https://arxiv.org/pdf/2203.13064.pdf

Paper review

Ensembling and Knowledge Distilling of Large Sequence Taggers for Grammatical Error Correction (ACL 2022)

Presentation: **Jeiyoon Park**
6^{th} Generation, TAVE

TAVE Research

Outline

1. Contribution
2. Method
3. Experiments
4. Conclusion

Outline

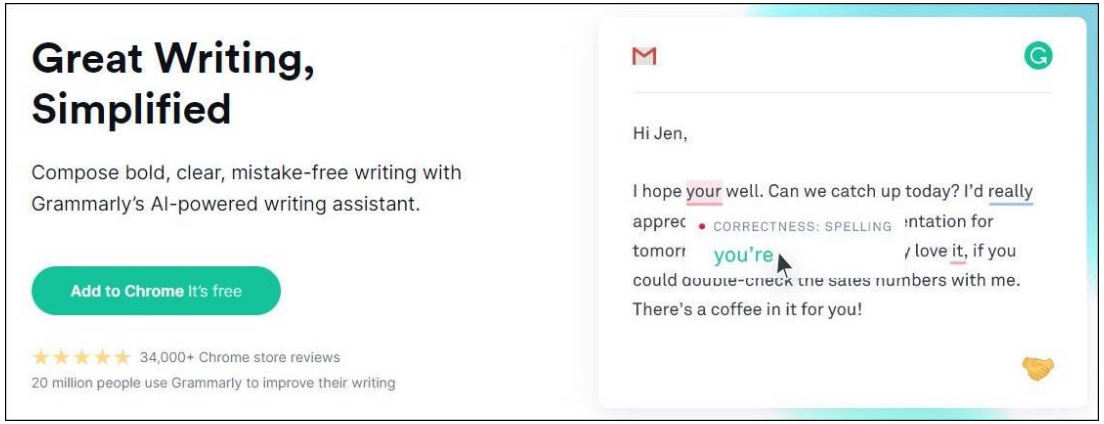
1. Contribution
2. Method
3. Experiments
4. Conclusion

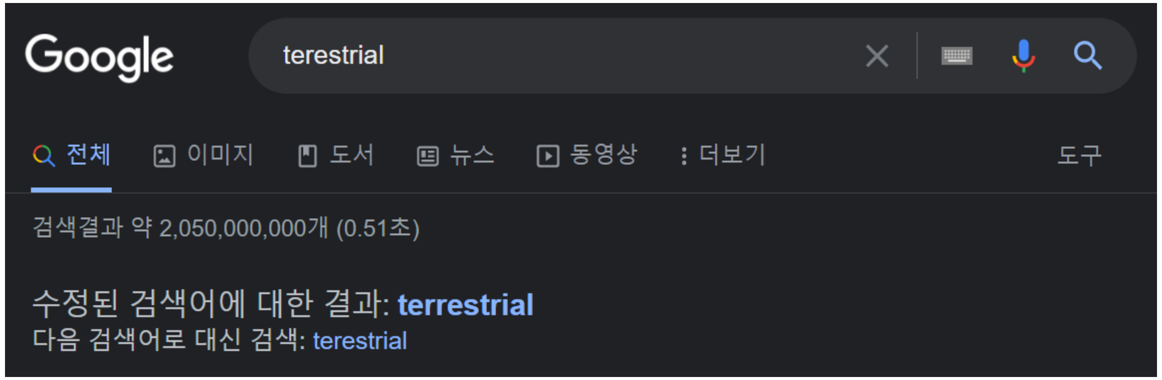
Detour: Grammatical Error Correction

1. Grammatical Error Correction (GEC)

24. 다음 글의 밑줄 친 부분 중, 어법상 틀린 것은? [3점]

In some communities, music and performance have successfully transformed whole neighborhoods as ① profoundly as The Guggenheim Museum did in Bilbao. In Salvador, Brazil, musician Carlinhos Brown established several music and culture centers in formerly dangerous neighborhoods. In Candeal, ② where Brown was born, local kids were encouraged to join drum groups, sing, and stage performances. The kids, energized by these activities, ③ began to turn away from dealing drugs. Being a young criminal was no longer their only life option. Being musicians and playing together in a group looked like more fun and was more ④ satisfying. Little by little, the crime rate dropped in those neighborhoods; the hope returned. In another slum area, possibly inspired by Brown's example, a culture center began to encourage the local kids to stage musical events, some of ⑤ them dramatized the tragedy that they were still recovering from.





Detour: Grammatical Error Correction

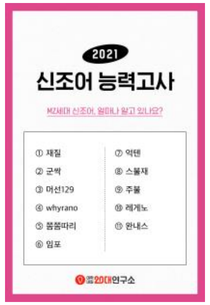
1. Grammatical Error Correction (GEC)

1) GEC is the task of fixing grammatical errors in text, such as typos, tense and article mistakes

2) Training a model for GEC requires a set of labeled *(ungrammatical / grammatical)* sentence pairs, which are expensive to obtain

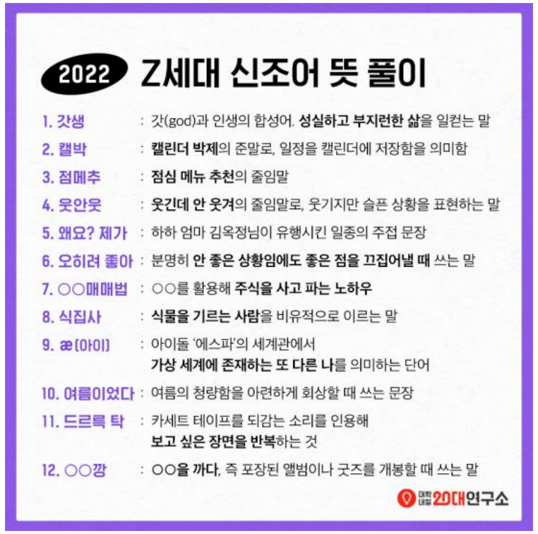
Iteration #	**Sentence's evolution**	**# corr.**
Orig. sent	A ten years old boy go school	-
Iteration 1	A ten**-**years old boy **goes** school	2
Iteration 2	A ten-**year-**old boy goes **to** school	5
Iteration 3	A ten-year-old boy goes to school**.**	6

Detour: Grammatical Error Correction



2. Challenges

1) Due to the unrestricted mutability of language, it is hard to design a model that is capable of correcting all possible errors made by non-native learners, especially when error patterns in new text are not observed in training data.



