データ拡張手法を用いたスペイン語文法誤り訂正

飯島 与喜[1] Tad Gonsalves[1]
[1]上智大学 理工学研究科 理工学専攻 情報学領域
t-iijima-2n3@eagle.sophia.ac.jp t-gonsal@sophia.ac.jp

概要

スペイン語の文法誤り訂正では，人の手によって訂正されたコーパスが限られ，英語と比較して使用できるデータが少ない．データ不足の問題を解決する方法としてデータ拡張手法がある．既存のデータ拡張手法ではランダムにデータを生成するため，コーパスに収録されているスペイン語学習者から取得したデータとは全く異なるものになる．そこで，本研究ではコーパスから誤りパターンを抽出し，データに反映させることで，コーパスに収録されているスペイン語学習者の誤りを再現するデータ拡張手法を提案する．この手法によって訓練用のデータを生成し,限られたデータで高性能なモデルを構築する．

1 はじめに

文法誤り訂正（Grammatical Error Correction : GEC）はスペルや句読点，文法的な誤りを検出して訂正する自然言語処理のタスクである．このタスクでは誤りを含む文（誤文）を GEC のモデルに入力し，その誤文に含まれる誤りを訂正した文（正文）を出力する．本研究では，スペイン語の誤りを訂正する．

文法誤り訂正では，主に誤文と正文をペアとした学習者のコーパスを教師データとしてモデルを学習させる手法が用いられる．スペイン語の学習者コーパスには Lang8[1]と COWSL2H[2]があるが，これらのデータセットだけでは英語のコーパスと比較するとデータ数が少ない．そこで，本研究では文法誤り訂正固有のデータ拡張手法である人工誤り生成手法を用いて，より高性能なモデルを構築する．人工誤り生成手法は単一言語データから誤文を生成する．生成した誤文と単一言語データの正文を組にして事前学習を行う．事前学習をした後にファインチューニングする手法が有効であると[3]で報告されている．

また，人工誤り生成手法で単一言語データに誤りを注入する際に，ランダムに誤りを注入するよりも学習者が犯した誤りに近いものを注入した方がより高性能なモデルを構築できると考えられる．そこで本研究では，学習者コーパスから誤りパターンを抽出し，単一言語データに誤りを注入することで，スペイン語学習者の誤りを再現するデータ拡張手法を提案する．学習者コーパスから抽出した誤りパターンを単一言語データに注入して誤りを生成することで，学習者コーパスに含まれる特定の品詞誤りや語形変化だけでなく様々な誤りを再現することができる．誤りを注入する際にコーパス内の分布に従って誤りを生成することで，ランダムに誤り生成するよりも学習者が書いたものに近い文を得ることができる．また，誤りをコーパスから抽出する手法を用いることで，誤りの規則を定義する手間を軽減することが可能である．

第３章でデータ拡張手法の詳細を説明し，第４章の実験でその有効性を確認するための実験設定について述べる．第５章の結果では，提案手法が既存研究[2]や[4]よりも高性能であった結果とデータ拡張手法の 1 文あたりの誤り率を変化させた場合のモデルの性能、誤りタグごとのモデル性能を検証した結果を示す.

2 関連研究

スペイン語文法誤り訂正の研究として Davidson ら[2]が行なったものがある．この研究では COWSL2H データセットを提案し，GEC モデルを学習させた結果を示している.スペイン語 GEC のベースラインとして”Bi-directional LSTM encoder + LSTM decoder”のモデルを使用している．COWLS2H には，スペイン語の L2 学習者（第二言語としてスペイン語を学ぶ者）が書いたエッセイとそれを人間が訂正したものが含まれる．モデルの訓練に用いる際には，エッセイを 1 文ごとに分割し，誤文と正文のペアにしたものが用いられている．さらに，1 文ごとに分割しペアにしたものを訓練用データ，検証

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

用データ,テストデータの 3 つに分割して使用する．

また，Flachs ら[4]が行なった研究もある．この研究では人工誤り生成手法を用いている．誤りを生成する際は[5]で提案された規則ベースの生成手法を用いて挿入，交換，置換，削除の操作を行い，誤りを生成する．置換の操作には Aspell Dictionary と Unimorph[6]，またはそれら両方を用いた場合のモデル性能を比較している．スペイン語においては，Aspell Dictionary のみを使用した場合が最も高精度であった．さらに，ノイズの多いデータを活用方法についても言及している．WikiEdit[7]，Lang8[1]の 2 つのデータセットを用いた場合で検証し，事前学習の後これらのデータをファインチューニングしたモデルが最も性能が良いという結果が示されている．WikiEdit は 4,871,833 文，Lang8 は 1,214,781 文が使用されている．

