

仮想的な誤りタイプの割り当てによる解説文生成の性能向上

永田 亮

甲南大学/JST さきがけ

nagata-nlp2021 @ ml.hyogo-u.ac.jp.

埴 一晃

理研 AIP/東北大学

kazuaki.hanawa@riken.jp

1 はじめに

本稿では、解説箇所を精緻に認識することで解説文生成の性能を改善する汎用的な手法について述べる。解説文生成とは、与えられた文章に対して解説する箇所を決定し、ライティングに関する解説を生成するタスクである [8]。例えば、図 1 であれば、解説対象文 (S1)~(S3) の赤い下線部が解説箇所、(C1)~(C5) が対応する解説文である。解説文生成の性能は、解説箇所の推定精度に大きく左右される。5 で示すように、推定した解説箇所とオラクルの解説箇所とでは、生成性能に大きな差が生まれる。このことは、解説箇所推定に成功すると、解説文の生成にも成功する傾向にあることを示唆する。また、解説すべきでない箇所に対しては、どのような解説文を生成しても誤生成となることは明らかである。

解説箇所推定の改善法として、文法誤り情報付きコーパス [11, 13, 12] や疑似誤り生成 [17] を流用することが考えられるが、訓練データの不一致という問題が生じる。訓練データの不一致とは、解説箇所の訓練事例は存在するが対応する解説文が訓練データとして存在しないことをいう。例えば、疑似誤り生成で、『make+人+to 不定詞』のような使役動詞に関する誤り箇所が大量に得られたとしても、(C1) や (C2) のような関連した解説文が訓練データになれば、適切な解説文を生成することは困難である。そのような場合に無理に生成を行うと、無関係な解説文が生成されることは容易に想像できる。解説文生成においては、類似解説文が訓練データに存在する(言い換えれば、解説が可能な)ときのみ、解説箇所として同定することが重要となる。

そこで、本稿では、与えられた訓練データのみを用いて解説箇所推定の性能を向上させる手法を提案する(図 1 に基本アイデアを示す)。提案手法は、「同じ誤りタイプに分類される解説箇所は表層上類似した解説がなされる。」という仮説に基づく。例えば、図 1 では、(C1) と (C2) および (C3) と (C4) は

同じタイプの誤り箇所に対する解説文であり、表層上類似している。この仮説が正しければ、解説文間の類似度を手掛かりにして、解説箇所を誤りタイプごとにまとめ上げることができるはずである。複数の解説箇所を同一の誤りタイプとすることにより、「『使役動詞+人+to 不定詞』は解説箇所である」のような規則の獲得が促進されると期待できる。具体的な処理として、表層類似度に基づいて解説文をクラスタリングし、その結果得られたクラスターに仮想的な誤りタイプを割り当てる。得られた仮想誤りタイプを、訓練時に、解説箇所のラベルとして利用する。

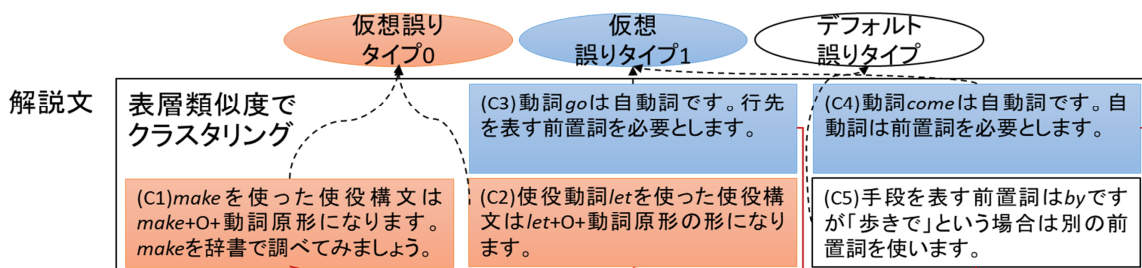
ここで強調しておきたいのは、あくまでも用いるのは仮想的な誤りタイプであるという点である。誤りの分類を人手で行う必要はないし、そもそも、なんらかの誤りの分類体系を仮定する必要もない。必要なのは、解説文が付与された英文データのみである。そのため、提案手法は幅広い機械学習アルゴリズムに汎用的に適用可能である。

