The 33rd Annual Conference of the Japanese Society for Artificial Intelligence, 2019

# ニューラル系列ラベリングを用いた音声認識誤り訂正

Neural Sequence-Labelling Models for ASR Error Correction

池田 大志 Taishi Ikeda　藤本 拓 Hiroshi Fujimoto　吉村 健 Takeshi Yoshimura　礒田 佳徳 Yoshinori Isoda

株式会社 NTT ドコモ
NTT DOCOMO, INC.

本稿では，音声認識処理の後処理として，音声認識誤りを訂正する手法を提案する．提案手法は，まず Bidirectional LSTM を用いて音声認識結果から音声認識誤り箇所を検出する．次に，誤り検出の結果と辞書を用いて単語ラティスを構築し，識別モデルにより最適な単語列を選択することで音声認識誤りを訂正する．評価実験の結果，提案手法により音声認識結果の文字誤り率と単語誤り率が改善することを示す．

## 1. はじめに

近年，音声認識システムと自然言語処理の技術を応用した音声エージェントや音声翻訳などのアプリケーションの普及を背景として，ディープニューラルネットワークによる音声認識手法が盛んに研究されている．一例として，電話会話を録音した Switchboard 音声コーパスを用いた実験では，人手による書き起こしに匹敵する認識結果が報告されている [1]．しかし，実世界で音声認識システムを利用する場合，集音マイクの性能や周囲の雑音など，使用環境による認識精度の低下が依然課題となっている．

このような音声認識誤りは，音声認識システムを利用したアプリケーションの自然言語理解（対話システムや機械翻訳）に悪影響を及ぼすことが報告されている．例えば，Sano らの研究 [2] では，音声エージェントにおいて，ユーザーとシステムによる対話のログデータを分析した結果，音声による入力には 31.7% の音声認識誤りが含まれており，自然言語理解における解析誤りの 57.2% は音声認識誤りが原因であると報告されている．また，Yoshikawa らの研究 [3] では，音声認識誤りが依存構造解析に悪影響を及ぼすことを報告されており，音声認識誤りは自然言語理解に必要な基礎解析の精度低下を招くため，その後段処理である応用アプリケーションの精度低下の原因となる．

一般的な音声認識システムは，音響モデル・単語辞書・言語モデルを組み合わせ構成されている．ある特定の誤りに対しては，各構成要素のいずれかの部分に学習データを追加することで，音声認識誤りの対策ができる．しかし，音声認識システムは各構成要素が相互作用しており，音響モデルや言語モデルにチューニングを施すことは，再学習の時間や精度検証など運用上のコストが掛かる．また，一般に API などで提供されている音声認識システムは，ブラックボックスであることが多く，音声認識誤りを発見し，対策を施そうとしても，音声認識システム自体に手を加えることができない問題もある [4]．

そこで本研究では，音声認識処理の後処理として，音声認識誤りを訂正する手法を提案する．音声認識システムの 1-best の音声認識結果を入力として，音声認識誤りとなる単語を発話内容の正しい表記に変換するテキスト正規化の問題として，音声認識誤り訂正を考える．提案手法では，音声認識誤り訂正を二段階のパイプライン処理により行う．具体的には，まず Bidirectional LSTM（BiLSTM）を用いて音声認識結果から音声認識誤り箇所を検出する．次に，辞書を用いて訂正候補を含む単語ラティスを構築し，構造化パーセプトロンを用いて，最適な単語列を選択することで音声認識結果を訂正する．

本研究では，提案手法の有効性を音声認識アプリケーションのログデータを用いた実験により検証した．その結果，提案手法を用いることで音声認識結果の文字誤り率と単語誤り率が，それぞれ 0.28 ポイントと 0.52 ポイント改善することを確認した．本稿では，本手法では解決することができなかった事例について考察し，今後の課題とその対策について述べる．

Contact: taishi.ikeda.fz@nttdocomo.com

## 2. 関連研究

本研究では，テキスト正規化の問題として，系列ラベリングによる音声認識誤り検出と，構造化パーセプトロンによる識別モデルを用いた二段階のパイプライン処理により，音声認識誤り訂正を行う．そこで本章では，音声認識結果から音声認識誤りを検出する研究と日本語を解析対象としたテキスト正規化の研究について，それぞれ詳細を記述する．