3 手法

本研究では提案するデータ拡張手法は，コーパスの誤りの抽出とデータへの誤りの注入の 2 段階に分かれている．

誤りの抽出には，正文と誤文から訂正箇所と誤りタイプを取得できる ERRANT[8]というツールを用いる．ERRANT は英語用に作られたツールであるが，[2]で述べられている通りスペイン語でも問題なく使用できる．しかし，より正確に使用できるようにするために以下の項目について訂正を行った．

- ADJ:FORM を ADJ:INFL に変更
- Determiner-noun agreement に対応する DET:INFL の追加
- NOUN:POS の削除

ERRANT で誤文の誤り部分（単語または文の一部の複数語）に 3 つの誤りタグ Missing(M)，Unnecessary(U)，Replacement(R)を付与し，訂正箇所とともに誤りパターンとして抽出する．この誤りパターンから MUR それぞれの操作対象となる単語（または文の一部の複数語）をその出現確率とともにノイズスキームへ格納した．各コーパスの MUR の割合は表 1 の通りである．

誤りの注入では，抽出操作で獲得した誤りタグ MUR と注入操作 Delete，Add，Replace を対応させたノイズスキームを用いる．Delete と Replace の操作は操作対象の文中の単語の中でノイズスキームに格納された単語に対して行われる．Add の操作ではノイズスキームに格納された単語を確率に基づいて選出し，任意の位置に挿入する．

表 1　各コーパスの MUR の割合

コーパス	M	U	R	総タグ数
Lang8	0.22	0.13	0.65	278502
COWSL2H	0.15	0.10	0.75	20080
Lang8+COWSL2H	0.21	0.13	0.66	298582

誤りの生成を行う際に 1 文あたりの誤り率と 1 文あたりの拡張数を設定できる． 1 文あたりの操作回数は 1 文あたりの誤り率と文の長さの積で決定する．

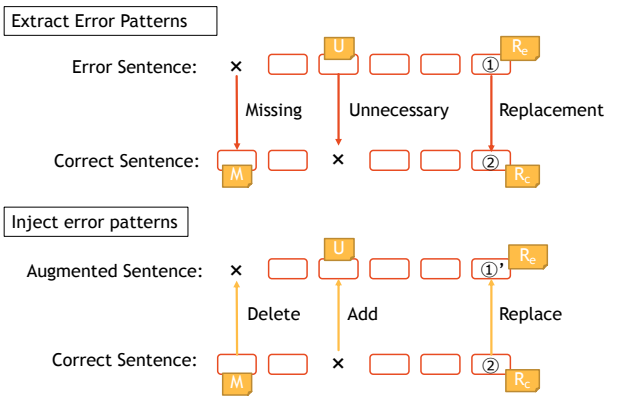

図 1　誤りタグと操作の対応

誤りパターンの抽出と誤りの注入操作は図 1 の通りに行われる．

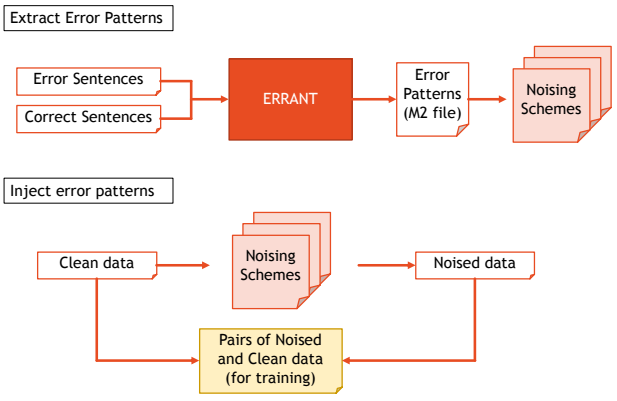

図 2　データ拡張手法の概略図

ERRANT を用いて抽出した誤りパターンから構築したノイズスキームを用いて訓練用の誤文と正文のペアを作成する過程を図 2 に示す．

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

4 実験

4.1 実験内容

まず，既存研究[4]で使用されている Aspell Dictionary を用いた手法と既存研究[2]の結果を比較した．次に，1 文あたりの誤り率を変化させた場合にモデルの性能にどのような影響があるかを検証し，誤りタグごとのモデルの性能についても検証した．

4.2 データセット

事前学習には WMT News Crawl から抽出した 10,000,000 の正文を用いた．抽出したデータを拡張して誤文を生成し，誤文と正文を組にして訓練用データとし，事前学習を行った．

