文法誤り解説文生成とはどのようなタスクなのか？

永田　亮†　　塙 一晃††
† 甲南大学知能情報学部　†† 理研 AIP
E-mail: †nagata-nlp2020@ konan-u.ac.jp., ††kazuaki.hanawa@riken.jp

1. はじめに

解説文生成とは，与えられた文書に対してライティング技術に関する解説を生成するタスクである．例えば，

例 (1)：*The exercise is good to me.*

形容詞 *good* の後に前置詞 *to* を続けると「〜に対して親切だ」という意味になります。「〜にとってよい」となるように前置詞を選んでみましょう。

例 (2)：*I agreed ___ it.*

動詞 *agree* は自動詞ですので目的語の前に前置詞が必要です。辞書で *agree* を調べてみましょう。

のような解説を生成するタスクである．文献 [3] で，解説文付き学習者コーパスが公開され，今後より一層，解説文生成研究が盛んになることが予想される．

一方で，解説文生成タスクそのものについて分かっていないことも多い．そもそも，解説文生成が真に生成タスクかどうかということすら確かではない．解説文は自然言語で記述されるので，一見，生成問題のように見える．しかしながら，解説文の種類数がそれほど多くなければ，分類問題として解けるかもしれない．仮に，生成問題であれば，現実的に解ける問題かどうかということも重要となる．更に，解説文生成手法をどのように効率的に評価するかという課題も残る．

そこで，本稿では，次の三つの疑問について探求する：

Q1： 解説文生成はどの問題クラスに分類されるのか？
Q2： 訓練データの量と性能の関係は？
Q3： 効率的な評価方法は存在するのか？

これらの疑問に答えるため解説文を人手で分類した．具体的には，解説文付きコーパス [3] に収録された，前置詞の用法に関する解説文 3,355 に対して階層的な分類コードを人手で付与した．その結果を利用して，何割の解説文が分類問題として解けるかを示す．Q2 ついては，解説文の総数と種類数の関係をグラフとして表す．その結果から，検索に基づいた手法の性能上限を示す．Q3 については，BERT を利用した自動評価の可能性を調査する．

2. 基本アプローチ

2.1 対象データと付与方法

解説文付きコーパス [3] の 1,070 のエッセイに付与された 3,355 の解説文を対象データとした．エッセイのトピックは，アルバイトに関するものと喫煙に関するものの二種類である（以降，それぞれ PTJ と SMK と省略する）．

付与作業前に全ての解説文に一度目を通した．その結果，多くの解説文は階層的に分類されることがわかった．そこで，階層的なコードを作成することとした．また，階層コードが捉えられない情報を含めるために補助コードも作成した．

作成後，階層コードと補助コードを各解説文に付与した．付与作業の過程で，両コードに適宜修正を加えた．その後もう一度，付与作業とダブルチェックを行った．

2.2 階層コード

本節では，階層コードの概要を述べる．図 1 に，作成した階層コードの一部を示す（括弧内の数字は，その分類コードが適用された解説文の数を表す (注 1)）．以降，1. で導入した例 (1) と (2) を適宜参照してコードの概要を説明する．

2.2.1 第一階層: 係りタイプ

解説文の内容は，対象となっている前置詞が係る単語に大きく影響を受ける．例えば，動詞に係る前置詞は，動詞特有の解説文が付与されることが多い．

そこで，第一階層では係る単語の品詞情報をコードとして表すことにする．品詞体系として Penn Treebank POS tag を流用した．ただし，VBZ や VBP など細分類が重要でない場合は，代表のタグを用いた（この例では VB）．例 (1) と (2) は，それぞれ，JJ と VB が付与される．

例外的なケースに対応するため，独自のコードも作成した．具体例には，ID (Idiom: イディオム内の前置詞)，CPP (Compound Prepositional Phrase; 複合前置詞．例：*according to*)，DPP (Deverbal Prepositional Phrase; 動詞派生前置詞．例：*including*）などがある (注 2)．解説文によっては，係り先の情報が明示されていない場合やルートノードに係る場合がある．その場合は，デフォルトコード PP を使用した．

