Vol.2022-NL-252 No.10
2022/6/29

フレーズアライメントと文構造に基づくデータ拡張を用いた頑健な自然言語生成

山本 賢太[1,a)]　河野 誠也[2,b)]　河原 達也[1,c)]　吉野 幸一郎[2,d)]

概要：自然言語生成タスクは意味表現を入力として対応するテキスト（発話）を生成するタスクで，対話システムにおける重要なタスクのひとつである．近年は，ニューラルネットワークを用いた手法により，自然な言語生成が実現されている．ニューラルネットワークを用いた言語生成は学習データへの依存性が大きく，しばしば与えた意味表現にない情報を生成文に含めてしまう過生成の問題が生じる．そこで，本研究では，言語生成タスクのデータセットから意味表現を削除した学習データのバリエーションを作成するデータ拡張手法により，より頑健な自然言語生成を実現する．具体的には，学習データに含まれる意味表現に対して一部のスロットの削除を行い，この意味表現に対応して学習データ中の文を編集することで，既存の学習データに含まれない意味表現の組み合わせに対応したデータの拡張を行う．この対応取得のため，フレーズアライメントや注意機構の重みを用いる．また，文の編集を行う際にもとの文の構造を考慮する．実験では，提案法により過生成の問題を抑制しつつ，自然性も高い言語生成を行うことができることが確認された．

1. はじめに

近年，チャットボットやスマートスピーカなどの普及に伴い，対話システムの需要が増加している．レストラン検索 [25] や観光案内 [24] などのタスク対話では，ユーザの意図やシステムからの情報提示を機械に解釈可能な形で利用するため，意味表現の利用が重要である．意味表現は，いくつかのスロットを持つフレームとして定義され，スロットごとにエンティティなどのスロット値が設定される．自然言語で表現された発話文をこうした意味表現と相互に変換する課題 [17], [18] が言語理解と言語生成 [22] である．本研究ではこのうち，言語生成の課題に取り組む．言語生成のタスクにおいては，E2E NLG Challenge [3] などの共有データセットを用いて自然な言語生成を実現するための手法が研究されている．言語生成の課題では，ルールに基づく手法 [3]，テンプレートベースの手法 [12], [16]，ニューラルネットワークを用いる手法 [1], [4], [6], [7] などが提案されてきた．

これらのうち特に教師あり学習による言語生成の手法においては，過生成の問題が発生する．過生成は，言語生成の条件に指定されていない内容が生成文に含まれてしまう現象で，最初は統計的機械翻訳でその問題が指摘された [3], [9]. 特に対話システムのための言語生成のタスクにおいては，入力された意味表現に存在しない内容が発話文中で言及される問題となる．過生成により，システム設計者・利用者が意図しない言語生成が行われ，その結果システムに対する信頼が損なわれる．また，対話制御結果に対応しない発話が生成されるため，対話履歴に矛盾が生じる可能性もある．

こうした過生成の問題は，特にニューラルネットワークを用いた言語生成モデルでしばしば生じる．ニューラルネットワークを用いた生成モデルは学習データへの依存が強く，学習データに存在する意味表現の組み合わせに対しては非常に流暢な発話を生成することができる一方，学習データに存在しない意味表現の組み合わせに対してしばしば対応しない内容を生成する．そこで本研究では，データ拡張によってこうした過生成の問題に対処する．具体的には，学習データの各意味表現の中からいくつかエンティティを削除したデータを作成して拡張データとして用いる．こうしたデータ拡張を行う際，削除したエンティティに対応する箇所を正解文から削除して学習データとする必要がある．そこで，本研究ではアライメントツールを用い

[1] 京都大学大学院情報学研究科
Graduate school of informatics, Kyotu University
[2] 理化学研究所ガーディアンロボットプロジェクト
GRP, RIKEN
[a)] yamamoto@sap.ist.i.kyoto-u.ac.jp
[b)] seiya.kawano@riken.jp
[c)] kawahara@i.kyoto-u.ac.jp
[d)] koichiro.yoshino@riken.jp

表 1　意味表現と生成文の例（レストラン案内）

情報	値
意味表現（入力）	name[the Wrestlers] priceRange[cheap] customerRating[low]
生成文（出力）	The wrestlers offers competitive prices, but isn't highly rated by costomers.

表 2　意味表現のスロットの例（レストラン案内）

スロット名	スロット値の例
name	*Eagle*, ...
eatType	*restaurant*, *pub*, ...
familyFriendly	*Yes* / *No*
priceRange	*cheap*, *expensive*, ...
food	*French*, *Italian*, ...
near	*Zizzi*, *Cafe Adriatic*, ...
area	*riverside*, *city center*, ...
customerRating	*1 of 5 (low)*, ...

