

漸進的な言語処理のためのRNNを用いた残存文長の推定

河村 天暉^{†,a)}大野 誠寛^{†,b)}松原 茂樹[‡][†]東京電機大学大学院未来科学研究科, [‡]名古屋大学情報連携総括本部{^{a)}18fmi07@ms, ^{b)}ohno@mail}.dendai.ac.jp

1 はじめに

同時通訳や字幕生成などのリアルタイム音声言語システムでは、入力に対して漸進的に処理を行う必要があり、処理の正確さを保ちつつ、遅延時間を抑えることが求められる。このような処理を実現するにあたり、意味的なまとまりをもつ文が今後どれだけ続くかという情報は、重要な手がかりとなりうる。例えば、文がもう少しで終わることが分かれば、処理の正確さを保つため、同時通訳における訳出タイミング [1] や、読みやすい字幕とするための改行挿入タイミング [2] を遅らせるという判断が可能となる。

これまでに著者らは、文節が入力されるごとに文の残りの長さ（以下、残存文長）をSVMを用いて推定する手法（以下、従来手法）を提案している [3]。この手法は、残存文長が1文節、2~3文節、4文節以上の3クラスのいずれであるかを推定できる。しかし、その推定精度は十分ではなく、また残存文長をより細かい粒度で推定するための拡張が容易ではない。

そこで、本稿ではRNN (Recurrent Neural Network) [4] を用いた残存文長の推定手法を提案する。提案手法では、RNNを用いることにより推定精度の向上が期待できるだけでなく、RNNの出力層の次元数を、最も長いと想定する文の長さ（文節数）とし、連続した自然数の上で残存文長を推定することができる。

2 残存文長

日本語文を構成する要素には、単語、文節、節などが存在するが、本研究では文の長さを測る単位として、文節を使用する。文 s が n_s 個の文節から成り、文頭から x 番目の文節 b_x まで既に入力されているとき（すなわち、既入力文節数が x であるとき）の残存文長 $RL(s, x)$ を、式 (1) により定義する。

$$RL(s, x) = n_s - x \quad (1)$$

3 RNNを用いた残存文長の推定

提案手法では、1文 ($s = b_1 \dots b_{n_s}$) を構成する文節が入力されるごとに、文頭から現在入力された文節までの単語系列（形態素系列）をRNNに入力し、そのときの残存文長（文節数）を推定する。この推定を、文節 b_1 が入力されてから文節 b_{n_s-1} が入力されるまで繰り返す。

提案手法の概要を図1に示す。図1は、以下の

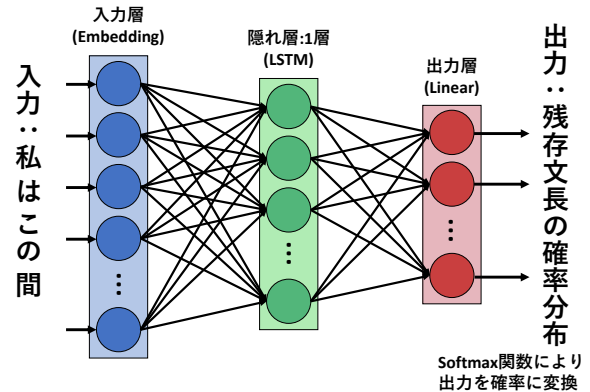


図1: 提案手法の概要

文 s : 私 は / この 間 / 学校 に / 行っ た。

の中の文節 b_2 「この 間」が入力されたときの残存文長 $RL(s, 2)$ をRNNを用いて推定する様子である*1。RNNへの入力は、文頭から現在入力された文節までの単語系列（図1では、「私 は この 間」）とし、RNNの出力は、そのときの残存文長 $RL(s, 2)$ の確率分布となる。この確率分布に基づいて、どの値を最終的な推定結果とするかについては様々な方法が考えられるが、提案手法では最大の確率値をとる残存文長を推定結果とする。また、RNNの出力層の次元数は、最も長いと想定する文の長さ（文節数）とし、連続した自然数の上で残存文長を推定する。

