

Grammatical Error Correction: A Survey of the State of the Art

- リンク
 - <https://arxiv.org/abs/2211.05166>
- 概要
 - Grammatical Error Correction ≡ Language Error Correction
 - Errorの例
 - Replacement
 - Omission
 - Insertion
 - Data
 - Minimal vs Fluent Correction
 - Minimal correction – annotators should make the minimal number of changes to make a text grammatical
 - This can lead to corrections that sound unnatural
 - Fluent Correction

Original		I want explain to you some interesting part from my experience.
Minimal		I want <u>to</u> explain to you some interesting <u>parts of</u> my experience.
Fluent		I want <u>to tell you about</u> some interesting <u>parts of</u> my experience.

MinimalとFluentの線引きが難しい

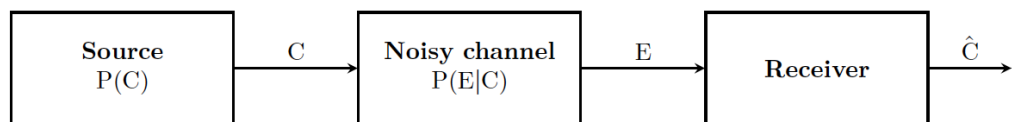
- TMU Evaluation Corpus for Japanese Learners
 - 日本語のGCEのデータセット
- Core Approaches
 - Classifiers

- 例：文中のすべての名詞句の前に{a/an, the, ..} のいずれかを予測する分類器を構築する
- エラーごとに分類器を作成する必要がある
- モデルの例
 - naive bayes, maximum entropy, decision trees, support-vector machines, the averaged perceptron
 - (Word2Vec, Glove) + (CNN, RNN, pointer networks)
- Statistical Machine Translation (SMT)
 - 専門的な知識や特徴量エンジニアリングなしに全てのエラーを修正できる
 - noisy channel modelから発想を得た

$$\hat{C} = \arg \max_C P(C|E) = \arg \max_C \frac{P(E|C)P(C)}{P(E)} = \arg \max_C P(E|C)P(C)$$

language model (LM) - $P(C)$

translation model (TM) - $P(E|C)$



- 例
 - 誤文は正解の文と似ているという性質を用いたもの
 - Levenshtein distance
 - edit operations
 - 異なるアルゴリズムによる重みの調整
 - minimum error rate training (MERT)
 - the margin infused relaxed algorithm (MIRA)
 - pairwise ranking optimization (PRO)

- 一つの言語を用いてLLMを追加学習する
 - データ : the British National Corpus (BNC), Wikipedia, Common Crawl
- ニューラルネットワークモデルを用いた手法
 - neural network global lexicon model (NNGLM)
 - neural network joint model (NNJM)
- 欠点
 - 局所的に整形されたフレーズを生成する傾向があり、全体で文法を見ると正しくないものが生成されることがある
 - 元のフレーズが正しくても、より頻度の高いフレーズに変更する傾向があり、その結果、不必要な修正が発生する
 - 長距離依存関係を処理できない
 - 特定のエラータイプに限定するのが難しい？
- Neural Machine Translation (NMT)

$$p(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{T'} p(y_t|\{y_1, y_2, \dots, y_{t-1}\}, \mathbf{x})$$

- 単一の大規模なニューラルネットワークを使用
- 膨大な学習データが必要
- Edit-based approaches
 - 文中の全トークンの代わりに編集操作を生成することで、編集生成アプローチは一般的に推論速度が速く、文全体を生成するGECシステムよりも5倍から10倍速いと言われている
- Language Models for Low-Resource and Unsupervised GEC
 - Language models as Discriminators
 - LM-Critic
 - Language models as Generators

- 最近のLMに基づくGECのアプローチは、ゼロショットまたは少数ショットのジェネレータとして言語モデルを使用し、プロンプトでノイズの多い入力文が与えられた場合に補正を生成する
- Additional techniques
 - iterative decoding / multi-turn decoding
 - GECシステムの出力をもう一度GECシステムの入力とする
- データ拡張
 - 元の綺麗な文章にノイズを追加して、文法誤り文を作成する
- 評価指標
 - MaxMatch

		TP		FN	FP	
Original	I	likes	to	drive	a	bicycle .
Hypothesis	I	like	to	drive	the	bicycle .
Reference	I	like	to	ride	a	bicycle .

$$(3) \quad P = \frac{TP}{TP + FP} \quad (4) \quad R = \frac{TP}{TP + FN} \quad (5) \quad F_\beta = (1 + \beta^2) \times \frac{P \times R}{(\beta^2 \times P) + R}$$

- ERRANT
 - MaxMatchとは異なり、エラーの種類によってGECシステムを評価できる
- GLEU
 - 原文ではなく、参照文と重複しているn-gramに報酬を与え、参照文ではなく、原文と重複しているn-gramにはペナルティを与える

$$p_n = \frac{\sum_{i=1}^{|H|} \left(\sum_{g \in \{h_i \cap r_i\}} \text{count}_{h_i, r_i}(g) - \sum_{g \in \{h_i \cap o_i\}} \max[0, \text{count}_{h_i, o_i}(g) - \text{count}_{h_i, r_i}(g)] \right)}{\sum_{i=1}^{|H|} \sum_{g \in \{h_i\}} \text{count}_{h_i}(g)}$$

