

2021 年度

卒業論文

ビジネス・社会の問題解決に向けた  
AI によるシステム開発

橋口 駿亮

(学籍番号：20224004)

2022 年 2 月 14 日

九州工業大学 情報工学部  
物理情報工学科 電子物理工学コース

指導教員：小田部 荘司 教授

# 目次

第 1 章	はじめに	1
第 2 章	LINE を使ったアプリケーション開発	3
2.1	序論 . . . . .	3
2.2	理論・サービス . . . . .	4
2.3	すごい外国語ペラペラばりぐっどくん . . . . .	8
2.4	ドキュメント翻訳ばりぐっどくん . . . . .	12
第 3 章	MLOps 導入に向けた LSTM による翌日電力需要予測	17
3.1	序論 . . . . .	17
3.2	理論 . . . . .	18
3.3	前処理 . . . . .	22
3.4	従来手法 . . . . .	23
3.5	提案手法 . . . . .	24
3.6	学習条件 . . . . .	25
3.7	予測結果 . . . . .	27
第 4 章	結論	32
	謝辞	33
	研究業績	34
	参考文献	35



# 目次

1.1	すごい外国語ペラペラばりぐっどくんのアイコン . . . . .	1
1.2	ドキュメント翻訳ばりぐっどくんのアイコン . . . . .	1
1.3	電力需要予測のイメージ図 . . . . .	2
2.1	Oauth2.0 によるユーザのデータ取得のフロー . . . . .	5
2.2	認可コードフロー . . . . .	6
2.3	日本語の長文による翻訳の仕組み (1) . . . . .	8
2.4	日本語の長文による翻訳の仕組み (2) . . . . .	9
2.5	日本語の長文による翻訳の仕組み (3) . . . . .	9
2.6	日本語の長文による翻訳の仕組み (4) . . . . .	10
2.7	すごい外国語ペラペラばりぐっどくんのアーキテクチャ . . . . .	10
2.8	すごい外国語ペラペラばりぐっどくんの出力結果 . . . . .	11
2.9	すごい外国語ペラペラばりぐっどくんの句点付与失敗例とその 対策 . . . . .	12
2.10	ドキュメント翻訳ばりぐっどくんのアーキテクチャ . . . . .	13
2.11	ドキュメント翻訳ばりぐっどくんの Google Drive 設定の画面遷 移図 . . . . .	15
2.12	ドキュメント翻訳ばりぐっどくんの翻訳の画面遷移図 . . . . .	16
3.1	機械学習システムに存在する技術的負債 . . . . .	18
3.2	LSTM のモデル図 . . . . .	19

3.3	従来手法のモデル図：従来手法では翌日 0 時～7 時までの電力データの平均を予測するモデル，翌日の電力データの主成分得点を予測するモデル，翌日 24 時間の電力データの予測をするモデルの 3 つのモデルを用いて翌日の電力需要予測を行っている。	25
3.4	提案手法のモデル図	25
3.5	東京電力の電力データを主成分分析したときの寄与度と累計寄与度：東京電力のデータを主成分分析すると，第 4 主成分得点までで累計寄与度が 100% 近傍まであることが確認できる。	27
3.6	提案手法による翌日の電力需要予測の結果：origin は元の電力データ，pred. は LSTM で電力データを予測したときの予測値を表している。2018 年 6 月 10 日～6 月 16 日の 1 週間分の翌日の電力需要予測を行った結果，正しく予測できていることが確認できる。	28
3.7	従来手法による翌日の電力需要予測の結果	29
3.8	提案手法のアンサンブル法による各モデルの損失の変化：model__番号はアンサンブル学習によるモデルを表しており，合計で 10 個のモデルを学習させている。左図は訓練データによる RMSE のグラフ，右図は評価データによる RMSE のグラフである。両者とも損失の収束に向かっていることから，学習ができていることが確認できる。	29

- 3.9 エポック数を対数にしたときの従来手法と提案手法の学習損失：左図が訓練データによる損失の変化，右図が評価データによる損失の変化を表してる．青線が従来手法，赤線が提案手法を表している．また，横軸のエポック数は対数を取っている．グラフから，提案手法の方が学習の収束が早いことが確認できる．また，エポック数が増えていくにつれて，右図は左図に比べてノイズが混じっているようなグラフとなっている（RMSE がスムーズに減少していない）．これは，評価データの RMSE において，徐々に最適解につれて，局所解に陥っているのではないかと考えられる． . . . . . 30
- 3.10 電力データの次元数による RMSE の変化：電力データの次元数を減らしても RMSE の変化があまり見られないのが確認できる．この結果から，元の 24 次元の電力データを次元圧縮しても翌日の電力需要予測ができています．これは図 3.5 より，第 4 主成分までで累計寄与度が 99% 以上であったため，次元数を減らしても翌日の電力需要予測ができたと考えられる． . . . . . 31

# 表目次

2.1	「すももももももものうち」を形態素解析したときの結果 . . .	4
3.1	学習条件 . . . . .	26
3.2	東京電力の電力需要予測における RMSE の結果と学習時間 . . .	28

# 第 1 章

## はじめに

AI 技術の発展により，様々なアプリケーションやデバイスなどに AI が搭載されている．特に，私たちが普段使用しているスマートフォンには音声認識が使われており，日常生活に広く使われている．本研究では，そういった日常生活で利用できる LINE アプリケーション開発と，社会・環境問題に適応するための AI 開発に取り組んだ．アプリケーション開発では LINE を使って，音声翻訳とドキュメント翻訳の 2 つの LINE アプリケーションを開発した．社会・環境問題では，CO2 による地球温暖化の防止に向けて，発電所の発電量を調整することを目的とした電力需要予測のモデル開発を行った．また，この電力需要予測モデルは従来手法をベースに開発を行ったが，今回提案手法として，クラウドサービスなどに ML システムとして導入することを踏まえて，様々な工夫を施した．



図 1.1 すごい外国語ペラペラばりぐっどくんのアイコン

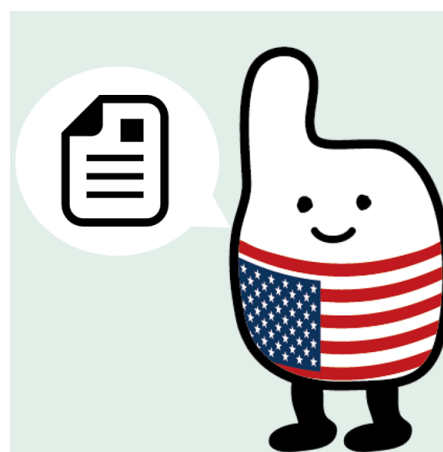
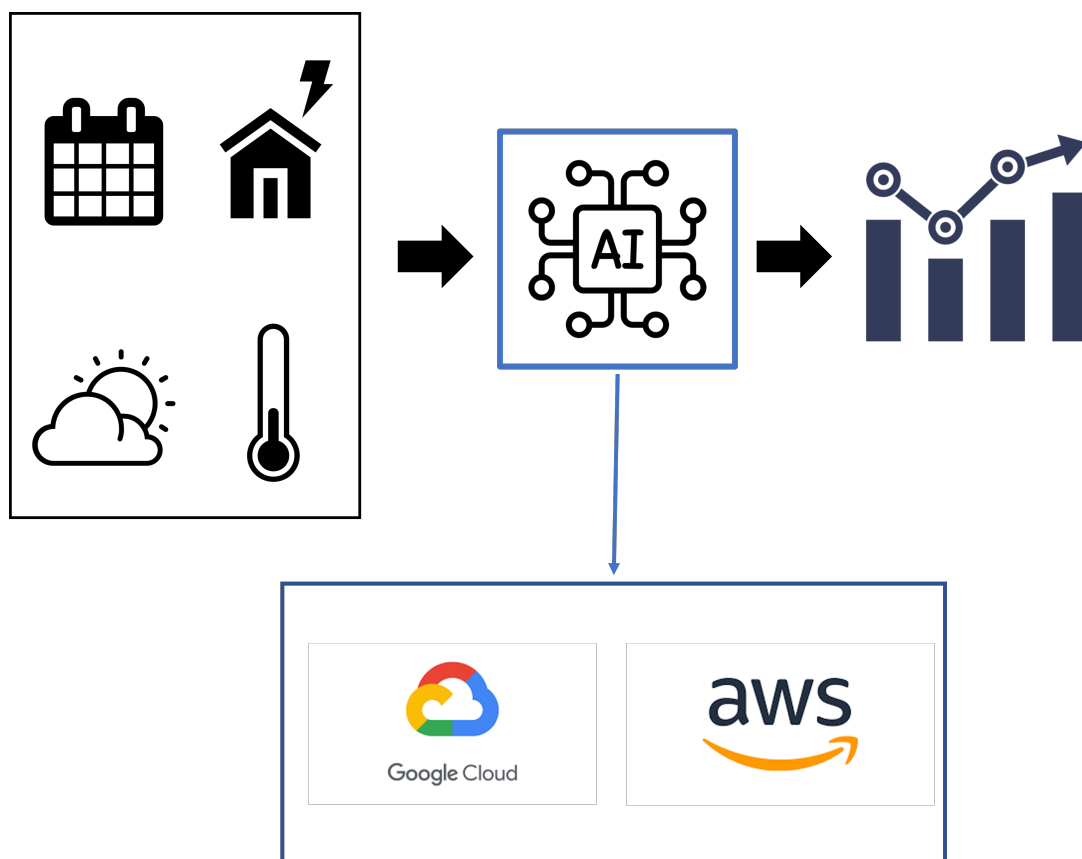