なお、提案手法で用いるRNNの隠れ層は1層とし、LSTM (Long Short Term Memory) [5] により構成されるものとする。また、RNNに入力される各単語はone-hotベクトルで表現されているものとする。さらに、単語系列の入力方向については、文頭から順に入力していく順方向 (FW)、文末から逆順に入力していく逆方向 (BW)、順方向と逆方向の出力を連結させた両方向 (DW) の3種類を試すこととし、それぞれによる推定を行う。

4 評価実験

提案手法の有効性を評価するために、新聞記事文を用いて残存文長の推定実験を実施した。特に本実験では、提案手法がどの程度、連続した自然数の上で残存文長を推定できるのかについて検証する。なお、同時通

*1 本論文では、単語境界を半角スペース、文節境界をスラッシュで表す。

表 2: ChanceRate と提案手法 (BW) の適合率 (P), 再現率 (R), F 値 (F)

		ChanceRate			提案手法 (BW)		
		P(%)	R(%)	F	P(%)	R(%)	F
残 存 文 長	1 文節	11.12 (305/2,742)	13.33 (305/2,291)	12.13	35.71 (941/2,635)	41.07 (941/2,291)	38.21
	2 文節	10.86 (290/2,670)	12.97 (290/2,239)	11.82	17.89 (513/2,867)	22.91 (513/2,239)	20.09
	3 文節	10.33 (261/2,527)	12.28 (261/2,129)	11.22	13.12 (409/3,117)	19.21 (409/2,129)	15.59
	4 文節	9.68 (225/2,325)	11.31 (225/1,989)	10.43	10.39 (228/2,194)	11.46 (228/1,989)	10.90
	5 文節	8.82 (184/2,087)	10.13 (184/1,817)	9.43	9.45 (163/1,725)	8.97 (163/1,817)	9.20
	6 文節	7.84 (143/1,825)	8.86 (143/1,615)	8.31	8.97 (151/1,683)	9.35 (151/1,615)	9.16
	7 文節	6.87 (107/1,558)	7.57 (107/1,408)	7.20	8.30 (124/1,494)	8.81 (124/1,408)	8.55
	8 文節	5.86 (77/1,313)	6.37 (77/1,213)	6.11	6.37 (83/1,302)	6.84 (83/1,213)	6.60
	9 文節	5.08 (55/1,082)	5.27 (55/1,045)	5.17	6.09 (54/886)	5.17 (54/1,045)	5.59
	10 文節	4.18 (37/886)	4.32 (37/867)	4.25	5.06 (29/573)	3.34 (29/867)	4.03

表 1: 46 クラス分類の文節単位正解率 (%)

ChanceRate	9.33 (1,757/20,593)
提案手法	FW 11.50 (2,369/20,593)
	BW 13.49 (2,777/20,593)
	DW 11.84 (2,438/20,593)

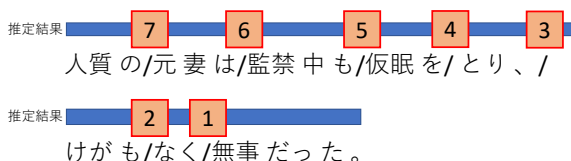


図 2: 1 文全体を正しく推定できた例

訳などの音声言語システムへの応用を考える場合は、話し言葉特有の非文法的な言語などにも対処する必要があるが、本稿では、端緒として、整った文を対象に検証する。

4.1 実験概要

実験データには、京都大学テキストコーパス Ver.4.0*2 (毎日新聞 95 年 1 月 1 日から 17 日までの全記事と、1 月から 12 月までの社説記事) のうち、文の長さが 1 文節であるものを除いた 38,115 文を使用した。それらのデータのうち、1 月 4 日、1 月 5 日の全記事 2,291 文をテストデータ、1 月 9 日の全記事 1,212 文を開発データ、残りの 34,612 文を学習データとして用いる。形態素情報や文節境界情報は、学習時・テスト時ともに、京都大学テキストコーパスのものをそのまま利用した。学習データにおける異なり語は 29,724 語であったため、これに未知語タグを加えて、単語の one-hot ベクトルの次元数を 29,725 とした。テスト

*2 <http://nlp.ist.i.kyoto-u.ac.jp/index.php?京都大学テキストコーパス>

データにおける未知語率は 2.88% であった。また、学習データにおける最長の文の長さは 47 文節であったため、RNN の出力層の次元数を 46 とした。

提案手法との性能比較のために、学習データより、各入力文節位置における 1 文の残存文長を集計し、その割合に従ってランダムに残存文長を出力する手法 (以下、**ChanceRate**) を用意した [3]。

