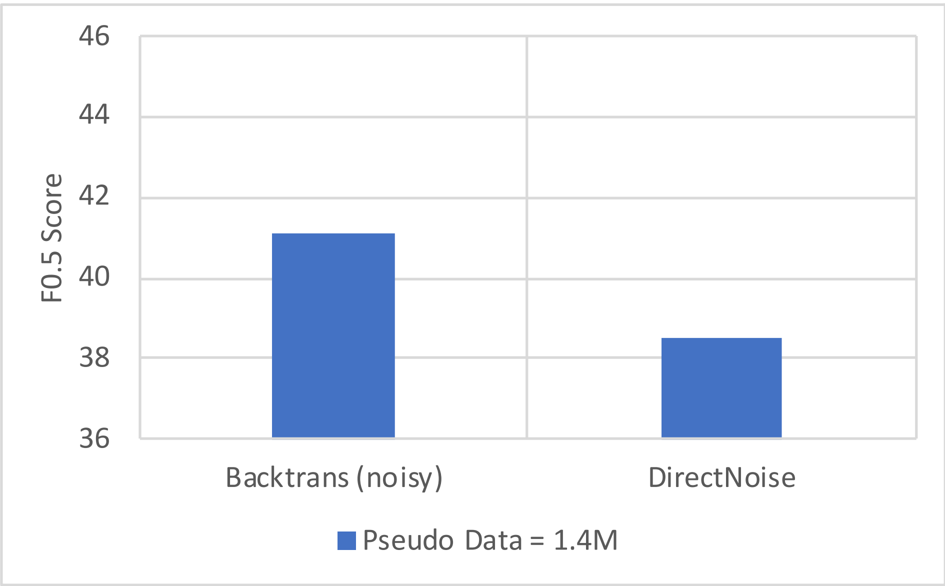
A1: Use Gigaword

- Seed corpus has **minor influence on $F_{0.5}$ score**
- **Gigaword** is an ideal option
  - **clean text** is more important than **domain**?

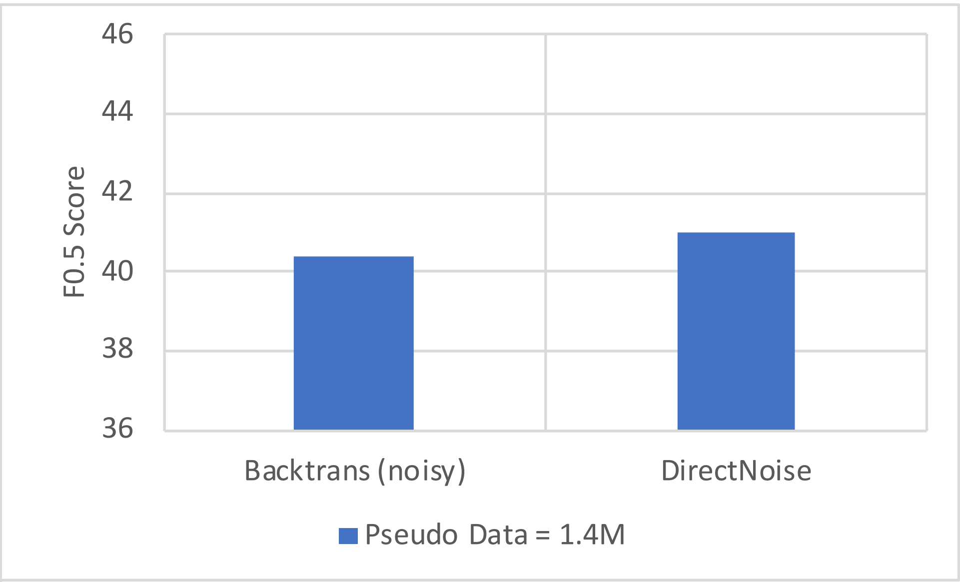
Experiment 2: Utilization of Pseudo Data

