

## 音声認識誤り訂正のための疑似誤り付与によるデータ拡張

鈴木悠生

綱川隆司

西田昌史

静岡大学

## 1. はじめに

音声認識システムにおいて生じる認識誤りを軽減するための誤り訂正モデルの学習においては、誤り事例の学習データが不足するという課題がある。本研究では疑似誤りを付与した対話データと T5 を利用し、対話形式の発話における音声認識誤りの訂正をテキストベースで行う。

疑似誤りについては、杉山らによる疑似音声認識誤り付与を行った研究[1]を参考に新たな生成手法を提案する。「単語の部分変化（単語の一部が別の音と認識される誤り）」や「同音異義語」、「分割ミス（単語の間違った区切り方による誤り）」に注目し、疑似誤りデータを作成した。

また、作成したデータを訂正モデルの転移学習に利用し、訂正性能の評価をした。中村らによる前後の発話を文脈情報として訂正に活用する研究結果[2]から、本研究でも入力文として訂正対象発話の前後の発話も追加した場合の性能を比較した。そして、その中で最も性能のよかった手法において学習データを追加した際の性能評価を行った。

## 2. 提案手法

## 2.1 疑似誤り生成手法

疑似誤りデータを作成するために、まずは音声認識誤りの特徴を分析する。2, 3 人が対話している音声に対して Google Speech-to-Text<sup>1</sup>を用いて音声認識を行い、認識結果を観察した。雑音や音声として認識されなかったことなどによる大きな認識抜けを除けば、正しい文章のものと発音が似ているような、部分変化に分類される認識誤りが多かった。この結果をもとに疑似誤り生成を行う。

図 1 に疑似誤り生成方法の例を示す。形態素解析器 janome<sup>2</sup>を用いて文章の読みを取得し、訓令式ローマ字に変換する。1 文字ずつ一定の確率で削除/置換させることで部分変化を再現した。その後ひらがなに変換し、さらに Google Transliterate<sup>3</sup>を用いて最終的な文章に変換する。その際に同音異義語の誤りや、文章を一括で変換させるため分割ミスの誤りも生じさせることができる。

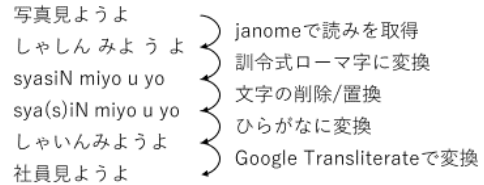


図 1: 疑似誤り生成例

## 2.2 誤り訂正モデル

本研究では T5 を用いて訂正を行う。T5 は Google が 2019 年に発表した、テキストをテキストに変換するという形式で、多岐にわたる自然言語処理タスクに適用できる、深層学習に基づく言語モデルである[3]。事前学習のモデルとトークナイザには日本語の T5 事前学習済みモデルの sonoisa/t5-base-japanese<sup>4</sup>を利用する。そして、疑似誤りを付与した発話データで転移学習を行う。

また、訂正対象の発話だけではなく、文脈情報として前の発話、前後の発話も加えた入力形式のモデルを作成する。

## 3. 実験設定

## 3.1 疑似誤りデータの作成

## 3.1.1 発話データの前処理

発話データには名大会話コーパス<sup>5</sup>を利用する。発話内容以外の情報（話者コードや括弧書きなど）の部分は削除する。また、聞き取れない部分が含まれている発話は利用しないようにした。名前や地名などの固有名詞はコードなどで示されているため、名字由来 net<sup>6</sup>の全国名字ランキングの上位 500 位からランダムに置き換えた。さらに音声認識を考慮し、句読点や記号等も削除した。

以降の実験には、120 対話分 (119,035 発話) を学習用に、2 対話分 (1,274 発話) を評価用のデータとして用いる。

## 3.1.2 疑似誤り付与

2.1 節の手法に従って疑似誤りを付与する。単純な相槌や返事など非常に短い文章は実際の認識時でも誤りにくいものと考え、本実験では 5 文字以上の発話を誤り付与対象とした。子音や母音の削除/置換については、誤り付与の程度について観察した結果から、本実験では 3% の確率で削除/置換を行うようにした。

Data Augmentation by Pseudo Error Generation for Speech Recognition Error Correction

YUSEI SUZUKI, Shizuoka University  
TAKASHI TSUNAKAWA, Shizuoka University  
MASAFUMI NISHIDA, Shizuoka University

<sup>1</sup> <https://cloud.google.com/speech-to-text>

<sup>2</sup> <https://mocobeta.github.io/janome/>

<sup>3</sup> <https://www.google.com/inputtools/services/features/transliteration.html>