図 1.2 ドキュメント翻訳ばりぐっどくんのアイコン





**電力需要予測モデルをGCPやAWSといった  
クラウドサービスで実装するためのモデル開発を行う**

図 1.3 電力需要予測のイメージ図

## 第 2 章

# LINE を使ったアプリケーション開発

### 2.1 序論

近年人工知能の技術向上が目覚ましく、その例として音声翻訳、ドキュメント翻訳がある。音声翻訳は、異なる言語を話す人々との会話を可能にする技術である。この技術により、旅行やビジネスなどで海外の人々と円滑に物事を進めることができるようになった。しかし、翻訳の精度が向上しても、(1) 音声翻訳の使用の複雑さや (2) 長文に対する音声翻訳の曖昧さといったデメリットが存在する。本研究では、この 2 つのデメリットに着目し、1 つ目のデメリットに関しては、日頃私たちが使用している LINE を用いて、日本語と英語の音声翻訳の実装を行った。また、2 つ目のデメリットに関しては、日本語の長文に対応できるように、無音部分の判定および形態素解析を行い、句点を自動で付与することによって、翻訳の曖昧さの改善を行った。ドキュメント翻訳は、PDF や Word などのドキュメントファイルに書かれている内容を別の言語に翻訳する技術である。日本語から他言語に翻訳したり、日本語に資料を翻訳したりする場合、Google 翻訳や DeepL 翻訳などを使って、文字を翻訳するのが主流になってきている。しかし、同じような作業を何回もするには手間がかかってしまう。近年ではそのような手間を省くドキュメント翻訳がサービスとして登場し始めている。本研究では、昨年 5 月に発表された Google Translate API のドキュメント翻訳を LINE Bot 上で実装した。また、翻訳されたドキュメントファイルを Google Drive 自動で保存するために、Oauth2.0 の機能を使用した。

## 2.2 理論・サービス

### 2.2.1 形態素解析

形態素解析は、文章に含まれる語彙を形態素ごとに分割する手法である。形態素とは、何らかの意味を持つ最小限の文字の集まりである。例えば、「すもももももものうち」は以下の表のように形態素を分割できる。

表 2.1 「すもももももものうち」を形態素解析したときの結果

文字	品詞
すもも	名詞, 一般
も	助詞, 係助詞
もも	名詞, 一般
も	助詞, 係助詞
もも	名詞, 一般
の	助詞, 連体化
うち	名詞, 非自立, 副詞可能

今回、形態素解析は「すごい外国語ペラペラばりぐっどくん」で使用しているが、形態素解析器は高速に形態素解析ができる MeCab を採用した。

### 2.2.2 Oauth 2.0

Oauth2.0 はアプリケーションによる HTTP サービスへの限定的なアクセスを可能にする認可フレームワークで、技術文書 RFC6749 で定義されている。この限定的なアクセスを実現するために必要なのがアクセストークンである。アクセストークンはユーザごとに保持している固有のトークンである。具体的には、クライアントアプリケーションがアクセストークンを生成する認可サーバに対してアクセストークンを要求し、認可サーバはクライアントアプリケーションにアクセストークンを送信する。この流れを標準化したものが Oauth2.0 である。ユーザの情報を取得するまでの流れを、クライアントアプリケーションとユーザのデータを管理するリソースサーバを使って説明する (図 2.1)。まず、ユー

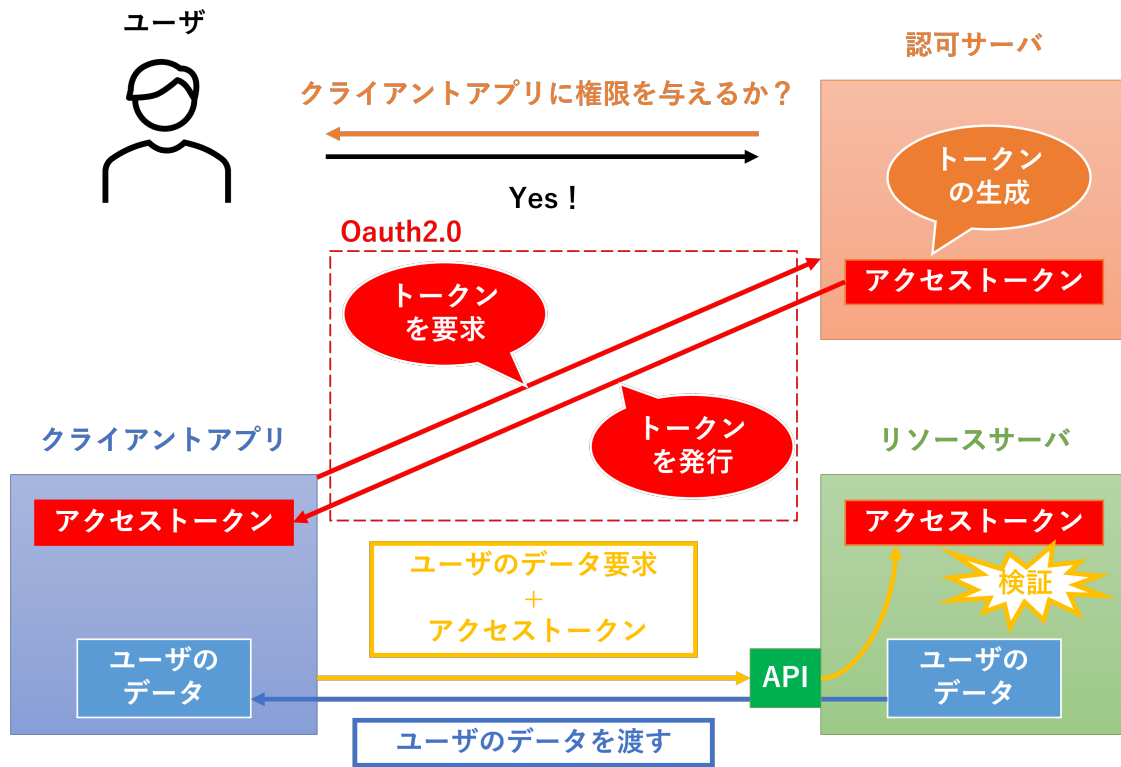


図 2.1 OAuth2.0 によるユーザのデータ取得のフロー

ザのデータをやり取りするために、リソースサーバに API (Application Program Interface) を設置する。次に、クライアントアプリケーションはリソースサーバに対してデータを要求すると同時に、予めクライアントアプリケーションに保持しているアクセストークンを提示する。そして、リソースサーバはリクエストに含まれているアクセストークンを取り出して、ユーザのデータを利用できるかどうか確認する。最後に、ユーザがデータを使用できる権限があることを確認した後に、ユーザのデータをクライアントアプリケーションに渡して処理が完了する。

### 2.2.3 認可コードフロー

前述では OAuth2.0 の紹介をしたが、実際にアクセストークンを取得するフローとしては「認可コードフロー」、「インプリシットフロー」、「リソースオーナー・パスワード・クレデンシャルズフロー」、「クライアント・クレデンシャルズフロー」の 4 つがある。その中でセキュリティ上推奨されているフローである認可コードフローを今回「ドキュメント翻訳ばりぐっどくん」に使用した。認可

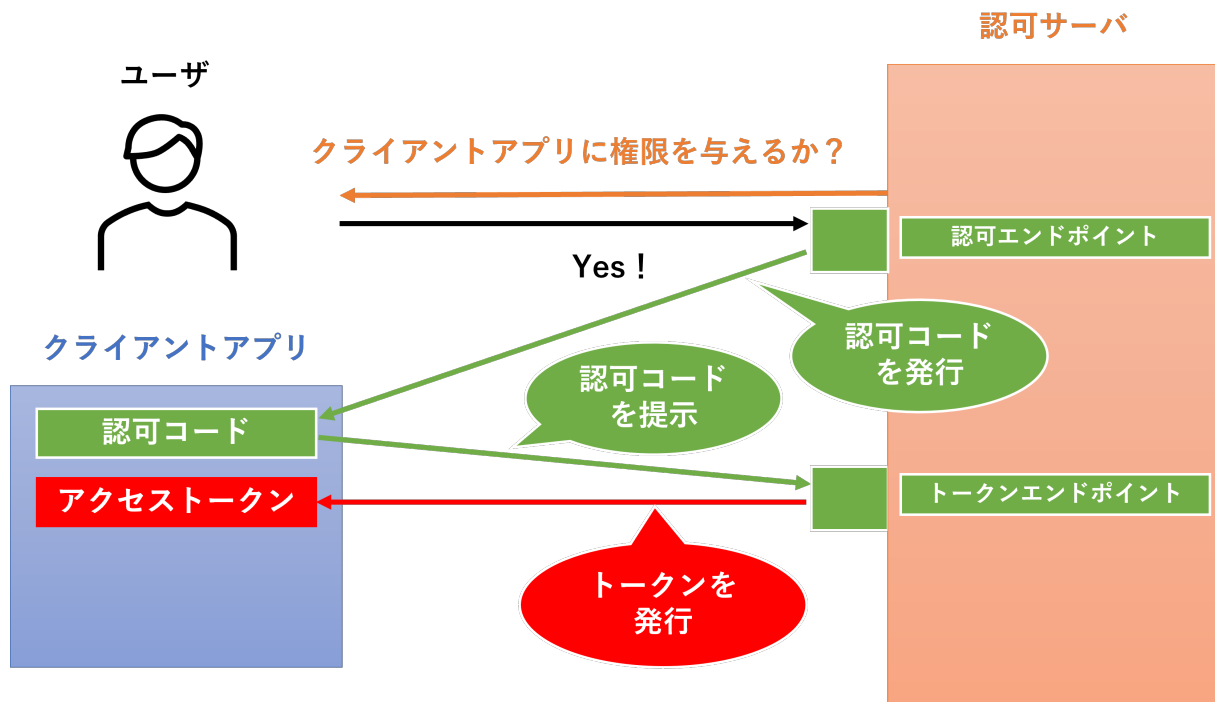


図 2.2 認可コードフロー

コードフローは RFC6749, 4.1 Authorization Code Grant で定義されているフローである。認可コードフローを図 2.2 に示す。認可サーバの認可エンドポイントにリクエストを投げ、応答として 1 時間の有効期限がある認可コードを受け取り、その認可コードを認可サーバのトークンエンドポイントでアクセストークンと交換するフローである。