ファインチューニングには GEC タスク用の誤文と正文が収録されている Lang8，COWSL2H を用いた．Lang8 にはスペイン語以外の言語が含まれるため，スペイン語のみを抽出し，Training と Validation に分割した．本研究では Lang8 で使用可能な部分を用い，データ数はそれぞれ 189,190，10,611 とした．COWSL2H は訓練用データ，検証用データ，テストデータに分割した．それぞれデータ数は 10,071，1,000，1,000 とした．

また，それぞれのデータについてデータクリーニングを行った．数字とスペイン語に使用される文字と記号以外の文字を削除し半角に変換した，単語と記号間を半角スペースで統一した．さらに，URL などの文字列を削除し，連続記号をその記号 1 文字に変換した．

4.3 データ拡張の設定

各操作の割合は GEC コーパスの Lang8 と COWSL2H から抽出したノイズスキームの各誤りタグの割合に従う．1 文あたりの誤り率は[4]との比較を行うため 0.15 に設定した．誤りを生成する際の候補の単語はコーパス中で３回以上誤っているものを使用した．

4.4 モデル

本研究では GEC を行う Encoder-Decoder モデルとして Transformer を用いた．具体的な設定として Vaswami ら[9]が定義した Transformer big を使用した．実装は fairseq 0.12.2 で行った．また，事前学習，ファインチューニングで設定したハイパーパラメータは表２と表３の通りである．ファインチューニング時には Early stopping を用いて検証損失が最も小さいモデルをテストに用いた．

表 2　事前学習時のハイパーパラメータ

Hyper parameters	値
Epochs	20
Max tokens	2048
Optimizer	Adafactor
Learning rate	3e-5
Warm up	8000 steps
Lr-Scheduler	Inverse Square Root
Loss function	Label Smoothed Cross Entropy
Dropout	0.3
Beam size	5

表 3　ファインチューニング時のハイパーパラメータ

Hyper parameters	値
Epochs	30
Max tokens	2048
Optimizer	Adafactor
Learning rate	3e-5
Lr-Scheduler	Inverse Square Root
Loss function	Label Smoothed Cross Entropy
Dropout	0.3
Beam size	5

4.5 評価方法

本研究では，Precision，Recall，F0.5 を評価手法として用いる．既存研究[4]と[2]の結果と比較を行うために F0.5 を MaxMatch(M2) scorer[10]と ERRANT[8]の２通りの方法で算出した．

5 結果

5.1 既存研究との比較

既存研究[4](Aspell_10m)の結果と本研究の提案手法(Errant_10m)を用いた結果, 既存研究と提案手法を合わせて用いた結果(A+E_10m)について比較を行った結果を表４に示す．全てのモデルを COWSL2H でファインチューニングした．＋(L)は Lang8 を追加でファインチューニングしたことを表す．

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

また，既存研究[2](Bi-LSTM+LSTM)との比較を行なった結果を表５に示す．ここでは，F0.5 を ERRANT で算出した．

表 4　既存研究[4]との比較

モデル	Prec.	Rec.	F0.5(M2)
Aspell_10m[4]	-	-	48.22
Aspell+WikiEdit+Lang8[4]	-	-	57.32
A+E_10m	66.81	31.42	54.53
Errant_10m	68.95	34.05	57.22
Errant_10m+(L)	**70.63**	**35.72**	**59.08**

表 5　既存研究[2]との比較

モデル	Prec.	Rec.	F0.5(ER)
Bi-LSTM+LSTM[2]	25.4	15.3	22.4
A+E_10m	61.42	30.31	50.96
Errant_10m	63.01	32.50	53.05
Errant_10m+(L)	**64.78**	**34.21**	**54.96**

表４と表５の結果からいずれの既存研究よりも本研究の提案手法を用いた場合の方が Precision, Recall, F0.5 の値が高い．既存研究[4]ではファインチューニング時に本研究の提案手法よりも多くのデータを用いているが，提案手法はより少ないデータでより高い性能を実現している．本研究ではファインチューニングに用いるデータと同質の誤りを事前学習用データに注入することでより良い結果を得ることができた．

5.2 1 文あたりの誤り率とモデル性能

提案手法で 1 文あたりの誤り率を変化させた結果を表６に示す. モデルは Errant_10m+(L)を使用した．

表 6　1 文あたりの誤り率とモデル性能

誤り率	Prec.	Rec.	F0.5(M2)
0.10	**70.81**	34.97	58.76
0.15	70.63	35.72	**59.08**
0.20	67.74	**35.96**	57.57

誤り率を大きくするほど Precision が低くなり，Recall が高くなった．F0.5 は 0.15 の場合に最大になった．このことから 1 文あたりに注入する誤り数を増やすと誤りの検出精度は上がるが，検出した誤りを正しく訂正できなくなると考えられる．特に誤り率 0.2 の場合の Precision が比較的大きく低下していることから，誤り率から算出される注入する誤り数が多くなりすぎることによって精度に悪影響を及ぼすと考えられる．また、F0.5 が最大となった誤り率 0.15 が提案手法において最適な値であった．

