

# 地方創生を指向した LINE BOT を用いた文字起こし業務 効率化の試み

辻本天翔<sup>1</sup> 小田部荘司<sup>1</sup> 兵頭悠生<sup>2</sup> 宮里賢史<sup>3</sup> 星野美緒<sup>3</sup>

**概要**：地方における事務作業等を IT 技術で支援することを目的に、既存の文字認識アプリ「文字起こしぱりぐっどくん」の派生開発として「切り取って文字起こしぱりぐっどくん」を制作した。本アプリは、コミュニケーションアプリ「LINE」上の BOT として稼働している。これに画像を送信すると、その画像内の全ての文字を認識してテキストとして返信する。また、送信前に任意の領域を線で囲むことで、領域内の文字のみを認識させることができる。ユーザーは、特定のメッセージを送ることで切り取る線の色を変更することが可能である。

**キーワード**：OCR, LINE, チャットボット

## Attempts to Improve Transcription Work for Regional Revitalization using LINE BOT

TAKATO TSUJIMOTO<sup>†1</sup> EDMUND SOJI OTABE<sup>†1</sup>  
YUSEI HYODO<sup>†2</sup> TAKAFUMI MIYASATO<sup>†3</sup> MIO HOSHINO<sup>†3</sup>

### 1. はじめに

近年の高齢化社会における地方では、様々な業務について人手不足が懸念されている。このうち、文字を手打ちでテキストに起こすような単純な事務作業については、現在の OCR 技術であれば容易に任せられることができるため、機械が人間に取って代わるべきである。しかしながら、非技術者にとっては導入が難しい、もしくは技術の存在が知られておらず、有効活用されていないのが現状である。そこで、OCR 技術をより簡単に、また、より多くの人に利用させるべく開発されたのが、アプリ「文字起こしぱりぐっどくん」 [1] である。本アプリは、利用者 8900 万人を越えるコミュニケーションアプリ「LINE」 [2] 上のチャットボットとして稼働しており、多くの人が簡単に使用できるという要件を十分に満たしている。友だち登録を行い、チャットボット宛てに画像を送信することで、本アプリは画像内の文字を認識し、テキストデータとして返信する。本稿執筆時点で友だち登録者数は 21 万人となっており、その高い需要がうかがえる。

今回の制作物は、「文字起こしぱりぐっどくん」のユーザービリティをさらに高めるべく機能を追加したアプリ「切り取って文字起こしぱりぐっどくん」である。

従来のアプリは性質上、使用者の意図していない部分まで認識してしまうことがあった。例えば、書類をテキスト化する際に、隣に置いていたキーボードまで検知して余計にアルファベットをテキスト化してしまうといった事が起

こりうる (図1)。

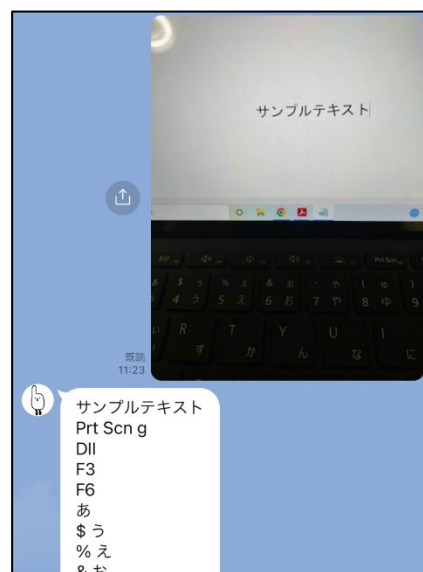


図 1 文字起こしぱりぐっどくんチャット画面

今回は、このような問題を解決すべく、範囲選択機能を追加した。

「切り取って文字起こしぱりぐっどくん」宛てに画像を送信する際、送信の直前に任意の領域を線で囲むことで、線に囲まれた閉領域内のみがテキスト化される (図 2)。このときユーザーは、線の描画には「LINE」アプリに備え付けの落書き機能を使用することとする。落書き機能で用いられる線の色は決まっているため、この RGB 値を事前に

1 九州工業大学  
Kyusyu Institute of Technology  
2 株式会社 YouLib  
YouLib, Inc.