2.2.2 第二階層: 誤りタイプ

訂正タイプは，当該前置詞の誤りタイプの情報をコード化する．具体的には，抜け（MISSING），余剰（EXTRA），置換（CHOICE）の三タイプがある．この三タイプは，文法誤り訂正の分野で頻繁に使用されるものである．例えば，例 (1) と (2) は，それぞれ，JJ-CHOICE と VB-MISSING とコード化される．

(注 1)：一部の階層コードは省略されているため，括弧内の数字の総和は解説文の総数（3,355）に一致しない．また，表記が一部変則的であるので注意が必要である．第一階層の総和が，第二，第三階層の総和と一致するような表記となっている．
(注 2)：係り先ではないものもあるが便宜的に係りタイプとした．

2.2.3 第三階層: 誤りサブタイプ

誤りタイプを細分類し，サブタイプとしてコード化する．18 種類のサブタイプが存在するが，ここでは主要なものについて説明する(注 3)．

コード VT と VI は，他動詞と自動詞の使い分けに関する誤りである．学習者は，しばしば両者の使い分けを誤る．例えば，他動詞と目的語の間に前置詞を置く誤りがある．逆に，例 (2) のように，目的語の前の前置詞を省略する誤りもある（この例は VB-MISSING-VI とコード化される）．

CAT は，前置詞の抜けにより同じカテゴリに属する二つの句が連結した誤り（例：**a kind animal*; *a kind of animal* の意）を表す．具体例として，

> 名詞で別の名詞を前から修飾する場合には、両者を連結するための前置詞が必要です。

を挙げることができる．

SPP (Subjective Proportional Phrase) は，主語に前置詞を付けた誤りを表す（例：**At the store is near my place.*）．ある種の言語では，主語を表すために前置詞（あるいは後置詞）を用いるが，英語の場合，通常は無標識である．この規則に従わない誤りに SPP を用いる．

NVP (Nominal Verb Phrase) とは，動詞句を名詞句のように使用した誤りである．典型例に，動詞句を主語として使用した誤り（例：** ___ Learn English is difficult.*）がある．

細分類不能な誤りについては，デフォルトの誤りサブタイプを用いる．単純に，上層の誤りタイプ（MISSING，EXTRA，CHOICE）を用いる．例えば，例 (1) には，CHOICE が適用され，その結果，JJ-CHOICE-CHOICE となる．

この階層までで，解説文の内容をある程度表せる．例えば，

> PP-MISSING-NVP
> 動詞句は、そのままでは名詞句として使用できません。*to* を加えることで名詞化する必要があります。

のようにコードで解説文の内容を表すことが可能である．

2.2.4 第四階層：係り先

第四階層では，前置詞の係り先の語をコードとして用いる．例えば，例（1）と（2）では，それぞれ *good* と *agree* がコードとなる（コード全体は，JJ-CHOICE-CHOICE-good と VB-MISSING-VI-agree となる）．

解説文中に係り先が明示されていない場合がある．そのような場合は，ANY というコードを用いる．また，特定できない場合は?を使用する．

2.2.5 第五階層：訂正情報

第五階層では，訂正情報をコード化する．訂正情報は誤りの前置詞と正しい前置詞とからなる（例：*to→for*）．例（1）であれば，JJ-CHOICE-CHOICE-good-*to→for* というコードを付与する．

(注 3)：詳細については，コード化したデータと共に公開予定のガイドラインに収録している．

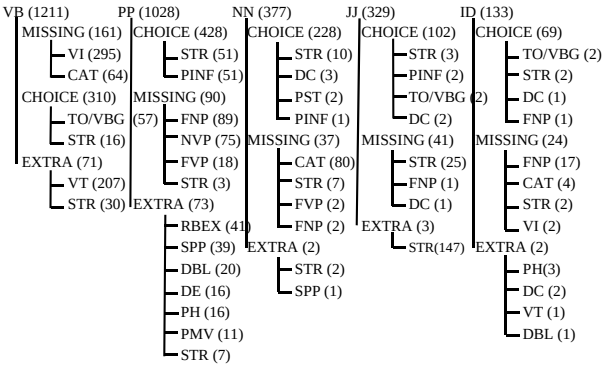

図 1： 作成した階層コード（一部）とその使用頻度．

前置詞の抜けおよび削除は NONE を用いて表す (例：*NONE→with*)．例（2）であれば VB-MISSING-VI-*agree-NONE→with* となる．

