K-017

深層学習を利用した盲ろう者向け自動要約筆記システムの提案

Proposal of an Automated Summary Writing System for the Deafblind using Deep Learning

市川 涼介† 村田 勇樹† 堀江 則之† 巽 久行†
Ryosuke Ichikawa Yuki Murata Noriyuki Horie Hisayuki Tatsumi

1. はじめに

盲ろう者とは視覚・聴覚障害のある者を指し、現在全国に 23,200 人程度と推定されている[1]。また、障害の発生順により、“先天盲ろう”,“盲ベース”,“ろうベース”に分類される。本研究では盲ベース（または、点字使用可能な盲ろう者）の触覚刺激によるコミュニケーションの 1 つである要約筆記について考察する。要約筆記は主に聴覚障害者の情報保障として用いられている。しかし、視覚で情報を取得する聴覚障害者と、触覚（指点字など）で情報を取得する盲ろう者とでは、要約筆記に含まれる情報量や理解への調整が異なることから、盲（点字）の知識を持った要約筆記者が付くことが望ましい。そこで本研究は、専門の要約筆記者が付かなくとも、盲ろう者への情報保障を可能とする自動要約筆記システムを提案する。

2. システムの概要

提案するシステムは“音声認識”,“要約”,“出力”の、以下 3 工程にて構成される。

2.1 音声認識

現在、音声認識を可能とする API は Google の“Speech-to-Text”や IBM の“Watson-API”など数多く存在する。その中でも本研究では障害者などのバリアフリーを支援しているアプリケーションである UD トークを使用する。本アプリケーションの利点を以下に示す。

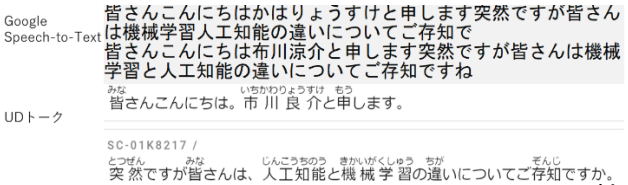

図 1　Google-Speech-to-Text と UD トークの比較

図 1 は Google が提供している API である Speech-to-Text（以下 Google-API）と UD トークを同マイク・同発話内容を文字化したものである。UD トークでは、漢字の誤変換があるものの、それ以外は完全な形で文字化されている。Google-API では、名前の認識や最終センテンスの終助詞に誤りがある。また、Google-API は認識候補が出力されることに加え、句読点が打たれないため、データの扱いやすさに関しても UD トークが優れている。その他にも、UD トークはマルチプラットフォームに対応しており、PC だけでなく、スマートフォン（Android、 iOS）などを用いても文字化が可能である。以上のことから、本システムにおいては UD トークの使用が最良であると判断した。

†筑波技術大学　Tsukuba University of Technology

2.2 話し言葉から読み言葉への要約

話し言葉から読み言葉への変換に関しては、Google が 2020 年に発表した機械学習モデル T5(Text-to-Text Transfer Transformer)[2]をベースに 101 か国の言語に対応するべく事前学習された mT5(multilingual T5)を用いる[3]。本モデルは事前学習モデルとして Wikipedia などの大量の情報を学習したのちに、転移学習をすることで高い精度を実現したモデルである。本研究では Google にて公開されている事前学習済みモデルを使用し、ファインチューニングとして盲ろう者向け要約筆記のデータを学習させることで、話し言葉から読み言葉への変換を試みる。

2.3 要約した読み言葉の出力

本研究では出力の一例として、“LINE Messaging API”を用いて出力を行う。しかし、障害者、特に盲ろう者の場合、使用しやすい媒体は大きく異なるため、出力する媒体を一意に定めず、使用者に合わせチューニングすることがユーザビリティ向上の観点において重要であると考える。

3. データセットについて

データセットの原文は「日本語話し言葉コーパス(Corpus of Spontaneous Japanese：CSJ)」を使用する。本データセットをもとに本学にて盲ろう者などへの要約筆記を行っている PCY298 が発行しているマニュアル[4]に則り要約文の作成を行う。つまり、データセットの形式は Input を“原文（話し言葉）”、Target を“読み言葉”としたものを 2,000 件程度作成し、学習データ：テストデータ：検証データとして 8:1:1 に分け、5,000Step の学習を行う。また、1,000Step ごとに checkpoint を設定し重みを保存する。

4. 提案モデル

作成したファインチューニング済みモデルをシステムの要約エンジンとして構築する。UD トークから情報を受け取る際は、動的構造の web ページを読み取れる python import である“selenium”を使用する。次に mT5 を使用するのだが、mT5 のシステム構造上モデルロードと Decode が同一セッションで行われているため、1 文の要約に 1 分程度かかってしまう。そこで、ファインチューニング済みモデルを PyTorch にコンパイルし Simple-Transformers にマウントする手法を用いる。本手法によって、1 分かかっていた Decode 時間を 5~10 秒まで短縮することができる。

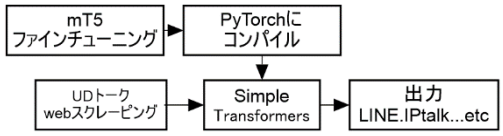

図 2　提案モデル

5. 評価

盲ろう者向け要約筆記にてファインチューニングしたモデル（以下ファインチューニング済みモデル）を評価するために、テキスト要約の評価指標である ROUGE-N、ROUGE-L を用いる[5]。以下に Recall の式を示す。

$$ROUGE_N = \frac{\sum_{S \in refs} \sum_{gram_n \in S} count_{match}(gram_n)}{\sum_{S \in refs} \sum_{gram_n \in S} count(gram_n)}$$

$$ROUGE_L = \frac{LSC(summary_{words}, refs_{words})}{refs_{words}}$$

ROUGE-N は N-gram 単位で一致度を算出する指標である。ROUGE-1 の場合は uni-gram(1 単語)、ROUGE-2 の場合は bi-gram(2 単語)での一致度を表すものである。ROUGE-L は一致する最大シーケンス (Longest Common Subsequence：LCS) の割合でスコアを算出する指標である。