3 株式会社西海クリエイティブカンパニー  
Saikai Creative Company.

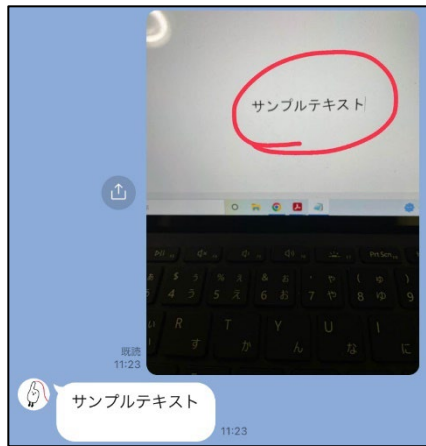


図 2 切り取って文字起こしぱりぐっどくんチャット画面

記録しておき、2 値化処理に利用することで誤りのないトリミングを実現する。

この方法では、元来のトリミング機能とは異なり、長方形以外の領域、また、複数の領域に対応できるのも利点である。

以上の機能によって、より一層文字起こし業務が効率化され、また、より多くの人に利用されることを目指す。

## 2. 開発概要

開発には以下の環境、サービスを用いた。

- LINE 公式アカウント [3]
- Google App Engine [4]
- Cloud Vision API [5]
- Canvas API [6]

### 2.1 アーキテクチャ

アプリのアーキテクチャを図 3 に示す。

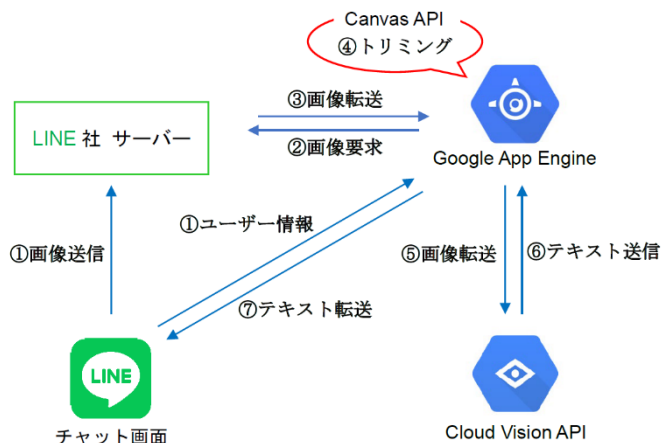


図 3 アプリのアーキテクチャ

- ① 公式アカウント「切り取って文字起こしぱりぐっどくん」宛てにユーザーが画像を送信すると、そのユーザー情報が GAE に送信される。画像送信の直前に加工画面が現れるため、ここでユーザーは切り取りたい部分を特定の色の線で囲む。画像データは LINE 社サーバーに送信される。
- ② GAE から LINE 社サーバーに画像データを要求する。①で得られたユーザー情報から参照している。
- ③ LINE 社サーバーから GAE に画像データが送信される。
- ④ ユーザーが線を描画している場合は画像の切り取り処理を行う。あらかじめ記録しておいた RGB 値によって 2 値化を行い、線の外側を除去することでトリミングを実現する。
- ⑤ 切り取った画像を Cloud Vision API（画像認識 AI）に送信する。
- ⑥ 画像認識によって文字起こしされたテキストデータが GAE に返される。
- ⑦ 得られたテキストデータを、公式アカウントのメッセージとして LINE 上に返信する。

### 2.2 画像の切り取り処理

前節の④における切り取り処理について、図 4 に示す。



図 4 切り取り処理の概要

- ① 特定の RGB 値付近をしきい値として、2 値化する。
- ② 線の外側をバケツ塗りつぶし (Flood Fill) する。
- ③ 塗りつぶされた領域を、元画像から除去する。

### 2.3 あいさつメッセージ

本制作物は、非技術者であっても容易に利用できることを目標としているため、より使いやすくなるようにシステム面以外でも工夫を行った。具体的には、「友だち追加」時に定型文を送信することができる「あいさつメッセージ」機能を用いた。