本研究の貢献は次の三点である。第一に、仮想的な誤りタイプを自動的に決定し、解説箇所推定の性能を向上させる手法を提案する。基本的には解説文のクラスタリングというシンプルなアイデアに基づくが、解説文の言語的特徴を考慮した手法としている。第二に、提案手法が解説箇所推定と解説文の生成の両方に有意に効果があることを示す。第三に、実験結果の分析を通じて、仮想誤りタイプにより性能が向上する理由を明らかにする。

2 関連研究

解説文付きコーパスが利用可能になりつつある。文献 [8, 9] では、既存の学習者コーパスに対して、解説箇所と解説文の付与が行われている。文献 [14] では、linking word の用法に着目した解説文を含む学習者コーパスを構築している。

コーパスの公開と共に、解説文生成手法も増えつつある。ルールに基づいた手法 [5, 10]、テンプレートに基づいた手法 [7]、検索に基づいた手法 [8] など



解説対象文 (S1) It makes students to study. (S2) They let their child to go college. (S3) They came the place by foot.

図1 提案手法の基本アイデア：同じ色の解説文は表層類似度が高く、同じクラスタに分類されることを意味する。

が知られている。より一般的には、解説文生成は、解説対象文と解説箇所を入力とした言語生成問題と捉えることができる。したがって、各種のニューラル言語生成モデルが本タスクに適用可能である。そのような研究に文献 [19] がある。

3 解説箇所推定手法

提案手法は、次の5ステップからなる：

- Step (1) 解説文間距離の算出
- Step (2) 仮想誤りタイプの同定
- Step (3) 仮想誤りタイプの訓練データへの反映
- Step (4) 解説箇所推定モデルの訓練
- Step (5) 解説箇所の推定

なお、5の実験で用いるデータセットを踏まえ、以降は、解説対象文と解説文はそれぞれ英語と日本語であるとする。ただし、提案手法は解説対象文、解説文とも言語を選ばない汎用的な手法である。

Step (1) では、訓練データとして与えられた解説文の全ての組み合わせに対して距離を計算する。1で述べたように、表層上類似した解説文をまとめ上げたいので、編集距離¹⁾を距離の基礎とする。ただし、解説文の言語的特徴を考慮するため、次の二つの工夫を行う。

まず、解説文中の英単語列を特殊トークン（例：E1）に置換する²⁾。表層が同じ英単語列は同じ特殊トークンに置換する。例えば、図1の(C2)は、「使役動詞 E1 を使った使役構文は E1+E2+動詞原形の形になります。」のようになる。この処理には、参照している英単語は異なるが内容は類似する解説文間の編集距離を小さくする効果がある。

二つ目の工夫として、文間のアライメントをとり距離を求める。この処理は、文数が異なる解説文間の編集距離が必要以上に大きくなるのを避けるため

- 1) 本稿では文字ベースの編集距離とする。ただし、トークン分割を行えば単語ベースの編集距離も適用可能である。いずれの場合も、長さで正規化した編集距離を用いる。
- 2) 英語の場合でも、データセット [9] では、引用単語列を表す特殊なタグが付与されており同様の処理を行うことができる。

に行う。例えば、図1(C1)と(C2)は同種の誤りに対する解説文であるが、前者は二文からなるため、その分、編集距離は大きくなる。しかしながら、一文目のみに注目すると編集距離はずっと小さい。この問題を解決するために、編集距離が小さい文の組から順にアライメントする。ただし、一つの文は最大でも一つの文にしかアライメントできないとする。(C1)、(C2)では、一文目同士がアライメントされ、(C1)の二文目はアライメントされない。最終的な距離は、アライメントされた文間の編集距離の平均とする。したがって、(C1)、(C2)では、一文目のみで距離が決定されることになる。