$\text{count}_a(g) = \# \text{ occurrences of n-gram } g \text{ in } a$

$\text{count}_{a,b}(g) = \min(\# \text{ occurrences of n-gram } g \text{ in } a, \# \text{ occurrences of n-gram } g \text{ in } b)$ (6)

$$\text{BP} = \begin{cases} 1 & \text{if } l_h > l_r \\ \exp(1 - l_r/l_h) & \text{if } l_h \leq l_r \end{cases} \quad (7)$$

$$\text{GLEU}(O, H, R) = \text{BP} \cdot \exp \left(\frac{1}{N} \sum_{n=1}^N \log p_n \right) \quad (8)$$

■ Reference-based

• I-measure

- MaxMatchでは、(TP=0, FP>0)と(TP=0, FP=0)を同様の精度と見なしてしまうが、I-measureではそれを改善した

• GMEG

- MaxMatch, ERRANT, GLEU, I-measureのアンサンブル

• GoToScorer

- 多くのGECシステムが正しく直せた問題については、重きを小さくし、ほとんどのGECシステムが間違っただけのものについては、重きを大きくした。

■ Reference-less

• GBMs

- 他のGECシステムを使用して文法誤りを評価する

• GFM

- Grammatically, Fluency, Meaning preservation

• USim

- 原文と予測の文章をセマンティックグラフに落とし込んで距離を測る?

• SOME

- GFMをBERTのファインチューニングを加えることで改善
- Scribendi Score
 - 原文と予測の文書を比べて、GPT2にとってどちらの方が発生確率が高いかを計算する
- IMPARA
 - 学習時はreference-basedで学習し、推論時はreference-lessで評価
- GECシステムの比較

System	Synthetic Sents	Corpora	Pre-trained Model	Architecture	Techniques	CoNLL14 M2	BEA19 ERRANT
Qorib, Na, and Ng (2022)	-	W (dev)	Various ¹	T5-large, RoBERTa-base, XLNet-base, Transformer-big	SC	69.5	79.9
Lai et al. (2022)	9m	N+F+L+W	RoBERTa, XLNet	RoBERTa-base, XLNet-base	ENS+PRT+MTD	67.0	77.9
Sorokin (2022)	9m	cL+N+F+W	RoBERTa	RoBERTa-large	RE+MTD	64.0	77.1
Tarnavskiy, Chernodub, and Omelanchuk (2022)	-	N+F+L+W	RoBERTa, XLNet, DeBERTa	RoBERTa-large, XLNet-large, DeBERTa-large	VT+PRT+MTD	65.3	76.1
Rothe et al. (2021)	-	cL	T5-xxl	T5-xxl	-	68.9	75.9
Sun and Wang (2022)	300m	N+F+L+W	BART	BART (12+2)	PRT	-	75.0
Stahlberg and Kumar (2021)	546m	F+L+W	-	Transformer-big	ENS	68.3	74.9
Cao, Yang, and Ng (2023)	200m	cL+N+F+W	-	Transformer-big	ENS	68.5	74.8
Omelanchuk et al. (2020)	9m	N+F+L+W	BERT, RoBERTa, XLNet	BERT-base, RoBERTa-base, XLNet-base	ENS+PRT+MTD	66.5	73.7
Lichtarge, Alberti, and Kumar (2020)	340m	F+L+W	-	Transformer-big	ENS	66.8	73.0
Zhang et al. (2022c)	-	cL+N+F+W	BART	BART-large	-	67.6	72.9
Sun et al. (2021)	300m	N+F+L+W	BART	BART (12+2)	-	66.4	72.9
Yasunaga, Leskovec, and Liang (2021)	9m	N+F+L+W	XLNet	XLNet-base	PRT+MTD	65.8	72.9
Parnow, Li, and Zhao (2021)	9m	N+F+L+W	XLNet	XLNet-base	PRT+MTD	65.7	72.8
Yuan et al. (2021)	-	N+F+L+W+CLC	ELECTRA	Multi-encoder, Transformer-base	RE	63.5	70.6
Stahlberg and Kumar (2020)	346m	F+L+W	-	Seq2Edits (modified Transformer-big)	ENS+RE	62.7	70.5
Kaneko et al. (2020)	70m	N+F+L+W	-	Transformer-big	ENS+RE	65.2	69.8
Mita et al. (2020)	70m	N+F+L+W	-	Transformer-big	ENS+RE	63.1	67.8
Chen et al. (2020a)	260m	N+F+L+W	RoBERTa	Transformer-big	-	61.0	66.9
Katsumata and Komachi (2020)	-	N+F+L+W	BART	BART-large	ENS	63.0	66.1

¹ Combines [Rothe et al. \(2021\)](#); [Omelanchuk et al. \(2020\)](#); [Kivono et al. \(2019\)](#); [Grundkiewicz, Junczys-Dowmunt, and Heafield \(2019\)](#); [Choe et al. \(2019\)](#).

Table 8

Top-performing systems since 2020. The symbols in the Corpora column are N: NUCLE, F: FCE, L: Lang-8, W: W&I, cL: cLang-8, and CLC: Cambridge Learner Corpus. The symbols in the Techniques column are ENS: ensemble, MTD: multi-turn decoding, PRT: precision-recall trade-off, RE: re-ranking, SC: system combination, and VT: voting combination.

- データの量とパフォーマンスに明確な関連性があるとはまだ言えない
- 学習済みモデルを使用する際には、複数使用してアンサンブルした方が良い