2.2.6 第六階層：前置詞の目的語

解説文によっては，前置詞の目的語に言及するものがある．例（1）であれば，前置詞 *with* の目的語 *me* に言及している．このような場合，目的語そのものをコードとして使用する．したがって，例（1）には， JJ-CHOICE-CHOICE-good-*to→for-me* というコードを与える．目的語が明記されていない場合には NONE を用いる．

以上が，全ての階層である．これらのコードを用いると，二つの解説文が同一であるかどうかある程度判定できる．例えば，表層は異なる次の二つの解説文：

> 動詞 *agree* は自動詞ですので、目的語 *opinion* は前置詞を必要とします。

と

> 動詞 *agree* は自動詞です。したがって、*opinion* の前に適切な前置詞が必要となります。

は，同一のコード VB-MISSING-VI-*agree-NONE→with-opinion* が与えられるため，同一内容であると判定できる．

2.3 補助コード

階層コードに加え補助コードも用いる．補助コードは，階層的でない，(多くの場合，オプショナルとなる）情報をコード化するのに用いる．本節では，本稿に関係するコードの概要について述べる．

2.3.1 正誤意味情報

解説文は，正しい／誤りの前置詞に関する意味情報に言及することがある．例えば，例（1）は，正しい前置詞を用いた場合の意味（i.e.,「〜にとってよい」）と誤りの前置詞を用いた場合の意味（i.e.,「〜に対して親切だ」）を説明している．

この情報をコード化するために，COR_MEANING（正しい前置詞）と INCOR_MEANING （誤りの前置詞）を用いる．例（1）は両コードとも適用される．一方，例（2）どちらのコードも付与されない．

2.3.2 残りの情報

以上のコードを用いても，解説文内の全ての情報を表現で

きない場合がある．解説文の分類という観点からは，表現できていない情報があるということを明示するのは重要である．

そこで，表現できていない情報がある場合は，補助コード REMAINING_INFO を付与する．例えば，

> 動詞 *face* は他動詞であるため，目的語の前に前置詞 *with* を必要としません．ただし，受動態の場合は，動作主を表すために前置詞 *by* を用います．

に対して，コード VB-EXTRA-VT-*face-with*→*NONE-ANY* を付与したとすると，一文目の情報は全てカバーされる．しかしながら，二文目の情報は全くカバーされない．このような場合，REMAINING_INFO を用いる．

3. Q1：どの問題クラスに分類されるのか？

3,355 の解説文に階層コードを付与したところ，96.9%の解説文に第六階層までのコードを付与することができた．残りの 3.1%については，準備したコードのいずれにも該当しなかった．多くの場合，二種類以上の文法規則を記述しており，単一の規則を想定している本研究のコードではうまく取り扱うことができなかった．また，前置詞誤りを対象にしてはいるが，前置詞を使わず，文全体を書き換えるように勧める解説文も存在した．

96.9%の解説文のうち 43.3%は，デフォルト（MISSING, CHOICE，EXTRA）以外の誤りサブタイプが割り当てられた．これらについては，コードに対応する既定の解説文を作成することで，内容をある程度解説できる．例えば，コード VB-MISSING-VT-*approach-to*→*NONE* に対して，

> 動詞 *approach* は他動詞なので目的語は前置詞 *to* を必要としません。

という解説文を作成すれば，少なくとも誤りの理由の一部を説明できる．言い換えれば，分類問題として解ける．ただし，43.3%のうち，5.2%については分類問題としては解決できない情報も含んでいた（すなわち，補助コード REMAINING_INFO が付与されている）．したがって，これら 5.2%については，人間と同様な解説文を実現しようとすると，生成問題として解く必要がある．

残りの 53.6%（=96.9 − 43.3）のうち 36.4%については，補助コード COR_MEANING または INCOR_MMEANING が付与されていた．すなわち，正しい／誤りの前置詞を使用した場合の意味情報が記載されていた．これらの内容については，例えば，辞書に記載されている当該の意味（もしくは用法）を適切に参照することで生成できる可能性がある．これはある種のグランディング問題と捉えられる．以上をまとめると，解説文生成問題のうち約 80%（43.3+36.4）については，分類もしくは辞書へのグランディングとして解けることを示唆する．ただし，先ほどと同様に，80%のうち 11.4%については REMAINING_INFO が付与されていた．

