Masked Language Model による系列確率に基づく文法誤り検出

土肥 康輔　須藤 克仁　中村 哲
奈良先端科学技術大学院大学
{doi.kosuke.de8, sudoh, s-nakamura}@is.naist.jp

概要

言語モデルに基づく文法誤り訂正手法は，少量のラベル付きデータしか必要としないという長所があるが，既存手法では訂正対象の誤り種類が限られているという課題があった．本研究では，対象の誤り種類を拡大する第一歩として，言語モデルにより算出される系列確率に基づく文法誤り検出手法を提案する．文法的に正しい文での各語の確率を計算して辞書に格納し，推論時には，誤り検出対象の文での確率と辞書での確率を比較することで，その語が誤りであるかを判別する．実験の結果，$F_{0.5}$ スコアは既存モデルに及ばなかったものの，Recall が Precision より高くなり，今後適切な制約を加えることで性能を向上させられる可能性が示された．

1　はじめに

文法誤り検出 (GED) は，テキスト中の誤りを自動的に検出するタスクである．Bi-LSTM ベースのモデルが主流であったが [1, 2, 3, 4, 5]，近年では事前学習済み言語モデルを用いる手法も提案されている [6, 7]．これらの研究は GED を系列ラベリングタスクとして解いているが，利用可能なラベル付きデータは多くなく，疑似データの利用も行われている．

少量のラベル付きデータしか必要としない手法に，言語モデルに基づくアプローチがある．文法誤り訂正 (GEC) タスクにおいて，言語モデルの利用は主要なアプローチの 1 つだったが [8]，機械翻訳に基づく手法の台頭によりあまり見られなくなっていった[1)]．Bryant ら [9] や Alikaniotis ら [10] の研究により，言語モデルに基づく手法が再度注目を浴びたが，[9][10] の手法では，訂正できる誤りの種類が限定されていることが課題となっていた．そこで本研究では，言語モデルに基づく GEC 手法で扱える誤り種類を拡大する第一歩として，事前学習済み言語モデルによる系列確率に基づく GED を提案する．

実験では，提案手法が評価セット中のすべての誤り種類を検出対象とできていることが確認された．しかし，性能では既存モデルに及ばず，特に Precision に課題があった．一方で，提案手法は比較的高い Recall を達成しており，今後適切な制約を加えることで性能を向上させられる可能性がある．

2　関連研究

ニューラルネットワーク手法を初めて GED に用いたのは Rei ら [1] である．CNN，RNN，Bi-LSTM を比較した結果，Bi-LSTM が最も良い性能を示したと報告されている．以降は，[1] のモデルを改良する形で研究が進められ，[2] では文字レベルの情報を追加で用い，[3] ではラベルを予測する前後のトークンを言語モデルで同時に予測することで，性能の向上が図られた．[5] は文脈化された単語分散表現を追加の特徴量として用いることで，さらに性能を向上させた．また，疑似データを用いることで性能が向上することも報告されている [4]．

Bi-LSTM に基づくこれらの手法に対して，Kaneko ら [6] は事前学習済み言語モデルを用いることを提案した．事前学習済み言語モデルをファインチューニングして用いる場合，最終層の出力のみを予測に用いるのが一般的だが，[6] は BERT [11] の最終層だけでなく中間層の出力も利用するモデルを構築した．Yuan ら [7] は事前学習手法が GED タスクに類似している ELECTRA [12] を用いるほうが，BERT を含む他の事前学習済み言語モデルを用いるよりも高い性能となることを示した．さらに，誤り箇所の誤りカテゴリも予測する多クラス GED を提案し，GED の出力を GEC タスクの追加入力として用いることで，GEC の性能が向上することを示した．

言語モデルに基づくアプローチでは，低い確率の系列には誤りが含まれていると考える．[9] は母語話者コーパスから構築した N-gram 言語モデルを用いることで，少量の誤り情報付きデータしか必要としない GEC 手法を提案した．[10] は BERT 等の最

1）ただし，GEC システムの一部分としては使い続けられた．

新のニューラル言語モデルを用いることで，同手法の性能を向上させた．しかし，[9][10] のモデルの訂正対象は，存在しない語，形態（例：名詞の数，動詞の時制），冠詞，前置詞に限定されており，誤りの20%程度を占める不足誤りを訂正できないという問題点があった．

3 提案手法

本研究では，訓練データでの各トークンの確率を算出し，トークンを見出し，確率を値とする確率辞書を作成する．推論時には，誤り検出対象の文での確率と辞書での確率を比較することで，そのトークンが誤りであるかを判別する．以下の節で，確率辞書作成と推論の手法について説明する．

