6N-02

句読点を同時予測する End-to-End 音声認識

野崎 樹文[1]　稲熊 寛文[2]　河原 達也[2]

[1] 京都大学 工学部情報学科　[2] 京都大学 大学院情報学研究科

1. はじめに

ディープニューラルネットワークの登場による精度向上により、音声認識システムの実用化が進んでいる。しかし、これらのシステムの出力するテキストには一般に句読点が含まれていない。これは音声認識結果の可読性を下げる要因の一つであり、また、機械翻訳などの後続の自然言語処理タスクへの入力としても望ましくない。この問題を解決するために、近年、音声認識結果のテキストに対してディープラーニングモデルを用いて句読点を付与する研究が盛んである [1, 2]。ただし、これらの研究は音声認識の出力結果に対して句読点予測を行うもので、音声認識誤りの影響を受ける一方、音声の情報を活用できていない。

これに対して本研究では、音声を入力として句読点の付いた音声認識結果を一つのモデルで直接出力可能なシステムを提案する。具体的には Transformer [3] の枠組みを使用して、単語系列と句読点系列を同時に予測するモデルを構築する。また、句読点の予測には、句読点が挿入される位置よりも後の文脈の情報が重要であることに着目し、句読点系列の予測を単語系列の予測よりも数ステップ遅らせる方法の検証も行う。さらに、従来のテキスト情報のみから句読点を予測するシステムとの性能や総パラメータ数の違いを比較する。

2. 句読点付き音声認識

2.1 問題設定

本研究では、句読点付き音声認識を、音響特徴量 X を入力とし、句読点を含むトークン系列 Y を出力する問題と定義する。句読点としては、コンマ (,)、ピリオド (.)、クエスチョン (?) の 3 種類のみを考える。

2.2 カスケード接続型モデル

句読点付き音声認識では、音声認識と句読点予測をそれぞれ別々に学習されたモデルを用いて行う手法が主流である [1]。具体的には、まず、Y の部分列であり、句読点以外の単語系列である $W = \{w_i\}_{i=1}^{N}$ を音声認識モデルを用いて予測する。ここで N は単語系列 W の長さを表す。その後、W の各要素 w_i の直後に挿入される句読点を表す系列 $P = \{p_i\}_{i=1}^{N}$ $(p_i \in \{O, COMMA, PERIOD, QUESTION\})$ を句読点予測モデルを用いて予測する。$p_i = O$ は、w_i の後にどの句読点も入らないことを表す。本研究では、句読点予測モデルとして ELECTRA [4] をファインチューニングしたものを用いる。ELECTRA を用いた句読点予測モデルは、W の各要素 w_i を、対応する p_i に 4 クラス分類する問題として学習する。

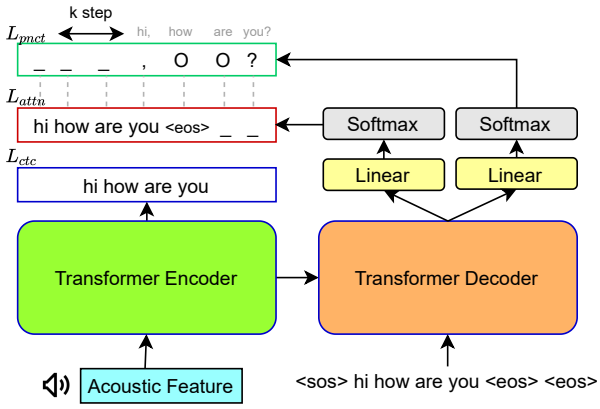

図 1: 提案モデルの概要図

また、句読点付き音声認識を一つのモデルで行うナイーブな方法として、X と Y をペアデータとして直接 End-to-End に音声認識モデルを学習する方法が考えられる。本研究では、以上に述べたカスケード型接続モデルとナイーブな End-to-End モデルをベースラインとして用い、提案モデルと比較する。