#### 2.2.4 Google Cloud Platform(GCP)

Google が提供しているクラウドコンピューティングサービスである。Google Cloud を利用すれば、あらゆるクラウド、あらゆる環境でデータを活用してアプリを実行することができる。また、料金はアクセスの数に応じて支払われる従量課金制となっているため、チーム開発だけでなく、個人開発としても便利なプラットフォームである。この GCP 環境では、今回のアプリ開発で使用した Google Cloud Engine(GAE), Cloud Functions, Google Cloud Storage(GCS), Cloud Firestore などといったマイクロサービスが多く存在する。

## 2.2.5 Google App Engine(GAE)

Google App Engine(GAE)とは、大規模なウェブアプリケーションを開発ができるサーバーレスプラットフォームである。GAEを使うことでインフラの管理を気にせず、アプリの開発だけに集中することができる。また、プログラムを実行する上で必要なライブラリのバージョンに対する頑健性が強いのが特長である。

## 2.2.6 Cloud Functions

Cloud Functions は、GAEと同じサーバーレス環境で開発ができるプラットフォームである。GAEと同様に、インフラ管理を気にせずに開発ができるだけでなく、負荷に応じて自動でCPUのコア数を変更できたり、ロールごと・関数ごとに組込みのセキュリティをかけたりすることができる。しかし、ライブラリのバージョンに対する頑健性は弱く、バージョンの統一性が必要になる場合がある。

## 2.2.7 Google Cloud Storage(GCS)

Google Cloud Storage(GCS)は、クラウド上にデータを保存することができるストレージサービスである。保存できるデータ量の制限はなく、必要に応じてデータを取得することができる。

## 2.2.8 Cloud Firestore

Cloud Firestore は、FirebaseとGoogle Cloudからのモバイル、ウェブ、サーバ開発に対応したNoSQLクラウドデータベースである。Cloud Firestoreは、上層からコレクション、ドキュメント、コレクションという階層になっており、json形式でデータを格納することができる。

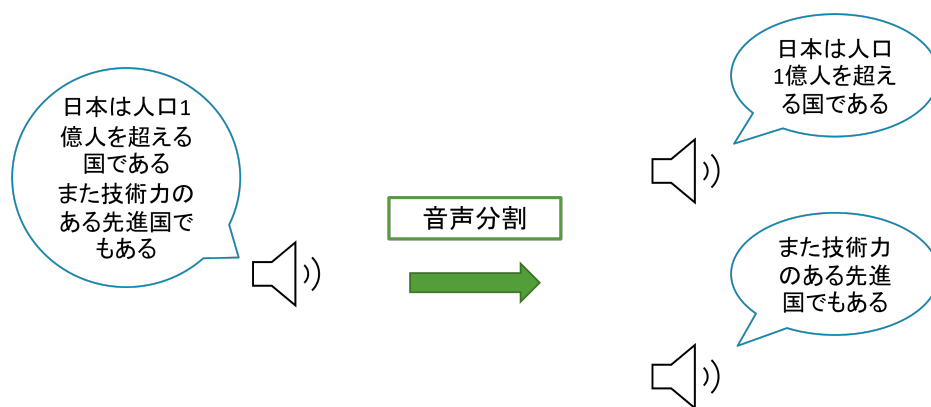
## 2.3 すごい外国語ペラペラばりぐっどくん

### 2.3.1 概要

すごい外国語ペラペラばりぐっどくんは過去に製作された「外国語ペラペラばりぐっどくん」の改良版である。外国語ペラペラばりぐっどくと異なるところは、音声による翻訳を追加し、また、日本語による長文にも対応できるような仕組みを形態素解析と音声分割で実現している点である。

### 2.3.2 日本語の長文による翻訳の仕組み

日本語の長文による翻訳の仕組みについて図を用いて説明する。まず、元の日本語の音声データから無音時間を判定して、音声の分割を行う。図 2.3 より、音声データの内容は 2 文で構成されているため、2 つの音声データに分割される。次に、分割された音声をそれぞれ文字起こしして、形態素解析を行い、形態素ごとの品詞を特定する（図 2.4）。そして、形態素解析を行った結果から、文章末尾の形態素の品詞が動詞、あるいは助動詞であれば文の最後に句点を付与する（図 2.5）。句点が付与された文は、文章として他の文と結合させる。最後に句点が付与された文章を英訳して、文字から音声を生成する（図 2.6）。



① 日本語の音声データ入力して、音声データから無音部分を判定し、音声を分割する

図 2.3 日本語の長文による翻訳の仕組み (1)

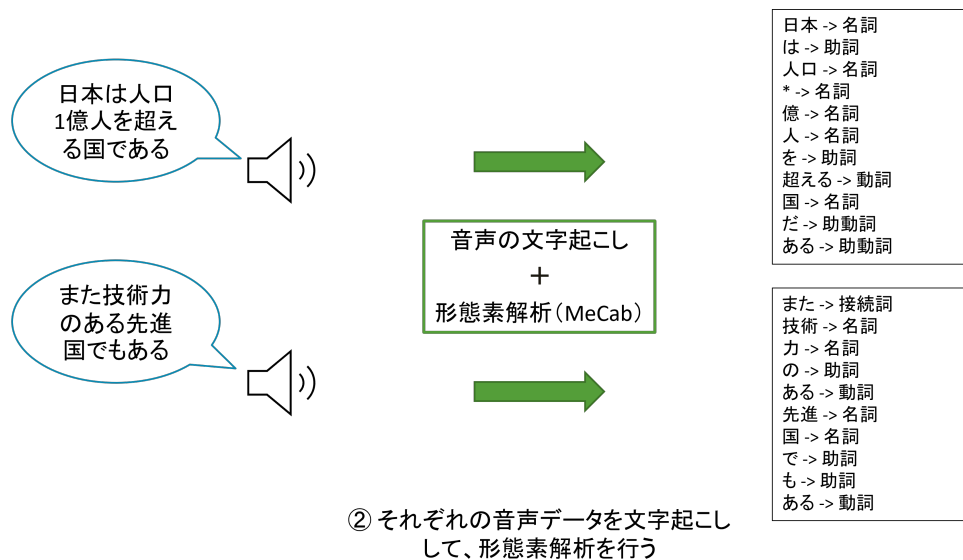


図 2.4 日本語の長文による翻訳の仕組み (2)

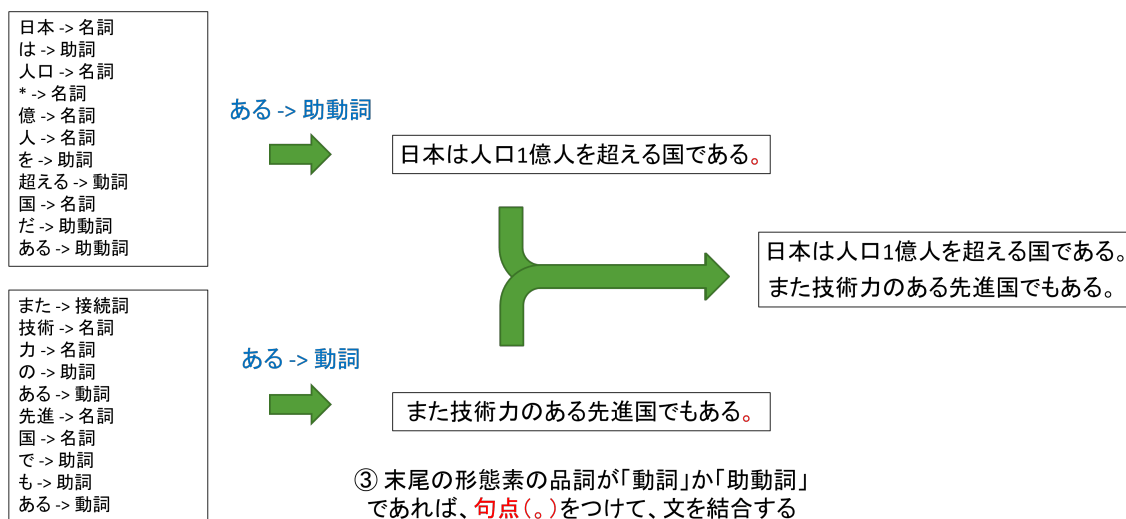
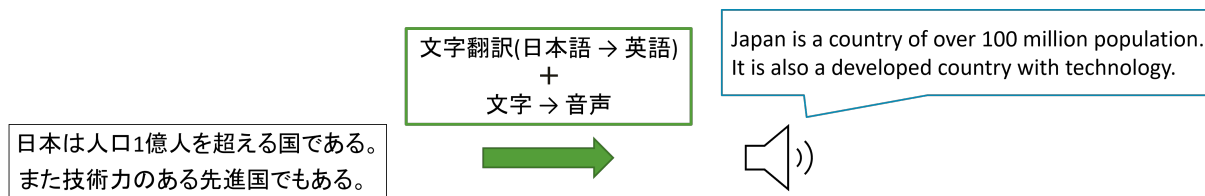


