キーワードに基づくニューラル文生成のためのリランキング

尾形朋哉 * [1]　叶内晨 [1]　高谷智哉 [2]　小町守 [1]

[1] 首都大学東京　[2] トヨタ自動車株式会社

1 はじめに

近年，LSTM を用いた Encoder-Decoder モデルなどの長距離の依存関係を表現可能なニューラルネットワークの研究 [1][2] が盛んに行われている．これらの研究には，2 言語間の翻訳を行うニューラル機械翻訳 [3] や，ある発話文に対応する応答文を生成するニューラル対話生成，文書要約においても抽象型の文書要約などがあり，様々なタスクにおいて成功を収めつつある．

一方で，文書要約が文書から部分的な情報を出力するのに対して，対話行為のような構造化された情報から適切な自然言語を生成する研究がある [4] [5] ．これらの先行研究では入力として対話行為のアノテーションが必要であるが，日本語において対話行為を持つ大規模なデータセットは存在しない．また先行研究の手法 [5] では，システムの目的言語側の語彙に含まれていないキーワードを出力できないという未知語の問題と，入力で指定したキーワードが必ずしも出力文に含まれない問題がある．

そこで，本研究では対話行為を使わずにキーワードのみから文を生成するというタスクに取り組む．先行研究と本研究の入力および出力は表 1 に示す通りである．先行研究と本研究は出力が文生成であるという点が共通しているが，本研究は対話行為のアノテーションやオントロジーを必要とせず文生成を行う．

提案手法は，キーワードさえ収集できれば学習したデータから文を生成することが可能である．未知語の問題に対しては，原言語側から未知語をコピーする機構を持つ Encoder-Decoder モデルを応用した Positional Unknown モデル [6] により対処した．また，入力のキーワードが出力に含まれているかどうかで文のリランキングを行い，入力で指定したキーワードが出力文に含まれるようにした．

本研究の主な貢献は以下の通りである．

- 対話行為を用いずにキーワードのみから文の生成を行うタスクを提案した．

*ogata-tomoya1@ed.tmu.ac.jp

表 1: 部分的な情報からの文生成タスク

	入力	出力
先行研究	“注文”(“料理” = “炒飯”, “数” = “1”)	炒飯を 1 皿ください
本研究	降る 雨 明日	明日は雨が降るだろう

- キーワードによるリランキングを用いてキーワードを含む文を出しやすくした．
- キーワードに基づく文生成の未知語問題に対して，機械翻訳における Positional Unknown モデルを適応した．

2 関連研究

文書要約が文書から部分的な情報を出力する研究であるのに対して，部分的な情報から適切な自然言語を生成する研究として Wen ら [4] や Dusek ら [5], Konstas ら [7] がある．Wen ら [4] や Dusek ら [5] は，対話行為を入力として指示された文を生成している．対話行為はドメインにおけるその文のタイプや属性など，生成する文が持つべき情報を指示するものである．例えば，表 1 の対話行為は文のタイプに “注文” を取り，“注文” の属性の一つである “料理” に “炒飯” を取り，もう一つの属性である “数” に “1” を取っているので，この対話行為は “炒飯を 1 皿ください” のような，炒飯を注文する際の文を生成するという指示を表している．しかし，対話行為を持つ日本語の大規模なデータセットはなく，特定のドメインに限られる．そこで本研究では対話行為を用いずキーワードのみを与えることで文の生成を行うという新しいタスクに取り組む．

一方，対話行為を入力とする Dusek ら [5] のシステムはシステムの目的言語側の語彙に含まれていないキーワードを出力できないという問題がある．Encoder-Decoder の未知語の問題を解決するためにデコード時に原言語側の単語をコピーすることで，未知語を出力できるようにするといった手法が機械翻訳 [6] や要約 [8][9]，対話 [9] において用いられているが，本研究の行った自然言語文生成のタスクには用いられていない．そこで，本研究では，Luong らの Positional Unknown モデル [6] を用いてこの未知語の問題に対応する．

