2U-07

機能語のテンプレートに基づく N-gram による文章生成

新美佳吾 † 亀谷由隆 ††
† 名城大学大学院理工学研究科情報工学専攻 †† 名城大学理工学部情報工学科

1 はじめに

会話ボットの発話等を目的とした N-gram による文章生成に関する研究が長年行われている．平滑化を施していない N-gram の場合，小さなコーパスから学習するとコーパス中の文章をそのまま生成することが多い．一方，平滑化を施すと，意味の通らない文章が生成されやすくなる．本稿では，小さなコーパスから自然な文章を生成することを目的とし，機能語のテンプレートを生成する N-gram を用いた文章生成手法を提案する．生成されたテンプレートでは内容語の入る場所が空欄となっており，内容語は前後の単語を考慮して後から埋められる．また，空欄に入る内容語を品詞の活用型・活用形まで区別することで自然な文章を生成させる．Twitter への投稿文を用いた評価実験の結果を報告する．

2 提案手法

本研究では，(1) ツイートの取得，(2) 内容語の抽出とタグ化，(3) N-gram の構築と文章生成，(4) 評価を順に行う．

2.1 ツイートの取得

まず，Twitter API を利用し，特定人物のツイートを取得する．その後，取得したツイートに対して正規化を行う．URL やリプライ，ハッシュタグの要素を削除する．削除した内容を下記に示す．

- URL：「http://」もしくは「https://」で始まる半角英数字と:/.-_の記号
- リプライ：@から連続する半角英数字と_の記号
- ハッシュタグ：#から連続する文字列
- 引用：「RT」の文字列および二重引用符の中の：で始まる文字列

2.2 内容語の抽出とタグ化

正規化を行ったツイートに対し，形態素解析器 MeCab による文章の分割，内容語のタグ化を行う．動詞と形容詞の自立語に対して，活用型・活用形を英数字の組み合わせでタグに置き換える．動詞または形容詞の活用型・活用形については IPA 品詞体系を元に分類し，1 番目の英大文字が品詞，2 番目の i が自立語，次の 1 文字以上の英字の組み合わせが活用型，最後の数字が活用形を示す．表 1 に内容語の品詞とそれを一般的に置き換えたタグの対応の例を示す．また，この際に品詞, 活用型・活用形毎に出現回数と出現確率とともに内容語を辞書に記録する．

表 1: 内容語の品詞と置き換えタグの対応の例

数字	[NUM]
固有名詞 (人名)	[NAME]
普通名詞 (一般，サ変可能)	[N]
形容動詞 (一般)	[B]
形容詞自立・アウオ段連用タ接続	[AiA20]
動詞自立可変・来る・未然形	[ViC10]

2.3 N-gram の構築と文章生成

次に， N-gram の構築と文章生成を行う．今回，機能語のテンプレートを生成することを目的とした N-gram と内容語を埋める際に使用する N-gram を構築する．便宜上，前者を F-gram (Function word N-gram)，後者を C-gram (Content word N-gram) と呼ぶ．F-gram は内容語をタグ化を施した文章を元にした長さ 1 から 4 の前向き N-gram， C-gram はタグ化を施していない文章を元にした前向き 2-gram と後ろ向き 2-gram である [1]．また F-gram には Kneser-Ney 平滑化 [2] を施す．Kneser-Ney 平滑化は現在の平滑化手法の中で精度が良いとされ，これを適用することで，平滑化処理の施されていない N-gram と比べて，コーパスには出現しなかった文字列が文章として生成できる．

文章生成は，機能語のテンプレートの生成，内容語の選択の順に行う．N-gram による文章生成と同じく，出現確率を元に F-gram からテンプレートを生成する．内容語を穴埋めする際は，前後の形態素から内容語を選択する．生成したテンプレートに以下のような形態素列が見つかったと仮定する．

$$w_1\ [t]\ w_2$$

w_i は機能語または穴埋めされた形態素，$[t]$ は置き換えられたタグのことを示す．C-gram を元に式 1 から内容語の出現確率を計算し，内容語を選択する．

$$\begin{aligned} P(t|w_1,w_2) &= P(t,w_1,w_2)/P(w_1,w_2) \\ &= P(t,w_1,w_2)/\sum_{t'} P(t',w_1,w_2) \\ &\propto P(t)P(w_1|t)P(w_2|t) \end{aligned} \quad (1)$$

またテンプレートを生成する F-gram は平滑化を施しているため，式 1 では出現確率を計算できない場合がある．その場合は，タグに対しての出現確率を元に内容語を選択する．

2.4 生成した文章の評価

評価方法として，Kneser-Ney 平滑化を施した 2-gram を用いてパープレキシティと文章の形態素数で散布図をプロットし，その分布を比較する．コーパスの文章と生成した文章を形態素解析器 MeCab で形態素毎に分かち書きし， 2-gram の出現確率を元にパープレキシティを算出する．形態素数 K の文章のパープレキシティは以下の式 2 で計算する．

$$perplexity = \frac{1}{K}\sum_{k=1}^{K} \log P(w_k|w_{k-1}) \quad (2)$$

Sentence Generation using an N-gram Model based on Function-Word Templates
† Niimi Keigo
†† Kameya Yoshitaka
Division of Information Engineering, Graduate School of Science and Technology, Meijo University (†)
Department of Information Engineering, Faculty of Science and Technology, Meijo University (††)

このパープレキシティの値が小さいほど，構築された N-gram から文章がかけ離れていないことを示す．

3 実験

今回，実際に用いるコーパスは芸能人 A のツイートデータを使用する．Twitter API で取得した 3,200 ツイートから正規化を施し，形態素数 3 以下のツイートを除いた合計 2,529 ツイートをコーパスとして用いる．表 2 にコーパスの一部と，その文章のパープレキシティと形態素数を示す．最左のカラムはパープレキシティを，中央のカラムは形態素数を示す．