評価では、残存文長を示す文節数 (全 46 クラス) などの推定における適合率、再現率、F 値、および、全推定における正解率 (以下、文節単位正解率) を測定した。さらに、1 文中の全ての推定において正解した文の割合 (以下、文単位正解率) も測定した。なお、正解の残存文長が 47 文節以上である場合は、46 文節と推定した場合に正しく推定したと判定した。

RNN の学習は、Chainer V4.0*3 を介して行った。学習アルゴリズムには Adam を採用した。パラメータの更新はミニバッチ学習 (学習率 0.001, バッチサイズ 128) により行い、更新時にユニットを 0.1 の確率でドロップアウトさせた。エポック数は 50 とした。入力層の出力ベクトルと隠れ層の次元数は統一したうえで、順方向 (FW) では 400, 逆方向 (BW) では 800, 両方向 (DW) では 800 とした。これらの値は、200, 400, 600, 800, 1000 の 5 通りで学習を行い、開発データの推定において最も正解率の高かったものを採用した。

4.2 実験結果

表 1 に文節単位正解率を示す。提案手法のいずれも ChanceRate を上回り、その中で BW は 13.49% と最も高い文節単位正解率を達成した。一方、提案手法における文単位正解率は、FW が 1.05% (24/2,291), BW が 0.56% (15/2,291), DW が 0.65% (13/2,291) であった。1 文中の全推定において正解することは難しいといえるが、提案手法によって 1 文全体を正しく推定できた例もいくつか存在した。そのような例を図 2 に示す。8 文節からなる文に対して、提案手法 (FW) が全推定を正解した例である。

*3 <https://chainer.org/>

表 3: 3 クラス分類の実験結果 (適合率: P, 再現率: R, F 値: F, 文節単位正解率: A)

	残存文長: 1 文節			残存文長: 2~3 文節			残存文長: 4 文節以上			A(%)
	P(%)	R(%)	F	P(%)	R(%)	F	P(%)	R(%)	F	
Chance Rate	11.42 (269/2,420)	11.74 (269/2,291)	11.42	21.19 (978/4,615)	22.39 (978/4,368)	21.77	67.67 (9,173/13,556)	65.83 (9,173/13,934)	66.74	50.60
SVM	26.66 (1,511/5,667)	65.95 (1,511/2,291)	37.97	30.36 (1,321/4,394)	30.24 (1,321/4,368)	30.15	79.17 (8,338/10,532)	59.84 (8,338/13,934)	68.16	54.25
TCM -FW	24.34 (834/3,426)	36.40 (834/2,291)	29.18	25.01 (1,029/4,114)	23.56 (1,029/4,368)	24.26	73.94 (9,651/13,053)	69.26 (9,651/13,934)	71.52	55.91
TCM -BW	35.30 (800/2,266)	34.92 (800/2,291)	35.11	28.73 (1,639/5,670)	37.29 (1,639/4,368)	32.46	76.68 (9,706/12,657)	69.66 (9,706/13,934)	73.00	58.93
TCM -DW	26.26 (836/3,183)	36.49 (836/2,291)	30.54	25.85 (1,146/4,433)	26.24 (1,146/4,368)	26.04	74.97 (9,729/12,977)	69.82 (9,729/13,934)	72.31	56.87
TCE -FW	26.17 (714/2,731)	31.17 (714/2,291)	28.43	26.11 (1,039/3,979)	23.79 (1,039/4,368)	24.90	73.92 (10,263/13,883)	73.65 (10,263/13,934)	73.79	58.35
TCE -BW	36.54 (768/2,102)	33.52 (768/2,291)	34.96	29.46 (1,605/5,448)	36.74 (1,605/4,368)	32.70	76.62 (9,994/13,043)	71.72 (9,994/13,934)	74.09	60.05
TCE -DW	27.66 (745/2,693)	32.52 (745/2,291)	29.90	26.50 (1,147/4,329)	26.26 (1,147/4,368)	26.38	74.89 (10,163/13,571)	72.94 (10,163/13,934)	73.90	58.54

表 2 に, 残存文長の長さごとの適合率・再現率, F 値を示す. 表 2 は, 提案手法の中で最も高い文節単位正解率を達成した BW に着目して, ChanceRate と比較したものである. また, 残存文長の長さのうち, 1 文節~10 文節の評価結果のみを抜粋している. F 値を比較すると, 約 8 割のクラス (表 2 の 10 クラス中 8 クラス, 全クラスにおいても, 両手法の F 値が 0 であったものを除いた 24 クラス中 19 クラス) で, 提案手法 (BW) が ChanceRate を上回っていた.