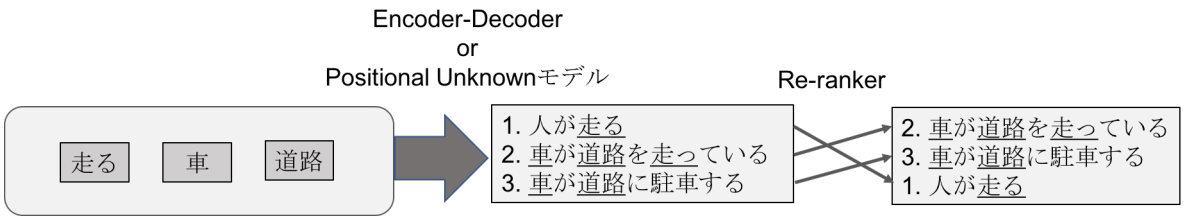

図 1: キーワードに基づく文生成のリランキング

また，Dusek ら [5] は出力された文において，入力された対話行為の属性値が一致していないものに対してペナルティをかけ，リランキングを行い文の生成をしている．本研究のタスクでは対話行為を持たないので，本研究では Jaccard 係数による文の類似度によりペナルティをかけ，リランキングをして文の生成を行った．

3 キーワードに基づく文生成

本研究では，対話行為がアノテーションされていないデータにおいても，キーワードのみを与えることで文の生成を行える．これにより，対話システムで発話中のキーワードを用いて応答文を出力したり，レコメンドサイトでキーワードを入力することで適切なレビュー文を生成できるなど様々なタスクに応用できる．しかし，キーワードに未知語が入ることがあり，Encoder-Decoder ではそのキーワードを出力することはできない．

そこで本研究では Encoder-Decoder の未知語の問題を改善し，未知語のキーワードを出力できる Positional Unknown モデル [6] を用いた．本研究におけるキーワードに基づく文生成の全体的な処理の流れを図 1 に示す．キーワードを入力として 3.2 節の Positional Unknown モデルまたは Encoder-Decoder を適用し，n-best の生成文を出力する．その後，3.3 節で説明するリランカを用いることで，生成文に含まれるキーワードの数に応じてリランキングを行う．

3.1 コーパス

対話行為のような構造化されたデータを用いた文生成が行われているが，日本語では，実際にそのようなデータはほとんどない．本研究では，対話文を集めたコーパスから tf.idf などに基づき，キーワードを抜き出し，キーワードと元の文の対となるコーパスを作成し，モデルを学習する．本研究では，ここで作成したコーパスを用いて，モデルにキーワードを入力として与えた際に，そのキーワードを含むような文を出力するというタスクに取り組んだ．本研究では対話行為を持たないようなコーパスでも，キーワードさえ抜き出すことができればデータセットを作成できる．

3.2 Positional Unknown モデル

Positional Unknown モデル [6] は，ソース側から未知語をコピーする機構を持つ Encoder-Decoder モデルである．Encoder-Decoder は Encoder と Decoder からなるモデルで，任意長の入力列から任意長の出力列を出力するように学習する．Encoder では入力から隠れ層ユニットを更新し，Decoder は対応する出力を一つずつ出力する．本研究で用いた Encoder-Decoder はアテンション機構付きのモデル [3] である．

Positional Unknown モデルでは，トレーニングデータにおいて，ターゲット側の未知語がソース側に含まれる場合，未知語を unk_(ソース側の位置) という形でソース側の位置に対応する記号に置き換えて学習する．ここで，ソース側の位置はターゲット側の未知語がソース側の文に現れる時，ソース側の文におけるその未知語の位置を表している．テスト時，このモデルは生成の語彙において unk_(ソース側の位置) が選ばれた際に，対応するソース側の単語を出力する．

3.3 リランカ

Positional Unknown モデルはキーワードが未知語の場合でもキーワードを含めた文を生成できるが，その文が実際にキーワードを含んでいるかは保証できない．そこで，本研究ではデコード時にビームサーチを行い，ビームサイズ個の保持している文の中から，どれだけキーワードを含んでいるかのスコアをつけ，このスコアをもとに並び替えをし，スコアの最も高い文を出力するようにした．本研究では出力文がキーワードを含むかどうかのスコアリングに Jaccard 係数を用いる．

$$Jaccard\ 係数 = \frac{|SetX \cap SetY|}{|SetX \cup SetY|} \quad (1)$$

また，ニューラルネットワークを用いた文生成では同じ単語を複数回出力してしまうという問題点が指摘されている [10]．そこで，重複するキーワードを含む候補は出にくくなるように式 (2) によるペナルティを課し，最終的なスコアは式 (3) のように計算される．

$$ペナルティ = 1 - \frac{|\ 重複する内容語の出現回数\ |}{|\ 文中の内容語の出現回数\ |} \quad (2)$$

表 2: BLEU と Adequacy による対話文生成の自動評価
（表の左側はキーワード 2 つ，右側はキーワード 3 つで文を生成した際の評価を表している）