### 2.1 音声認識誤り検出

Byambakhishig らの研究 [5] では，条件付き確率場を用いてコンフュージョンネットワーク上から音声認識誤りを検出し，リランキングにより音声認識誤り単語を訂正することで，音声認識システムの改善を行っている．それに対し，本研究では，ブラックボックスの音声認識システムを利用すると仮定し，コンフュージョンネットワークを利用せず，音声認識システムの 1-best の認識結果を入力とする音声認識誤り訂正タスクに取り組む．

Ghannay らの研究 [6] では，系列ラベリング手法を用いて，音声認識結果の各単語に対して，「正解」または「誤り」のラベルを付与することで，音声認識誤りを検出する手法を提案している．それに対し，本研究では，音声認識誤りを訂正することを目的とするため，音声認識結果の各単語に対して，「操作なし」「置換」「削除」の三種類のラベルに拡張したラベル付与を行う．これらのラベルと辞書を用いて訂正候補を含む単語ラティスの構築を行い，音声認識誤りの訂正を行う．

### 2.2 テキスト正規化

近年，Web テキストを対象としたテキスト正規化手法が盛んに研究されている [7, 8, 9]．テキスト正規化の研究では，入力文中の表記揺れや口語表現を辞書に存在する正規表記に変換

することを目的としている．本研究では，音声認識誤り訂正タスクを，音声認識結果の認識誤りを発話内容の正しい表記に変換するテキスト正規化の問題として考える．

テキスト正規化の手法としては，Encoder-Deocder モデルを用いて，音声認識誤りを正しい表記に翻訳する問題として取り込むことができる．しかし，音声認識結果と音声認識誤りを訂正したペアデータを大量に収集することは難しく，小規模なデータを用いて Encoder-Deocder モデルを学習する場合，データ拡張などの手法が必要となる [9, 10, 11]．そこで本研究では，Kaji ら [7] や Saito らの研究 [8] を参考として，辞書と識別モデルを用いた音声認識誤りの訂正を行う．

## 3. 提案手法

図 1 に提案手法の概要を示す．図 1 では，音声認識結果が「セキュリティ祖父とにログイン」，書き起こしが「セキュリティソフトにログイン」の例を示している．この例では，音声認識結果の「祖父と」の部分が音声認識誤りである．図 1 下部は，BiLSTM を用いて，音声認識結果から音声認識誤りを検出する過程を示している．図 1 上部は，辞書を用いて単語ラティスを構築し，識別モデルにより最適な単語列を選択する過程を示している．以下，それぞれ詳細を記述する．

### 3.1 系列ラベリングを用いた音声認識誤り検出

本研究では，音声認識結果から音声認識誤りを検出するため，形態素解析後の単語列に対して系列ラベリングを行う．また，音声認識誤りを検出するだけではなく，訂正することを目的とするため，音声認識結果の単語列に対して「操作なし」「置換」「削除」の三種類のラベル付与を行う．各単語に対するラベルは，動的計画法を用いて，音声認識結果と書き起こしのアライメントを求め，ラベル付与を行う．

図 2 にラベル付与の例を示す．具体的には，単語が一致する場合はスコア 0 を与え，削除と置換の操作が必要な場合にはスコア 1 を各単語に与える．各単語のスコアを足し合わせアライメントのスコアを計算し，ビタビアルゴリズムによりスコアが最小となるアライメントを求める．得られたアライメント結果を元に，対応する単語が一致する場合は「操作なし」ラベルを付与し，不一致となる場合は「削除」または「置換」ラベルを付与する．図 2 の例では，書き起こしの「セキュリティ」と音声認識結果の「セキュリティ」が一致するため「操作なし」ラベルを付与する．また,「ソフト」と「祖父」が不一致となるため「置換」ラベルを付与し,「と」に対してはアライメント対象が存在しないため「削除」ラベルを付与する．