図 2.5 日本語の長文による翻訳の仕組み (3)

### 2.3.3 アーキテクチャ

開発環境は LINE の他に、音声翻訳の処理を行う GAE、翻訳された音声データを保存する GCS、翻訳したい言語を保存する Firestore の 3 つを使って、音声翻訳の実装を行った。LINE から翻訳結果を得るまでの流れを説明する。まず、LINE は GAE にユーザ情報を送信して、同時に LINE Server に音声データを送信する。次に、GAE は Firestore に予めユーザに設定した翻訳したい言語を取得し、LINE Server から音声データを取得する。そして、GAE 内で翻訳の処理を実





④ 文章翻訳して、翻訳した文章  
を音声データに変換

図 2.6 日本語の長文による翻訳の仕組み (4)

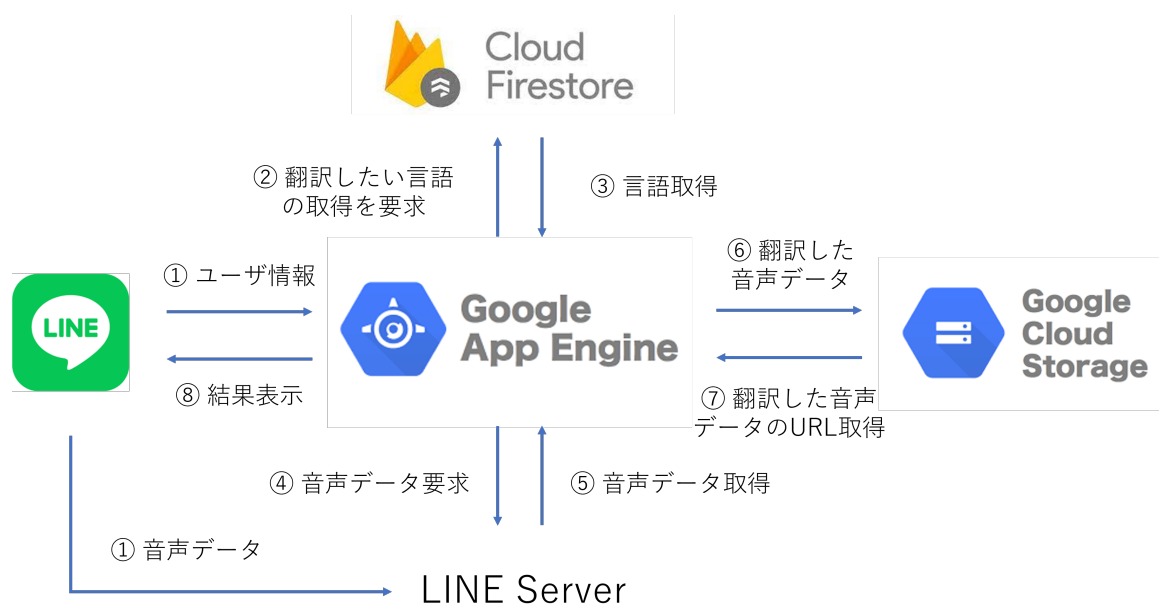


図 2.7 すごい外国語ペラペラばりぐっどくんのアーキテクチャ

行し、GCS に翻訳された音声データを保存する。最後に、翻訳された音声データの URL を取得し、LINE に結果を表示する。

### 2.3.4 アプリ画面

アプリの実行結果を図 2.8 に示す。「日本は人口一億人を超える国である また技術力のある先進国でもある」という原文の音声を入力すると、「日本は人口一億人を超える国である」と「また技術力のある先進国でもある」の末尾に句点が付与されていることが確認できる。



図 2.8 すごい外国語ペラペラばりぐっどくんの出力結果

### 2.3.5 句点付与失敗例

句点付与の失敗例を図 2.9 に示す。失敗例では、「日本は人口一億人を超える国である」の末尾に句点が付与されていない。これは、音声分割の基準となる無音時間よりも、実際に送信された音声の 2 つの文の間の無音時間の方が短かったため、音声分割ができず、句点付与がうまくいかないという結果になってしまった。この解消として、文字起こしの句点の付与についてユーザに提示して、文字起こしが妥当だったかどうかを判定してもらい、その結果に応じて、音声分割の基準となる無音時間をユーザごとに再設定する機能を付けている。しかし、この方法を使っても句点付与がうまくいかないことが多く、場合によっては文字起こした内容が変化することもある。また、この方法はユーザにとっても手間をかけさせてしまっているのが現状である。

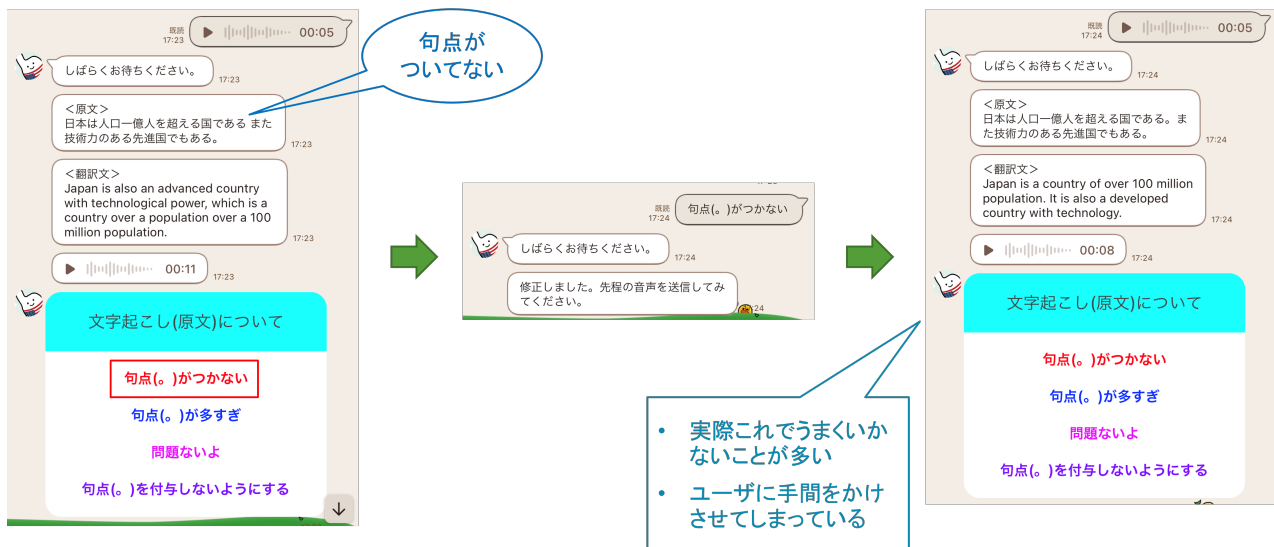


図 2.9 すごい外国語ペラペラばりぐっどくんの句点付与失敗例とその対策

## 2.4 ドキュメント翻訳ばりぐっどくん

### 2.4.1 概要

ドキュメント翻訳ばりぐっどくんは、LINE にドキュメントファイルを送信することで、ユーザが指定した言語に翻訳するアプリである。翻訳されたドキュメントファイルは、Google Drive に自動で保存することができる。

### 2.4.2 アーキテクチャ

ドキュメント翻訳ばりぐっどくんのアーキテクチャを図 2.10 に示す。ここでは、PDF ファイルの送信の場合について示している。ユーザは、翻訳したいドキュメントファイルを Google Drive から LINE に送信する。LINE 側は、Cloud Functions にドキュメントファイルのバイナリデータを送信する。そして、Cloud Functions はドキュメントファイルのバイナリデータと、ユーザがあらかじめ設定した翻訳したい言語を Firestore から取得して、Google Translate API のドキュメント翻訳を実行することで、翻訳されたドキュメントファイルを生成する。最後に、翻訳されたドキュメントファイルを Google Drive に送信して、LINE に Google Drive の URL を送信する。