3.1 確率辞書作成

確率辞書を作成するための元データには，母語話者コーパス，または誤り訂正済みの学習者コーパスを用いる．辞書に保存されている確率値が推論時の正誤判断基準となるため，文法的に正しい文を用いて辞書を作成する．本研究では，トークンの確率算出に BERT を用いる．入力文中の各トークンを [MASK] し，元のトークンが返される確率を計算する手法 [10, 13] によってトークンの確率を得る．訓練データから 1 文ごとにトークンの確率を算出し，そのトークンを見出し，確率を値として辞書に登録していく．訓練データの文は BERT のトークナイザーで分割されるため，見出しはサブワードとなり，訓練データで複数回登場するサブワード（見出し）には複数の確率（値）が登録される．訓練データ中での頻度が N 以上であるサブワードのみを確率辞書に残すというパラメータ min_freq を設定し，本研究では $min\_freq \in \{1, 5, 10\}$ とする．

3.2 推論

推論時には，確率辞書作成時と同じ手法でトークンごとの確率を算出し，推論対象の文中での確率と辞書の確率を比較する（図 1）．確率辞書には，1 つの見出しに対して複数の値が登録されているため，基準となる値（std_{token}）を決める必要がある．本研究では，登録されている値の平均値，または中央値とする．また，確率辞書の見出しにないトークンが推論対象の文に現れた場合の基準値（std_{all}）は，全見出しの値の平均値，または中央値とする．

しかし，元トークンが文法的に正しい場合でも，

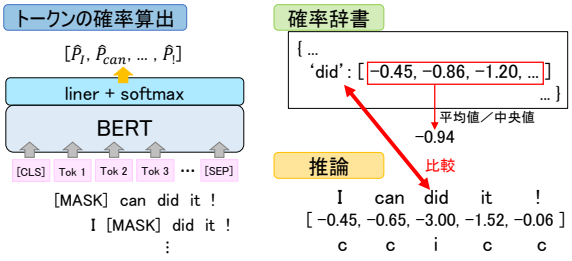

図 1 提案手法での推論の例

言語モデルにとってより「自然な」候補が存在する場合，言語モデルが元トークンに与える確率は相対的に低いものとなってしまう．そこで [10] に従い，元トークンが文法的に正しいというバイアスを加えるパラメータ τ を設定する．あるトークン T の推論対象の文での確率を $P(T_s)$，確率辞書での基準値を $P(T_d)$ とすると，そのトークンに付与される $label(T_s)$ は，以下のようになる．

$$label(T_s) = \begin{cases} \text{i} & \text{if } \log P(T_s) + \log \tau < \log P(T_d) \\ \text{c} & \text{otherwise} \end{cases} \quad (1)$$

“i” は誤りあり，“c” は正しいと判断されたことを表す．本研究では，$\tau \in \{0, 2, 4, 6, 8, 10, 15, 20\}$ とする．

上記のように $label(T_s)$ を得ると，ラベルは BERT のトークナイザーで分割されたサブワード単位で付与されるため，元のトークン列との対応を取る必要がある．両者の対応付けは以下のアルゴリズムで行う．サブワードと元トークンを要素とするリストを作成し，リストから要素を 1 つずつ読み込む．文字列が一致した場合は，$label(T_s)$ を元トークンに対応付け，両リストから次の要素を 1 つずつ読み込む．一致しない場合は，2 つの文字列長を比較し，短いほうの文字列のリストから次の要素を読み込み，その文字列の後ろに結合する．この際，サブワード分割を表す “##” は削除したうえで読み込む．これを，2 つの文字列が一致するまで続ける．一致するまでに使ったサブワードに対応する $label(T_s)$ を確認し，全てが “c” であれば元トークンに “c” を対応付け，そうでなければ “i” を対応付ける．

4 実験

実験では，確率辞書作成方法の違いによる誤り検出性能の差を検証する．辞書作成に用いるデータの種類（母語話者 vs. 学習者）と量，辞書に登録する語の訓練データ中での最低頻度，基準値の決め方の影響を評価する．また，τ の違いによる誤り検出性能の差を検証する．