る方法と注意機構を用いる方法を提案する．これらの手法では，スロット値に対応する文中の箇所を取得し，その情報に基づき文を編集することで，スロット値に相当する内容が削除された文を拡張データとして作成する．この際，構文的・意味的におかしい文章を生成しないような制約を文の構文構造を考慮しつつ与えることで，品質の良い学習データを追加可能となる．評価実験では，このデータ拡張により，未知の意味表現パターンを含むデータに対して過生成を抑制されることができ，また生成文の自然性も向上することが確認された．

2. 言語生成と過生成

本研究では，対話システムなどで用いられる意味表現を入力とした言語生成タスクと，そのタスクにおける過生成の問題解消に焦点を置く．この節ではタスクの定義と，既存の研究で具体的にどのような問題が生じるかについて説明する．

2.1 言語生成タスク

本研究で扱う言語生成タスクは，構造化された意味表現をもとに，発話文を生成するタスクである．言語生成モデルは，複数のスロット値を入力とし，そのスロット値の内容を全て反映した生成文を出力する．レストラン案内タスクにおける意味表現と生成文の例を表 1 に示す．レストラン案内タスクの意味表現では，E2E NLG チャレンジ [3] に従い，表 2 に示すようなレストランに関係する複数の意味表現がスロットが用意されている．これらのスロットが具体的な値と共に与えられ，その入力内容をもとに表 1 のような生成文を生成することが目標である．このようなデータは，入力する意味表現に対応するよう人手で生成文を作成するのが一般的で，こうしたデータの入出力から文生成を行うモデルを構築するのが言語生成タスクである．

2.2 過生成

このような言語生成タスクは，ニューラルネットワークを用いた単語列の生成モデルを利用することが一般的である．しかし，既存の言語生成モデルの多くは，過生成という問題を抱えている．(図 1)．過生成とは，入力された意味表現に含まれないスロットの内容が生成文中に含まれてしまう現象である．過生成は言語生成モデルの制御性の問題と捉えることができる．つまり，生成モデルが本来意味表現で意図された以上の情報を含む文を生成することで，情報伝達を困難にする．

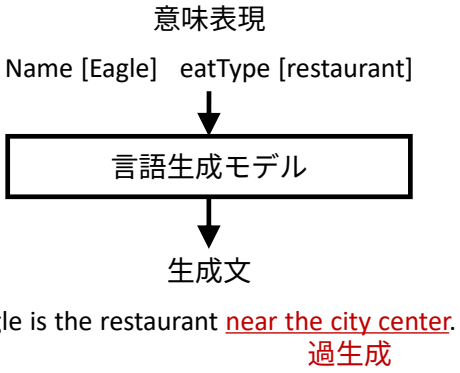

図 1　言語生成タスクにおける過生成の例

言語生成タスクにおける過生成の原因として，データに対する過学習があげられる．ニューラルネットワークを用いた言語生成モデルは，学習データへの依存度が大きい．つまり，学習データに含まれるような意味表現の組み合わせパターンについては流暢な言語生成を行うことができるが，学習データにない組み合わせ，特に学習データに存在する一部のスロットが欠けたような事例に対して過生成を行うようなモデルを学習してしまう．しかし，一般的にデータセットを作成する際には，スロットの組み合わせに対して発話文を人手で作成するため，過不足なくデータセットを作成することは困難である．

こうした過生成の問題に対する対処方法としては，デコーディングの工夫や外部データを用いたデータ拡張などの方法が提案されている [4]．データ拡張のアプローチは，実際に与えられる意味表現のバリエーションに対して，元々持つ学習データの分布を拡張データで補正しようとするものである [6], [7]．本研究はこの方針を採用し，過生成の問題に対して既存のスロットの組み合わせから一部のスロットを削除したデータを作成し，言語生成モデルの拡張学習データとして用いる．こうしたデータ作成をする際，削除したスロットの対応する生成文中の表現を削除する．これにより，学習データに欠けているスロットの組み合わせに対しても，対応する生成文が用意されるため，過学習が抑制されることが期待される．

3. アライメントと文構造を用いたデータ拡張

本研究で提案するデータ拡張の手法について述べる．まず，もとの学習データ中の意味表現から，機械的に一部を削除する．これに対応する生成文を用意するため，元の学習データの中の文を，意味表現との対応と文構造にもとづいて編集する．図 2 に提案手法の概要を示す．データセットがもとの学習データに含まれる組み合わせ，データセットが本手法によって新しく生成される拡張データセットの一部である．このデータの生成に 2 つのステップを踏む．具体的には，意味表現のスロット値と生成文の単語とのアライメントを取得する．そのアライメント結果に基づき，スロットの一部を削除した意味表現に対応するように生成文を編集する．その際に，生成文が不自然にならないように，構文解析を用いたフィルタリングを行う．以降では，これらの手順の詳細を説明する．

3.1 意味表現と生成文のアライメント

モデルに入力する意味表現のスロット値と生成文中の単語との対応関係を取得する．本研究では，対応関係を取得する方法として 2 種類の方法を比較した．一つ目の手法は，アライメントツール [10] を用いる方法で，二つ目は Transformer を用いた言語生成モデルの注意重み [21] をアライメントに用いる方法である．