2022 Z세대 신조어 뜻 풀이

1. 갓생 : 갓(god)과 인생의 합성어. 성실하고 부지런한 삶을 일컫는 말
2. 캘박 : 캘린더 박제의 준말로, 일정을 캘린더에 저장함을 의미함
3. 점메추 : 점심 메뉴 추천의 줄임말
4. 웃안웃 : 웃긴데 안 웃겨의 줄임말로, 웃기지만 슬픈 상황을 표현하는 말
5. 왜요? 제가 : 하하 엄마 김옥정님이 유행시킨 일종의 주접 문장
6. 오히려 좋아 : 분명히 안 좋은 상황임에도 좋은 점을 끄집어낼 때 쓰는 말
7. ○○매매법 : ○○를 활용해 주식을 사고 파는 노하우
8. 식집사 : 식물을 기르는 사람을 비유적으로 이르는 말
9. æ(아이) : 아이돌 '에스파'의 세계관에서 가상 세계에 존재하는 또 다른 나를 의미하는 단어
10. 여름이었다 : 여름의 청량함을 아련하게 회상할 때 쓰는 문장
11. 드르륵 탁 : 카세트 테이프를 되감는 소리를 인용해 보고 싶은 장면을 반복하는 것
12. ○○깡 : ○○을 까다, 즉 포장된 앨범이나 굿즈를 개봉할 때 쓰는 말

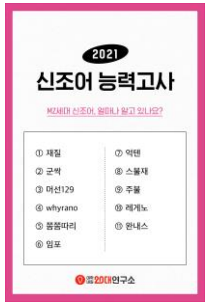
20대연구소

2) Unlike machine translation, a large amount of annotated ungrammatical texts and their corrected counterparts are not available.

3) The artificially generated data cannot precisely capture the error distribution in real erroneous data.

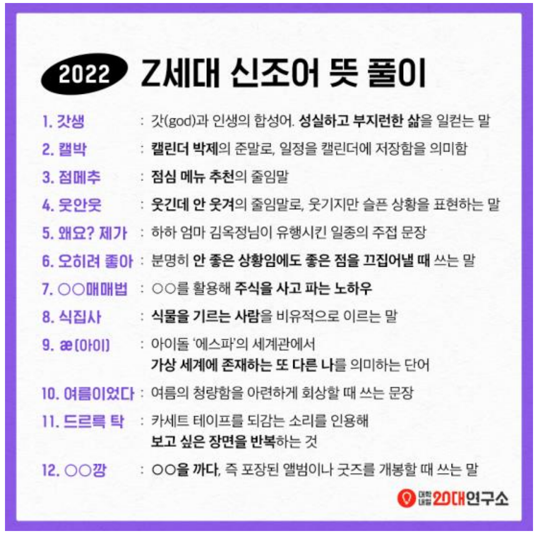
e.g.) We don't use *"a am I boy"* (*"I am a boy"*)

Detour: Grammatical Error Correction



2. Challenges

1) Due to the unrestricted mutability of language, it is hard to design a model that is capable of correcting all possible errors made by non-native learners, especially when error patterns in new text are not observed in training data.



2022 Z세대 신조어 뜻 풀이

1. 갓생 : 갓(god)과 인생의 합성어. **성실하고 부지런한 삶**을 일컫는 말
2. 캘박 : **캘린더 박제**의 준말로, 일정을 캘린더에 저장함을 의미함
3. 점메추 : **점심 메뉴 추천**의 줄임말
4. 웃안웃 : **웃긴데 안 웃겨**의 줄임말로, 웃기지만 슬픈 상황을 표현하는 말
5. 왜요? 제가 : 하하 엄마 김옥정님이 유행시킨 일종의 주접 문장
6. 오히려 좋아 : 분명히 **안 좋은 상황임에도 좋은 점을 끄집어낼 때** 쓰는 말
7. ○○매매법 : ○○를 활용해 **주식을 사고 파는 노하우**
8. 식집사 : **식물을 기르는 사람**을 비유적으로 이르는 말
9. æ(아이) : 아이돌 '에스파'의 세계관에서 **가상 세계에 존재하는 또 다른 나**를 의미하는 단어
10. 여름이었다 : 여름의 청량함을 아련하게 회상할 때 쓰는 문장
11. 드르륵 탁 : 카세트 테이프를 되감는 소리를 인용해 **보고 싶은 장면을 반복**하는 것
12. ○○깡 : **○○을 까다**, 즉 포장된 앨범이나 굿즈를 개봉할 때 쓰는 말

20대연구소

2) Unlike machine translation, a large amount of annotated ungrammatical texts and their corrected counterparts are not available.

3) The artificially generated data cannot precisely capture the error distribution in real erroneous data.

e.g.) We don't use *"a am I boy"* (*"I am a boy"*)

Contribution: Why This paper

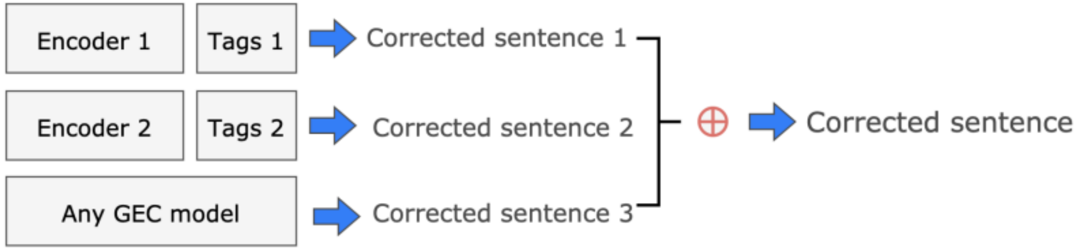
1. Ensembling of recent cutting-edge transformer-based models