Step (2) では、算出した距離を用いて解説文をクラスタリングする。クラスタリングの結果得られる各クラスタに、仮想的な誤りタイプ（例えば連番）を割り当てる。ただし、メンバ数が一定数以下のクラスタについては、解説箇所の推定が困難になることが予想されるため、まとめて一つの誤りタイプとする（以降、デフォルト誤りタイプと表記）。クラスタリングには、階層型クラスタリングの一種である single linkage clustering [3] を用いる。クラスタ間の距離に single linkage を用いるのは、同一内容の解説文でも様々な表層になりえることを考慮してのことである。すなわち、クラスタ内に一つでも類似した解説文が存在すれば、同じタイプの誤りとして認めるという方針である。クラスタリングの終了条件は、全てのクラスタ間の距離が閾値以上となったときとする。

Step (3) では、Step (2) で得られた仮想誤りタイプの情報を訓練データの解説箇所に反映する。単純に、仮想誤りタイプを訓練データ中の対応する解説箇所にラベルとして付与する。

Step (4) では、上述の訓練データを用いて解説箇所推定モデルの訓練を行う。解説箇所推定モデルとして、任意の分類（もしくは系列ラベリング）のための機械学習アルゴリズムを用いることができる。

本稿では、BiLSTM を用いた誤り検出手法 [6] を採用する。ネットワークの構成とハイパパラメータは付録 B に示すとおりである。

最終的に Step (5) で、解説箇所推定を行う。特別な処理はなく、通常の推論を行うだけである。

4 解説文生成手法

3 で提案した手法を用いると、解説すべきかどうかの情報だけでなく、(仮想的な) 誤りタイプの情報も得られる。この情報は、解説文生成にも有益であると予想される。幸いなことに、ニューラル言語生成モデルであれば、仮想誤りタイプの利用は容易である。本稿では、仮想誤りタイプの情報を分散表現としてデコーダへ入力する方法を採る。

ニューラル言語生成モデルとして、解説文生成の従来手法 [19] を参考にして、pointer-generator network [15] を利用する。ただし、解説文生成に次の二つの変更を加える(ネットワークの構成は付録 B の図 2 に示す)。解説箇所を考慮して解説文を生成するように、解説箇所に対応するエンコーダの隠れ状態³⁾をデコーダの初期状態とする。また、仮想誤りタイプもデコーダの初期状態とする。

5 評価実験

二つの観点(解説箇所推定と解説文の生成)で提案手法を評価した。両評価とも、前置詞解説文データセット [9] を用いた。同データは、トピック「アルバイト」(英文数 14,334; 解説箇所 2,906)と「喫煙」(英文数 14,495; 解説箇所 2,785)について書かれたエッセイを収録する。このデータをエッセイ単位で、85%, 7.5%, 7.5%の割合で分割し、それぞれ訓練、開発、評価データとした(詳細は付録 A に示す)。前置詞解説文では、1 単語からなる解説箇所が大部分を占める。解説箇所が複数単語にわたる場合は、中央の単語⁴⁾を解説箇所とし、常に解説箇所が 1 単語となるようにした。

各モデルの訓練はトピックごとに行った。仮想誤りタイプを得るためのクラスタリングでは、クラスタ間距離の閾値 0.10 と 0.15, 最小クラスタメンバー数 3 と 5 を用いて、開発データで解説箇所推定の $F_{1.0}$ 値が最大となる組み合わせを選択した。ネットワークのハイパパラメータの値も同様に決定した(詳細を付録 B に示す)。

3) 解説箇所が複数単語からなる場合は平均ベクトルとする。

4) 解説箇所の単語数が偶数となる場合は中央左側の単語を解説箇所とした。

比較のため、仮想誤りタイプを用いない手法の性能も評価した。仮想誤りタイプ以外は提案手法と同一とした。以降では、表記を簡潔にするため、この手法を便宜的に従来手法と呼ぶことにする。