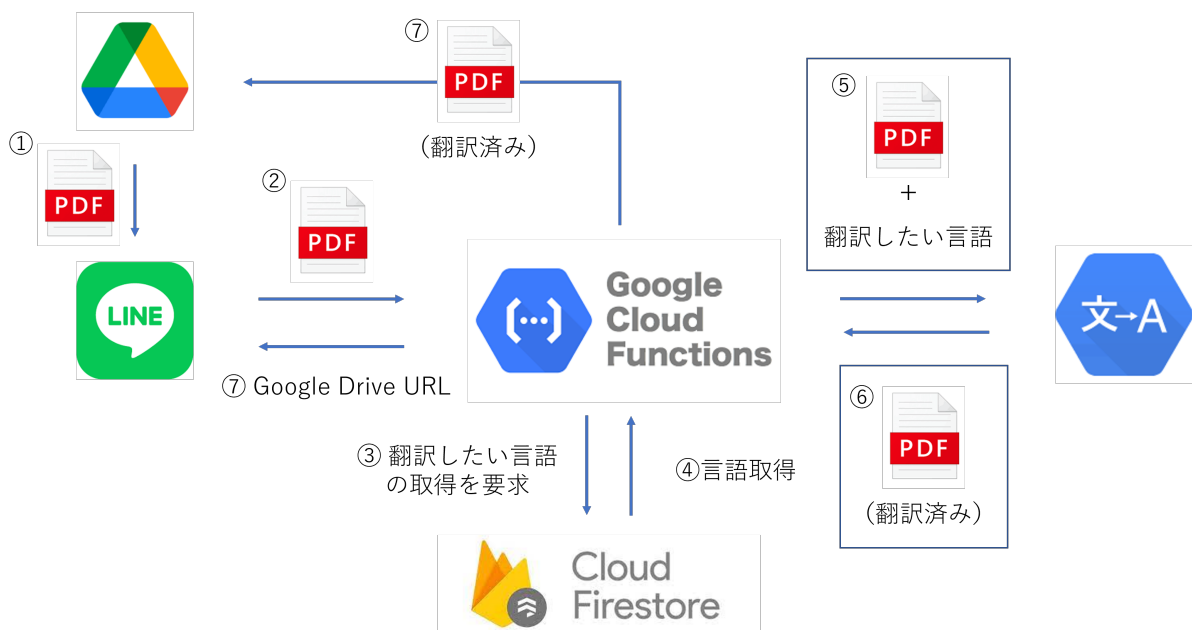


図 2.10 ドキュメント翻訳ばりぐっどくんのアーキテクチャ

### 2.4.3 アプリ遷移画面

アプリの画面遷移図を Google Drive の設定の場合（図 2.11）とドキュメントファイルの送信の場合（図 2.12）に分けて示す。

Google Drive 設定の画面遷移図から説明する。まず、LINE に「設定」とテキストメッセージを送る。すると、Google アカウント登録の URL が表示され、タッチすることで設定ページに遷移する。そして、Google アカウントを選択すると、Google Drive へのファイルのアップロードに関するアクセスを要求されるので、「continue」をタッチすると、Google Drive 設定に必要なコード（認可コード）が発行される。そして、これをコピーして LINE に送信することで Google Drive の設定が完了する。

ドキュメントファイル翻訳の画面遷移図について説明する。今回は厚労省の新型コロナウイルスに関する PDF ファイルを日本語から英語に翻訳する。まず、LINE のリッチメニューから、「英語に！」を選択すると、ドキュメントファイルの選択画面が表示される。今回は PDF ファイルなので、「PDF」をタッチすると、翻訳の設定が完了し、Google Drive のリンクが表示される。そして、Google Drive から翻訳したい PDF ファイルを選択して、ドキュメント翻訳ばりぐっどく

んに送信することで、翻訳が実行される。数十秒待つと、翻訳された PDF ファイルが Google Drive に保存されていることが確認できる。

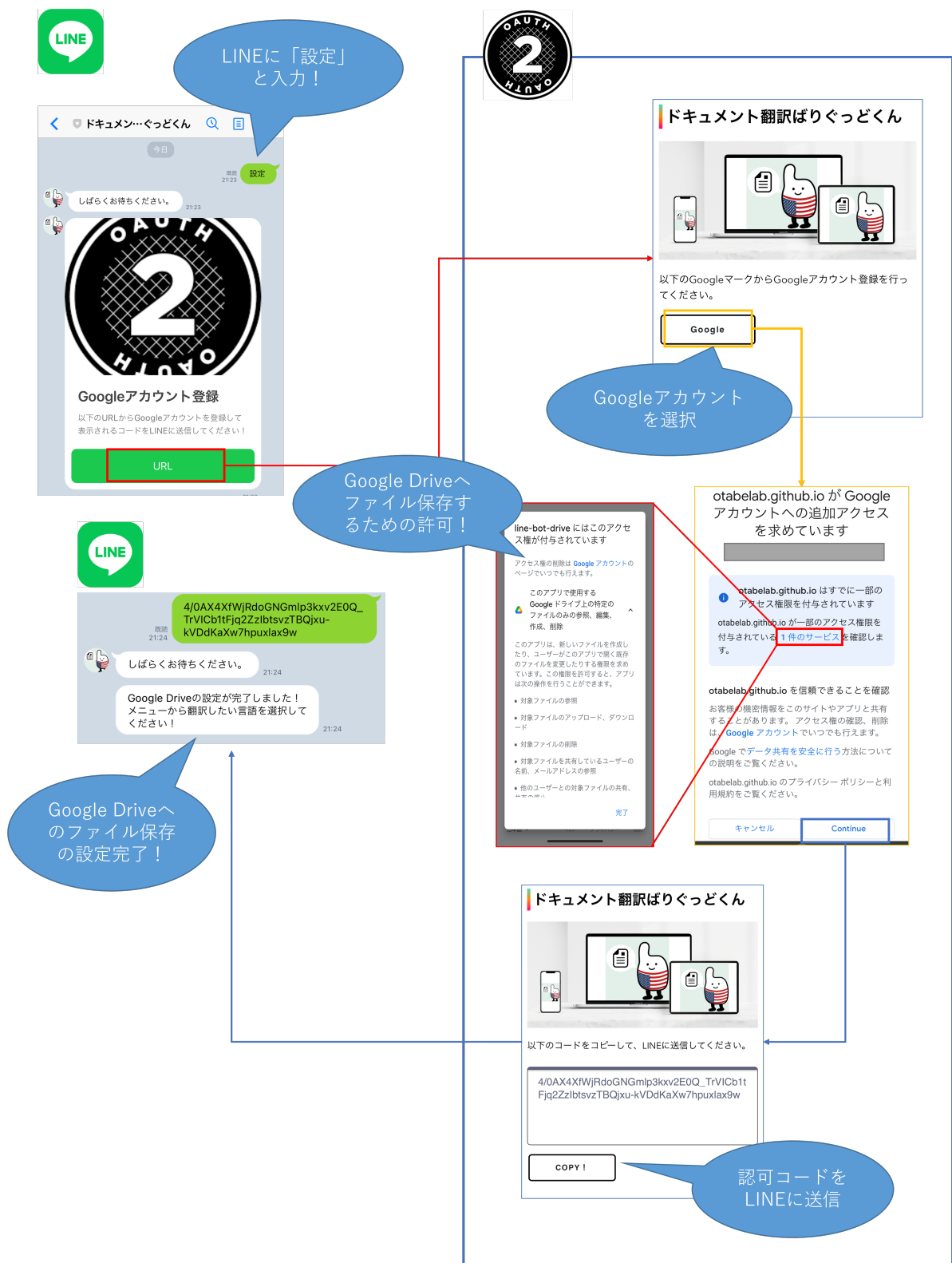


図 2.11 ドキュメント翻訳ばりぐっどくんの Google Drive 設定の画面遷移図

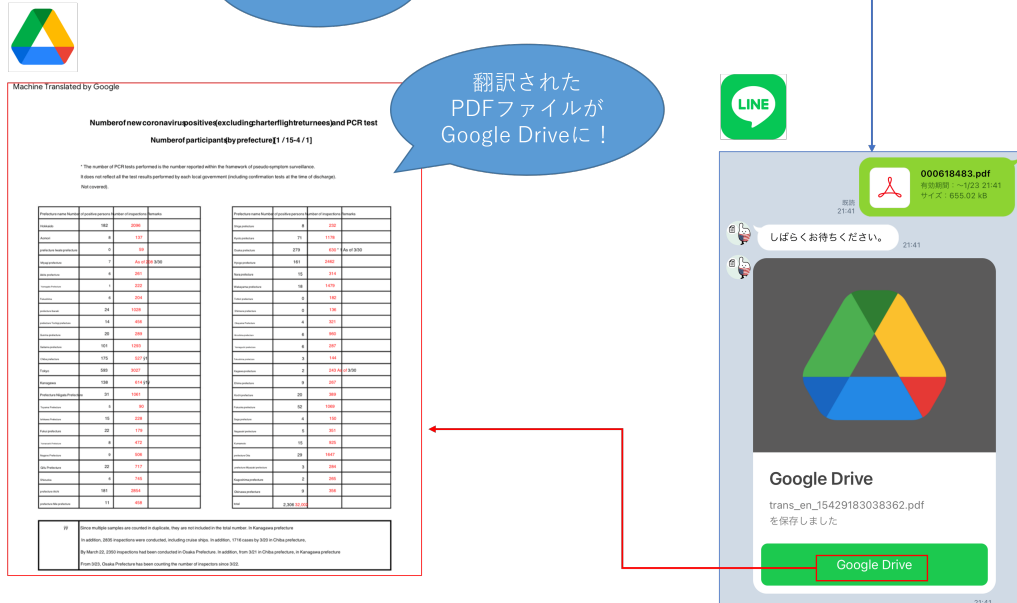
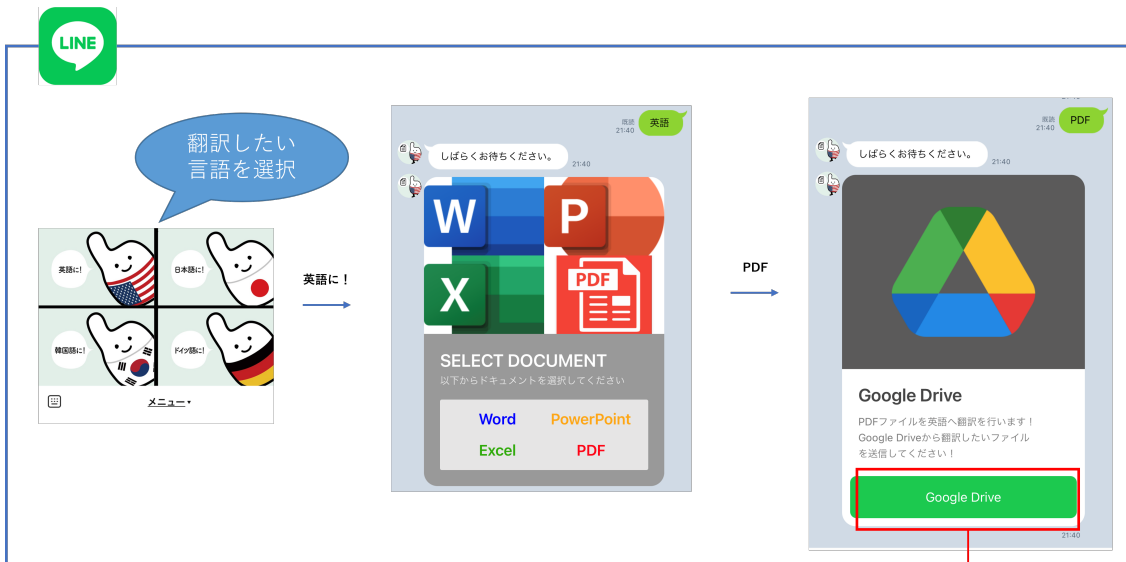