3.1.1 アライメントツールを用いる手法

アライメントツールとは，ニューラル機械翻訳以前の統計的機械翻訳などで用いられた文同士のフレーズ対応を推定するツールである．今回は，Giza++ [*1]から，IBM モデル 4 を用いて学習したモデルを用いて，対応する意味表現と生成文それぞれを構成する単語間のアライメント結果を出力する．このとき，意味表現については “スロット名 1 [スロット値 1] スロット名 2 [スロット値 2] ...” という表現をアライメントツールへの入力として用いる．

図 2 に示すように．アライメントツールによってスロットのごとに生成文中で対応する単語を取得することができる．これにより，スロットを削除した際に，生成文から削除されるべき単語を取得する．図 2 の例では，“near [The Sorrento]” のスロットを削除した際に，アライメント結果に基づいて “the Sorrento” を削除対象として推定する．しかしこの処理だけでは，“by” が生成文中に残ってしまい不自然な文となる．そこで，3.2 節で述べる構文構造を用いた補正を用いて，不自然な文を生成しないようにする．

3.1.2 注意機構を用いる手法

アテンションを用いる方法では，Transformer モデルの注意機構重みをアライメントに利用する．ここでは，2 層のエンコーダと 2 層のデコーダからなる Transformer を用いてアライメントを取得する．デコーダの 2 層目のデコーダの Cross Attention 層を意味表現と生成文との対応づけに用いる (図 3)．このため，意味表現と生成文のペアデータを用いて Transformer による言語生成モデルを学習する．学習したモデルに対して，訓練データを入力した場合の，Cross Attention の重みを抽出する．この際デコーダは入力文と同じ内容を出力する．行列の全ての要素の平均値を閾値として，閾値 $\bar{x}$ を超えている行列の要素 x_{ij} を取得する．その要素 x_{ij} に対して，意味表現の i 番目の単語と生成文の j 番目の単語が対応しているとみなす．以上の手順で，入力と出力間のアライメントを取得する．

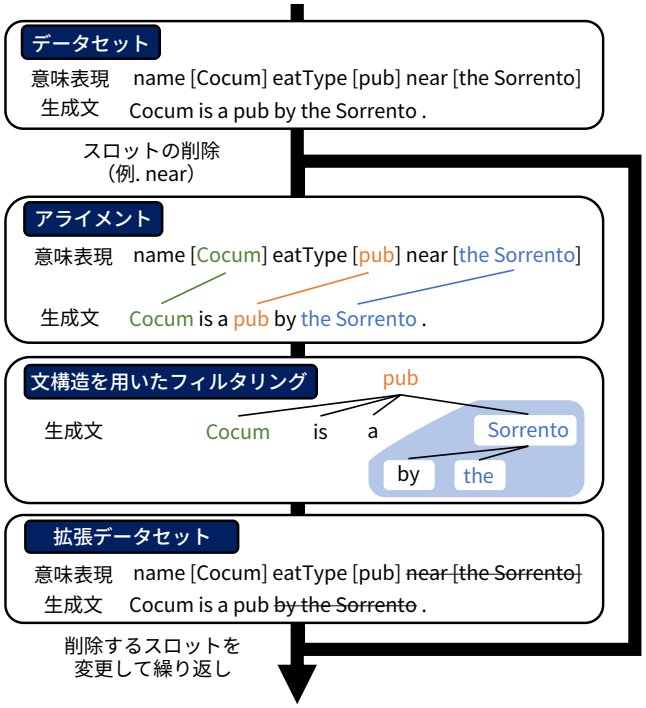

図 2 データ拡張手順

3.2 文構造を用いたフィルタリング

3.1 節で説明したアライメント結果を用いて，生成文を編集する．しかし 3.1 節で述べた通り，アライメント結果だけでは適切な文の編集を行うことができず，結果として不自然な文が生成されてしまう場合も多い．そこで，文構造をもとにデータをフィルタリングする．これによりアライメントだけでは捉えられていない，削除済みのスロットに対応する生成文中の単語を削除する．アライメント結果にはその対応に漏れが生じる場合があるが，構文構造を考慮することで削除対象単語を含むフレーズを適切に削除することができる．さらに，文構造を考慮することで，少なくとも係り受け関係が不明な文を生成することは抑制することができる．また，文構造の情報は削除するスロットの決定にも用いる．具体的には，生成文を構成する最低限の要素である用言と，その用言が持つべき必須格の情報は保持するように削除対象のスロットを決定する．