評価尺度として recall, precision, $F_{1.0}$ を用いた。解説箇所推定については、仮想誤りタイプの一致までは見ず、解説箇所かどうかの 2 値分類として評価した。解説文の生成については、第一著者と第二著者が、正解の解説文を参照しながら適切な解説文かどうかを独立に判定した。不一致箇所は相談の上、最終的な評価を決定した。

表 1 に、「アルバイト」と「喫煙」に対する出力をまとめてから性能値を求めた結果を示す(個別の評価結果は付録 C に示す)。この表から、仮想誤りタイプにより、recall, precision とともに改善することがわかる。実際、 $F_{1.0}$ の差は、どちらのタスクにおいても有意であった(permutation test; $p = 0.002$ (解説箇所推定), $p = 0.041$ (解説文生成))。

参考として、オラクルの解説箇所に対する従来手法の生成性能を評価したところ、 $F_{1.0} = 0.43$ となり、推定した場合とでは大きな差があることが明らかとなった。この値は、ある意味での性能限界と捉えることができる。提案手法は、従来手法と同一の訓練データを使用しているにもかかわらず、従来手法の性能と性能限界の中間地点付近まで性能を改善していると解釈することもできる。

6 考察

実験結果を分析したところ、提案手法で解説箇所推定に成功し、かつ、仮想誤りタイプが出力できたのは 72 箇所であった。その内 62 箇所については従来手法でも推定に成功した。したがって、仮想誤りタイプが出力できる箇所は、従来手法でも解説箇所推定に成功する傾向にあるといえる。デフォルト誤りタイプでなく仮想誤りタイプが出力されるということは、同種の誤り事例がそれなりの数、訓練データに存在することを意味する。なぜなら、メンバー数が一定以上のクラスタのみを仮想誤りタイプとして採択するからである。訓練事例が一定数ある誤りを解

表 1 性能評価結果 (R: Recall, P: Precision)。

タスク 手法	解説箇所推定			解説文生成		
	R	P	$F_{1.0}$	R	P	$F_{1.0}$
提案手法	0.35	0.71	0.47	0.27	0.53	0.36
従来手法	0.33	0.64	0.43	0.24	0.48	0.32

説箇所として認識するのは比較的容易であろう。

では、なぜ提案手法の性能は良いのであろう。実は、上述の理由が従来手法における precision の低下も説明する。一定数事例がある場合、規則の過剰一般化を引き起こすこともある。最悪の場合、解説箇所周辺の数単語を誤って規則としてしまう。実験結果からは、そのような傾向が確認された。例えば、“concentrating our attention on studying” のような誤検出が確認できた。訓練データに“attention on”という誤りが多数⁵⁾あるため、この単語列を規則とした可能性が高い⁶⁾。一方、仮想誤りタイプを利用すると、複数の事例を一つの誤りタイプとして認識できるため、解説箇所とそうでない箇所の相違がより明確になる。例えば、“pay attention on”や“pay attention for”などが同じ誤りタイプと認識されたため、この種の誤りが成立する必要条件に on は含まないと学習される可能性が高い(相対的に、pay が含まれる可能性が高くなる)。同様な例として、従来手法では、“We can just set a smoking place.”を誤検出した(『助動詞+to 不定詞』を過剰に一般化した結果であると分析できる)。一方、提案手法では、“can/must/will to 不定詞”など、必ず to を含んだ解説箇所が仮想誤りタイプのメンバとなったため、事例間の共通点を学習するのがより容易である。