上記の操作により，BiLSTM の入出力となる単語列 $W = w_1, w_2, \cdots, w_n$，品詞列 $P = p_1, p_2, \cdots, p_n$，ラベル列 $Y = y_1, y_2, \cdots, y_n$ を得る．ただし，$n$ は単語数を示す．また，単語列 $W$ と 品詞列 $P$ は，埋め込みベクトルに変換され，BiLSTM の入力となる．時刻 $i$ の入力となる $\mathbf{x}_i \in \mathbb{R}^{d^{word}+d^{pos}}$ は，$d^{word}$ 次元の単語ベクトル $\mathbf{w}_i \in \mathbb{R}^{d^{word}}$ と $d^{pos}$ 次元の品詞ベクトル $\mathbf{p}_i \in \mathbb{R}^{d^{pos}}$ を結合したものである．単語ベクトルは，Skip-gram などにより事前学習した単語埋め込みを初期値とし，品詞ベクトルは，一様分布によりランダムに初期化し，それぞれ学習中に更新する．BiLSTM は，各時刻の入力として $\mathbf{x}_i$ を受け取り，$d^{hidden}$ 次元のベクトル $\mathbf{h}_i \in \mathbb{R}^{d^{hidden}}$ を計算する．

$$\mathbf{h} = \mathbf{BiLSTM}(\mathbf{x})$$

ただし，$\mathbf{x} = \mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n$ は埋め込みベクトル系列，$\mathbf{h} = \mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_n$ は BiLSTM の出力ベクトル系列である．各ラベルの予測は，各時刻の $\mathbf{h}_i$ を入力として受け取り，アフィン変換とソフトマックス関数により各ラベルに対する確率値のベクトル $\mathbf{z}_i \in \mathbb{R}^{|T|}$ を得る．

$$\mathbf{z}_i = \mathbf{Softmax}(\mathbf{W}_{out}\mathbf{h}_i + \mathbf{b}_{out})$$

ただし，$|T|$ はラベル数，$\mathbf{W}_{out} \in \mathbb{R}^{d^{|T|\times hidden}}$ は重み行列，$\mathbf{b}_{out} \in R^{|T|}$ はバイアスベクトルである．また，目的関数としてクロスエントロピー，パラメーターの最適化には Adam ($\beta_1 = 0.9, \beta_2 = 0.999$) を用いる．

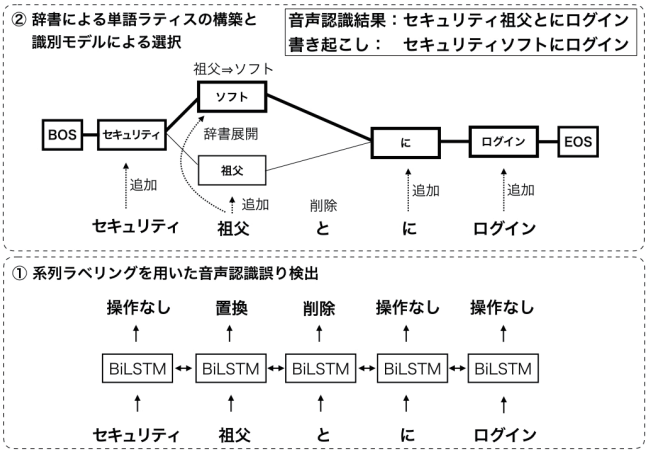


図 1: 提案手法の概要（単語ラティスの太線は正解系列を表す）

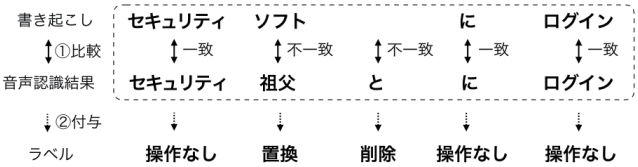


図 2: 動的計画法によるラベル付与

### 3.2 辞書による単語ラティスの構築と識別モデルによる選択

本研究では，辞書を用いて単語ラティスを構築し，構造化パーセプトロンにより最適な単語列を選択することで音声認識結果を訂正する．

まず，単語ラティスの構築方法について説明する．入力単語列 $W$ と BiLSTM による予測ラベル列 $\hat{Y}$ が与えられたとき，BiLSTM により「操作なし」ラベルが付与された単語は，単語ラティスに追加し,「置換」ラベルが付与された単語は辞書により展開した訂正候補とともに単語ラティスに追加する．また,「削除」ラベルが付与された単語は単語ラティスには追加しない．図 1 上部の例では,「操作なし」ラベルが付与された「セキュリティ」は，単語ラティスに追加し,「置換」ラベルが付与された「祖父」は，辞書を参照し，訂正候補が存在するのであれば訂正候補の「ソフト」と「祖父」を単語ラティスに追加する.「削除」ラベルが付与された「と」は単語ラティスには追加しない．本研究では，人手による辞書構築を行い，辞書には音声認識誤りとその訂正後の表記のペアデータが 129 件含まれる．