<sup>4</sup> <https://huggingface.co/sonoisa/t5-base-japanese>

<sup>5</sup> <https://mmsrv.ninjal.ac.jp/nucc/>

<sup>6</sup> <https://myoji-yurai.net/prefectureRanking.htm>

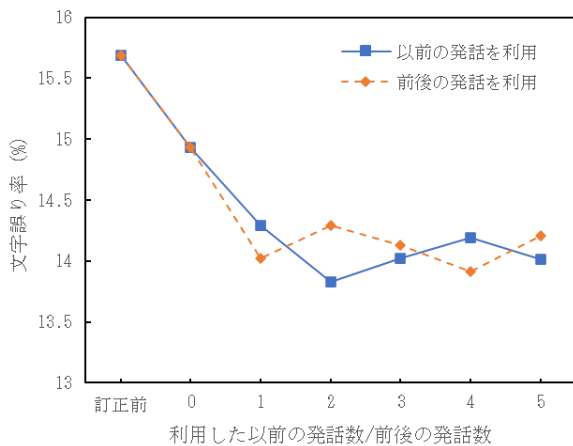


図 2 : 利用した発話数ごとの訂正性能比較

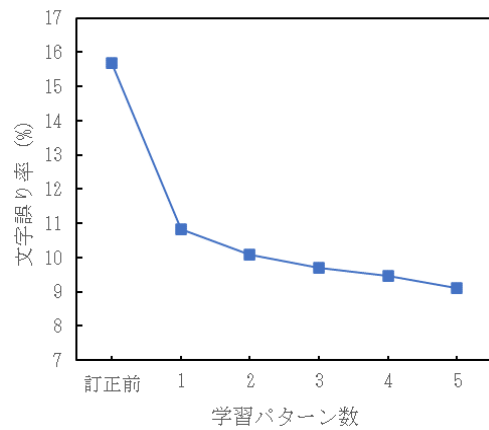


図 3 : 学習パターン数ごとの訂正結果

### 3.2 文脈情報を追加した訂正性能の比較

訂正対象発話の前後の発話を利用した訂正を行い、性能を比較する。対象発話のみを入力とした訂正，対象発話とそれ以前の 1~5 発話を利用した訂正，対象発話との前後の各 1~5 発話を利用した訂正の結果を比較する。本実験では，学習用データ中の 10 対話分 (11,670 発話) のデータで学習を行った。

### 3.3 過去 2 発話を利用した訂正で学習データを追加

3.2 節の実験の結果，訂正対象の発話とそれ以前の 2 発話を利用した訂正が最も良い結果となることがわかった。この手法で学習データを追加し，訂正を行う。本実験では学習用のデータに対して 5 パターン分 (1 パターン=120 対話=119,035 発話) の疑似誤り付与を行い，学習データとして用いた。

## 4. 実験結果

### 4.1 訂正結果の比較

3.2 節の実験結果を図 2 に示す。以後，評価には対話全体の平均文字誤り率を用いる。訂正作業前は 15.69% だった。対象発話以前の 2 発話を利用した訂正が 13.83%，前後 4 発話ずつを利用した訂正が 13.91% でそれぞれ最も小さい数値となった。結果としては以前の 2 発話を利用した訂正が最も良い性能を示した。

### 4.2 学習データの追加結果

3.3 節の実験結果を図 3 に示す。1 パターン分のデータを使った学習時は文字誤り率 10.83% で 5 パターン分での学習時は 9.12% となり，データを追加するごとに性能の改善傾向がみられた。

## 5. 考察

3.2 節の実験時に，利用できる情報は多いほど訂正性能が向上すると考えていた。しかし 4.1 節の結果を受け，以前や前後の発話を利用し始めたときは性能の向上が見ら

れるが，それ以降は同じような数値で上下するようになることがわかった。これは，離れた発話には訂正の手掛かりとなる情報が含まれにくく不要な情報も増えてしまうことや，入力の最大トークン数による入力上限（本研究では 512 とした）が理由であると考えられる。

訂正対象発話の以前の 2 発話を利用した訂正において，学習に使ったデータ量の増加に伴い訂正性能が向上していることが分かった。これはデータを効果的に拡張できているものだと考えられる。

## 6. おわりに

本研究では，会話コーパスに疑似誤りを付与させデータ拡張を行い，T5 を用いて音声認識誤りの訂正を行った。その結果，訂正対象発話の以前の 2 発話が訂正に効果的な影響を与えることがわかった。また，作成した疑似誤りデータを追加して学習を行わせ，訂正性能を向上させられることがわかった。

課題としては，本研究では訂正性能を疑似誤りデータで評価しているため，疑似誤りの範囲内での性能評価となってしまう点が挙げられる。実際の音声認識結果のデータに対する訂正も効果的に行えるか調べる必要があると考えられる。

## 参考文献

- [1] 杉山雅和, 吉村綾馬, 友松祐太, 小町守, "日本語音声認識誤り訂正のための疑似誤りデータ作成と評価", 2021 年度人工知能学会全国大会 (第 35 回), 2021.
- [2] 中村朝陽, 李聖民, 田村鴻希, 吉永直樹, "前後の発話を文脈として考慮するニューラル音声認識誤り訂正", 第 254 回自然言語処理研究会, 2022.
- [3] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J. Liu, "Exploring the Limits of Transformer Learning with a Unified Text-to-Text Transformer", JMLR, Vol.21, pp.1-67, 2020.