2. Knowledge distillation method to produce annotated data

3. When trained on the distilled data, GEC models show competitive performance

4. Code and datasets are available

Ensembling by majority votes on output edit spans



Outline

1. Contribution
2. Method
3. Experiments
4. Conclusion

https://arxiv.org/pdf/2005.12592.pdf

Detour: GECToR

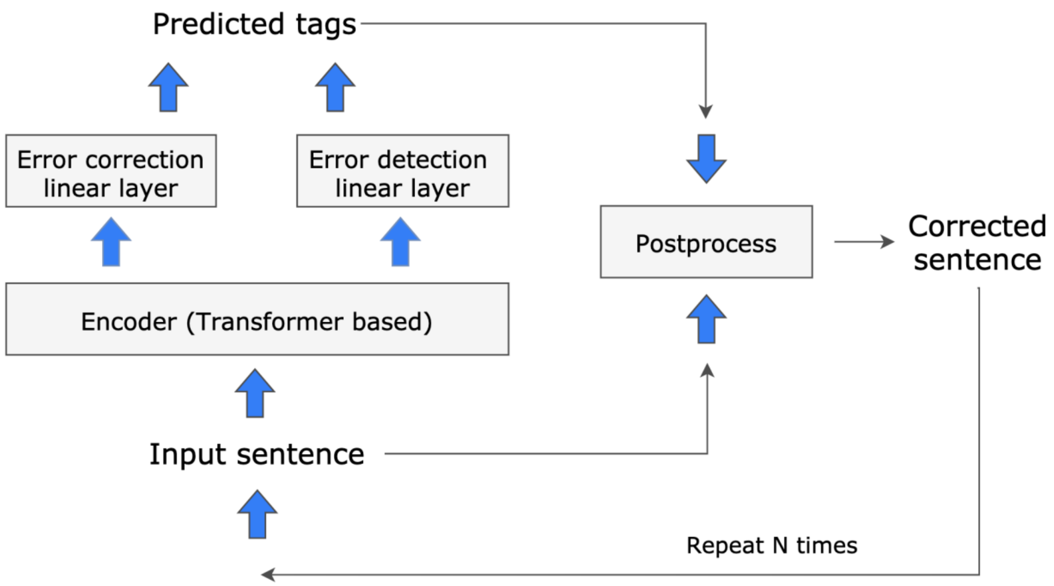
1. Overview

(1) GECToR?: Sequence tagging model using transformer encoder

- GECToR approaches the GEC task as a sequence tagging problem

(2) Model architecture

- Transformer-based encoder
- Two output linear layers
- A cross-entropy loss function
- Iterative postprocessing is performed



Iteration #	Sentence's evolution	# corr.
Orig. sent	A ten years old boy go school	-
Iteration 1	A ten**-**years old boy **goes** school	2
Iteration 2	A ten-**year-**old boy goes **to** school	5
Iteration 3	A ten-year-old boy goes to school**.**	6

https://arxiv.org/pdf/2005.12592.pdf

Detour: GECToR

2. Token-level transformations

(1) Basic transformations

- *$KEEP*
- *$DELETE*
- *$APPEND*
- *$REPLACE*

(2) g-transformations

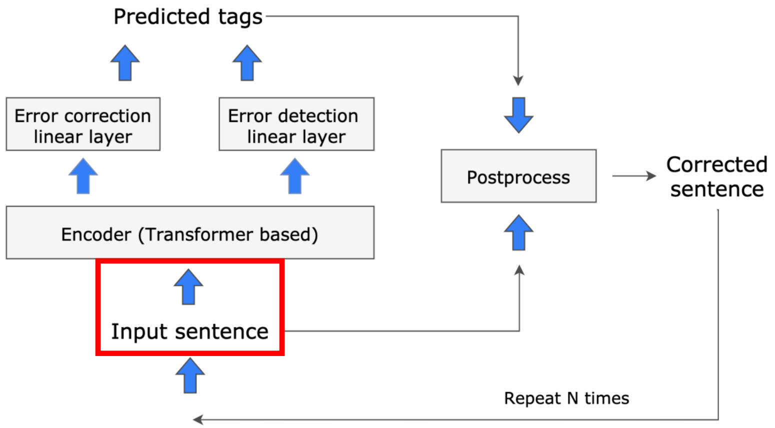
- *$CASE:* Change the case of the current token
- *$MERGE:* Merge the current token and the next token into a single one
- *$SPLIT: Split the current token into two new tokens*
- *$NOUN NUMBER*
- *$VERB FORM*

id	Core transformation	Transformation suffix	Tag	Example
basic-1	KEEP	∅	$KEEP	... many people want to travel during the summer ...
basic-2	DELETE	∅	$DELETE	... not sure if you are {**you** ⇒ ∅} gifting ...
basic-3	REPLACE	a	$REPLACE_a	... the bride wears {**the** ⇒ **a**} white dress ...
...	...	...	...	...
basic-3804	REPLACE	cause	$REPLACE_cause	... hope it does not {**make** ⇒ **cause**} any trouble ...
basic-3805	APPEND	for	$APPEND_for	... he is {**waiting** ⇒ **waiting for**} your reply ...
...	...	...	...	...
basic-4971	APPEND	know	$APPEND_know	... I {**don't** ⇒ **don't know**} which to choose...
g-1	CASE	CAPITAL	$CASE_CAPITAL	... surveillance is on the {**internet** ⇒ **Internet**} ...
g-2	CASE	CAPITAL_1	$CASE_CAPITAL_1	... I want to buy an {**iphone** ⇒ **iPhone**} ...
g-3	CASE	LOWER	$CASE_LOWER	... advancement in {**Medical** ⇒ **medical**} technology ...
g-4	CASE	UPPER	$CASE_UPPER	... the {**it** ⇒ **IT**} department is concerned that...
g-5	MERGE	SPACE	$MERGE_SPACE	... insert a special kind of gene {**in to** ⇒ **into**} the cell ...
g-6	MERGE	HYPHEN	$MERGE_HYPHEN	... and needs {**in depth** ⇒ **in-depth**} search ...
g-7	SPLIT	HYPHEN	$SPLIT_HYPHEN	... support us for a {**long-run** ⇒ **long run**} ...
g-8	NOUN_NUMBER	SINGULAR	$NOUN_NUMBER_SINGULAR	... a place to live for their {**citizen** ⇒ **citizens**}
g-9	NOUN_NUMBER	PLURAL	$NOUN_NUMBER_PLURAL	... carrier of this {**diseases** ⇒ **disease**} ...
g-10	VERB FORM	VB_VBZ	$VERB_FORM_VB_VBZ	... going through this {**make** ⇒ **makes**} me feel ...
g-11	VERB FORM	VB_VBN	$VERB_FORM_VB_VBN	... to discuss what {**happen** ⇒ **happened**} in fall ...
g-12	VERB FORM	VB_VBD	$VERB_FORM_VB_VBD	... she sighed and {**draw** ⇒ **drew**} her ...
g-13	VERB FORM	VB_VBG	$VERB_FORM_VB_VBG	... shown success in {**prevent** ⇒ **preventing**} such ...
g-14	VERB FORM	VB_VBZ	$VERB_FORM_VB_VBZ	... a small percentage of people {**goes** ⇒ **go**} by bike ...
g-15	VERB FORM	VBZ_VBN	$VERB_FORM_VBZ_VBN	... development has {**pushes** ⇒ **pushed**} countries to ...
g-16	VERB FORM	VBZ_VBD	$VERB_FORM_VBZ_VBD	... he {**drinks** ⇒ **drank**} a lot of beer last night ...
g-17	VERB FORM	VBZ_VBG	$VERB_FORM_VBZ_VBG	... couldn't stop {**thinks** ⇒ **thinking**} about it ...
g-18	VERB FORM	VBN_VB	$VERB_FORM_VBN_VB	... going to {**depended** ⇒ **depend**} on who is hiring ...
g-19	VERB FORM	VBN_VBZ	$VERB_FORM_VBN_VBZ	... yet he goes and {**eaten** ⇒ **eats**} more melons ...
g-20	VERB FORM	VBN_VBD	$VERB_FORM_VBN_VBD	... he {**driven** ⇒ **drove**} to the bus stop and ...
g-21	VERB FORM	VBN_VBG	$VERB_FORM_VBN_VBG	... don't want you fainting and {**broken** ⇒ **breaking**} ...
g-22	VERB FORM	VBD_VB	$VERB_FORM_VBD_VB	... each of these items will {**fell** ⇒ **fall**} in price ...
g-23	VERB FORM	VBD_VBZ	$VERB_FORM_VBD_VBZ	... the lake {**froze** ⇒ **freezes**} every year ...
g-24	VERB FORM	VBD_VBN	$VERB_FORM_VBD_VBN	... he has been went {**went** ⇒ **gone**} since last week ...
g-25	VERB FORM	VBD_VBG	$VERB_FORM_VBD_VBG	... talked her into {**gave** ⇒ **giving**} me the whole day ...
g-26	VERB FORM	VBG_VB	$VERB_FORM_VBG_VB	... free time, I just {**enjoying** ⇒ **enjoy**} being outdoors ...
g-27	VERB FORM	VBG_VBZ	$VERB_FORM_VBG_VBZ	... there still {**existing** ⇒ **exists**} many inevitable factors ...
g-28	VERB FORM	VBG_VBN	$VERB_FORM_VBG_VBN	... people are afraid of being {**tracking** ⇒ **tracked**} ...
g-29	VERB FORM	VBG_VBD	$VERB_FORM_VBG_VBD	... there was no {**mistook** ⇒ **mistaking**} his sincerity ...