4.1 データセット

確率辞書作成のための母語話者コーパスには the One Billion Word Benchmark (1B) [14] を用い，学習者コーパスのトークン数と同程度になるようにサンプリングして使用した．また，データ量の影響を検証するために，約 1,000 万トークンサンプリングしたデータを使用した．学習者コーパスは，W&I+L train [15] の 80%，FCE train [16]，NUCLE [17] を用いた．確率辞書作成に用いたデータの詳細は付録 A に示す．開発データには，W&I+L train の残りの 20%，FCE dev，CoNLL-2013 [18] を用い，評価データには W&I+L dev，FCE test，CoNLL-2014 [8] を用いた．

4.2 実験設定

文中の各トークンの確率算出には，[13] の実装[2)]を用いた．BERT は HuggingFace 社が公開している `bert-base-cased` を用い，訓練データの最大入力長は 250 トークンとした．確率辞書の作成はデータセットごと（付録 A の表の行ごと）に行った．確率辞書作成に先立ち，URL を含む文，メールアドレスを含む文，1 文に数字のみから成るトークンが 30 以上含まれる文を訓練データから取り除いた．

また，アポストロフィーを含む短縮形のトークン分割方法の違いのため，データをそのまま BERT のトークナイザーに入力すると，分割結果に影響があった．そのため，1B は WMT11[3)]で配布された `detokenizer.perl` を用いてデトークナイズした．それ以外のデータは，spaCy が例外規則でトークン分割している短縮形[4)]のみをデトークナイズした．

モデルの性能は，ERRANT [19] により算出される Precision，Recall，$F_{0.5}$ スコアにより評価した．

4.3 実験結果

開発セットでの結果に基づき，$min\_freq = 1$，std_{token} と std_{all} の組み合わせ $\in$ {{mean, mean}, {median, median}}，$\tau \in \{0, 8, 20\}$，確率辞書 $\in$ {FCE, Learner all, 1B_510K, 1B_2.1M, 1B_10M}について，評価セットで評価を行った．

表 1 は，FCE train で確率辞書を作成したときに，基準値の取り方と τ を変化させた結果である[5)]．

表 1 確率辞書の基準値，τ による比較

		W&I+L dev		
Std.	τ	Pre.	Rec.	$F_{0.5}$
(mean, mean)	0	20.06	74.29	23.49
	8	30.69	51.85	33.42
	20	35.37	43.01	36.67
(median, median)	0	15.57	84.44	18.61
	8	28.24	56.45	31.37
	20	32.94	47.98	35.14

Precision は $\tau = 0$ のときに最も低く，τ が大きくなるにつれて上昇した．Recall はその反対で，τ が大きくなるにつれて低下した．

τ が同じ値の場合，$F_{0.5}$ スコアは{mean, mean}のほうが高くなったが，これは，基準値を mean とした場合 (P_{avg}) と median とした場合 (P_{mdn}) に，$P_{avg} < P_{mdn}$ となることが多いためと考えられる．FCE で作成した確率辞書において，P_{avg} の平均値は −4.16 であったのに対して，P_{mdn} の平均値は −3.95 であった．また，頻度 3 以上の見出し 3,787 語において，3,047 語が $P_{avg} < P_{mdn}$ となっていた．P_{mdn} のほうが “i” ラベルが多くつくが，Precision が低いために $F_{0.5}$ スコアが悪化したと考えられる．

τ の値を大きくすると，誤り検出対象の文のすべての語において，ラベル判断に用いられる値が τ だけ一律に大きくなる（式 1）．この操作は，確率辞書の基準値を一律に下げる操作と同じである．多くの語で $P_{avg} < P_{mdn}$ となっている状況では，これら 2 つのハイパーパラメータは，実質同じものを操作していたことになる．τ のほうが細かい調整が可能なこと，基準値を{median, median}とすると，初期値として高い Recall からスタートできることから，今後は std_{token} と std_{all} の組み合わせは{median, median}で固定し，τ のみをハイパーパラメータとして保持するほうがよいと考えられる．

次に確率辞書間で比較をすると，辞書の作成に利用するデータの種類，量はスコアに影響を与えないことが明らかになった．例えば，基準値を{mean, mean}，$\tau = 20$ としたとき，W&I+L dev での $F_{0.5}$ スコアは，確率辞書の設定によらず 36 程度であった．つまり，確率辞書作成に用いるデータは必ずしも学習者コーパスである必要はなく，言語モデルと確率辞書を作成するための母語話者コーパスが十分にあれば，提案手法を用いて GED を行うことが可能となることを示唆している．$min\_freq = 1$ と設定しているため，データ量を増やすと確率辞書の見出し数が増加することが期待されるが，コーパスのトー