表 2: コーパスの一部

1.091	10	配信遅れて申し訳ありません今日の田中です。
1.872	8	本日ラジオ。10 月ラスト。。。
2.409	18	有吉の壁 第 5 回大会 ロケ終了。 今回も最高でした。感謝感謝。。。
3.264	25	紅茶に浮かぶレモンになりたい。 これを言ってるのが先輩オジサンで、 とりあえずビンタの下克上。。。

次に F-gram と C-gram を構築し，文章を生成する．今回は 4-gram で文章を 10,000 文生成した．表 3 に生成したテンプレートを，表 4 にテンプレートから内容語を穴埋めして生成した文章を示す．最左のカラムはパープレキシティを，中央カラムは形態素数を示す．

表 3: 生成したテンプレートの一例

配信 [ViI20] てごめんなさい。今日の [NAME] です。
今年も [N] 降臨。。。
配信 [ViI20] て [AiA30]。今日の [NAME] で ございます。また [N][N]。。。
ダンサーインザダーク。 [N] が [AiA30]。。。[ViI20] よう。。。
毎回 [NAME] さんの [N] は『[N]』土曜 [N]。
[N][N][N] 言い訳は色々 その後ぶっ放して旅続報必ず。。。
[N][N][N][N][N]

表 4: 生成した文章の一例

1.860	10	配信遅れてごめんなさい。今日の山根です。
2.183	7	今年もカリスマ降臨。。。
2.424	18	配信遅れてデカイ。今日の田中で ございます。またネイチャーボーイ。。。
2.718	13	ダンサーインザダーク。 目玉が良い。。。忘れよう。。。
3.411	12	毎回飯尾さんの俺は『艶』土曜 9 時。
5.938	15	これこれ会言い訳は色々 その後ぶっ放して旅続報必ず。。。
10.516	5	服装これ正気最高ダチョウ

次に評価を行う．図 1 に提案手法 (tmps_kn) で生成した文章とコーパスの文章のパープレキシティと形態素数をプロットした散布図を示す．縦軸がパープレキシティ，横軸が形態素数を示す．また今回は提案手法で生成した 10,000 文のうち，コーパスの文章数と同じ 2,529 文を無作為に選んでプロットした．

形態素数が 30 以下の部分では生成した文章とコーパスの文章の分布が重なっている．このことから，表 2 の 1 行目と表 4 の 1 行目，3 行目のように，また 3 連続の句点のように，コーパスと似た文章が生成していることがわかる．また提案手法で生成した文章は，パープレキシティの値がコーパスの文章より全体的に大きいことから，文章の幅を広げていると想定できる．

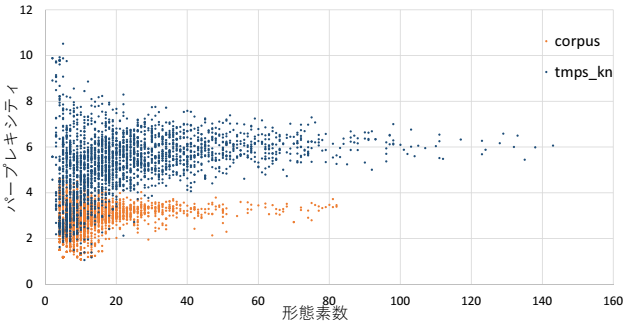

図 1: パープレキシティと形態素数による評価結果

次に， N-gram による文章生成手法はコーパスと同じ文章を生成することがあるため，同じ文章を生成する割合を調べる．提案手法と通常の N-gram による文章生成手法 (ngram)，Kneser-Ney 平滑化を施した N-gram による文章生成手法 (kn_ngram)，機能語の並びを生成し辞書から内容語を選択する文章生成手法 (tmp_kn) の 4 つの手法で比較する．表 5 に，それぞれの手法で 10,000 文の文章を生成し，コーパスと同じ文章の算出した割合を示す．中央のカラムは生成した文章 10,000 文に対してコーパスと同じ文章の割合を，最右のカラムはコーパス 2,529 文に対してコーパスと同じ文章の割合を示す．このことから，通常の N-gram による文章生成よりもコーパスと同一の文章を生成することが少なく，図 1 の分布が重なっている部分でも，同一の文章であることが少ないと考えられる．

表 5: 各手法によるコーパスとの同じ文章の割合

tmps_kn	1.67%	6.60%
ngram	18.10%	71.59%
kn_ngram	2.74%	10.83%
tmp_kn	0.01%	0.04%

4 おわりに

本研究では，機能語のテンプレートを生成する N-gram を用いた文章生成手法を提案した．内容語の出現確率だけでなく，前後の単語を考慮して内容語を穴埋めした．また内容語を品詞だけでなく，活用型・活用形まで区別することで，テンプレートを用いた文章生成の弱みである，活用型・活用形の違いによる文章の不自然さを解決することができた．

会話や発話における個性を再現する研究の一環として本稿では文章生成手法を提案した．先に機能語の並びを生成後に内容語を穴埋めするテンプレートによる文章生成手法は，個人の口癖パターンや価値観を文章に反映させやすい．会話や発話における個性を再現するために，そういった要素をどのように抽出し反映させるかが今後の課題である．

参 考 文 献

[1] 渡辺由貴，中田豊久: 双方向マルコフ連鎖を用いた文章自動生成，情報処理学会第 76 回全国大会 (2014)

[2] R. Kneser and H. Ney: Improved backing-off for for M-gram language modeling, Proc. of ICASSP-95, pp.181–184 (1995)