- Settings: Wikipedia as seed corpus

PRETRAIN



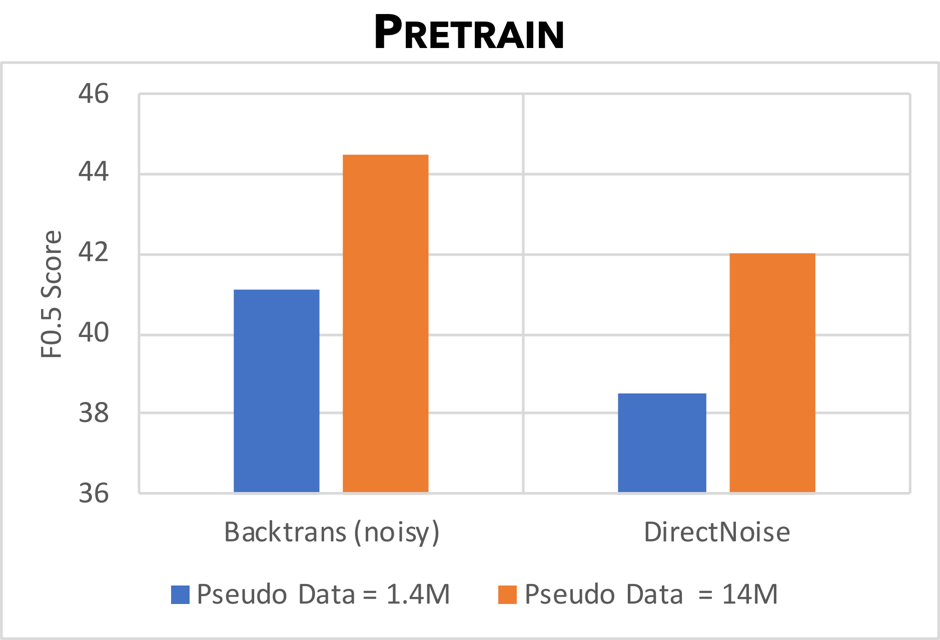
JOINT

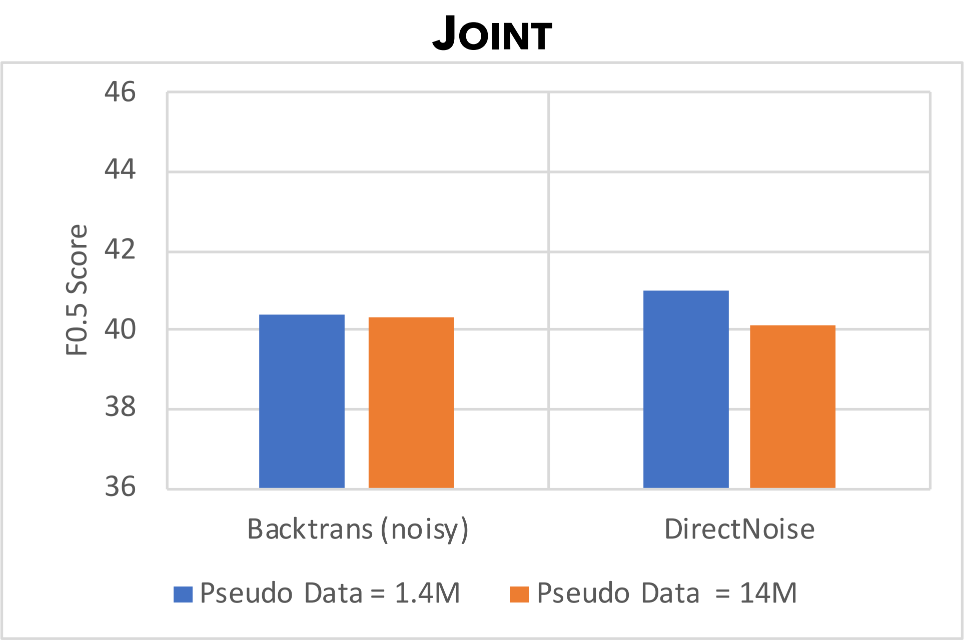


- If amount of pseudo data ≒ genuine data
  → PRETRAIN and JOINT are competitive

Experiment 2: Utilization of Pseudo Data