2) https://github.com/awslabs/mlm-scoring
3) https://www.statmt.org/wmt11/translation-task.html
4) https://github.com/explosion/spaCy/blob/master/spacy/lang/en/tokenizer_exceptions.py
5) スペースの都合上，詳細な結果は付録 B に掲載する．

表 2 既存モデルとの比較

		FCE test		
Dict.	τ	Pre.	Rec.	$F_{0.5}$
FCE	20	42.01	48.07	43.10
1B_510K	20	39.80	50.43	41.55
Yuan ら [7]		82.05	50.49	72.93
Rei ら [1]		46.1	28.5	41.1

クン数が増加しても，異なり語数は線形には増加しない．また，辞書に入る異なり語数が増加しても，評価セットでそれらの語が出てくるとは限らず，カバー率は劇的には上昇しない．これらの要因により，本手法ではデータ量によるスコア差が出なかったものと考えられる．

表 2 は，確率辞書を FCE，または 1B_510K で作成し，基準値を{median, median}，$\tau = 20$ としたときのスコアを，既存モデルと比較した結果を示している．[7] は，ELECTRA を FCE データでファインチューニングすることで，世界最高性能を達成したモデル，[1] は，追加の特徴量を用いていないナイーブな LSTM ベースのモデルである．[7] に対しては，$F_{0.5}$ スコアで 30 ポイント程度のビハインドがあるが，[1] に対しては，同程度か上回るスコアを達成している．誤り情報付きデータを確率辞書作成に用いていない 1B_510K においても，誤り情報付きデータを用いている [1] のモデルに迫るスコアを達成していることは特筆すべき点である．

提案モデルは，既存モデルと異なり Recall が Precision より高い傾向となった．Recall は [7] と同程度のスコアを達成しているが，Precision には大きな課題がある．実験結果から，τ は Precision と Recall の調整にある程度機能していたことがうかがえるが，同時に τ の値のみでは不十分であったことも示唆している．τ を上昇させたときに Recall が急激に低下する一方で，Precision の上昇は比較的緩やかなものとなっていた．先に述べたように，τ は誤り検出対象の文のすべての語において，ラベル判断に用いられる値を一律で増加させる．一律に変化させるのではなく，そのトークンが使われている文脈に応じて制約を課すしくみが必要と考えられる．

4.3.1 誤り種類ごとの検出性能

本手法の動機のひとつに，言語モデルベースのモデルで扱える誤り種類の拡大があった．本節では，FCE train で確率辞書を作成し，基準値{mean, mean}と設定したモデルの FCE test での誤り検出結果について，誤りタイプ，カテゴリごとの結果を述べる．

ERRANT の出力から，FCE test に存在する 15 種類すべての誤りカテゴリが検出対象となっていたことが確認できた[6]．また，提案モデルは不足誤りの箇所を検出することができていた．以下に例を示す．

Gold: Apart from that it takes long time to go somewhere .
$\tau = 0$: Apart from that it takes long time to go somewhere .
$\tau = 8$: Apart from that it takes long time to go somewhere .

下線は，その語に誤りラベルがついていることを表している．冠詞の不足誤りを提案モデルが捉えることができているのがわかる．一方で，上記の例の $\tau = 0$ のように，提案モデルは偽陽性の数が多くなっている．$\tau = 0, 8, 20$ で Precision が一貫して下位 5 位以内になる誤りカテゴリは，短縮形，名詞，代名詞であった．しかし，個別の例を確認していくと，GED データのラベル付け方法が原因で，ERRANT の誤りカテゴリの判別がうまくいっていない可能性があることがわかった．GED データでは，不足誤りはそのトークンが存在しないため，直後のトークンに誤りラベルを付与することが一般的である．そのため，ERRANT が名詞誤りや代名詞の誤りと判別していても，実際は別の誤りカテゴリの不足誤りである可能性がある．GED を誤りカテゴリごとに評価するには，不足誤りを正しいカテゴリに対応付ける必要があるが，現状のモデル出力では，不足，置換，余剰のいずれの誤りなのかはわからない．Yuan ら [7] の系列ラベリングベースのモデルのように，誤りタイプまで予測する GED モデルも出てきているが，言語モデルベースのモデルでそれが可能なのか検討することは今後の課題である．

5 おわりに

本研究では，BERT により算出される系列確率を用いることで，少量の誤り情報付きデータしか必要としない GED 手法を提案した．既存の言語モデルベースの GEC モデルが扱えていなかった誤り種類について，提案手法は検出対象とすることができていた．しかし，Recall は高いものの，Precision が低いという課題があった．また，現在の確率辞書では，同じ表層系で異なる品詞は区別できていない．文脈情報を利用した制約の導入と併せて検討し，Recall の低下を抑えながら Precision を向上させることが今後の課題である．

6) 検出性能の良し悪しは考慮していない．

謝辞

本研究の一部は JSPS 科研費 JP21H05054 の助成を受けたものである．

参考文献

[1] Marek Rei and Helen Yannakoudakis. Compositional sequence labeling models for error detection in learner writing. In **Proc. of ACL**, pp. 1181–1191, 2016.