仮想誤りタイプにより recall が向上する例も確認できた(例：“smoking is banned at anywhere”, “must to set”)。こちらも上と同様に説明できる。例えば、前者の場合は、訓練データに、“banned at all the restaurants”を含む文が 463 もあるため⁷⁾、従来手法では、単語列“banned at”は正しいと判定する規則が獲得された可能性が高い。一方、提案手法では仮想誤りタイプにより、“at/in anywhere/somewhere”などが同じ誤りタイプにまとめられたため、適切な規則『前置詞+anywhere のような単語』が獲得されたと分析できる。

仮想誤りタイプと同様な効果がマルチタスク学習により得られる可能性がある。すなわち、解説箇所推定と生成を同時に行うネットワークである。実際に、従来手法の二つのネットワークを組み合わせた手法を評価したところ、解説箇所推定で $F_{1.0} = 0.41$

5) 例えば、「アルバイト」では少なくとも 8 例確認できる。

6) ニューラルネットを用いているので、実際にこのようなハードな規則が獲得されているわけではないが、出力を分析するとそのような振る舞いが観測される。

7) トピック「喫煙」は、正確には“Smoking should be completely banned at all the restaurants in the country.”という英文で与えられる。そのため、書き手は同様な表現を頻繁に使用している。

となり、従来手法の性能すら超えられなかった。解説文は表層のバリエーションが多く(図 1(C1)のようにオプションな情報を含むものが多い)、訓練事例が独立に与えられるマルチタスク学習の設定では、解説文間の類似性を正確に把握するのはより難しい。言い換えれば、提案手法のように、文間のアライメントをとり直接的に類似度を算出することが重要となる。更に、限られた訓練データで、より複雑なネットワークの訓練を行うことの難しさも理由として挙げられる。

解説文の生成における仮想誤りタイプの効果も顕著である。仮想誤りタイプが出力された箇所とデフォルト誤りタイプが出力された箇所の生成正解率は、それぞれ 71.8%と 41.7%であり、大きく異なる。ただし、上述のとおり、仮想誤りタイプは一定数以上の訓練事例があることを意味するので、(頻度という意味で)比較的生成が容易な事例であることに注意する必要がある。すくなくとも、仮想誤りタイプが推定できた場合は、解説文の生成にも正解する可能性が高いということはいえる。このことは、1 で述べた解説文生成における要件「解説可能なときのみ解説箇所として同定することが重要」を満たすという点で好ましい特性である。

一方で、仮想誤りタイプ内の事例が同一の解説文に偏ったため生成の柔軟性が失われる傾向も確認した。例えば、「most of の後ろにはある特定の集団が入ります。単に学生一般を指す場合は、most を名詞ではなく形容詞として使いましょう。」という解説文の数が非常に多く、同種の別の誤り(例：most of people)に対しても、「学生一般」という表現を生成してしまっていた(本来は、「人々一般」などを生成しなければならない)。この例のように、少し書き換えると適切になる解説文も正しいと認めると生成正解率が 71.8%から 80.0%まで向上した。一方、デフォルト誤りタイプが出力された箇所の正解率は 41.7%から 45.5%となり変化が少ない。このことから、上述の傾向が確認できる。

7 おわりに

本稿では、解説文の言語的特徴を利用して仮想誤りタイプを割り当て、解説箇所推定と解説文生成の性能を向上させる手法を提案した。提案手法は、シンプルかつ汎用的な手法にもかかわらず、両タスクの性能を有意に改善することを示した。また、実験結果の分析により、性能改善の理由を示した。