- Settings: Wikipedia as seed corpus



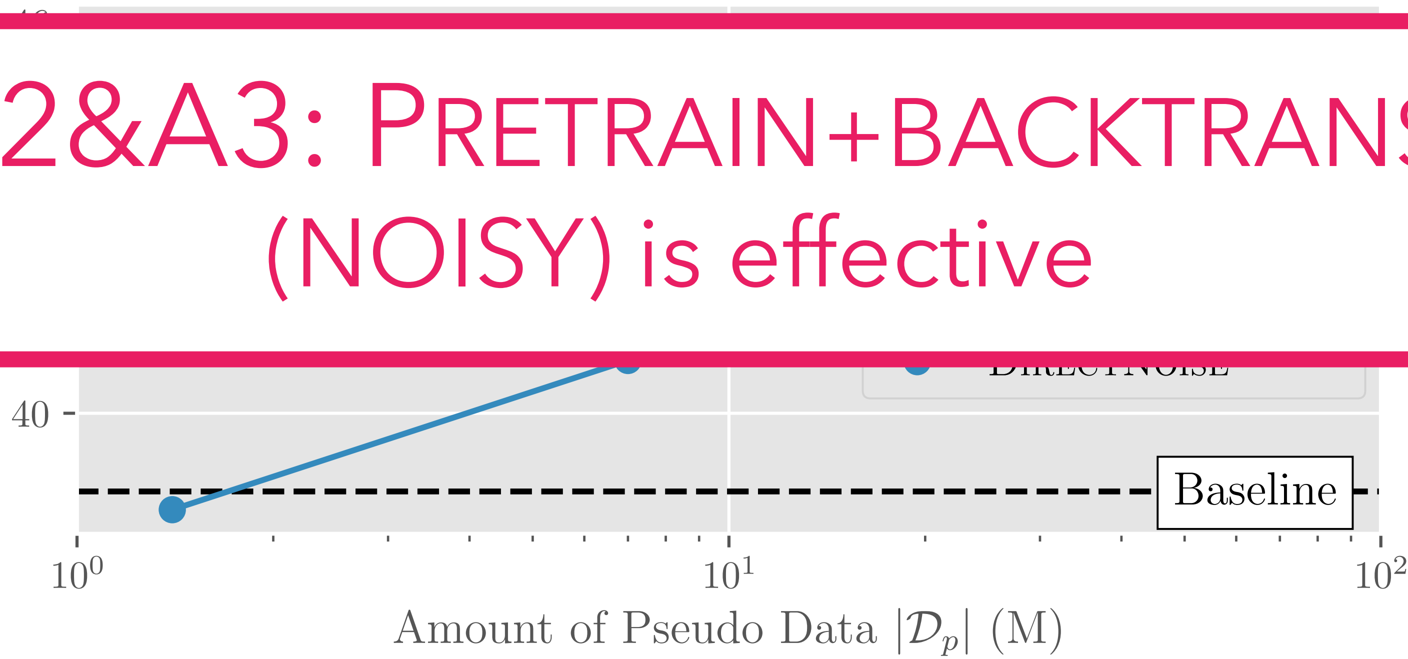


- Increasing amount of pseudo data improves the performance in PRETRAIN
- performance does not improve in JOINT
  - Pseudo data becomes dominant in JOINT