以上のことから, 連続する自然数の上での残存文長の推定を, 提案手法によって実現できる可能性を確認した.

5 従来手法との比較実験

前章では, 提案手法によって, 連続した自然数の上で残存文長を推定できることを検証した. 本章では, 従来手法 [3] との比較実験を実施し, 残存文長の推定精度の向上における, 提案手法の有効性を検証する.

従来手法は, 残存文長が 1 文節, 2~3 文節, 4 文節以上の 3 クラスのいずれであるかを推定するものであり, 提案手法のように, 連続した自然数の上での残存文長の推定はできない. 従来手法との精度比較を行うためには, 提案手法の出力を, 従来手法に合わせて 3 クラスに変換する必要がある. 3 クラスに変換する方法は多々考えられるが, 本実験では以下の 2 種類の交換方法をそれぞれ採用し, 従来手法と比較した.

・提案手法 [TCM (3-class Classification by Maximum value)]: RNN が出力する確率分布において, 最大の確率値をもつ文節数 (すなわち, 提案手法による交換前の推定結果) が, 交換先の 3 クラスの中で,

どのクラスに属するかを求め, そのクラスを推定結果とする.

・提案手法 [TCE (3-class Classification by Expected value)]: RNN が出力する確率分布の期待値 (小数点第 1 位を四捨五入) が, 交換先の 3 クラスの中でどのクラスに属するかを求め, そのクラスを推定結果とする.

5.1 実験概要

以下の 2 点を除き, 4.1 節と同じ設定で実験した.

(1) 比較手法として, 以下の 2 つの手法を用意した.

・ChanceRate: 学習データより, 各入力文節位置における 1 文の残存文長を集計し, その割合に従ってランダムに 3 クラスを出力する.

・従来手法 [SVM]: 学習データの文集合 \mathbf{S} における入力文節数ごとの平均残存文長 ($ARL(\mathbf{S}, x) = \frac{\sum_{s \in \{s | n_s > x, s \in \mathbf{S}\}} RL(s, x)}{|\{s | n_s > x, s \in \mathbf{S}\}|}$), 入力文節の語彙情報, 既に入力された文節列の構文情報を素性とし, SVM により 3 クラスを出力する [3]. なお, 学習データは 4 章のものと同じである.

(2) 提案手法の RNN における入力層と隠れ層の出力ベクトルの次元数は, TCM と TCE それぞれにおいて 4 章と同様の方法で調節を行った結果, 両方とも FW では 400, BW では 200, DW では 200 とした.

5.2 実験結果

表 3 に, 残存文長のクラスごとの再現率・適合率・F 値, また, 文節単位正解率を示す. クラス「1 文節」の F 値は従来手法 [SVM] が最大値を得ているものの, 他の

表 4: 3 クラス分類の文単位正解率 (%)

SVM	TCM			TCE		
	FW	BW	DW	FW	BW	DW
0.79 (18/2,291)	2.79 (64/2,291)	2.14 (49/2,291)	2.58 (59/2,291)	2.79 (64/2,291)	2.14 (49/2,291)	2.44 (56/2,291)

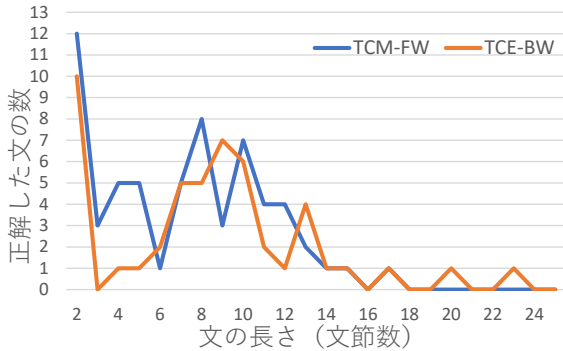


図 3: 文の長さごとの正解文数

クラスの F 値や文節単位正解率に着目すると、提案手法が従来手法を全体的に上回っていることがわかる。次に、文単位正解率を表 4 に示す。提案手法はいずれも、従来手法を上回る文単位正解率を達成している。