参考文献

- [1] Bies, Ann, et al. English web treebank LDC2012T13. Web Download, 2012.
- [2] Daniel Blanchard et al. ETS corpus of non-native written English, ldc2014t06. web download, 2014.
- [3] J.C. Gower and G.J.S. Ross. Minimum spanning trees and single linkage cluster analysis. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, Vol. 18, No. 1, pp. 54–64, 1969.
- [4] Sylviane Granger. The international corpus of learner English. In *English language corpora: Design, analysis and exploitation*, pp. 57–69. Rodopi, 1993.
- [5] Jun'ichi Kakegawa, Hisayuki Kanda, Eitaro Fujioka, Makoto Itami, and Kohji Itoh. Diagnostic processing of Japanese for computer-assisted second language learning. In *Proc. of 38th Annual Meeting of the Association for Computational Linguistics*, pp. 537–546, 2000.
- [6] Masahiro Kaneko, Yuya Sakaizawa, and Mamoru Komachi. Grammatical error detection using error- and grammaticality-specific word embeddings. In *Proc. of 8th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 40–48, 2017.
- [7] Yi-Huei Lai and Jason Chang. TellMeWhy: Learning to explain corrective feedback for second language learners. In *Proc. of 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pp. 235–240, 2019.
- [8] Ryo Nagata. Toward a task of feedback comment generation for writing learning. In *Proc. of 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 3197–3206, 2019.
- [9] Ryo Nagata, Kentaro Inui, and Shin'ichiro Ishikawa. Creating Corpora for Research in Feedback Comment Generation. In *Proc. of the 12th Language Resources and Evaluation Conference*, pp. 340–345, 2020.
- [10] Ryo Nagata, Mikko Vilenius, and Edward Whittaker. Correcting preposition errors in learner English using error case frames and feedback messages. In *Proc. of 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 754–764, 2014.
- [11] Ryo Nagata, Edward Whittaker, and Vera Sheinman. Creating a manually error-tagged and shallow-parsed learner corpus. In *Proc. of 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 1210–1219, 2011.
- [12] Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadwinoto, Raymond Hendy Susanto, and Christopher Bryant. The CoNLL-2014 shared task on grammatical error correction. In *Proc. 18th Conference on Computational Natural Language Learning: Shared Task*, pp. 1–14, 2014.
- [13] Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadwinoto, and Joel Tetreault. The CoNLL-2013 shared task on grammatical error correction. In *Proc. 17th Conference on Computational Natural Language Learning: Shared Task*, pp. 1–12, 2013.
- [14] Ildiko Pilan, John Lee, Chak Yan Yeung, and Jonathan Webster. A Dataset for Investigating the Impact of Feedback on Student Revision Outcome. In *Proc. of 12th Language Resources and Evaluation Conference*, pp. 332–339, 2020.
- [15] Abigail See, Peter J Liu, and Christopher D Manning. Get To The Point: Summarization with Pointer-Generator Networks. In *Proc. of 55th Annual Meeting of the Association for Computational Linguistics*, pp. 1073–1083, 2017.
- [16] Masatoshi Sugiura, Masumi Narita, Tomomi Ishida, Tatsuya Sakaue, Remi Murao, and Kyoko Muraki. A discriminant analysis of non-native speakers and native speakers of English. In *Proc. of Corpus Linguistics Conference CL2007*, pp. 84–89, 2007.
- [17] Ziang Xie, Guillaume Genthial, Stanley Xie, Andrew Ng, and Dan Jurafsky. Noising and denoising natural language: Diverse backtranslation for grammar correction. In *Proc. of 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 619–628, 2018.
- [18] Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. A new dataset and method for automatically grading ESOL texts. In *Proc. of 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 180–189, 2011.
- [19] 塙一晃, 永田亮, 乾健太郎. 高信頼度な文法誤り解説生成のための生成制御手法. 人工知能学会全国大会論文集, pp. 2D1GS903–2D1GS903, 2020.

A 評価実験に用いた前置詞解説文データセットの詳細

表 2 に、評価実験に用いた前置詞解説文データセットの詳細を示す。基本的には、文献 [9] の前置詞解説文データセットであるが、未公開のデータも使用した。これらのデータも公開予定である。

B ネットワークの構成とハイパラメータの設定

B.1 解説箇所推定モデル

BiLSTM の層数は 2 に固定し、隠れ状態の次元 200, 600, 800 をサーチした。それ以外は次のように固定した：バッチサイズ 32；early stopping（エポック数 100, patience 10）；ドロップアウトレート 0.5。訓練は 5 回行い、各訓練結果から得られる出力（確率）の平均値により最終的な判定を決定した。