[*1] https://github.com/moses-smt/giza-pp

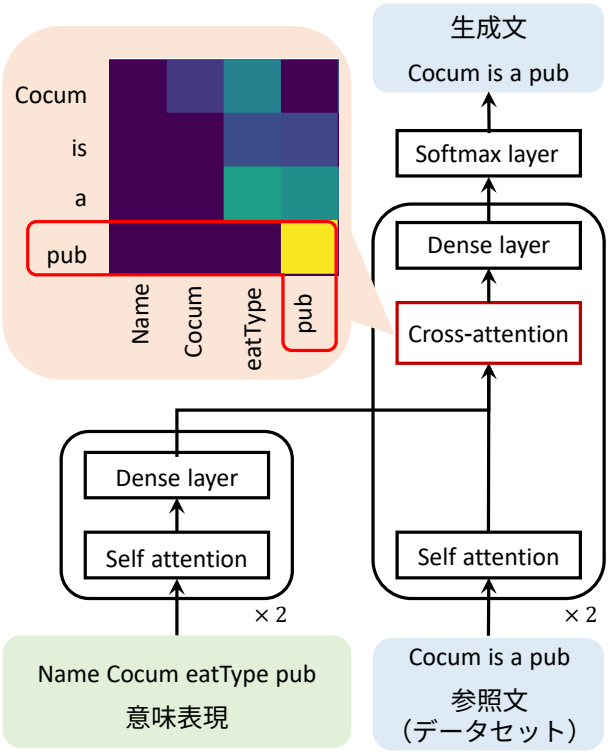

図 3 注意機構を用いたアライメント

構文解析では，Stanford Core NLP の Python のラッパーである Stanza [13] を用いた．この構文解析では，単語同士の係受け関係と単語の格情報を抽出する．まず，述語と必須格に対応するスロットは削除対象から除いた上で，ランダムに削除対象スロットを決定する．構文解析結果で，削除対象の単語が必須格に含まれている場合は，スロットを削除しない．削除対象とするスロットに対応する単語をマークし，それらの単語全てを被覆する部分木を特定する．この部分木全てを該当するスロットに対応する部分木であるとして削除する．この際，削除対象となる単語が，削除対象でないスロットと対応する単語を含む場合は，このスロットの組み合わせをスキップする．こうして，一部のスロットが削除された意味表現と生成文のペアを新しい拡張データとする．削除可能なスロットの組み合わせに対して，以上の手順を繰り返す．これにより，生成文の構文的な自然性を損なうことなく，データを拡張できる．

4. 評価実験

提案するデータ拡張手法によって実際に言語生成器の頑健性が向上するか，評価実験で確認を行う．実験には E2E チャレンジで用いられたデータセットを利用する．この際，過生成の問題が解消されたかを確認するため，E2E チャレンジにおけるテストデータに加えて，E2E チャレンジのデータを編集して一部のスロットを削除し，対応する生成結果の正解を新たに作成したテストデータを用いる．評価では，自動評価に加えて人手で自然性，有用性，過生成率について評価した．以下では実験の詳細について述べる．

4.1 E2E チャレンジデータセット

はじめに，本研究で用いる学習データについて述べる．学習データとして，E2E NLG チャレンジで用いられているデータセットを用いる．このデータセットでは，レストラン案内タスクにおける意味表現のスロットとそれに対応する生成文が用意されている．データの構成を表 3 に示す．このデータセットでは，同じ意味表現に対して複数の人手により生成文を作成しているため，意味表現の種類よりも多くの生成文が用意されている．

4.2 ベースラインモデルと非語彙化

今回，言語生成モデルとしては SLUG [6] を用いる．SLUG では、学習時に学習データの非語彙化が行われている．非語彙化は，スロット値と生成文の対応する単語をプレースホルダートークンに置き換える手続きである．これによりスロット値の語彙数が削減され，モデルの学習に必要なデータ量を削減することができる．生成文のプレースホルダートークンは，後処理で元の単語に置き換える．今回先行研究に従い，非語彙化するスロットは name, near, food の 3 つとした．これらのスロットに出現する単語は，同じ単語が生成文に出現しているためルールベースによる置換が可能である．一方，priceRange や area などのスロットにおいては，スロット値の “less than $20” が生成文では “cheap” と表現されている場合や，“riverside” が “by the river” に置き換わる場合など，生成文における表現方法がスロット値以外に存在するため，非語彙化を行わない．

単純に非語彙化を行うと，非語彙化されたスロットの周りの語彙の選択に影響が生じる場合がある．そこで，以下の 2 点に着目して，プレースホルダートークンを定義する．1 点目は，スロット値に含まれる名詞が単数形か複数形かという点である．また，単数形場合は冠詞 “a” を使うか “an” を使うかという問題も生じる．2 つ目のポイントは，“food” スロットの値が “food” を示す単語を含むかに着目する．例えば，“food[french]” では、このプレースホルダーを用いて文中の “French” という単語を置き換え，“serves French food” -¿ “serves food[french] food” のように表現することができる．一方で，“food[fast food]” というプレースホルダーを利用しようとすると，“serves fast food” -¿ “serves food[fast food]” となってしまい、それ以外のパターンと適合しないケースが出てくる．この 2 つのケースは，プレースホルダートークンに “cuisine” が使われているかどうかで区別する．

ベースライン，提案法いずれもこの非語彙化したデータを用いる．また，入力として与えられていない非語彙化されたスロットが生成されるケースが存在するが，こうしたパターンはデコーディング時に削除する．これは，デコーディング時に過生成を抑制する工夫のひとつである．

表 3　学習データの構成

データの種類	データ数	意味表現の種類
訓練データ	42,061	4,862
検証データ	4,672	547
評価データ	4,693	630
合計	51,426	6,039