それ以外の約 20%（分類不能の 3.1%を含む）については，階層コードと補助コードだけでは十分に情報がコード化できない結果となった．これらについては，生成問題として解く必要があることを示唆する．

4. Q2：訓練データの量と性能の関係は？

Q2 に答えるため解説文の総数と種類数の関係を調査した．次のように総数と種類数を求めた：(i) コード化した解説文を一度に一つランダムにサンプリング；(ii) それまでにサンプリングした解説文の総数と種類数を計数（同じ解説文かどうかの判定は次のパラグラフで述べる）；(iii) (i) と (ii) を 100 回繰り返し，平均値を求める．以上の手順を PTJ と SMK 別々に行った．同様に，SMK が与えられたときの PTJ における総数と種類数の関係も調査した（更に，逆の組み合わせでも行った）．これは，あるトピックについて既に一定量の解説文のデータが利用可能であるときに，両者の関係がどのように変化するかを見積もるためである．

同一の解説文とみなす条件は，(a) 二つの解説文のコードが第六階層まで同じであること，かつ (b) 両方とも補助コード REMAINING_INFO が付与されていないこととした．これは，二つの解説文が同じカテゴリに分類され，かつ，コード化されていない情報がないという規則に相当する．

図 2 に結果を示す．横軸と縦軸は，それぞれ解説文の総数と種類数を表す．図 2 より，総数と種類数の関係は線形でないことがわかる．また，同図により，例えば，1,600 の解説文のうち約 38%については一つ以上の重複があることもわかる．容易に予想されるように，別のトピックにおける解説文データが与えられると，更に重複度合いは大きくなる．

重複度合いは，性能限界についても示唆を与える．例えば，あるトピックについて 1,600 の解説文からなる訓練データが与えられたとき，同じトピックについて書かれたエッセイに対して解説文を生成することを考えてみよう．誤りの分布が変わらないという仮定が成り立つとすると，上の議論より，38%の誤りについては，適切な解説文が訓練データ中にあることになる．したがって，検索に基づいた手法の性能限界は，再現率 0.38，適合率 1，F 値は 0.55 となる．以上の考察は，Q1 に対して，解説文生成の一部は検索問題として解けるという別の回答も与える．

5. Q3：効率的な評価方法は存在するのか？

本節では，二つの解説文の内容が同一かどうかを判定する分類器について考える．もし，任意の二つの解説文の同一性が高精度に判定できれば，効率的な評価につながる．分類器の入力は，人手で作成した解説文と生成結果の二種類の解説文である．出力は，入力の二種類の解説文が同一内容である確率である．具体的には，まず，入力である二つの解説文を

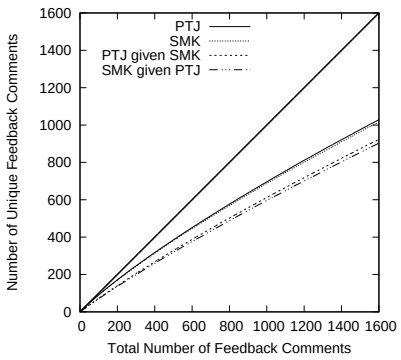

図 2： 解説文の総数と種類数の関係.

BERT でベクトルに変換する (注 4)．解説文が日本語で記述されているため，日本語の BERT(注 5)を用いる．日本語 BERT を用いると，英単語の大部分は未知語となる．そのため解説文中に出現した全ての英単語の分散表現を平均したベクトルも入力として利用する．この平均ベクトルと BERT で得られる解説文ベクトルを二層の順伝播型のニューラルネットに入力し上述の確率を推定する．

分類器の訓練は次のように行う．まず，4. の方法を用いて同一内容である解説文のペアを作成し，正例とする．負例については，正例でないものをランダムにペアにすることで得る．この正例と負例を一対一の比率で合わせたものを訓練データとする．この方法により，PTJ，SMK，その両方を組み合わせたもの，それぞれについて訓練を行う．