次に，構造化パーセプトロンを用いて最適な入力単語列を

表 1: 本研究での実験で用いたコーパス概要

| | 学習 | 評価 |
|---|---|---|
| 文数 | 9,724 | 4,490 |
| 誤り文数 | 4,213 | 2,068 |
| 単語数（音声認識結果） | 113,993 | 55,265 |
| 単語数（書き起こし） | 112,699 | 54,896 |

選択する方法について説明する．本研究では，単語列 $W$ と BiLSTM による予測ラベル列 $\hat{Y}$ に対し，音声認識誤りが訂正された単位列 $\hat{W}$ を求める問題を考える．この問題は次のように定式化できる．

$$\hat{W} = \underset{\ell \in \mathbf{L}(\mathbf{W}, \hat{\mathbf{Y}})}{\arg\max}\ w \cdot f(\ell)$$

ここで，$\ell \in L(W, \hat{Y})$ は，入力単語列 $W$ と BiLSTM による予測ラベル列 $\hat{Y}$ により構築された単語ラティス（各ノードは単語）である．$w \cdot f(\ell)$ は，重みベクトル $w$ と素性ベクトル $f(\ell)$ の内積を表す．最適系列は $w \cdot f(\ell)$ の値にしたがって選択される．

素性は，学習データの書き起こし bi-gram 素性を用いる．ここでは，書き起こしの単語列を選択するよう学習を行い，構造化パーセプトロンにおける重みベクトル $w$ の推定を行う．正解データの値 $w \cdot f(\ell)$ が，正解以外のどの値よりも大きくなるように $\hat{W}$ を求める．

重みベクトル $w$ の推定は，平均化パーセプトロン学習に基づいて行う．平均化パーセプトロンでは，正解である単語系列 $W$ が付与された $N$ 個の文が与えられたとき，現在のパラメータ $w^i$ に基づいて一文ずつ最適解 $\hat{W}$ を求め，もしこの系列が正解と異なる場合は重みパラメータ $w^{i+1}$ を下記の式により更新する．

$$w^{i+1} = w^i + f(\hat{\ell}) - f(\ell)$$

ただし，$f(\hat{\ell})$ は正解系列から得られる素性であり，$f(\ell)$ はパラメータ $w^i$ より求めた系列から得られる素性である．もし現在のパラメータに基づいて出力された最適解が正解と一致する場合にはパラメータの更新を行わない．最後に，文数と繰り返し回数の積で平均化した重みパラメータを計算する．以上の二段階のパイプライン処理により，音声認識誤り訂正を行う．

## 4. 実験

提案手法の有効性を検証するため，音声認識アプリケーションのログデータを用いて，音声認識誤り訂正の評価実験を行う．

### 4.1 実験設定

本研究では，重み付き有限状態トランスデューサを用いた音声認識システムを利用し，携帯電話に関する問い合わせに応答する音声認識アプリケーションからログデータを収集する．ここでは，一般に API などで提供されている音声認識システムを利用することを想定とする．そこで，ユーザーから収集した音声発話を用いて，1-best の音声認識結果を取得した．また，人手による音声発話の確認を行い，正解データとなる書き起こしの作成を行った．表 1 に学習と評価に用いたデータ数を示す．

評価データに対して，提案手法の誤り訂正処理を適応し，文字誤り率（Character Error Rate：CER）と単語誤り率（Word Error Rate：WER）がどの程度改善できるかを評価する．CER と WER は，音声認識結果と書き起こしの編集距離を計算し，文字単位の誤り数と単語単位の誤り数を元に求める．以下，CER と WER の定義を示す．