4.3　過生成評価のためのテストデータ

評価データとして，まず E2E チャレンジデータセットに含まれるデータを利用する（Test-O）．しかし今回問題とするケースは，本来の学習データで被覆されていないようなスロット値が少ない意味表現が与えられたときに過生成が行われるような場合である．こうしたケースでも過生成を行わず必要十分な生成が可能かどうかについて評価を行う．そこで，評価データの 630 種類の意味表現を用いて過生成について評価するための新しい評価データを用意した．具体的には，評価データの意味表現をランダムに削除し，この一部が欠けた意味表現に対して新しい評価データを用意した（Test-R）．ただし，削除する意味表現を決定する際，対応する単語が生成文作成に必須の単語であるかどうかを 3.2 節の方法で判断し，述語および必須格が最低限存在するよう意味表現の組を用意した．このようにして作成した意味表現のスロット値の組 630 組に対して，アノテータに新しい生成文を付与してもらった．具体的には，アノテータに，正解の生成文，削除する前の意味表現，削除される前のスロットの値の組を提示して，スロットを削除した新しい意味表現に対して適切な文を作成してもらった．

4.4　モデル構成

言語生成のモデルは SLUG の論文 [6] に基づいて構成した．ただし，エンコーダは元の論文で用いられている LSTM から Transformer に変更した．エンコーダとデコーダともに 2 層で構成されている (図 4)．

学習時のバッチサイズは 1024 で，潜在変数は 256 次元である．モデルの入力として用いる意味表現は，スロット名とスロットの値のペアを単語単位で並べたものを入力する．その際に，スロット名は 1 単語として扱い，“name_slot” のようにスロット名であることを明確にする．デコーダは生成文を単語単位で出力する．学習時には Teacher forcing を用いる．また，推論時に Top-K デコーディングを行い，その 1-best を利用した．ただし， 1-best がスロット名（“eattype”，“pricerange”，“customerrating”）が直接生成文中に出力される場合や，意味表現に含まれていない非言語化したスロットが含まれる生成文は候補から除外し，次の候補を生成文として出力する．以降は，元のデータセットと拡張したデータセットで学習させたこのモデルを評価をする．

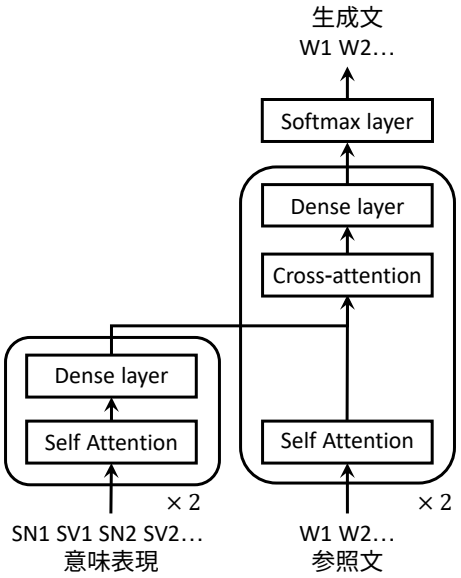

図 4　モデル構成：スロット名 (SN)，スロット値 (SV)，生成文の各単語 (W)

4.5　評価指標

4.5.1　自動評価での評価指標

まず，自動評価スコアを用いて提案するデータ拡張手法の有効性を検証する．評価は元の E2E チャレンジの評価データ（Test-O）と，過生成の評価用に新しく用意した評価データ（Test-R）を用いた．これらの評価データに対して，BLEU と Entity F1 の二つのスコアを用いて評価した．BLEU は、評価データの生成文 に対する一致度を測るものである．Entity F1 は，スロットの対応する単語が出力されているかどうかを評価するものである．適合率が低く，再現率が高い場合は，過生成が発生している可能性が高い．Entity F1 は以下のように算出に際しては，意味表現のスロットごとに予想される出力単語を用意する必要がある．非語彙化しているスロット “name”,“food”,“near” はスロット値が生成文に出力されているかを判定基準とした．その他のスロット値に関しては，事前に意味表現のスロットごとに想定される表現を列挙したリストを用意する．そのリストに含まれる表現が生成文に出力されているかを判定する．

4.5.2　人手評価での評価指標

人手評価では，アノテータにモデルが生成した文章の質を自然性，有用性，過生成率を評価してもらった．評価用データは，自動評価で用いた過生成評価用のテストデータ（test-R）から選択した意味表現 200 個を用い，それぞれのモデルに対応する生成結果を評価して貰った．アノテータには，生成文と意味表現を提示し，以下の項目について 5 段階で評定してもらった．

- 自然性: 生成文が自然かどうか
- 有用性: 生成文が意味表現を過不足なく反映しているかどうか

また，生成文に含まれていないスロットの，過生成の個数を数えて割合を算出した．人手による評価は，評価の質を保証するため，評価対象に正解文を含めてブラインドで実施した．