Table 9: List of token-level transformations (section 3). We denote a tag which defines a token-level transformation as concatenation of two parts: a *core transformation* and a *transformation suffix*.

https://arxiv.org/pdf/2005.12592.pdf

Detour: GECToR



2. Token-level transformations

(3) Preprocessing

Step 1) Map each token from source sentence to subsequence of tokens from target sentence:

[A $\mapsto$ A], [ten $\mapsto$ ten, -], [years $\mapsto$ year, -],

[old $\mapsto$ old], [go $\mapsto$ goes, to], [school $\mapsto$ school, .

- GECToR searches for best-fitting subsequence by minimizing Levenshtein distance (e.g. “process” → “profess” → “professo” → “professor”: Three Levenshtein dist.)

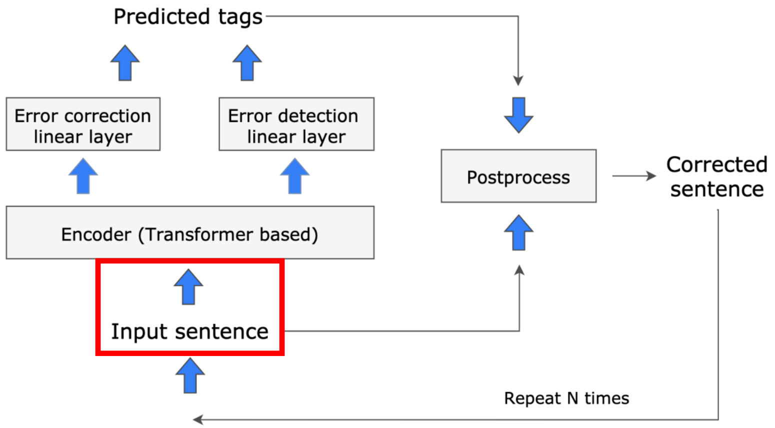
https://arxiv.org/pdf/2005.12592.pdf

Detour: GECToR

2. Token-level transformations

(3) Preprocessing

Step 2) Find token-level transformations which convert source to target subsequence:

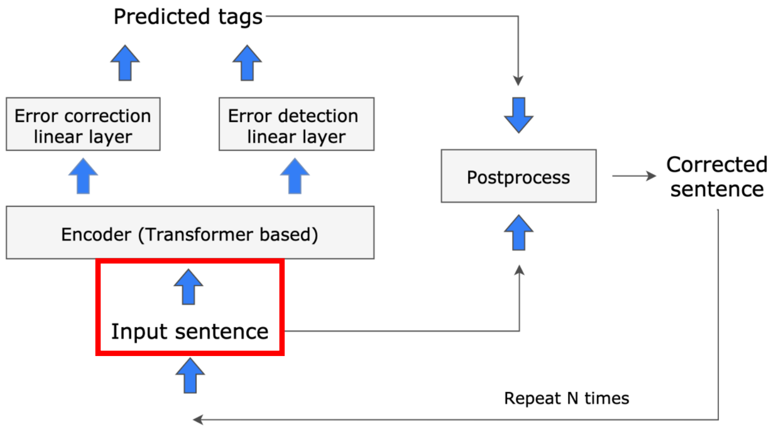


[A $\mapsto$ A]: \$KEEP, [ten $\mapsto$ ten, -]: \$KEEP, \$MERGE_HYPHEN,
[years $\mapsto$ year, -]: \$NOUN_NUMBER_SINGULAR, \$MERGE_HYPHEN],
[old $\mapsto$ old]: \$KEEP, [go $\mapsto$ goes, to]: \$VERB_FORM_VB_VBZ, \$AP- PEND_to,
[school $\mapsto$ school, .]: \$KEEP, \$AP- PEND_{.}].

Detour: GECToR

2. Token-level transformations

(3) Preprocessing



Step 3) Leave only one transformation for each Source token:

A ⇔ $KEEP,
ten ⇔ $MERGE_HYPHEN,
years ⇔ $NOUN_NUMBER_SINGULAR,
old ⇔ $KEEP,
school ⇔ $APPEND_{.}.

Iteration #	Sentence's evolution	# corr.
Orig. sent	A ten years old boy go school	-
Iteration 1	A ten-years old boy **goes** school	2
Iteration 2	A ten-**year**-old boy goes **to** school	5
Iteration 3	A ten-year-old boy goes to school**.**	6

- A single tag for each token
- In case of multiple transformations GECToR takes the first transformation, except *$KEEP*.