表 2: 音声認識誤り検出結果

| ラベル | 適合率 | 再現率 | F1 値 | 件数 |
|---|---|---|---|---|
| 操作なし | 0.95 | 0.98 | 0.96 | 50,220 |
| 削除 | 0.51 | 0.19 | 0.28 | 1,691 |
| 置換 | 0.61 | 0.43 | 0.50 | 3,354 |

表 3: 音声認識誤り訂正結果

| 訂正方法 | CER | WER | 相対向上数 |
|---|---|---|---|
| 操作なし | 10.10 | 11.59 | 0 (0-0) |
| 削除 | 10.01 | 11.22 | +10 (74-64) |
| 置換 | 9.954 | 11.43 | +34 (125-91) |
| 削除+置換 | **9.819** | **11.07** | +64 (188-124) |
| 削除（正解ラベル） | 8.068 | 8.517 | +298 (298-0) |
| 置換（正解ラベル） | 9.302 | 10.91 | +181 (181-0) |
| 削除+置換（正解ラベル） | 7.248 | 7.820 | +511 (511-0) |

$$\mathbf{CER} = \frac{\text{置換誤り文字数} + \text{削除誤り文字数} + \text{挿入誤り文字数}}{\text{全文字数}} \times 100$$

$$\mathbf{WER} = \frac{\text{置換誤り単語数} + \text{削除誤り単語数} + \text{挿入誤り単語数}}{\text{全単語数}} \times 100$$

また，訂正によって音声認識結果が書き起こしと一致した件数と書き起こしと一致せず差分が発生した件数の差である相対向上数を表 3 に示す．

音声認識結果と書き起こしの形態素解析には，MeCab[*1] とその辞書である Unidic[*2] を利用した．BiLSTM のハイパーパラメータは，学習データの 5%を開発データとして利用し，$d^{word} = 100$，$d^{pos} = 32$，$d^{hidden} = 100$ と設定した．開発データにおいて，WER が最小となるモデルを保存し，評価データで性能の評価を行う．また，BiLSTM のラベル予測には，三種類のラベルを利用するため，$|T| = 3$ となる．

### 4.2 結果

音声認識結果から音声認識誤りをどの程度検出することが可能か検証するため，評価データに対して，音声認識誤り検出の精度評価を行った．表 2 には音声認識誤り検出の精度評価を示す．ここでは，音声認識誤りである単語の検出率を確認するため，「削除」ラベルと「置換」ラベルの再現率に注目する．「削除」ラベルの再現率が 0.19，「置換」ラベルの再現率が 0.43 と各ラベルとも低く，多くの音声認識誤りの検出に失敗していることがわかる．この結果より，音声認識誤り検出の再現率をさらに上げる方法の検討が必要になると考えられる．

次に，音声認識誤りの検出結果から音声認識誤りをどの程度訂正することが可能か検証するため，評価データに対して，音声認識誤り訂正の精度評価を行った．表 3 には音声認識誤り訂正の精度評価を示す．音声認識誤り訂正の効果を測るため，以下の方法を比較することで精度評価を行った．

**操作なし** 音声認識結果に訂正を一切行わない場合．

*1 http://taku910.github.io/mecab/
*2 https://unidic.ninjal.ac.jp/

**削除** 「削除」ラベルを用いて，音声認識誤りを削除した場合．

**置換** 「置換」ラベルを用いて，識別モデルによる訂正を行った場合．

**削除＋置換** 削除と置換を同時に行った場合．

ここでは，**操作なし**と比較し，どの程度 CER と WER が改善したか確認することで，提案手法の有効性を検証する．提案手法である**削除＋置換**を評価データに適応した場合，**操作なし**の CER と WER と比較し，CER が 0.28 ポイント と WER が 0.52 ポイント改善したことから，提案手法により音声認識誤りの訂正が行われていることが確認できる．また，提案手法によって正しく訂正された例を示す．一番目と二番目の例では，提案手法によって助詞の重複や助詞の間違いを正しく訂正している事例を確認できた．

| | |
|---|---|
| 正解 | 暗証 番号 の |
| 認識結果 | 暗証 番号 を の |
| 訂正結果 | 暗証 番号 の |
| 正解 | サービス の 廃止 |
| 認識結果 | サービス が 廃止 |
| 訂正結果 | サービス の 廃止 |