図 2.12 ドキュメント翻訳ばりぐっどくんの翻訳の画面遷移図

## 第 3 章

# MLOps 導入に向けた LSTM による翌日電力需要予測

### 3.1 序論

近年 AI 技術の発展により，AI を使ったシステム開発が進められている．その AI を使ったシステムの 1 つとして需要予測がある．需要予測とは，対象の商品やサービスについて，将来的な需要を過去のデータから予測する取り組みである．需要予測の歴史は 1990 年代頃から製造業を中心として SCM(supply chain management) の一環として導入が開始されており，現在では，小売業や流通業など，様々な分野で応用が広がっており，電力もその 1 つである．電力需要は IT 技術の進展やエアコンの普及などによって，全世界において顕著である．しかしそれに伴い，地球温暖化防止や CO2 排出削減など，国際的にも持続可能な環境づくりを目指す動きがあり，経済成長と関係する部分が増えてきている．この問題の解決に，発電量を調整することを目的として，電力データや天候などのデータを使って電力需要モデルを構築して電力データを予測する研究が進められている．しかし，電力需要モデルをビジネスや社会に適用するためのシステム作りをするためにも，電力需要モデルの複雑化や学習に対する時間・メモリなどのコスト増大は回避しなければならない．本研究では，電力需要モデルを機械学習システムとして導入することを踏まえて，学習に使用する入力データの次元を小さくし，かつ単純なモデルを使って電力需要予測を行った．



## 3.2 理論

### 3.2.1 MLOps

MLOps は機械学習システムを円滑に進めるための取り組みである。ML システムを構築するためには、図 3.1 に示すように機械学習モデルはほんの一部でしかなく（図 3.1 の「ML Code」）、データの特徴量の変化やモニタリング、データの収集など周辺のインフラが大半を示している [2]。しかし、MLOps を導入することによって、この機械学習システムに存在する隠れた技術的負債を削減することができる。MLOps は「DevOps」からの派生で、従来のシステムに適用していた DevOps を機械学習システムに拡張した概念だと言われている（MLOps = ML + Dev + Ops）。MLOps は、機械学習システムによる一連のタスクにおいて、様々な要因（学習コスト、特徴量モニタリングなど）を考慮して、ビジネスに適用することが求められる。

### 3.2.2 LSTM

LSTM は Long Short-Term Memory の略で、RNN（Recurrent Neural Network）の長期記憶を保持するのが難しいという問題点を改善した時系列モデルである。LSTM は RNN の一種だが、ゲートと呼ばれる仕組みを導入することで、過去の情報を忘れるか忘れないかを判断しながら、必要な情報だけを次の時刻に引き継ぐことができる。

LSTM のモデル図を図 3.2 に示す。 $\sigma$  はシグモイド関数、「 $\times$ 」は要素ごとの

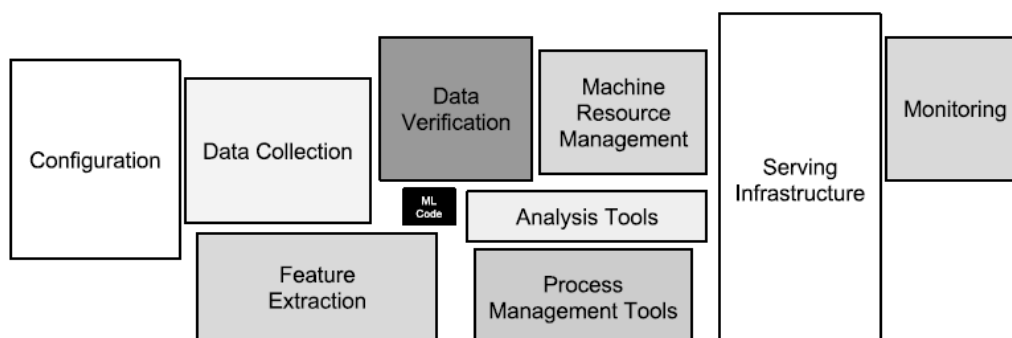


図 3.1 機械学習システムに存在する技術的負債

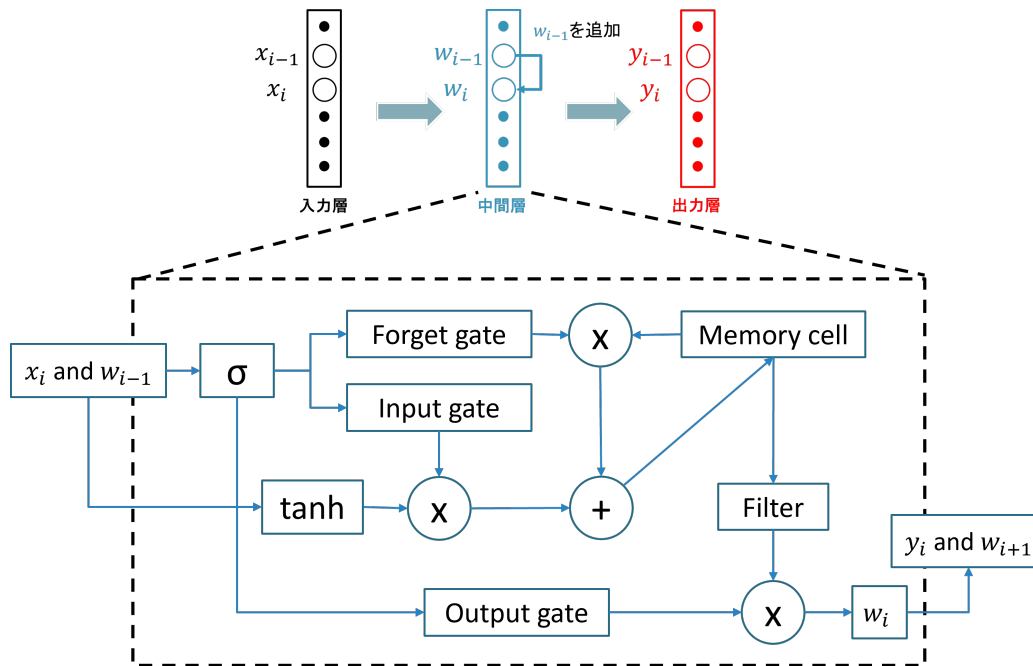


図 3.2 LSTM のモデル図

積を表している。また、LSTM ブロックの内部は 1 つ前 ( $w_{i-1}$ ) の時刻のデータを保持する記憶セル (Memory cell)、1 つ前の時刻からの入力を受け取るかどうかを判断する入力ゲート (Input gate)、過去の出力を入力にフィードバックする出力ゲート (Output gate)、記憶セルの内容をリセットする忘却ゲート (Forget gate) の 4 つのゲートが存在する。これらのゲートを用いることで、長期記憶を実現している。

### 3.2.3 機械学習

機械学習とは、人工知能の分野の 1 つで、人に備わる学習能力と同じような機能をコンピュータで再現しようとする技術である。機械学習は、様々なテクノロジー系の企業が、近年特に力を入れている分野で、例えば、検索エンジン、スパム検出、パターン認識など幅広い分野で応用されている。機械学習の手法としては、ニューラルネットワーク、サポートベクターマシン (SVM)、強化学習、決定木などが上げられる。

### 3.2.4 ニューラルネットワーク

ニューラルネットワークは、人間の脳にある神経細胞を模倣化した機械学習の1つのモデルである。ニューラルネットワークは、データを入力する入力層、データの特徴量を抽出する中間層、結果を得る出力層の3つの層で構成されている。また、近年話題のディープラーニングはニューラルネットワークの中間層が3層以上のものを指している。

### 3.2.5 最適化アルゴリズム

最適化アルゴリズムは、ある凸関数において大域最適解にたどり着くために用いるアルゴリズムである。最適化アルゴリズムは、これまでに多くの手法が提案されているが、その中で今回電力需要予測で使用した Adam を紹介する。