4.6 比較評価を行うモデル

評価では，ベースラインモデルとしてデータ拡張を行わない場合を利用した（Baseline）．これに対し，アライメントツールを使ってデータ拡張した場合（Tool），注意機構を使ってデータ拡張をした場合（Attention），双方を利用した場合（Both methods）の比較を行う．これらのデータ拡張では，最大で削除するスロットの数に応じて “削除したスロット数” を設定した．削除したスロット数が “1,2,3” では，データ拡張の際にランダムに削除するスロットを最大 3 個設定することができ，データ拡張の数が多くなる．Both methods では， Tool と Attention が同じ拡張データを出力した場合は 1 件のみを拡張に用いる．

また，既存の過生成を抑制する方法として，デコーディング時の N-best の利用が挙げられる．これは N-best 候補を生成し，その中で与えられた意味表現に対する Entity F1 スコアが一番高くなるものを出力とするものである．自動評価この N-best デコーディングをする場合，しない場合の評価をそれぞれ行う．人手評価においては提案手法の純粋な効果を測定するため， N-best デコーディングを行わない場合の評価を行う．

5. 評価

5.1 自動評価結果

N-best デコーディングを行わない場合の実験結果を表 4 に示す．はじめに，元の E2E チャレンジの評価データと (Test-O) と新しく用意した評価データ (Test-R) に対する評価結果を比較する．Test-R では，Test-O 対する評価と比較して EntityF1 スコアの適合率が低下しており，再現率が増加していることから，Test-R では過生成が発生している．EntityF1 スコアを見ると，BLEU スコアが低下しているものの，アライメントツールを用いた場合に，いずれの設定でも F1 スコアが最も高くなる．一方，注意機構を用いる手法では，BLEU スコアが少し改善された．さらに，注意機構を用いた手法では，アライメントツールを用いる方法に比べて，拡張データ数（表の訓練データ数）が少ない．これは，注意機構を用いることで，自然な拡張データが生成されていると考えられる．両方のデータを統合するとベースラインと同程度のスコアとなる．このことから，2 つの手法の中間的な結果になっている可能性がある．しかし，自動評価と人手評価の相関は限定的であるため，人による評価結果を確認する必要がある．

N-best デコーディングを行った場合の実験結果を表 5 に示す．N-best デコーディングは 1-best デコーディングの場合とを比較して，EntityF1 が向上し，BLEU スコアが低下している．これは，N-best デコーディングでは，EntityF1 を最大化するようにデコーディングしているためである．アライメントツールを用いた場合に最も EntityF1 のスコアが高く，注意機構を用いた場合に最も BLEU スコアが高くなることが示された．

5.2 人手評価結果

人手評価では，自然性，有用性，過生成率に着目した．前者 2 つは，言語生成の研究で広く用いられている基準であり，最後の基準は本研究の課題に焦点を当てた基準である．実験結果を表 6 に示す．まず，今回のアノテータは，正解文に対して高い自然性と有用性を与えており，この結果は人手の評価が適切におこわなれたことを示唆している．

注意機構を用いることで，有用性と過生成率で最も良い評価が得られた．また，自然性を若干低下させたが，有用性，過生成率においては Baseline との間に有意水準 0.01 で有意差がある．Baseline 以外の手法でも，両方の手法を組み合わせる方法以外では，過生成が抑制されていることがわかる．このことから提案手法によるデータ拡張は過生成の問題に対して効果があることが確認された．一方，Both methods では過生成を抑制することができなかった．これは，拡張データとテストデータの分布の間にギャップがあることが原因である可能性がある．つまり Both methods モデルでは，最も多くの拡張データを用いて学習しているため，モデルが出力する生成文がテストデータと異なる分布になっていると考えられる．

6. 関連研究

過生成を抑制するために，いくつかのデコーディング手法が提案されている [15], [19]．Tian ら [19] は，注意機構重みから信頼度スコアを過生成の抑制に利用している．Shen ら [15] は，与えられた意味表現の中から注目しているセグメンテーションに対応するスロットを推定し，そのスロット値を用いて過生成を抑止している．また，未知のパターンに対して頑健な言語生成ネットワークを用いることも提案されている [20]．過生成を抑制するために，言語生成モデルの学習のための目的関数がいくつか提案されている [8], [11], [14]．これらの研究では強化学習を導入し，与えられた意味表現を適切に表現していない生成文にペナルティを与えることで，言語生成モデルの制御性を向上させた．また，言語生成モデルの頑健性を保証するためのマルチタスク学習も提案されている [26]。彼らの考え方は我々のアプローチと似ており、学習データの MR パターンの多様性を向上させることで、過生成を抑制している．

一方，過生成を抑制するためのデータ増強法も提案されている [6], [7], [17], [23]．Juraska ら [6] は，参照文が複数の文からなる場合に，それらに対応する意味表現を持つ