[2] Marek Rei, Gamal Crichton, and Sampo Pyysalo. Attending to characters in neural sequence labeling models. In **Proc. of COLING**, pp. 309–318, 2016.

[3] Marek Rei. Semi-supervised multitask learning for sequence labeling. In **Proc. of ACL**, pp. 2121–2130, 2017.

[4] Sudhanshu Kasewa, Pontus Stenetorp, and Sebastian Riedel. Wronging a right: Generating better errors to improve grammatical error detection. In **Proc. of EMNLP**, pp. 4977–4983, 2018.

[5] Samuel Bell, Helen Yannakoudakis, and Marek Rei. Context is key: Grammatical error detection with contextual word representations. In **Proc. of BEA**, pp. 103–115, 2019.

[6] Masahiro Kaneko and Mamoru Komachi. Multi-head multi-layer attention to deep language representations for grammatical error detection. **Computacion y Sistemas**, Vol. 23, No. 3, pp. 883–891, 2019.

[7] Zheng Yuan, Shiva Taslimipoor, Christopher Davis, and Christopher Bryant. Multi-class grammatical error detection for correction: A tale of two systems. In **Proc. of EMNLP**, pp. 8722–8736, 2021.

[8] Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. The CoNLL-2014 shared task on grammatical error correction. In **Proc. of CoNLL: Shared Task**, pp. 1–14, 2014.

[9] Christopher Bryant and Ted Briscoe. Language model based grammatical error correction without annotated training data. In **Proc. of BEA**, pp. 247–253, 2018.

[10] Dimitris Alikaniotis and Vipul Raheja. The unreasonable effectiveness of transformer language models in grammatical error correction. In **Proc. of BEA**, pp. 127–133, 2019.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In **Proc. of NAACL**, pp. 4171–4186, 2019.

[12] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: Pre-training text encoders as discriminators rather than generators. In **ICLR**, 2020.

[13] Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. Masked language model scoring. In **Proc. of ACL**, pp. 2699–2712, 2020.

[14] Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. One billion word benchmark for measuring progress in statistical language modeling. In **Proc. of Interspeech**, pp. 2635–2639, 2014.

[15] Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. The BEA-2019 shared task on grammatical error correction. In **Proc. of BEA**, pp. 52–75, 2019.

[16] Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. A new dataset and method for automatically grading ESOL texts. In **Proc. of ACL**, pp. 180–189, 2011.

[17] Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. Building a large annotated corpus of learner English: The NUS corpus of learner English. In **Proc. of BEA**, pp. 22–31, 2013.

[18] Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetreault. The CoNLL-2013 shared task on grammatical error correction. In **Proc. of CoNLL: Shared Task**, pp. 1–12, 2013.

[19] Christopher Bryant, Mariano Felice, and Ted Briscoe. Automatic annotation and evaluation of error types for grammatical error correction. In **Proc. of ACL**, pp. 793–805, 2017.

A 確率辞書作成に用いたデータセット

Dataset	Sampled	Sentences	Tokens
1B	510K	20,250	513,103
	2.1M	83,900	2,126,760
	10M	394,100	10,000,636
W&I+L train	80%	27,444	513,437
FCE train		28,350	458,836
NUCLE		57,151	1,155,563
Learner all		112,945	2,127,836