Adam は Adaptive moment estimation の略で、2014 年に Kingma らが提唱したアルゴリズム [3] である。Adam は最適化アルゴリズムの1つである Adagrad と RMSProp の利点を併せ持った手法で、数式は以下のように表される。ただし、 $\beta_1$ ,  $\beta_2$ ,  $\eta$ ,  $\epsilon$  は定数、 $t$  は反復回数である。

$$\begin{aligned} m_0 &= v_0 = 0 \\ m_t &= \beta_1 m_{t-1} + (1 - \beta_1) \frac{\partial E}{\partial w} \\ v_t &= \beta_2 v_{t-1} + (1 - \beta_2) \left( \frac{\partial E}{\partial w} \right)^2 \\ \widehat{m}_t &= \frac{m_t}{1 - \beta_1^t} \\ \widehat{v}_t &= \frac{v_t}{1 - \beta_2^t} \\ w &\leftarrow w - \eta \frac{\widehat{m}_t}{\sqrt{\widehat{v}_t} + \epsilon} \end{aligned} \tag{3.1}$$

### 3.2.6 活性化関数

活性化関数は、ニューロンを興奮させるための関数である。ニューロンへの入力と重みをかけたものの総和にバイアスを足し合わせた値を、ニューロンの興奮

状態を表す記号に変換する。活性化関数がないと、ニューロンの演算は単なる積の総和になってしまい、ニューラルネットワークによる複雑な表現ができなくなってしまう。本研究では、モデルの出力と LSTM に  $\tanh$  関数と呼ばれる活性化関数を使用した。  $\tanh$  関数は以下の式で表される。  $\tanh$  関数は  $-1$  から  $1$  の間を滑らかに変化する関数である。

$$y = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)} \quad (3.2)$$

### 3.2.7 損失関数

損失関数は、出力と正解の誤差を定義する関数である。誤差は、あるべき状態との乖離の度合いである。誤差の値が大きければ、ニューラルネットワークが望ましい状態から離れていることになる。学習の際は、この誤差を最小にするように行われる。本研究では RMSE を使用する。RMSE は Root Mean Squared Error の略で、回帰問題でモデルの良さを測る指標の一つで、式は以下のようにあらわされる。

$$E = \sqrt{\frac{1}{n} \sum_t (y_t - f_t)^2} \quad (3.3)$$

ここで、  $f_t$  は予測値、  $y_t$  は実績値（正解値）であり、  $n$  は予測期間におけるデータの個数である。

### 3.2.8 主成分分析 (PCA : Principal Component Analysis)

主成分分析とは、代表的な次元削減手法の 1 つである。次元削減は、データの次元を、人間が理解しやすい低次元に削減することを目的としている。主成分分析はデータから分散が大きいベクトル（固有ベクトル）を抽出し、ベクトルを並べた行列と元データとの内積を取ることで、データの次元を圧縮できる。次元圧

縮したときの行列  $\mathbf{X}_{\text{pca}}$  は以下のようになる。

$$\mathbf{X} = \begin{bmatrix} x_{11} & \cdots & x_{1D} \\ \vdots & \ddots & \vdots \\ x_{N1} & \cdots & x_{ND} \end{bmatrix}$$
$$\mathbf{U} = \begin{bmatrix} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_d \end{bmatrix}^T \quad \mathbf{u}_d = \begin{bmatrix} u_{1d} \\ \vdots \\ u_{Dd} \end{bmatrix}$$
$$\mathbf{X}_{\text{pca}} = \mathbf{X}\mathbf{U} \tag{3.4}$$

ここで、固有ベクトル  $\mathbf{u}_1, \dots, \mathbf{u}_d$  は正規直交基底、 $D$  は総次元数 (今回は 24 次元)、 $d$  は圧縮次元数、 $N$  はデータ数、 $\mathbf{X}$  はデータ (電力データ) である。

### 3.2.9 選択的アンサンブル平均法

機械学習による予測を行う際、ただ 1 つのモデルの結果を採用するのではなく、複数のモデルの結果の平均を取ることでより予測性能が向上することが広く認識されている。この特性を利用し、複数のモデルに効率的に学習を行わせ、その結果を利用して精度の良い予測を行うことを目指すアンサンブル学習が多くの研究で用いられている。これに対して、Zhou et al. は選択的アンサンブル平均法を提唱した [4]。これは、複数のモデルを学習させ、そのモデルの中から評価誤差が小さいモデルをいくつか選択してアンサンブル平均をとる手法である。これにより予測性能・分類性能の向上の可能性を示した。

## 3.3 前処理

今回電力需要予測モデルを学習する上で、電力需要データ、暦データ、気象データを 3 つのデータを用いて学習を行う。また、データの前処理は日本エネルギー経済研究所の論文 [5] を元に行う。

### 3.3.1 電力需要データ

電力需要データは、一般公開されている東京電力の電力需要データを用いる [7]. また、電力データは以下の式で規格化を行う.

$$x_t = \frac{X_t - \bar{X}}{X_{\max} - X_{\min}} \quad (3.5)$$

ここで、 $X_t$  は元のデータ、 $\bar{X}$  および  $X_{\max}$ ,  $X_{\min}$  はそれぞれ  $X_t$  の平均値および最大値, 最小値,  $x_t$  はモデルの入出力データである.

### 3.3.2 暦データ

暦データは年月日の月, 曜日, 祝日の 3 つの特徴量を用いる. 月はその自然数そのものを使用し, 曜日は日曜を 0, 月曜を 1, ..., 土曜を 6 とする. 祝日は, その日が国民の祝日, もしくは 12 月 29 日~1 月 3 日, 8 月 13 日~16 日の期間内であれば 1, それ以外であれば 0 とする.

### 3.3.3 気象データ

気象データは気象庁が公開しているデータを使用する [8]. 今回は東京の気象データを使用する. この気象データから, 東京の天候, 最高気温, 最低気温の 3 つのデータをモデルの入力に用いる. 最高気温, 最低気温については式 (3.5) で規格化を行い, 天候は one-hot ベクトルで晴れ, 曇り, 雨, 雪の 4 つの変数を用意し, 天気概況にそれぞれ語が含まれていたら 1, そうでなければ 0 とする.

## 3.4 従来手法

本研究では日本エネルギー経済研究所の論文 [5][6] を参考にモデルを作成する. 従来手法のモデル図を図 3.3 に示す. 従来手法では, 3 つの全結合層 NN (Neural Network) を使っている. 図 3.3 より, 左のモデルから順に, 翌日 0 時~7 時までの電力データの平均を予測するモデル, 翌日の電力データの主成分得点を予測するモデル, 翌日 24 時間の電力データの予測をするモデルを表している. 従来手法のモデルはニューロン 30 個, 中間層が 3 層の全結合層で構成されてい

る。中間層の活性化関数は softplus 関数を用い、以下の式で表される。

$$y = \log(1 + \exp(x)) \quad (3.6)$$

選択的アンサンブル平均法では、それぞれ 20 個のモデルでアンサンブル学習を行い、その中から評価誤差が低いモデルを 5 個選択して、テストデータで予測を行っている。また、翌日の電力データの主成分得点を予測するモデルで、翌日の主成分得点を予測して、電力需要の近似曲線を得ているが、この近似曲線は、主成分分析で用いた  $d-1$  日までの電力データの時間ごとの平均値と、電力データの第 4 主成分と翌日の主成分得点との線形結合との和によって求めている。その式を以下に示す。

$$L_{d,t} = W_{0,t} + \sum_{i=1}^4 a_{i,d} W_{i,t} \quad (3.7)$$

ここで、 $W_{0,t}$  は主成分分析の対象とする過去の全ての日の時刻  $t$  の電力需要の平均値、 $a_{i,d}$  はモデルで予測した  $d+1$  日の主成分得点、 $W_{i,t}$  は主成分得点によって予め得られる第 1~4 主成分である。この式 3.7 によって、 $d+1$  日の電力需要の近似曲線を導出している。

従来手法による電力需要予測は高い性能を発揮しているが、役割が異なるモデルを 3 つ使用しているため、学習に時間がかかってしまう。これにより、クラウド環境で実装する上で、学習時間・メモリのコストがかかってしまう。

### 3.5 提案手法

提案手法のモデル図を図 3.4 に示す。提案手法では、LSTM のみで翌日 24 時間の電力データの予測を行う。また中間層の活性化関数は、従来手法では softplus 関数を使用していたが、提案手法では tanh 関数を使用した。また、従来手法・提案手法の出力層の活性化関数は tanh 関数を使用した。これは、電力需要の値を式 3.5 によって、 $-1 \sim 1$  の間で規格化しているため、この関数を使用する。