Detour: GECToR

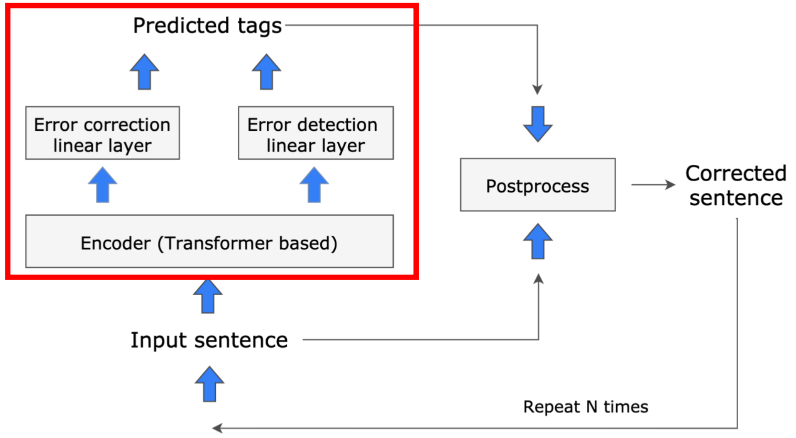
3. Tagging model architecture

(1) An encoder made up of pretrained transformer

(2) With two linear layers

- Error detection linear layer
- Error correction (a.k.a., error tagging) linear layer

(3) Softmax layers



A ⇔ $KEEP,
ten ⇔ $MERGE_HYPHEN,
years ⇔ $NOUN_NUMBER_SINGULAR,
old ⇔ $KEEP,
school ⇔ $APPEND_{.}.

Outline

1. Contribution
2. Method
3. Experiments
4. Conclusion

Experiments

1. Datasets

- Annotated data:
  - Lang-8
  - NUCLE
  - FCE
  - W&I
- Monolingual data, Distilled data:
  - 1BW
  - Blogs
  - starts with “Troy-”
- Synthetic data:
  - PIE (9M parallel sentences)

Dataset	Type	Part	# Sent.	# Tokens	% Edits
Lang-8*	Ann	Train*	1.04M	11.86M	42%
NUCLE*	Ann	Train*	57k	1.16M	62%
FCE*	Ann	Train*	28k	455k	62%
W&I*†	Ann	Train*	34.3k	628.7k	67%
		Dev	3.4k	63.9k	69%
		Test†	3.5k	62.5k	N/A
LOCNESS†	Ann	Dev	1k	23.1k	52%
		Test†	1k	23.1k	N/A
1BW‡	Mon	N/A	115M	0.8B	N/A
Blogs‡	Mon	N/A	13.5M	171M	N/A
Troy-1BW	Dis	Train	1.2M	30.88M	100%
Troy-Blogs	Dis	Train	1.2M	21.49M	100%
PIE‡	Syn	Train	1.2M	30.1M	100%

Table 1: Description and statistics of datasets used in this work. Dataset types: (Ann)otated, (Syn)thetic, (Mon)olingual, and (Dis)tilled. *Combined, these datasets form the *Joint Train Dataset*. †BEA-2019 dev/test parts are concatenations of W&I and LOCNESS dev/test parts. ‡Only parts of the original corpora from the cited sources are used in our work.

Dataset	# sentences	% errorful sentences	Training stage
PIE-synthetic	9,000,000	100.0%	I
Lang-8	947,344	52.5%	II
NUCLE	56,958	38.0%	II
FCE	34,490	62.4%	II
W&I+LOCNESS	34,304	67.3%	II, III

Table 1: Training datasets. Training stage I is pretraining on synthetic data. Training stages II and III are for fine-tuning.

Experiments

2. Evaluation

- ERRANT
    - $F_{0.5}, precision, recall$

- On dev and test datasets from W&I + LOCNESS Corpus

3. Tokenization

- AllenNLP’s → Too slow
- HuggingFace Transformers’ → Not provide a BPE-to-words mapping
- ✔ SentencePiece

Experiments

Dataset	# sentences	% errorful sentences	Training stage
PIE-synthetic	9,000,000	100.0%	I
Lang-8	947,344	52.5%	II
NUCLE	56,958	38.0%	II
FCE	34,490	62.4%	II
W&I+LOCNESS	34,304	67.3%	II, III

Table 1: Training datasets. Training stage I is pretraining on synthetic data. Training stages II and III are for fine-tuning.

4. Training stages

1) StageI (optional): The model is pretrained on synthetic datasets
2) StageII: Carry out warm-up training on the *Joint Train Dataset* (Lang-8 + NUCLE + FCE + W&I)
3) StageIII: Fine-tuning on the W&I dataset

- During the first two epochs they train only the linear layers (so-called "cold epochs"); later make all model's weights trainable

- Too many sentences without edits lead to reducing the appearance rate of the tagger
- StageII: Filter out edit-free sentences
- StageIII: Unfiltered version of W&I

Dataset	Type	Part	# Sent.	# Tokens	% Edits
Lang-8*	Ann	Train*	1.04M	11.86M	42%
NUCLE*	Ann	Train*	57k	1.16M	62%
FCE*	Ann	Train*	28k	455k	62%
W&I*†	Ann	Train*	34.3k	628.7k	67%
		Dev	3.4k	63.9k	69%
		Test†	3.5k	62.5k	N/A
LOCNESS†	Ann	Dev	1k	23.1k	52%
		Test†	1k	23.1k	N/A
1BW‡	Mon	N/A	115M	0.8B	N/A
Blogs‡	Mon	N/A	13.5M	171M	N/A
Troy-1BW	Dis	Train	1.2M	30.88M	100%
Troy-Blogs	Dis	Train	1.2M	21.49M	100%
PIE‡	Syn	Train	1.2M	30.1M	100%

Table 1: Description and statistics of datasets used in this work. Dataset types: (Ann)otated, (Syn)thetic, (Mon)olingual, and (Dis)tilled. *Combined, these datasets form the *Joint Train Dataset*. †BEA-2019 dev/test parts are concatenations of W&I and LOCNESS dev/test parts. ‡Only parts of the original corpora from the cited sources are used in our work.

Experiments

4. Training stages

1) StageI (optional): The model is pretrained on synthetic datasets
2) StageII: Carry out warm-up training on the *Joint Train Dataset* (Lang-8 + NUCLE + FCE + W&I)
3) StageIII: Fine-tuning on the W&I dataset

- Inference tweaks: Introducing additional hyperparameters for balancing between the precision and recall
(Additional confidence to the probability for the *$KEEP* tag and minimum error probability for correction tags)

Training stage #	**Base**			**Large**		
	P	**R**	**$F_{0.5}$**	**P**	**R**	**$F_{0.5}$**
Stage I.	N/A	N/A	N/A	N/A	N/A	N/A
Stage II.	50.12	34.04	45.79	52.11	37.34	48.29
Stage III.	53.77	**39.23**	50.06	54.85	**42.54**	51.85
Inf. tweaks	**62.49**	32.26	**52.63**	**65.76**	33.86	**55.33**

Table 2: Performance of our system with a RoBERTa encoder (in Base and Large configurations) after each training stage and inference tweaks on BEA-2019 (dev). Pre-training on synthetic data (Stage I) as was done in (Omelianchuk et al., 2020) is not performed.