Methods	BLEU	N=1	N=2	動詞	名詞	BLEU	N=1	N=2	N=3	動詞	名詞 1	名詞 2
Encoder-Decoder	0.147	0.936	0.562	0.884	0.614	0.290	0.982	0.868	0.562	0.908	0.726	0.778
Encoder-Decoder + リランカ	0.158	0.984	0.684	**0.939**	0.729	0.292	0.993	0.905	0.643	0.946	0.818	0.866
PU モデル	0.156	0.967	0.674	0.886	0.755	**0.319**	0.994	0.955	0.733	0.924	0.860	0.898
PU モデル + リランカ	**0.164**	**0.990**	**0.793**	0.935	**0.848**	0.302	**0.998**	**0.986**	**0.813**	**0.951**	**0.909**	**0.937**

$$\text{スコア} = \text{ペナルティ} \times \textit{Jaccard}\ \text{係数} \quad (3)$$

4 実験

4.1 実験設定

実験には opensubtitles.org[1]の日本語字幕データ（約 153 万文）を用いた．本実験では前処理として文節が 3 つ以上，名詞が 2 つ以上，動詞を 1 つ含む文のみを抽出したものを作成した．ここで，文節の検出には CaboCha（version: 0.69）と MeCab（辞書 : IPADic）を用いた．その後，抽出した文に対して tf.idf による文節の先頭の単語の重要度に基づいて複数キーワードを選び，キーワードと元の文を対にしたコーパスを作成した．この時，キーワードは動詞を 1 つと，ひらがなと数字 1 文字以外の名詞から tf.idf の高い順に 1 つまたは 2 つ抽出した．ここで作成したコーパスは 195,364 文である．この内，training データとして 193,364 文，dev セットとして 1,000 文，test データとして 1,000 文を用いた．

動詞 1 つと名詞 1 つまたは動詞 1 つと名詞 2 つからなる文のキーワードを入力として入れた時に，そのキーワードを含む文を正しく出せるかを Encoder-Decoder，Positional Unknown モデル（表 2 中では PU モデルと表記）でそれぞれ実験する．Encoder-Decoder と Positional Unknown モデルはそれぞれリランキングありとなしの場合で比較する．

それぞれのモデルに対する自動評価は BLEU（bi-gram までの一致率）と，文ごとにキーワードを N 個（N=1，2，3）以上出せたら 1，それ以外 0 とした時にキーワードをどのくらい出力することができたかの Adequacy，および各キーワード（動詞または名詞）をどれだけの割合で出せたのかの Adequacy で評価する．また，流暢性はテストデータによる出力のうちランダムに 100 件サンプルしたものを人手で評価する．

ニューラルネットワークのハイパーパラメータは，それぞれ入力語彙を 30,000，出力語彙を 10,000，埋め込み層を 512，隠れ層を 512，アテンションサイズを 512 として実験を行った．また，エポックは 15 まで回し，各 dev セットで BLEU が最大のエポック数を用い，単語ベクトルの初期値には training データで学習した word2vec，最適化のアルゴリズムは Adagrad，学習率は 0.01 を用いた．

[1] http://www.opensubtitles.org/ja (2016 年 12 月 14 日)

4.2 実験結果

表 2 に示すように Encoder-Decoder と Positional Unknown モデルともに，リランカを用いると BLEU はほぼ変わらず，キーワードを出す割合が増えている．また，Encoder-Decoder に比べ Positional Unknown モデルのほうが BLEU，Adequacy ともに良くなっている．例えば，表 3 の事例 1 では正しい位置でキーワードをコピーできている．表 2 の左側と右側を比較すると，キーワードを 2 つから 3 に増やすことで，BLEU が約 2 倍になることが分かる．

表 3 に各モデルの出力例を示す．表 3 の事例 2 において，リランキングなしの Encoder-Decoder とリランキングありの Encoder-Decoder の結果を比較すると，リランキングなしの結果で見られたキーワードの重複がリランキングありの結果ではなくなっている．

出力された 100 件の流暢性を人手で評価したところ，そのうち 50 件は流暢性に問題はなかった．一方，残り 50 件のうち 64%にあたる 32 件が文末が体言で終わるなどの非文で，36%が意味による誤りであった．

5 考察

人手で評価した時のエラー分析の結果として，文末が体言で終わっているようなエラーが多く見られた．これは，今回使用したトレーニングデータの正解データに，最後の文節に動詞または助動詞を含まない非文が 22%含まれていたことが原因だと考えられる．また，名詞を 2 つ以上出さずに文が終わってしまっているエラーも見られたが，これは EOS と相関が高い単語が早い段階で出力されることが原因だと考えられる．したがって，入力を EOS との相関の低い順に並び替えて与えるなどすることで改善できると考えられる．