以上から、3 クラス上での残存文長の推定における精度向上に提案手法が有効であることを確認した。

5.3 考察

本節では、提案手法の中で推定精度を比較し、6 種類 (3 クラスへの変換方法 2 種×入力方向 3 種) の各方法の特徴を考察する。

まず 3 クラスへの変換方法 TCM と TCE との間で比較する。表 3 の文節単位正解率に着目すると、全ての入力方向 (FW, BW, DW) において TCM より TCE が上回っていた。また、表 3 の各クラスでの F 値をみると、クラス「1 文節」では TCM が高い F 値を達成しているが、他のクラスでは TCE が優勢であることが確認できる。一方、表 4 の文単位正解率では両者にほとんど差はなかった。これらのことから、TCE が比較的優位であると考えられる。

次に、入力方向 FW, BW, DW の間で比較する。表 3 の文節単位正解率、また、各クラスでの F 値に着目すると、3 クラスへの変換方法の両方法 (TCM, TCE) において、BW が最も高く、FW が最も低かった。一方、表 4 の文単位正解率では、BW が最も低く、FW が最も高い結果となり、逆の結果が得られた。

そこで以下では、このように逆の結果が得られた理由を考察するため、文節単位正解率が最も低いが文単位正解率が最も高かった TCM-FW と、文節単位正解率が最も高いが文単位正解率が最も低かった TCE-BW とを、より詳細に比較分析する。

図 3 に、TCM-FW と TCE-BW の各手法において、1 文中の全推定が正解した文の数を文の長さごとに集

計したグラフを示す。横軸は文の長さ (文節数) を、縦軸は正解した文の数を表している。図 2 の横軸が 2 文節から 5 文節の範囲に着目すると、TCM-FW が TCE-BW の正解文数を大きく上回っていることがわかる。TCE-BW と比べて、TCM-FW は比較的短い文の推定に有効であるといえ、そのことが TCM-FW の高い文単位正解率につながったと考えられる。

6 おわりに

本論文では、RNN を用いて、1 文を構成する文節が入力されるごとに残存文長を推定する手法を提案した。連続した自然数の上での残存文長の推定実験では、提案手法が ChanceRate を上回る文節単位正解率 (13.49%) を達成し、その実現可能性を示すことができた。従来手法との比較実験では、1 文節、2~3 文節、4 文節以上の 3 クラス上での推定において、従来手法を 5.80% 上回る文節単位正解率 60.05% を達成し、提案手法の有効性を確認した。

今後は、Attention の導入などによる精度向上、また、話し言葉への対応に取り組みたい。

謝辞 本研究は、一部、科学研究費補助金基盤研究 (B) (No. 26280082) 及び (C) (No. 16K00300) により実施した。

参考文献

- [1] 笠 浩一朗, 松原 茂樹, 稲垣 康善, “英日同時翻訳のための依存構造に基づく訳文生成手法,” 信学論, Vol. J92-D, No. 6, pp. 921–933, 2009.
- [2] 村田 匡輝, 大野 誠寛, 松原 茂樹, “読みやすい字幕生成のための講演テキストへの改行挿入,” 信学論, Vol. J92-D, No. 9, pp. 1621–1631, 2009.
- [3] 河村 天暉, 大野 誠寛, 松原 茂樹, “漸進的な言語処理のための残存文長の推定,” FIT2018 講演論文集, Vol. 17 No. 2, pp. 183–184, 2018.
- [4] Yoshua Bengio, Rejean Ducharme, Pascal Vincent, and Christian Jauvin, “A Neural Probabilistic Language Model,” Journal of Machine Learning Research, Vol. 3, pp. 1137–1155, 2003.
- [5] Martin Sundermeyer, Ralf Schlueter, and Hermann Ney, “LSTM Neural Networks for Language Modeling,” Proc. of INTERSPEECH 2012, pp. 194–197, 2012.