あいさつメッセージには、

- チャットボットの使い方を簡単に説明するテキストメッセージ
- チャットボットを実際に使っている様子を写真でより詳しく説明するカードタイプメッセージ

の 2 点を設定した (図 5)。

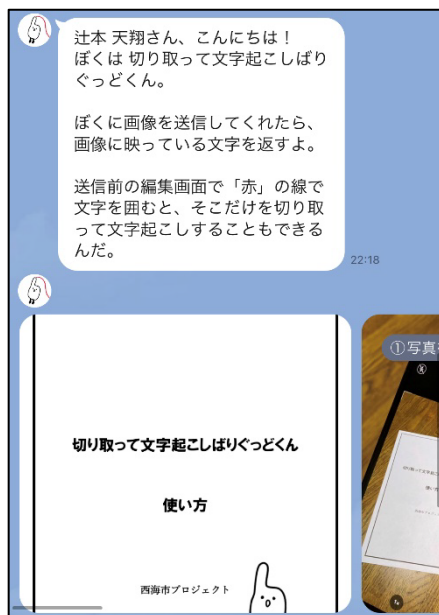


図 5 あいさつメッセージ

### 2.4 色の切り替え

当初、本アプリは特定の RGB 値の「赤」でのみトリミングすることを想定していたが、稀に撮影した写真自体に RGB 値の近い「赤」が含まれていた場合、誤作動を起こすことがあった。そこで、ユーザーからのメッセージに応じてトリミングを行う色を変更できることとした。図 6 は、切り取る色を「青」に変更して、実際にトリミングされている様子である。

### 3. まとめ

本制作では、文字起こし業務を効率化すべく、また、多

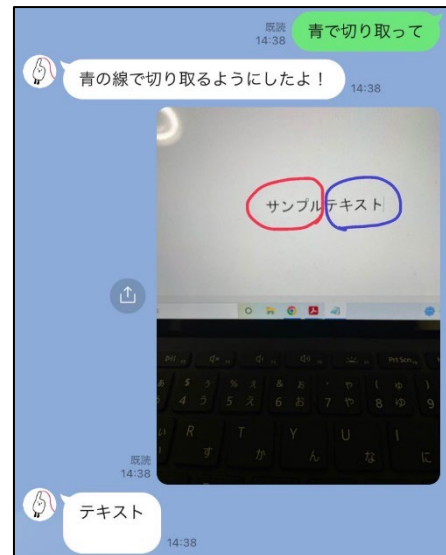


図 6 切り取る線の色変更

くの人に利用させるべく、既存のアプリについて派生開発を行った。画像の必要な領域のみを選択する機能を実現することができ、かつ、非技術者であっても感覚的に分かりやすい手法を取ることができた。

今回、技術的に難しいことはしていないが、非技術者の方々に対して、このように技術を利用するための導線を引き出すことができた、というのが大きな収穫だと思っている。実際に、「文字起こしぱりぐっどくん」は市役所の業務で利用され、役に立ったとの声を聞くことができています。また、チャットボットに対して感謝を述べたというユーザーもあり、ローカルメディアのキャラクター「ぱりぐっどくん」を採用したことも功を奏したと言えるだろう。今後は、本アプリの利点を保ちながらも新しい要素を加えていくことでユーザビリティを高めていく。ひいては文字起こし業務以外の分野についても展開が期待される。

**謝辞** 本制作を遂行するにあたり、協力していただいた西海クリエイティブカンパニーをはじめとする皆様に、厚く感謝申し上げます。

### 参考文献

- [1] “文字起こしぱりぐっどくん” . <https://varygoodkun.net/app/line-mojiokoshi/>, (参照 2022-05-18).
- [2] “LINE” . <https://line.me/ja/>, (参照 2022-05-18).
- [3] “LINE 公式アカウント” . <https://www.linebiz.com/jp/service/line-official-account/>, (参照 2022-05-18).
- [4] “App Engine アプリケーション プラットフォーム” . <https://cloud.google.com/appengine/>, (参照 2022-05-18).
- [5] “Vision AI” . <https://cloud.google.com/vision/>, (参照 2022-05-18).
- [6] “キャンバス API” . [https://developer.mozilla.org/ja/docs/Web/API/Canvas\\_API](https://developer.mozilla.org/ja/docs/Web/API/Canvas_API), (参照 2022-05-18).