コーパス 生成される文に含まれやすいキーワードの傾向としては，動詞のキーワードが含まれやすく，次に名詞のキーワードが含まれやすい．特にキーワードが 3 つの時には，1 つ目の名詞よりも 2 つ目の名詞の方が含まれやすいという傾向があった．1 つ目のキー

表 3: Encoder-Decoder と Encoder-Decoder + リランカ，Positional Unknown モデルの比較
（単語の前の P はその単語が入力側からコピーされたことを示している）

Methods	事例 1：名乗る ネロ 彼	事例 2：犯す すべて 罪	事例 3：思う お前 俺
正解	彼 は ネロ と 名乗っ た	犯し た 罪 の すべて を	お前 は 俺 を 馬鹿 だ と 思っ て いる の か ！
Encoder-Decoder	彼 が 憧れ てる の は 久しぶり クール	罪 の 罪 を 犯し た	俺 は お前 を どう 思う ？
Encoder-Decoder + リランカ	彼 は とんだ と 名乗っ た	罪 の すべて を 犯し た	俺 は お前 だ と 思う
Positional Unknown モデル	彼 は P:ネロ と 名乗っ てる	すべて の 罪 を 犯し た	お前 が 俺 だ と 思っ て
Positional Unknown モデル+リランカ	彼 は P:ネロ と 名乗っ た	すべて の 罪 を 犯し た	お前 が 俺 だ と 思う

ワードの方が 2 つ目のキーワードよりも tf.idf が高いので，その文の特徴的な語よりも，普遍的なキーワードの方が出されやすい可能性が考えられる．

表 3 の事例 3 のように主語と目的語の位置が入れ替わっても文が成立するような文は正しく生成できないことがあった．入力に助詞の情報を追加して文の生成を行うことでこのような文も正しく出力できるようになると考えられる．また，全体の出力結果からキーワードに含まれない単語はほとんど出力せず，内容語としてキーワードのみを含むシンプルな文を出力する傾向がある．キーワードの単語を拡張し，キーワードを増やして入力とすることで，複雑な文も出力できるようになると考える．

Positional Unknown モデル Positional Unknown モデルは Encoder-Decoder と比較して，システムの出力側の語彙に含まれないキーワードを出力することができるようになり，BLEU と Adequacy は向上した．キーワードを含んでいなかった出力文として，言語モデルに基づき異なる単語を出力してしまっているというものが多く見られた．今回の実験では，ターゲット側の語彙に含まれない時のみソース側の単語をコピーする学習になっており，言語モデルの学習の方が重視されていることが原因であると考える．学習時にターゲット側の単語がソース側に含まれる場合は，すべてソース側からコピーするというように優先的にコピーをすることで，このようなエラーを減らせると考える．

リランキング リランキングを用いることでキーワードを含む文を出しやすくなり，Encoder-Decoder と Positional Unknown モデルの両方で BLEU を下げることなく Adequacy を向上させることができた．また，表 3 の事例 2 では，リランキングなしの出力とリランキングありの出力はキーワードを同じタイプ数だけ含み，内容語のトークン数も同じである．しかし，リランキングなしの出力が重複する単語を出力しているのに対し，リランキングありの出力では重複を含まない文を出力できているので，リランカのペナルティが効いていると考えることができる．

6 おわりに

本研究ではキーワードを与えることで，文を生成する手法を提案した．この手法は対話行為を用いた手法と比べ，対話行為などのアノテーションの必要がなく簡単に文生成ができる．今回は，簡単化のために動詞を 1 つのみ含む単文の生成を行ったが，今後は複文の生成の実験を行いたい．

参考文献

[1] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *NIPS*. 2014.

[2] Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. In *SSST*, 2014.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *ICLR*, 2015.

[4] Tsung-Hsien Wen, Milica Gasic, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. Semantically conditioned LSTM-based natural language generation for spoken dialogue systems. In *EMNLP*, 2015.

[5] Ondřej Dušek and Filip Jurcicek. Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings. In *ACL*, 2016.

[6] Thang Luong, Ilya Sutskever, Quoc Le, Oriol Vinyals, and Wojciech Zaremba. Addressing the rare word problem in neural machine translation. In *ACL*, 2015.

[7] Ioannis Konstas and Mirella Lapata. Concept-to-text generation via discriminative reranking. In *ACL*, 2012.

[8] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *CoNLL*, 2016.

[9] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. Incorporating copying mechanism in sequence-to-sequence learning. In *ACL*, 2016.

[10] Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. Modeling coverage for neural machine translation. In *ACL*, 2016.