5.1 事前学習済みモデルごとの Loss 値とスコア

mT5 には複数の事前学習済みモデルが存在している。パラメータが小さい順から“small”，“base”，“large”，“xl”，“xxl”であり、パラメータ数は順に“3 億”，“5.8 億”，“12 億”，“37 億”，“130 億”となっている。本研究では、同一の学習データにて small~xl までの事前学習済みモデルのファインチューニングを行い、どのモデルが最適かを探る（xxl に関しては、PC スペックの問題でファインチューニングができなかったため除外した）。

表 1　Loss 値最小の際の Step

Model	Step	Loss
small	4604	1.73E-04
base	4865	6.04E-05
large	4637	2.20E-05
★ xl	4842	**1.05E-05**

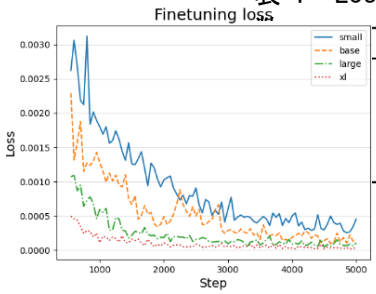

図 3　ファインチューニングの際の Loss 値

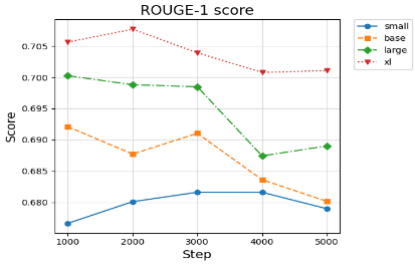

図 4　Step ごとの ROUGE-1 スコア

表 2　ROUGE スコア最大値と Step 数

Model	Step	Loss	ROUGE-1	ROUGE-2	ROUGE-L
small	3000	0.00077	0.68161	0.55956	0.67935
base	1000	0.00115	0.69210	0.56457	0.69084
large	1000	0.00055	**0.70026**	0.58576	0.69900
★ xl	1000	0.00020	**0.70770**	**0.59493**	**0.70612**

表 3　checkpoint ごとの Loss 値

Model	1000 step	2000 step	3000 step	4000 step	5000 step
small	0.0016915	0.00106	0.00077	0.00054	0.00038
base	0.0011513	0.00037	0.00030	0.00021	0.00011
large	0.0005525	0.00025	0.00013	0.00010	**0.00006**
★ xl	0.0001987	0.00009	**0.00005**	**0.00002**	**0.00005**

5.2 考察

5,000Step の学習における Loss 値最小は、xl の 4,842Step 目の 1.05E-5 であった。また、ROUGE の各指標の最大スコアは 1,000Step 目の xl であり ROUGE-1 においては 0.707 と非常に高いスコアとなった。つまり、ROUGE のスコアはパラメータ数に大きく依存している。しかし、large と xl のスコア差は 0.01~0.001 と非常に小さく、パラメータ数の増大と計算コストを考えた際に効率が良い方は large であると考察する。また、small と xl でのスコアの差は、ROUGE-2 において 0.035 に上るが、即時性が求められる要約筆記において、mT5 を処理するスペックを充分に用意できない環境ならば、small の事前学習済みモデルを使用し、レスポンス速度を優先するという選択も有用であると考える。

図 4 から、4,000Step 以降 ROUGE のスコアが低下傾向を示した。表 3 から、3,000~4,000Step にかけて Loss 値が総じて減少しているため、上記現象は過学習を起こしていると推察する。

日本語においての ROUGE は句読点によっても、スコアが大きく変動するため、0.6~0.7 というスコアは人が要約した文章に非常に近いと判断できる。

6. おわりに

本研究では、盲ろう者向けの自動要約筆記システムを提案した。今回の実験では、データセットの少なさによる過学習と思われる現象が観測されたが、ROUGE を使用した評価では、高スコアとなったことから、人の要約に近い出力の可能性が期待できる。また、要約筆記に重要な発話者の判断を現状手動で行っているため、MFCC などを用いて発話者を識別することで、更なるユーザビリティ向上につながると考えている。

本研究にて提案したシステムは、他の要約筆記の場面においても応用・転用できる。例としては、会議の議事録作成の際などに本システムを活用することで作成者の負担を大幅に軽減することが可能であると考える。

謝辞

本研究では UD トーク開発者である青木秀仁氏からアプリケーションの提供を受けた。ここに深く謝意を表する。

参考文献

[1] “東京盲ろう者友の会” http://www.tokyo-db.or.jp/?page_id=106 (参照 2021-6-16)
[2] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena,Yanqi Zhou,Wei Li, Peter J. Liu, ”Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer” (2020)
[3] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua,Colin Raffel,"mT5: A massively multilingual pre-trained text-to-text transformer” (2020)
[4] 特定非営利活動法人 PCY298, “PC 通訳基礎知識 2020 年版”
[5] Chin-Yew Lin,“ROUGE: A Package for Automatic Evaluation of Summaries”(2004)