Experiments

5. Upgrading to Large encoders

- Most likely, Base configurations were chosen due to the better inference speed/quality ratio.

- XLNet, RoBERTa, and BERT show best performance

Encoder	Base			Large		
	P	R	$F_{0.5}$	P	R	$F_{0.5}$
BERT	57.21	29.93	48.39	61.18	31.26	51.35
DeBERTa	**64.22**	31.87	**53.38**	**66.35**	32.77	55.07
RoBERTa	62.49	**32.26**	52.63	65.76	33.86	**55.33**
XLNet	63.16	30.59	52.07	64.27	**35.17**	55.14

Table 3: Performance of our single system on BEA-2019 (dev) for different encoders from pretrained Transformers in Base and Large configurations.

Encoder	Time (sec)		# Params	
	Base	Large	Base	Large
BERT	19.28	49.17	120M	350M
DeBERTa	23.75	58.32	150M	410M
RoBERTa	19.05	45.66	129M	360M
XLNet	30.46	71.19	120M	345M

Table 4: Inference times and model sizes for our single tagging models. Inference time for NVIDIA Tesla P100 on BEA-2019 dev, single models, batch size=128, averaged over 5 inferences.

Experiments

6. Exploring tag vocabulary sizes

- Most of the tag-encoded edits are token-specific (e.g., *$APPEND_it*, and *$REPLACE_the*)
- Thus, the tag vocabulary size matters

Encoder	P	R	$F_{0.5}$
$\text{DeBERTa}_{5K}^{(L)}$	**66.35**	32.77	55.07
$\text{RoBERTa}_{5K}^{(L)}$	65.76	33.86	**55.33**
$\text{XLNet}_{5K}^{(L)}$	64.27	**35.17**	55.14
$\text{DeBERTa}_{10K}^{(L)}$	**65.46**	34.59	55.55
$\text{RoBERTa}_{10K}^{(L)}$	64.72	**36.04**	**55.83**
$\text{XLNet}_{10K}^{(L)}$	64.12	34.02	54.48

Table 5: Performance on BEA-2019 (dev) for varied tag vocabulary sizes and encoders in their (L)arge configurations. Subscripts encode the models' tag vocabulary sizes from the set (5K, 10K).

headcanon 😅

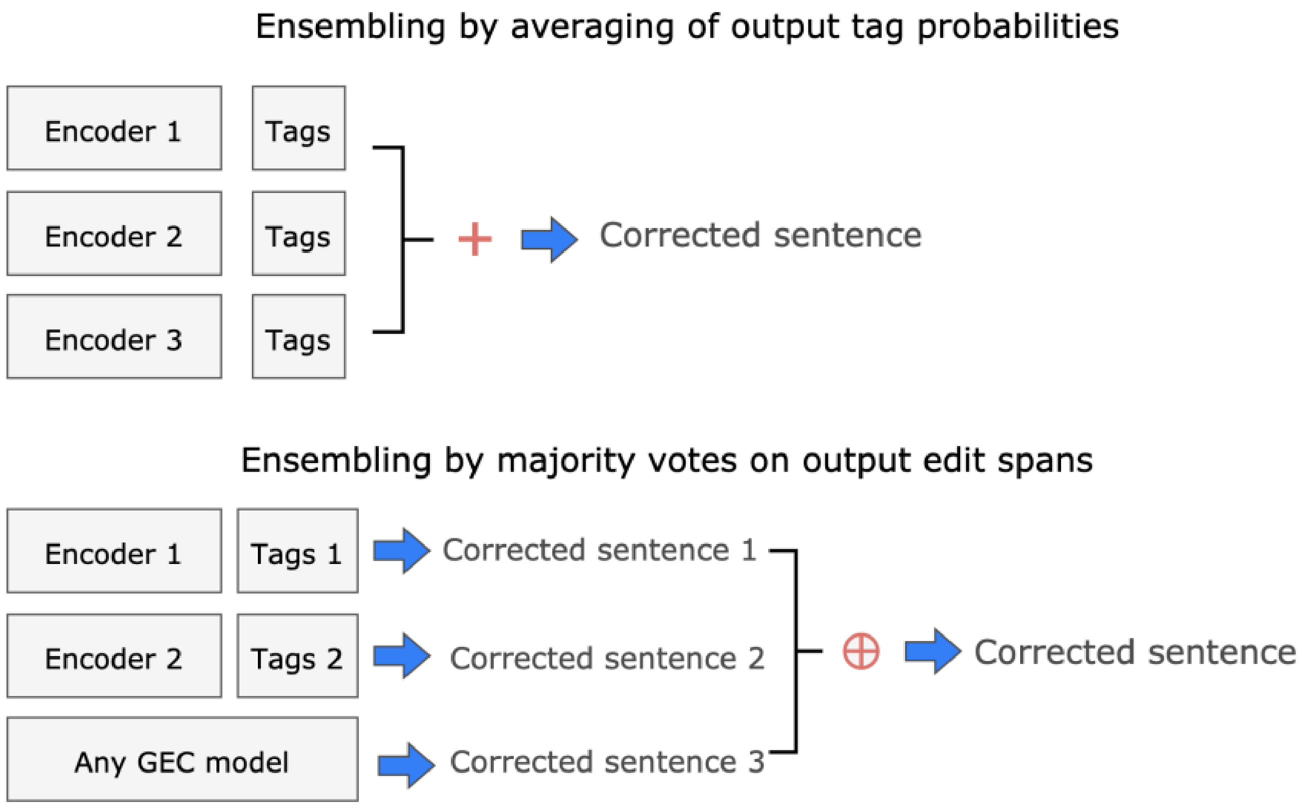
Encoder	Time (sec)		# Params	
	Base	Large	Base	Large
BERT	19.28	49.17	120M	350M
DeBERTa	23.75	58.32	150M	410M
RoBERTa	19.05	45.66	129M	360M
XLNet	30.46	71.19	120M	345M

Table 4: Inference times and model sizes for our single tagging models. Inference time for NVIDIA Tesla P100 on BEA-2019 dev, single models, batch size=128, averaged over 5 inferences.