表 4 人手評価結果

モデル名	削除した スロット数	訓練 データ数	拡張テストデータ (Test-R) BLEU	Entity 適合率	Entity 再現率	Entity F 値	E2E テストデータ (Test-O) BLEU	Entity Precision	Entity Recall	Entity F1
Baseline	0	42,061	0.632	0.953	0.928	0.940	0.694	1.0	0.900	0.947
Tool	1	63,055	0.634	0.953	0.913	0.932	0.689	0.999	0.903	0.950
Tool	1,2	84,184	0.629	0.952	0.916	0.933	0.687	0.998	0.899	0.946
Tool	1,2,3	97,137	0.612	0.993	0.907	**0.949**	0.681	1.0	0.892	0.942
Attention	1	49,239	0.623	0.922	0.921	0.922	0.687	0.999	0.903	0.949
Attention	1,2	52,055	0.635	0.944	0.921	0.932	0.690	1.0	0.905	0.950
Attention	1,2,3	52,721	**0.637**	0.941	0.918	0.929	0.686	1.0	0.899	0.947
Both methods	1	70,233	0.635	0.930	0.904	0.917	0.687	0.999	0.900	0.947
Both methods	1,2,3	107,797	0.632	0.952	0.919	0.935	0.690	1.0	0.906	0.950

表 5 人手評価結果 (N-best デコーディング)

モデル名	削除した スロット数	訓練 データ数	拡張テストデータ (Test-R) BLEU	Entity 適合率	Entity 再現率	Entity F 値	E2E テストデータ (Test-O) BLEU	Entity Precision	Entity Recall	Entity F1
Baseline	0	42,061	0.615	0.964	0.934	0.949	0.672	1.0	0.908	0.951
Tool	1	63,055	0.516	0.969	0.931	0.950	0.650	0.999	0.911	0.953
Tool	1,2	84,184	0.567	0.968	0.930	0.948	0.656	0.999	0.909	0.952
Tool	1,2,3	97,137	0.587	0.994	0.928	**0.960**	0.659	0.999	0.908	0.952
Attention	1	49,239	0.609	0.963	0.926	0.944	0.665	1.0	0.911	0.953
Attention	1,2	52,055	0.617	0.970	0.931	0.950	0.653	1.0	0.916	0.956
Attention	1,2,3	52,721	**0.623**	0.981	0.930	0.955	0.670	1.0	0.909	0.952
Both methods	1	70,233	0.530	0.963	0.933	0.948	0.645	0.999	0.913	0.954
Both methods	1,2,3	107,797	0.602	0.987	0.931	0.958	0.643	0.999	0.907	0.951

表 6 人手による生成文の質の評価と過生成率

モデル	人手評価 自然性	人手評価 有用性	過生成率 (平均過生成数)	
Baseline	**4.945***	4.050	12.6%	**(0.305)**
Tool	4.500	3.900	15.3%	**(0.455)**
Attention	4.900	**4.245****	**7.0%****	**(0.185)**
Both methods	4.350	3.930	21.3%	**(0.595)**
Reference (gold)	4.880	4.855	0.6%	**(0.020)**

自然性と有用性はウィルコクソンの符号順位検定
過生成率は対応ありの両側 t 検定
3 つの検定は Baseline と Attention 間を検定
($* < 0.05$, $** < 0.01$)

別々の学習サンプルに分割することで学習データを拡張している．他の研究 [2], [7], [23] では，様々な意味表現パターンを用いて事前に学習した言語生成モデルからサンプル文をデータ拡張に利用している．データ拡張の考え方は，言語理解のタスクおいても重要である [5]．しかし，学習済み言語生成モデルからのサンプリングに基づく従来手法は元の学習データ分布に強く依存してしまう．そのため，出現度の低い意味表現パターンに対応した生成が困難である点や，データ拡張に用いる生成文の不自然さが問題となる．これに対して，提案手法で得られる拡張データは，実際の人間の文章に基づく自然なものでありながら，多種多様な意味表現のパターンに対応可能である．さらに，削除するスロットを制御可能であるため，従来のデータ拡張手法に比べて柔軟なデータ拡張が可能である．また，本手法は他のデータ拡張手法と容易に併用可能な手法である．

7. まとめ

本研究では，自然言語生成タスクにおける過生成の問題に着目した．この問題を解消するため，フレーズアライメントと文構造を利用したデータ拡張方法を提案した．対応関係を取得する方法として，アライメントツールと言語生成モデルの注意機構を用いた．実験では，注意重みに基づく手法で最も高い BLEU スコアを達成し，人手評価でも過生成の抑制で最も高い評価を得られた．また，N-best デコーディングなどの他の過生成抑制手法との併用も可能であることが確認された．

今後の課題としては，より多様なデータセットで提案法を検証することが挙げられる．また，提案法はシンプルで他のデータ拡張やデコーディングなどの過生成抑制手法と併用することが容易であるため，併用の効果についても調査していく必要がある．

参考文献

[1] Agarwal, S., Dymetman, M. and Gaussier, É.: Char2char Generation with Reranking for the E2E NLG Challenge, *Proceedings of the 11th International Conference on Natural Language Generation (INLG)*, pp. 451–456 (2018).

[2] Du, W., Chen, H. and Ji, Y.: Self-augmented Data Selection for Few-shot Dialogue Generation, *arXiv preprint arXiv:2205.09661* (2022).

[3] Dušek, O., Novikova, J. and Rieser, V.: Evaluating the State-of-the-Art of End-to-End Natural Language Generation: The E2E NLG Challenge, *Computer Speech and Language*, Vol. 59, No. C, pp. 123–156 (2020).