B 評価セットでのスコア

		W&I+L dev			FCE test			CoNLL-2014-id0			CoNLL-2014-id1		
Dict.	τ	Pre.	Rec.	$F_{0.5}$	Pre.	Rec.	$F_{0.5}$	Pre.	Rec.	$F_{0.5}$	Pre.	Rec.	$F_{0.5}$
(mean, mean)													
FCE	0	20.06	74.29	23.49	27.65	74.30	31.62	18.31	70.09	21.48	24.73	69.45	28.39
	8	30.69	51.85	33.42	40.00	52.04	41.94	26.15	46.26	28.64	35.24	45.72	36.94
	20	35.37	43.01	36.67	44.53	43.63	44.35	30.03	37.60	31.29	40.56	37.25	39.85
Learner all	0	19.91	75.40	23.35	27.27	74.97	31.25	17.99	71.70	21.16	24.47	71.54	28.18
	8	30.54	52.44	33.32	39.57	52.71	41.65	26.45	47.57	29.03	35.39	46.68	37.19
	20	35.50	43.77	36.89	44.39	44.54	44.42	30.41	38.70	31.77	40.77	38.06	40.20
1B_510K	0	19.56	76.06	22.97	26.84	75.53	30.81	17.68	72.14	20.82	24.26	72.60	27.98
	8	29.72	54.00	32.66	38.06	53.39	40.38	25.67	48.61	28.35	34.72	48.20	36.77
	20	34.15	44.67	35.84	42.52	45.66	43.11	29.56	39.98	31.19	39.36	39.04	39.30
1B_2.1M	0	19.71	76.05	23.14	26.97	75.64	30.95	17.70	72.17	20.85	24.30	72.67	28.03
	8	29.96	53.82	32.87	38.44	53.50	40.73	25.68	48.74	28.36	34.84	48.50	36.92
	20	34.57	44.69	36.21	43.16	45.56	43.62	29.78	40.05	31.39	39.64	39.09	39.53
1B_10M	0	19.76	76.18	23.19	26.97	75.73	30.96	17.75	72.24	20.90	24.37	72.75	28.11
	8	30.06	53.85	32.97	38.68	53.66	40.97	25.71	48.61	28.38	34.87	48.35	36.93
	20	34.69	44.75	36.33	43.29	45.66	43.74	29.82	39.95	31.42	39.57	38.87	39.43
(median, median)													
FCE	0	15.57	84.44	18.61	21.50	84.47	25.27	14.98	80.60	17.89	20.17	79.62	23.71
	8	28.24	56.45	31.37	37.21	56.46	39.93	24.37	51.29	27.23	32.84	50.69	35.33
	20	32.94	47.98	35.14	42.01	48.07	43.10	27.91	42.23	29.94	37.41	41.51	38.16
Learner all	0	15.00	86.58	17.98	20.84	86.12	24.57	14.33	82.98	17.17	19.30	81.95	22.78
	8	27.86	57.78	31.08	37.00	57.99	39.89	24.13	52.67	27.07	32.86	52.58	35.52
	20	32.49	48.82	34.82	41.94	49.12	43.20	28.18	43.74	30.34	37.46	42.64	38.39
1B_510K	0	14.73	86.20	17.66	20.69	86.17	24.39	14.16	83.08	16.97	19.15	82.42	22.62
	8	26.88	58.69	30.15	35.18	58.58	38.24	23.10	53.44	26.06	31.80	53.94	34.64
	20	31.40	50.14	33.93	39.80	50.43	41.55	26.99	44.91	29.33	36.21	44.19	37.57
1B_2.1M	0	14.76	86.43	17.70	20.72	86.23	24.43	14.14	83.25	16.96	19.18	82.77	22.66
	8	26.93	58.68	30.20	35.53	58.94	38.59	23.08	53.71	26.06	31.86	54.36	34.73
	20	31.59	50.34	34.13	40.00	50.56	41.75	27.05	45.28	29.42	36.45	44.76	37.86
1B_10M	0	14.76	86.46	17.70	20.71	86.33	24.43	14.15	83.12	16.96	19.14	82.47	22.61
	8	26.96	58.74	30.23	35.41	59.05	38.49	23.23	53.71	26.20	32.01	54.28	34.87
	20	31.65	50.38	34.19	40.27	50.80	42.01	27.09	45.08	29.44	36.29	44.29	37.65
Yuan ら [7]		72.81	46.85	65.54	82.05	50.49	72.93	55.15	39.78	51.19	76.44	40.13	64.73
Rei ら [1]		-	-	-	46.1	28.5	41.1	-	-	-	-	-	-

C 実際の誤り種類と ERRANT の判別が一致していない例

以下の例で，提案モデル（確率辞書 = FCE，基準値 = {mean, mean}，$\tau = 0$）の検出結果は，正解と一致している．GED データを元に ERRANT で作成した M2 ファイルでは，“information” は名詞誤りと分類されていたが，GEC データで確認すると，実際は形容詞の不足誤りであった．

Gold: If you need any information , please let me know .
Model: If you need any information , please let me know .
Corrected: If you need any more information , please let me know .