Experiments

7. Ensembling the GEC taggers

- (1) Averaging of output tag probabilities vs. (2) majority votes on output edit spans



Experiments

7. Ensembling the GEC taggers

- (1) Averaging of output tag probabilities vs. (2) majority votes on output edit spans

	Ensemble	P	R	$F_{0.5}$
(B)	RoBERTa$^{(B)}$ + DeBERTa$^{(B)}$	53.44	**34.91**	48.31
	RoBERTa$^{(B)}$ + XLNet$^{(B)}$	53.45	34.3	48.08
	RoBERTa$^{(B)}$ + DeBERTa$^{(B)}$ + XLNet$^{(B)}$	54.78	34.87	49.17
	RoBERTa$^{(B)}$ + BERT$^{(B)}$ + DeBERTa$^{(B)}$ + + XLNet$^{(B)}$	**56.34**	33.76	**49.69**
(B+L)	RoBERTa$^{(B)}$	50.12	34.04	45.79
	RoBERTa$^{(L)}$	52.11	**37.34**	48.29
	RoBERTa$^{(B)}$ + RoBERTa$^{(L)}$	**54.83**	35.93	**49.61**
(L)	RoBERTa$^{(L)}$ + DeBERTa$^{(L)}$	54.12	39.77	50.48
	RoBERTa$^{(L)}$ + XLNet$^{(L)}$	53.83	38.65	49.91
	RoBERTa$^{(L)}$ + BERT$^{(L)}$ + DeBERTa$^{(L)}$	**57.31**	37.41	51.8
	RoBERTa$^{(L)}$ + DeBERTa$^{(L)}$ + XLNet$^{(L)}$	54.30	**39.95**	50.66
	RoBERTa$^{(L)}$ + BERT$^{(L)}$ + DeBERTa$^{(L)}$ + + XLNet$^{(L)}$	56.97	38.52	**51.99**

Table 6: Comparison of ensembles by averaging of output tag probabilities after Stage II for (B)ase and (L)arge encoders with a tag vocabulary size of 5K. Benchmark is BEA-2019 (dev).

Stage	Ensemble	P	R	$F_{0.5}$
St. I	RoBERTa$^{(L)}$ + DeBERTa$^{(L)}$ + XLNet$^{(L)}$	N/A	N/A	N/A
St. I	RoBERTa$^{(L)}$ $\oplus$ DeBERTa$^{(L)}$ $\oplus$ XLNet$^{(L)}$	N/A	N/A	N/A
St. II	RoBERTa$^{(L)}$ + DeBERTa$^{(L)}$ + XLNet$^{(L)}$	54.3	**39.95**	50.66
St. II	RoBERTa$^{(L)}$ $\oplus$ DeBERTa$^{(L)}$ $\oplus$ XLNet$^{(L)}$	**56.74**	38.53	**51.84**
St. III	RoBERTa$^{(L)}$ + DeBERTa$^{(L)}$ + XLNet$^{(L)}$	58.08	**43.17**	54.33
St. III	RoBERTa$^{(L)}$ $\oplus$ DeBERTa$^{(L)}$ $\oplus$ XLNet$^{(L)}$	**60.58**	41.92	**55.63**
In.tw.	RoBERTa$^{(L)}$ + DeBERTa$^{(L)}$ + XLNet$^{(L)}$	68.45	**35.56**	57.76
In.tw.	RoBERTa$^{(L)}$ $\oplus$ DeBERTa$^{(L)}$ $\oplus$ XLNet$^{(L)}$	**69.67**	34.51	**57.88**

Table 7: Performance of selected ensemble for averaging of output tag probabilities ("+") and majority votes on output edit spans ("⊕") ensembling types. Ensembles are not pre-trained on synthetic data (Stage I), tag vocabulary size of 5K. Benchmark is BEA-2019 (dev).

Experiments

7. Ensembling the GEC taggers

- Majority quorum: Minimum # of votes for triggering the edit
- Increasing N_{min} filters out more edits where single models disagree
- $1 \leq N_{min} \leq N_{single\_models}$
- Best performance when $N_{min} = N_{single\_models} - 1$

Ensemble	N_{single_models}	N_{min}	P	R	$F_{0.5}$
RoBERTa$_{5K}^{(B)}$ ⊕ RoBERTa$_{5K}^{(L)}$ ⊕ RoBERTa$_{10K}^{(L)}$	3	1	44.49	**41.96**	43.96
RoBERTa$_{5K}^{(B)}$ ⊕ RoBERTa$_{5K}^{(L)}$ ⊕ RoBERTa$_{10K}^{(L)}$	3	2	57.96	41.79	53.79
RoBERTa$_{5K}^{(B)}$ ⊕ RoBERTa$_{5K}^{(L)}$ ⊕ RoBERTa$_{10K}^{(L)}$	3	3	**67.54**	30.99	**54.65**
RoBERTa$_{5K}^{(B)}$ ⊕ RoBERTa$_{5K}^{(L)}$ ⊕ RoBERTa$_{10K}^{(L)}$ ⊕ DeBERTa$_{10K}^{(L)}$	4	1	40.21	41.68	40.50
RoBERTa$_{5K}^{(B)}$ ⊕ RoBERTa$_{5K}^{(L)}$ ⊕ RoBERTa$_{10K}^{(L)}$ ⊕ DeBERTa$_{10K}^{(L)}$	4	2	55.02	**43.14**	52.15
RoBERTa$_{5K}^{(B)}$ ⊕ RoBERTa$_{5K}^{(L)}$ ⊕ RoBERTa$_{10K}^{(L)}$ ⊕ DeBERTa$_{10K}^{(L)}$	4	3	64.48	37.49	**56.36**
RoBERTa$_{5K}^{(B)}$ ⊕ RoBERTa$_{5K}^{(L)}$ ⊕ RoBERTa$_{10K}^{(L)}$ ⊕ DeBERTa$_{10K}^{(L)}$	4	4	**71.71**	27.89	54.57
RoBERTa$_{5K}^{(B)}$ ⊕ RoBERTa$_{5K}^{(L)}$ ⊕ RoBERTa$_{10K}^{(L)}$ ⊕ DeBERTa$_{10K}^{(L)}$ ⊕ XLNet$_{10K}^{(L)}$	5	1	37.20	40.88	37.88
RoBERTa$_{5K}^{(B)}$ ⊕ RoBERTa$_{5K}^{(L)}$ ⊕ RoBERTa$_{10K}^{(L)}$ ⊕ DeBERTa$_{10K}^{(L)}$ ⊕ XLNet$_{10K}^{(L)}$	5	2	51.77	**43.65**	49.92
RoBERTa$_{5K}^{(B)}$ ⊕ RoBERTa$_{5K}^{(L)}$ ⊕ RoBERTa$_{10K}^{(L)}$ ⊕ DeBERTa$_{10K}^{(L)}$ ⊕ XLNet$_{10K}^{(L)}$	5	3	61.89	41.43	56.33
RoBERTa$_{5K}^{(B)}$ ⊕ RoBERTa$_{5K}^{(L)}$ ⊕ RoBERTa$_{10K}^{(L)}$ ⊕ DeBERTa$_{10K}^{(L)}$ ⊕ XLNet$_{10K}^{(L)}$	5	4	56.43	34.43	**56.43**
RoBERTa$_{5K}^{(B)}$ ⊕ RoBERTa$_{5K}^{(L)}$ ⊕ RoBERTa$_{10K}^{(L)}$ ⊕ DeBERTa$_{10K}^{(L)}$ ⊕ XLNet$_{10K}^{(L)}$	5	5	**73.12**	26.00	53.67

Table 8: Exploring the impact of N_{min} ("majority quorum"), a minumum number of votes to trigger the edit in majority votes ensembling. Benchmark is BEA-2019 (dev).