[4] Elder, H., Gehrmann, S., O'Connor, A. and Liu, Q.: E2E NLG Challenge Submission: Towards Controllable Generation of Diverse Natural Language, *Proceedings of the 11th International Conference on Natural Language Generation (INLG)*, pp. 457–462 (2018).

[5] Glass, M., Rossiello, G., Chowdhury, M. F. M. and Gliozzo, A.: Robust Retrieval Augmented Generation for Zero-shot Slot Filling, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 1939–1949 (2021).

[6] Juraska, J., Karagiannis, P., Bowden, K. and Walker, M.: A Deep Ensemble Model with Slot Alignment for Sequence-to-Sequence Natural Language Generation, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pp. 152–162 (2018).

[7] Kedzie, C. and McKeown, K.: A Good Sample is Hard to Find: Noise Injection Sampling and Self-Training for Neural Language Generation Models, *INLG* (2019).

[8] Li, Y., Yao, K., Qin, L., Che, W., Li, X. and Liu, T.: Slot-consistent NLG for Task-oriented Dialogue Systems with Iterative Rectification Network, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 97–106 (2020).

[9] Müller, M. and Sennrich, R.: Understanding the Properties of Minimum Bayes Risk Decoding in Neural Machine Translation, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 259–272 (2021).

[10] Och, F. J. and Ney, H.: A Systematic Comparison of Various Statistical Alignment Models, *Computational Linguistics*, Vol. 29, No. 1, pp. 19–51 (2003).

[11] Perez-Beltrachini, L. and Lapata, M.: Bootstrapping Generators from Noisy Data, *16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, pp. 1516–1527 (2018).

[12] Puzikov, Y. and Gurevych, I.: E2E NLG Challenge: Neural Models vs. Templates, *Proceedings of the 11th International Conference on Natural Language Generation*, pp. 463–471 (2018).

[13] Qi, P., Zhang, Y., Zhang, Y., Bolton, J. and Manning, C. D.: Stanza: A Python Natural Language Processing Toolkit for Many Human Languages, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations* (2020).

[14] Rebuffel, C., Soulier, L., Scoutheeten, G. and Gallinari, P.: PARENTing via Model-Agnostic Reinforcement Learning to Correct Pathological Behaviors in Data-to-Text Generation, *Proceedings of the 13th International Conference on Natural Language Generation*, pp. 120–130 (2020).

[15] Shen, X., Chang, E., Su, H., Niu, C. and Klakow, D.: Neural Data-to-Text Generation via Jointly Learning the Segmentation and Correspondence, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 7155–7165 (2020).

[16] Smiley, C., Davoodi, E., Song, D. and Schilder, F.: The E2E NLG Challenge: A Tale of Two Systems, *Proceedings of the 11th International Conference on Natural Language Generation(INLG)*, pp. 472–477 (2018).

[17] Su, S.-Y., Huang, C.-W. and Chen, Y.-N.: Dual Supervised Learning for Natural Language Understanding and Generation, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5472–5477 (2019).

[18] Su, S.-Y., Huang, C.-W. and Chen, Y.-N.: Towards Unsupervised Language Understanding and Generation by Joint Dual Learning, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 671–680 (2020).

[19] Tian, R., Narayan, S., Sellam, T. and Parikh, A. P.: Sticking to the Facts: Confident Decoding for Faithful Data-to-Text Generation, *CoRR*, Vol. abs/1910.08684 (online), available from ⟨http://arxiv.org/abs/1910.08684⟩ (2019).

[20] Tseng, B.-H., Budzianowski, P., Wu, Y.-C. and Gasic, M.: Tree-Structured Semantic Encoder with Knowledge Sharing for Domain Adaptation in Natural Language Generation, *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pp. 155–164 (2019).

[21] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u. and Polosukhin, I.: Attention is All you Need, *Advances in Neural Information Processing Systems* (Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S. and Garnett, R., eds.), Vol. 30 (2017).

[22] Wen, T.-H., Gasic, M., Mrkšić, N., Su, P.-H., Vandyke, D. and Young, S.: Semantically Conditioned LSTM-based Natural Language Generation for Spoken Dialogue Systems, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1711–1721 (2015).

[23] Xu, X., Wang, G., Kim, Y.-B. and Lee, S.: AugNLG: Few-shot Natural Language Generation using Self-trained Data Augmentation, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1183–1195 (2021).

[24] Yoshino, K., Suzuki, Y. and Nakamura, S.: Information navigation system with discovering user interests, *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pp. 356–359 (2017).

[25] Young, S., Gašić, M., Keizer, S., Mairesse, F., Schatzmann, J., Thomson, B. and Yu, K.: The hidden information state model: A practical framework for POMDP-based spoken dialogue management, *Computer Speech & Language*, Vol. 24, No. 2, pp. 150–174 (2010).

[26] Zhu, C., Zeng, M. and Huang, X.: Multi-task learning for natural language generation in task-oriented dialogue, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 1261–1266 (2019).