3. 提案モデル

句読点付き音声認識を一つのモデルで行う枠組みを提案する。その概要図を図 1 に示す。具体的には、Transformer のデコーダの最終層の出力の隠れ層に対し、二つの線形層を接続し、それぞれ単語系列 W と句読点系列 P の予測に用いる。p_i は、w_i の直後に入る句読点を表すので、left-to-right のデコードでは、w_{i+1} を予測するのと同じタイミングで p_i を予測するのが自然である。しかし、句読点 p_i の予測には、その後の単語系列 $w_j (i < j)$ の情報が重要であると考え、p_i の予測を k ステップ遅らせて、w_{i+1+k} と同時に予測する手法を提案する。学習時の誤差関数は以下の様に表される。

$$\mathcal{L}_{\text{total}} = \lambda_{\text{ctc}}\mathcal{L}_{\text{ctc}} + \lambda_{\text{attn}}\mathcal{L}_{\text{attn}} + \lambda_{\text{pnct}}\mathcal{L}_{\text{pnct}} \quad (1)$$

$\mathcal{L}_{\text{ctc}}, \mathcal{L}_{\text{attn}}$ はそれぞれ単語系列 W の CTC 誤差とクロスエントロピー誤差、$\mathcal{L}_{\text{pnct}}$ は句読点系列 P のクロスエントロピー誤差を表す。$\lambda_{\text{ctc}}, \lambda_{\text{attn}}, \lambda_{\text{pnct}}$ はそれぞれの誤差にかかる係数である。

4. 実験

4.1 データセット

評価実験のためのデータセットとして、MuST-C コーパス [5] を用いる。英独音声翻訳のデータセットの内、英語音声と句読点付き英語スクリプトをペアデータとして用いる。提供されているデータセットの内、train、dev、

Punctuation-Aware End-to-End Speech Recognition: *Jumon Nozaki, Hirofumi Inaguma, and Tatsuya Kawahara (Kyoto Univ.)*

Copyright ©2021 Information Processing Society of Japan.
All Rights Reserved.

表 1: ベースラインモデルと提案モデルの比較。下 3 つが提案モデル。ピリオドとクエスチョンの予測の F1 スコアは、発話の末尾に存在するものと末尾以外に存在するものに対する値をこの順にコンマで区切って載せている。提案モデルの $k = 4$ は発話の末尾以外でクエスチョンを出力しなかったため、その F1 スコアは計算不能であった。

モデル		WER(%)↓	句読点予測精度 (F1)↑			総パラメータ数 (M)
			コンマ	ピリオド	クエスチョン	
カスケード接続型		12.1	75	97, 51	83, 64	167
ナイーブ End-to-End		14.5	70	99, 50	83, 63	58
提案モデル	$\mathbf{k = 0}$	12.0	71	97, 49	84, 61	58
	$\mathbf{k = 1}$	12.0	71	99, 49	84, 58	58
	$\mathbf{k = 4}$	12.2	70	99, 42	83, n/a	58

tst-COMMON をそれぞれ学習・開発・評価データとして用いる。

前処理として、スクリプトは全て小文字に変換する。また、Tilk ら [1] に倣い、エクスクラメーション (!) とセミコロン (;) はピリオド (.) に、ダッシュ(-) とコロン (:) はコンマ (,) に変換し、それらとクエスチョン (?) 以外の記号は削除する。

4.2 実験設定

カスケード接続型モデルのための音声認識モデルとして Transformer を用いる。音声と句読点を全て除いたスクリプトをペアデータとして学習する。学習時は音声のスピード変換によるデータ拡張を行う。推論時は句読点なしのテキストで学習した LSTM 言語モデルと shallow fusion を行う。句読点予測モデルは、 Hugging Face 社[§] が公開している事前学習済みの ELECTRA の base サイズの generator を用いる。最終層に句読点予測のための線形層を追加し、誤差関数に Focal Loss を用いてファインチューニングを行う。