Experiments

7. Ensembling the GEC taggers

Ensemble	P	R	$F_{0.5}$
$\text{DeBERTa}_{5K}^{(L)} \oplus \text{RoBERTa}_{5K}^{(L)} \oplus \text{XLNet}_{5K}^{(L)}$	69.67	34.51	57.88
$\text{DeBERTa}_{10K}^{(L)} \oplus \text{RoBERTa}_{10K}^{(L)} \oplus \text{XLNet}_{10K}^{(L)}$	70.13	34.23	57.97
$\text{DeBERTa}_{5K}^{(L)} \oplus \text{RoBERTa}_{10K}^{(L)} \oplus \text{XLNet}_{5K}^{(L)}$	**70.71**	33.78	58.02
$\text{DeBERTa}_{10K}^{(L)} \oplus \text{RoBERTa}_{10K}^{(L)} \oplus \text{XLNet}_{5K}^{(L)}$	70.32	**34.62**	**58.30**

Table 9: Performance of the best single models ensembled by majority votes on output edit spans. Subscripts encode the models' tag vocabulary sizes from the set (5K, 10K). Benchmark is BEA-2019 (dev).

System	P	R	$F_{0.5}$
Single models			
(Kiyono et al., 2019)	65.5	59.4	64.2
(Omelianchuk et al., 2020)	79.2	53.9	72.4
(Kaneko et al., 2020)	67.1	60.1	65.6
(Stahlberg and Kumar, 2021)	72.1	**64.4**	70.4
(Rothe et al., 2021)	N/A	N/A	**75.88**
$\text{RoBERTa}_{5K}^{(L)}$, multi-stage training (this work)	**80.70**	53.39	73.21
$\text{RoBERTa}_{5K}^{(L)}$, one-stage training (this work)	80.55	52.27	72.69
Ensembles			
(Grundkiewicz et al., 2019)	72.3	60.1	69.5
(Kiyono et al., 2019)	74.7	56.7	70.2
(Omelianchuk et al., 2020)	79.4	57.2	73.7
(Kaneko et al., 2020)	72.3	61.4	69.8
(Stahlberg and Kumar, 2021)	77.7	**65.4**	74.9
$\text{DeBERTa}_{10K}^{(L)} \oplus \text{RoBERTa}_{10K}^{(L)} \oplus \text{XLNet}_{5K}^{(L)}$ (this work)	**84.44**	54.42	**76.05**

Table 10: Comparison of our best single tagging models and ensembles with related work on BEA-2019 (test).

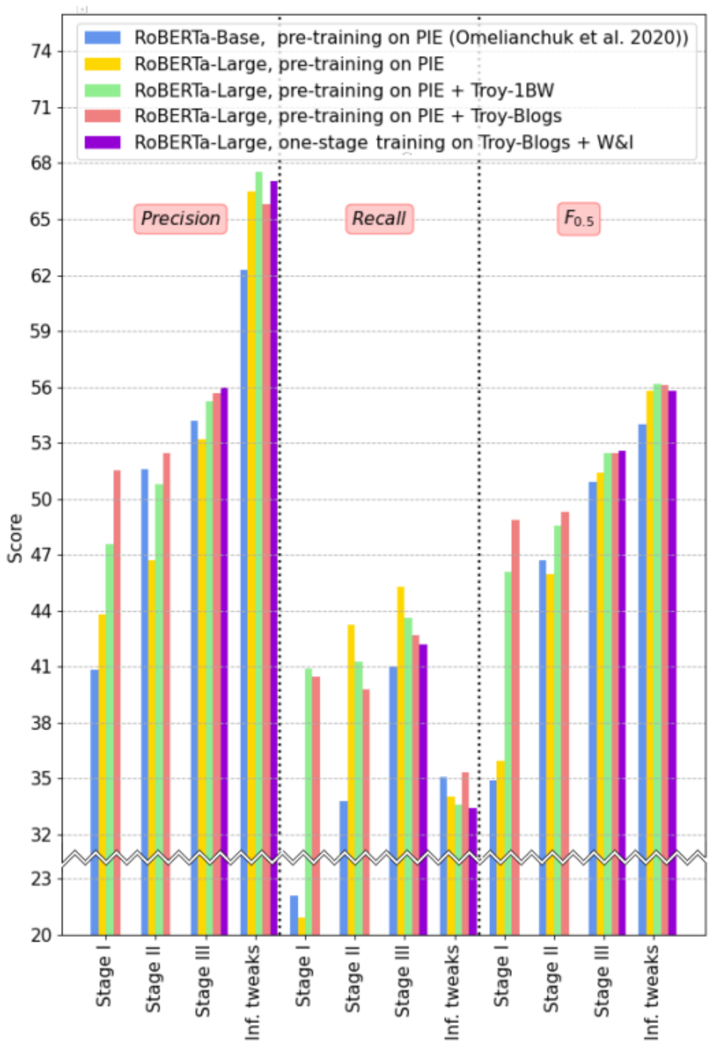
Experiments

8. Knowledge distillation

$\text{DeBERTa}^{(L)}_{10K} \oplus \text{RoBERTa}^{(L)}_{10K} \oplus \text{XLNet}^{(L)}_{5K}$ (this work)	**84.44**	54.42	**76.05**

- Teacher: Best ensemble model
- Student: A single sequence tagger
- The ensemble receives erroneous texts and generates their corrected version
- These input-output pairs of sentences are used for training single models

- 5% of 1BW (Troy-1BW)
- 28% of Blogs (Troy-Blogs)



Outline

1. Contribution
2. Method
3. Experiments
4. Conclusion

Conclusion

1. Any drawbacks?

Thank you

https://jeiyoon.github.io/