生成結果を得るための手法として， 文献 [1] の手法を用いる．この手法自身の訓練については，解説文データセット [3] を用いた（詳細は表 1 の通り）．

自動評価のための分類器の性能評価は正解率を用いた．正解率は，入力の二つの解説文が同一内容かどうかを人手で判定した結果と分類器の判定結果の一致率とした．人手評価は，英語指導（2 年間）と英語統語アノテーション（10 年以上），両方に経験がある作業者に依頼した．以上に加えて，一方のトピック（PTJ または SMK）についての判定結果を用いて分類器をファインチューニング (注 6)，もう一方のトピックについて正解率を求めた．

表 2 に結果を示す．一見すると，ファインチューニングを行った場合の正解率は十分に高いように見える．しかしながら，この結果はベースラインの正解率が高いということを反映しているに過ぎない；人手の判定では，PTJ および SMK

(注 4)：正確には，BERT が出力する [CLS] に対応するベクトルを解説文ベクトルとした．ハイパパラメータは次の通り：英単語の分散表現：GloVe；順伝播型ニューラルネット：二層（200 次元の ReLU と softmax），バッチサイズ：32；最適化アルゴリズム：Adam (ステップサイズ 0.001，一次と二次のモーメント：0.9 と 0.999)；エポック数：開発データにより決定.

(注 5)：http://nlp.ist.i.kyoto-u.ac.jp/nl-resource/JapaneseBertPretrainedModel/Japanese_L-12_H-768_A-12_E-30_BPE.zip

(注 6)：二層の順伝播型のニューラルネットのみファインチューニングした.

表 1： 解説文生成手法 [1] の実装に用いたデータに関する統計量.

Topic	PTJ			SMK		
スプリット	訓練	開発	評価	訓練	開発	評価
エッセイ数	779	72	72	778	72	72
文数	11,848	1,058	1,065	11,966	1,135	1,079
コメント数	2,407	212	208	2,271	229	206

表 2： 自動評価の結果

トピック	チューニング	正解率	再現率	適合率	F 値
PTJ	あり	0.913	0.500	0.722	0.591
	なし	0.726	0.500	0.228	0.313
SMK	あり	0.898	0.357	0.769	0.488
	なし	0.757	0.786	0.333	0.468

それぞれにおいて，87.4%と 86.4%の事例について同一内容でないという結果であった．PTJ と SMK を併合して，分類器の正解率とベースラインの正解率の差を検定したところ有意であったが（有意水準 5%；マクネマー検定），その差は小さい（分類器の正解率：0.906，ベースラインの正解率：0.870）．このことは，適合率と再現率にも反映されている．

比較のため，BLEU-4 による評価についても調査を行った．まず，生成結果と人手で作成した解説文の間の BLEU を求めた．その値と同一内容かどうかを人手で判定した結果（1 or 0）の相関係数を求めたところ，0.730（PTJ）と 0.678（SMK）であった．これは，ファインチューニングした分類器の出力する確率と人手による判定結果との間の相関係数より高い：0.595（PTJ），0.519（SMK）．以上のことは，分類器の性能を更に改善する必要があることを示唆する．

6. おわりに

本稿では，解説文生成における，問題クラス，訓練データの量と性能に関する疑問について検討した．3,355 の解説文に対して，人手でコードを付与し分類した．その結果，40%の解説文については，分類問題として解くことで，ある程度内容を生成できることを明らかにした．また，辞書へのグランディングと合わせることで，80%の解説文に対応できることも明らかにした．更に，分類の結果に基づいて，解説文の総数に対して種類数がどの程度増加するかも明らかにした．評価に関しては，BERT に基づいた自動評価手法の有効性を検証した．その結果，正解率 0.906 が得られたが，改善の余地が大きく残ることを確認した．

参考文献

[1] Hashimoto et. al., “A retrieve-and-edit framework for predicting structured outputs,” Advances in Neural Information Processing Systems 31, pp.10052–10062, 2018.

[2] Y.H. Lai et. al, “TellMeWhy: Learning to explain corrective feedback for second language learners,” Proc. of EMNLP (System Demonstrations), pp.235–240, 2019.

[3] R. Nagata, “Toward a task of feedback comment generation for writing learning,” Proc. of EMNLP, pp.3197–3206, 2019.