ナイーブな End-to-End モデルの推論時は、句読点付きのテキストで学習した言語モデルと shallow fusion を行う。提案モデルの学習時の誤差関数の $\lambda_{\text{ctc}}, \lambda_{\text{attn}}, \lambda_{\text{pnct}}$ はそれぞれ、0.15、0.55、0.3 とする。提案モデルの推論時の単語系列 W の予測には、句読点なしのテキストで学習した言語モデルと shallow fusion を行う。それぞれのモデルで用いる Transformer は、エンコーダの層数は 12、デコーダの層数は 6、隠れ層の次元は 256 とする。

4.3 評価方法

音声認識の精度を測る指標として単語誤り率 (WER) を用いる。計算の際には出力されたテキストの句読点を除いてから計算する。句読点予測の精度を測る指標としては、コンマ・ピリオド・クエスチョンのそれぞれに対する F1 スコアを用いる。モデルが出力する予測系列は音声認識誤りを含むため、単純に正解系列と比較して F1 スコアを計算することはできない。そこで、予測系列と正解系列のアラインメントをとり、削除誤りと挿入誤り以外の箇所のみを計算の対象とする。また、MuST-C コーパスに含まれるピリオドとクエスチョンはそのほとんどが発話の末尾にあるため、精度が高く出てしまう。そこで、ピリオドとクエスチョンに関しては発話の末尾に含まれるものと、発話の末尾以外に含まれるもので別々に F1 スコアを計算する。その他、ベースラインモデルと提案モデルの総パラメータ数を比較する。

4.4 実験結果

実験結果を表 1 に示す。まず、ナイーブに句読点付きテキストを予測する End-to-End モデルは、カスケード接続型モデルと比較して WER が相対 19.8%悪化した。それに対し、提案モデルは句読点を同時に一つのモデルで予測しながら、音声認識精度はカスケード接続型モデルと同程度を保った。さらに、総パラメータ数もカスケード接続型モデルと比較して大幅に少ない。しかし、句読点予測の精度に関してはカスケード接続型モデルよりも性能が低下した。大規模コーパスで学習された事前学習済モデルの効果が大きいためである。また、提案モデルにおいて、句読点系列の予測を遅らせるステップ数 k が 4 の時に、$k = 0, 1$ の時と比較して句読点予測の精度が低下した。これは k ステップ遅らせることで単語系列と句読点系列で着目する音響特徴量の位置が離れるために学習が難しくなったことが原因であると推測する。

5. おわりに

本研究では、音声を入力として句読点の付いた音声認識結果を一つのモデルで直接出力可能なシステムを提案した。提案モデルは音声認識精度を落とすことなく、カスケード接続型のモデルよりも少ないパラメータ数で句読点を同時予測できることを示した。また、提案手法はリアルタイム出力が可能と考えられる。今後は、異なるアーキテクチャを用いたモデルの改善や、事前学習済モデルから知識蒸留を行う手法 [6] を検討していく予定である。

参考文献

[1] O. Tilk *et al.*: “Bidirectional Recurrent Neural Network with Attention Mechanism for Punctuation Restoration.” *Interspeech*, 3047–3051, 2016.

[2] J. Yi *et al.*: “Self-attention based model for punctuation prediction using word and speech embeddings” *ICASSP*, 7270-7274, 2019.

[3] A. Vaswani *et al.*: “Attention Is All You Need” *NeurIPS*, 5998–6008, 2017.

[4] K. Clark *et al.*: “Electra: Pre-training text encoders as discriminators rather than generators” *ICLR*, 2019.

[5] MA Di. Gangi *et al.*: “MuST-C: a Multilingual Speech Translation Corpus” *NAACL-HLT*, 2019.

[6] H. Futami *et al.*: “Distilling the Knowledge of BERT for Sequence-to-Sequence ASR” *INTERSPEECH*, 2020.

[§] https://huggingface.co/

Copyright ©2021 Information Processing Society of Japan.
All Rights Reserved.