しかし，提案手法により悪化した事例も存在する．例えば，一番目の例のように，不必要に単語を削除してしまう事例が多く存在した．また，提案手法では，二番目の例のような挿入操作に対応できないため，今後の課題として，挿入操作への対応を検討する必要がある．

| | |
|---|---|
| 正解 | 五 月 十 五 日 |
| 認識結果 | 五 月 十 五 日 |
| 訂正結果 | 月 五 日 |
| 正解 | アプリ が ダウンロード でき ませ ん |
| 認識結果 | アプリ がた の でき ませ ん |
| 訂正結果 | アプリ がた の でき ませ ん |

また，表 3 では，BiLSTM の各単語の予測ラベルを全て本来の正解ラベルとした場合の CER と WER を示している．**削除（正解ラベル）**の場合は，**操作なし**と比較し，CER では 2.03 ポイント，WER では 3.07 ポイントと大幅に誤り率が減少している．しかし，**置換（正解ラベル）**の場合は，**操作なし**と比較し CER では 0.79 と WER では 0.67 ポイントと誤り率が少ないことがわかる．この結果から，辞書の網羅性が低かったと考え，読み情報を利用した辞書構築など辞書構築方法の再検討が必要であると考えられる．

## 5. おわりに

本研究では，BiLSTM を用いて音声認識結果から音声認識誤り箇所を検出し，誤り検出の結果と辞書を用いて単語ラティスを構築後，識別モデルにより最適な単語列を選択することで音声認識誤りを訂正する手法を提案した．音声認識アプリケーションのログデータを用いた実験では，提案手法を用いることで音声認識結果の文字誤り率と単語誤り率が改善することを確認した．

今後の課題として，音声認識誤り検出に対する再現率の向上と辞書の構築方法の検討が考えられる．

## 参考文献

[1] Wayne Xiong, Lingfeng Wu, Fil Alleva, Jasha Droppo, Xuedong Huang, and Andreas Stolcke. The microsoft 2017 conversational speech recognition system. In *Proc. ICASSP*. IEEE, 2018.

[2] Shumpei Sano, Nobuhiro Kaji, and Manabu Sassano. Predicting causes of reformulation in intelligent assistants. In *Proc. SIGdial*, 2017.

[3] Masashi Yoshikawa, Hiroyuki Shindo, and Yuji Matsumoto. Joint transition-based dependency parsing and disfluency detection for automatic speech recognition texts. In *Proc. EMNLP*, 2016.

[4] Dong Yu, Jinyu Li, and Li Deng. Calibration of confidence measures in speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 2011.

[5] E Byambakhishig, Katsuyuki Tanaka, Ryo Aihara, Toru Nakashika, Tetsuya Takiguchi, and Yasuo Ariki. Error correction of automatic speech recognition based on normalized web distance. In *Proc. ACL*, 2014.

[6] Sahar Ghannay, Yannick Estève, and Nathalie Camelin. Task specific sentence embeddings for asr error detection. In *Proc. Interspeech*. ISCA, 2018.

[7] Nobuhiro Kaji and Masaru Kitsuregawa. Accurate word segmentation and pos tagging for japanese microblogs: Corpus annotation and joint modeling with lexical normalization. In *Proc. EMNLP*, 2014.

[8] Itsumi Saito, Kugatsu Sadamitsu, Hisako Asano, and Yoshihiro Matsuo. Morphological analysis for japanese noisy text based on character-level and word-level normalization. In *Proc. COLING*, 2014.

[9] Taishi Ikeda, Hiroyuki Shindo, and Yuji Matsumoto. Japanese text normalization with encoder-decoder model. In *Proc. WNUT*, 2016.

[10] Itsumi Saito, Jun Suzuki, Kyosuke Nishida, Kugatsu Sadamitsu, Satoshi Kobashikawa, Ryo Masumura, Yuji Matsumoto, and Junji Tomita. Improving neural text normalization with data augmentation at character-and morphological levels. In *Proc. IJCNLP*, 2017.

[11] Jun Harashima and Yoshiaki Yamada. Two-step validation in character-based ingredient normalization. In *Proceedings of the Joint Workshop on Multimedia for Cooking and Eating Activities and Multimedia Assisted Dietary Management*. ACM, 2018.