最終的に決定されたハイパラメータは、次のとおりである。トピック「アルバイト」については、BiLSTM の隠れ状態の次元 600, 仮想誤りタイプを割り当てるクラスターの最小メンバ数 5, クラスタ間距離の閾値 0.15 となった。同様に、「喫煙」では、同じ順で、800, 3, 0.15 となった。更に、従来手法では、BiLSTM の隠れ状態の次元は、800（「アルバイト」）と 600（「喫煙」）となった。

B.2 解説文生成モデル

図 2 に、pointer-generator に基づいた解説文生成モデルの概要を示す。濃い赤で示された部分が、オリジナルの pointer-generator からの変更点である。

ネットワークのハイパラメータは文献 [19] を参考にして次のように設定した。エンコーダは BiLSTM（層数 1, 隠れ状態の次元 300）とした。デコーダは、LSTM（総数 1, 隠れ状態の次元 300）とした。バッチサイズ 32, エポック数 50, ドロップアウトレート 0.5。訓練は 5 回行い、開発データ中の正解解説文との BLEU の平均値が最大となる訓練結果を採択した。

B.3 両モデルに共通する設定

英単語の分散表現は、アルゴリズムを CBOW, サイズを 200 次元に固定し、次のコーパスから得た：評価実験で用いた訓練、開発データの解説対象英文, CEEJUS⁸⁾, CLC FCE [18], ETS [2], ICLE [4],

8) <http://language.sakura.ne.jp/s/doc/projects/CEEJUS.pdf>

NICE [16]（以上、学習者コーパス）, EWT [1]（母語話者コーパス）。この分散表現を解説対象文の英単語の分散表現とした。また、解説文中に出現する英単語の分散表現としても使用した。その他の日本語単語については、ランダムな値で初期化した。日本語の形態素解析器として MeCab⁹⁾を用いた。

各モデルの最適化アルゴリズムは Adam（step size: 0.001, first/second moment: 0.9/0.999）とした。

C 付録：詳細な評価結果

表 3 と表 4 に、トピック「アルバイト」と「喫煙」を別々にして評価した性能を示す。

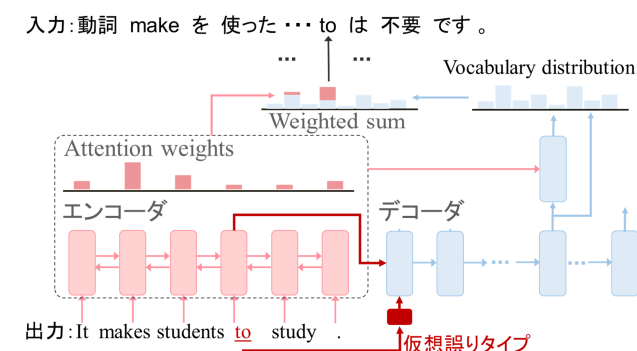


図 2 解説文生成モデルの構成。

表 2 評価実験に用いたデータの統計量。

トピック 分割	アルバイト			喫煙		
	訓練	開発	評価	訓練	開発	評価
英文数	12,163	1,129	1,042	12,312	1,160	1,023
解説箇所	2,439	244	224	2,342	230	214

表 3 トピック別の解説箇所推定性能。

トピック 手法	アルバイト			喫煙		
	R	P	F _{1.0}	R	P	F _{1.0}
提案手法	0.353	0.687	0.466	0.355	0.731	0.478
従来手法	0.326	0.635	0.431	0.322	0.651	0.431

表 4 トピック別の解説文の生成性能。

トピック 手法	アルバイト			喫煙		
	R	P	F _{1.0}	R	P	F _{1.0}
提案手法	0.250	0.487	0.330	0.285	0.587	0.384
従来手法	0.237	0.461	0.313	0.248	0.500	0.331

9) <http://taku910.github.io/mecab/>