Experiment 3: More Pseudo Data

- Backtrans (noisy) significantly outperforms DirectNoise

A2&A3: Pretrain+Backtrans (noisy) is effective



Summary

A1: Use Gigaword

A2&A3: PRETRAIN+BACKTRANS (NOISY) is effective

LARGEPRETRAIN

Comparison to Current Top Models

Model	Ensemble	CoNLL-2014 (M^2 scorer) Prec.	CoNLL-2014 (M^2 scorer) Rec.	CoNLL-2014 (M^2 scorer) $F_{0.5}$	BEA-test (ERRANT) Prec.	BEA-test (ERRANT) Rec.	BEA-test (ERRANT) $F_{0.5}$
Chollampatt and Ng (2018)	✓	65.5	33.1	54.8	-	-	-
Junczys-Dowmunt et al. (2018)	✓	61.9	40.2	55.8	-	-	-
Lichtarge et al. (2019)	[illegible]	[illegible]	[illegible]	60.4	-	-	-
Zhao et al. (2019)	✓	71.6	38.7	61.2	-	-	-
Grundkiewicz et al. (2019)	✓	-	-	64.2	72.3	60.1	69.5
LargePretrain		67.9	44.1	61.3	65.5	59.4	64.2
LargePretrain+SSE+R2L	[illegible]	[illegible]	[illegible]	**65.0**	72.1	**61.8**	69.8
LargePretrain+SSE+R2L+SED	✓	**73.3**	44.2	64.7	**74.7**	56.7	**70.2**

Previously Published Results

LargePretrain Results

(1) Strong Single Model Results

Model	Ensemble	CoNLL-2014 (M^2 scorer) Prec.	Rec.	$F_{0.5}$	BEA-test (ERRANT) Prec.	Rec.	$F_{0.5}$
Chollampatt and Ng (2018)	✓	65.5	33.1	54.8	-	-	-
Junczys-Dowmunt et al. (2018)	✓	61.9	40.2	55.8	-	-	-
Lichtarge et al. (2019)	✓	66.7	43.9	60.4	-	-	-
Zhao et al. (2019)	✓	71.6	38.7	61.2	-	-	-
Grundkiewicz et al. (2019)	✓	-	-	64.2	72.3	60.1	69.5
LargePretrain		67.9	44.1	61.3	65.5	59.4	64.2
LargePretrain+SSE+R2L	✓	72.4	**46.1**	**65.0**	72.1	**61.8**	69.8
LargePretrain+SSE+R2L+SED	✓	**73.3**	44.2	64.7	**74.7**	56.7	**70.2**

Our best **single model** outperforms the **other ensemble models** except for [Grundkiewicz+2019]

(2) Additional Techniques Improve Result

Model	Ensemble	CoNLL-2014 (M^2 scorer) Prec.	Rec.	$F_{0.5}$	BEA-test (ERRANT) Prec.	Rec.	$F_{0.5}$
Chollampatt and Ng (2018)	✓	65.5	33.1	54.8	-	-	-
Junczys-Dowmunt et al. (2018)	✓	61.9	40.2	55.8	-	-	-
Lichtarge et al. (2019)	✓	66.7	43.9	60.4	-	-	-
Zhao et al. (2019)	✓	71.6	38.7	61.2	-	-	-
	✓	-	-	64.2	72.3	60.1	69.5
LargePr			44.1	61.3	65.5	59.4	64.2
LargePretrain+SSE+R2L	✓	72.4	**46.1**	**65.0**	72.1	**61.8**	69.8
LargePretrain+SSE+R2L+SED	✓	**73.3**	44.2	64.7	**74.7**	56.7	**70.2**

SSE: Synthetic Spelling Error

SED: Sentence-level Error Detection

R2L: Right-to-Left Reranking

Ensemble of 4 Models

Our best model achieves the best performance on **CoNLL2014 ($F_{0.5}$=65.0) and BEA-test ($F_{0.5}$=70.2)**

Conclusions

- Investigated 3 questions regarding incorporating pseudo data into GEC model.

Q1: Choice for seed corpus?
Q2: Methods for generating pseudo data?
Q3: How to use pseudo data during training?

- Discovered settings suitable (LARGEPRETRAIN)
  - justified by SOTA performance on benchmark datasets
- Code and pretrained model are available