5.3 誤りタグごとのモデル性能

誤りタグごとのモデル性能を表７に示す．モデルは Errant_10m+(L)を使用した．

表 7　誤りタグごとのモデル性能

誤りタグ(操作割合)	Prec.	Rec.	F0.5(ER)
M (0.21)	50.00	29.29	43.80
U (0.13)	55.26	24.05	43.87
R (0.66)	69.19	36.63	58.75
ALL	64.78	34.21	54.96

Recall に着目すると誤りの注入回数が多いほど誤りをより多く検出できると考えられる．また，誤りの注入回数は U より M の方が多いのにも関わらず，F0.5 の値は M の方が低くなっている．これは Precision の値が U に比べ低いことが原因であるため，検出した誤りを正しく訂正できるよう，M に対する操作 Delete を改善する必要がある．また，U に対する操作 Add についても誤りをより検出できるよう改善することでよりモデルの性能を向上させることができると考えられる．

6 おわりに

本研究では，スペイン語の文法誤り訂正において学習者の誤りを再現するデータ拡張手法を提案し，事前学習用データを拡張して学習を行うことで既存研究よりも高い性能を実現した．スペイン語のようなデータが少ない言語の文法誤り訂正においてデータ拡張手法の有用性を示すことができた．

本研究ではコーパスの誤りを抽出，注入することでモデルの性能を向上させたが，今後はどのような誤りが性能に寄与するか，スペイン語特有の誤りがモデルの性能にどのように影響するかについて詳しい分析や検証が必要である．より詳細な分析を行うために本研究で使用しなかったスペイン語 ERRANT の誤りカテゴリ(ADJ，NOUN など)の認識性能を改良する必要がある．モデル性能の向上とともに今後取り組みたい．

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

参考文献

[1] **Tomoya Mizumoto, Mamoru Komachi, Masaaki Nagata and Yuji Matsumoto.** Mining Revision Log of Language Learning SNS for Automated Japanese Error Correction of Second Language Learners. In Proceedings of the 5th International Joint Conference on Natural Language Processing(IJCNLP), pp.147-155. Chiang Mai, Thailand, November 2011.

[2] **Sam Davidson, Aaron Yamada, Paloma Fernández Mira, Agustina Carando, Claudia H. Sánchez Gutiérrez, and Kenji Sagae.** Developing NLP Tools with a New Corpus of Learner Spanish. In Proceedings of the Twelfth Language Resources and Evaluation Conference, pages 7238-7243, Marseille, France. European Language Resources Association. 2020.

[3] **Jared Litchtarge, Chris Alberti, Shankar Kumar, Noam Shazeer, Niki Parmar, and Simon Tong.** Corpora generation for grammatical error correction. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 3291-3301, 2019.

[4] **Simon Flachs, Felix Stahlberg, and Shankar Kumar.** Data Strategies for Low-Resource Grammatical Error Correction. In Proceedings of the 16th Workshop on Innovative Use of NLP for Building Education Applications, pages 117-122, Online. Association for Computational Linguistics. 2021.

[5] **Jakub Náplava and Milan Straka.** Grammatical Error Correction in Low-Resource Scenarios. In Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019), pages 346-356, HongKong, China. Association for Computational Linguistics. 2019.

[6] **Christo Kirov, Ryan Cotterell, John Sylak-Glassman, Géraldine Walter, Ekaterina Vylomova, Patrick Xia, Mannal Faruqui, Sebastian Mielke, Arya McCarthy, Sandra Kübler, Davaid Yarowsky, Jason Eisner, and Mans Hulden.** UniMorph 2.0: Universal morphology. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan. European Language Resources Association (ELRA). 2018.

[7] **Adriane Boyd.** Using Wikipedia edits in low resource grammatical error correction. In Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text, pages 79-84, Brussels, Belgium. Association for Computational Linguistics. 2018.

[8] **Christopher Bryant, Mariano Felice, and Ted Briscoe. Automatic Annotation and Evaluation of Error Types for Grammatical Error Correction.** In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume1: Long Papers), pages 793-805, Vancouver, Canada. Association for Computational Linguistics. 2017.

[9] **Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin.** Attention is All you Need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, NeurIPS, pages 5998-6008. Curran Associates, Inc. 2017.

[10] **Daniel Dahlmeier and Hwee Tou Ng.** Better Evaluation for Grammatical Error Correction. In Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 568-572, Montréal, Canada. Association for Computational Linguistics. 2012.

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).