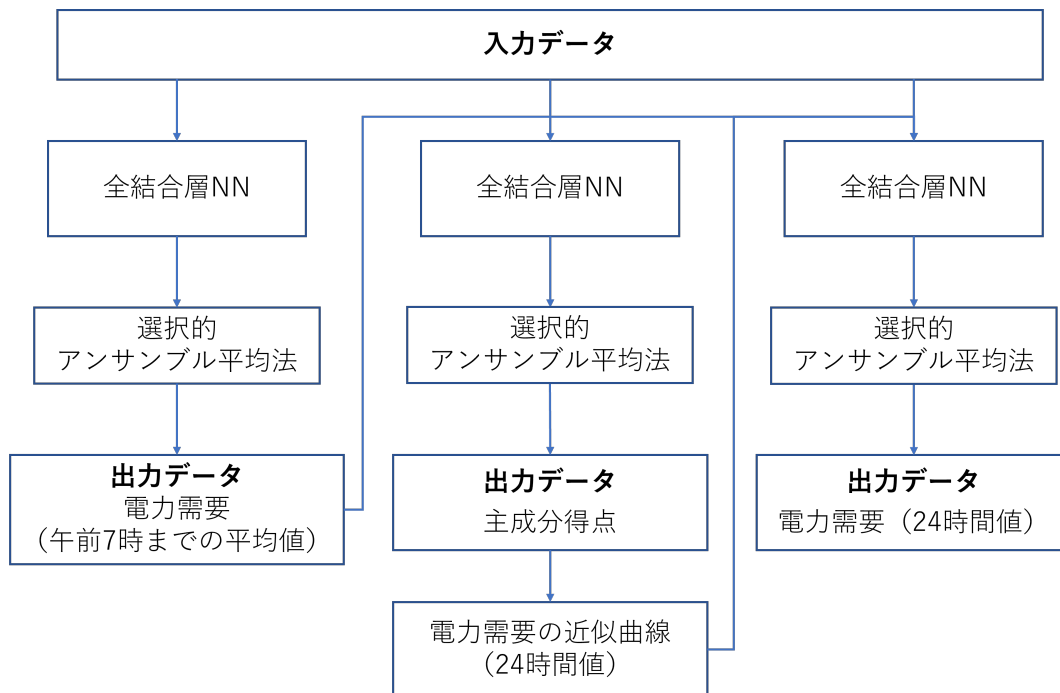


図 3.3 従来手法のモデル図：従来手法では翌日 0 時～7 時までの電力データの平均を予測するモデル，翌日の電力データの主成分得点を予測するモデル，翌日 24 時間の電力データの予測をするモデルの 3 つのモデルを用いて翌日の電力需要予測を行っている。

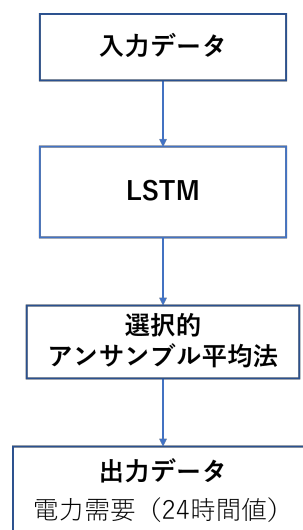


図 3.4 提案手法のモデル図

### 3.6 学習条件

今回学習の条件は表 3.1 のように設定した。中間層の活性化関数は従来手法では softplus 関数を使用していたが，提案手法では tanh 関数を使用した。本研究



では、ビジネスに適応できる電力需要予測モデルを構築するため、AI フレームワークは研究面で利用される「Pytorch」ではなく、「Tensorflow」の Keras を用いて実装を行う。また、エポック数に関しては EarlyStopping と呼ばれる Callback 関数を用意し、評価誤差が連続で大きくなったら、学習を止め、最も評価誤差が小さいときのモデルの重みを保持する。学習データとテストデータに関しては、2016年4月1日～2017年6月30日までを学習データ、2017年7月1日～2018年6月30日までをテストデータとして、モデルの学習・予測を行う。また、学習データに関しては、訓練データと評価データにそれぞれ8割、2割にデータを分割した。

表 3.1 学習条件

条件	従来手法	提案手法
エポック数	100	100
バッチサイズ	16	16
アンサンブル	10	10
最適化関数	adam	adam
誤差関数	RMSE	RMSE
活性化関数（中間層）	softplus	tanh
活性化関数（出力層）	tanh	tanh

また、従来手法と提案手法に入力するデータを以下に示す。この入力データの総次元数は 13 次元である。

- 電力需要データ： $d$  日（4 次元）
- 暦データ（月，曜日，祝日・休暇）： $d + 1$  日
- 気象予報データ（最高気温，最低気温，天候）： $d + 1$  日

東京電力の電力データを主成分分析すると、累計寄与度が第 4 主成分までで 99 % 以上になることが従来手法で確認されている。図 3.5 は今回学習データとして使用する東京電力の電力データを 4 次元に次元圧縮した際の寄与度と累計寄与度を表している。グラフから、第 4 主成分で累計寄与度が 100% 近くまであることが確認できる。つまり、電力データを 24 次元から 4 次元に次元圧縮して

も、元のデータを表現できているということになる。この結果を用い、電力需要（24 時間値）予測のモデルは、24 次元の電力データではなく、4 次元に次元圧縮した電力データを入力データとして使用した。また、この主成分分析で、翌日の電力需要（24 時間値）を予測するモデルで使用する訓練データ・テストデータは、訓練データはそのまま主成分分析を行うが、テストデータに関しては、この訓練データの主成分分析で得た固有ベクトルと内積をとって次元圧縮を行う。これは、機械学習システムとして実現する場合、固有ベクトルをクラウドサービスなどのストレージに保存して、翌日の電力データを予測をする際に、ストレージから固有ベクトルを取得して、当日の電力データと内積を取って次元圧縮を行うことを想定しているためである。

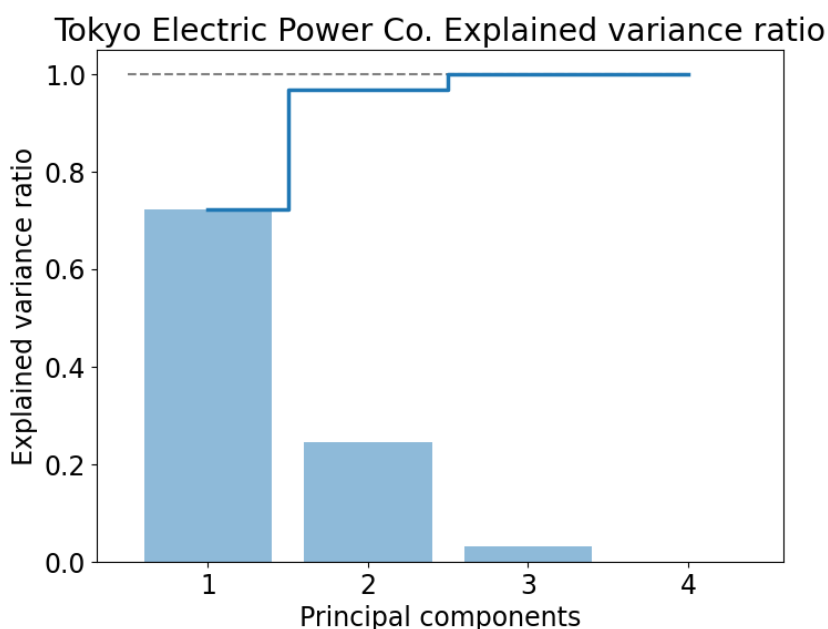


図 3.5 東京電力の電力データを主成分分析したときの寄与度と累計寄与度：東京電力のデータを主成分分析すると、第 4 主成分得点までで累計寄与度が 100% 近傍までであることが確認できる。

### 3.7 予測結果

提案手法による翌日の電力需要予測のグラフを図 3.6 に、従来手法による翌日の電力需要予測のグラフを図 3.7 に、提案手法のアンサンブル法による各モデルの損失の変化を図 3.8 に、従来手法と提案手法の RMSE と学習時間を表 3.2 に

表 3.2 東京電力の電力需要予測における RMSE の結果と学習時間

	従来手法	提案手法
RMSE 万 kW	196.8	154.0
学習時間 秒	273.9	152.3

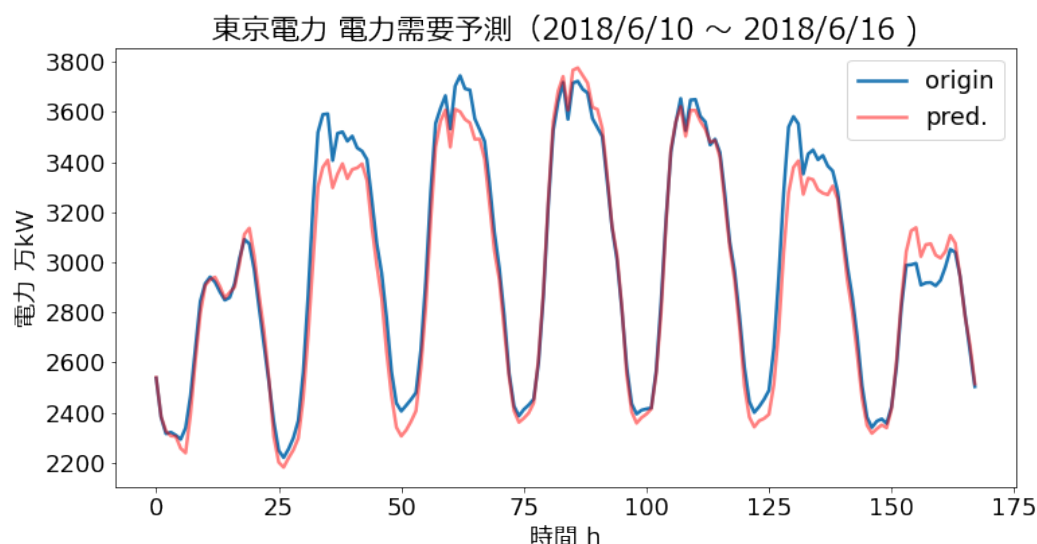


図 3.6 提案手法による翌日の電力需要予測の結果：origin は元の電力データ，pred. は LSTM で電力データを予測したときの予測値を表している。2018 年 6 月 10 日～6 月 16 日の 1 週間分の翌日の電力需要予測を行った結果，正しく予測できていることが確認できる。

示す。

予測結果の図 3.6，図 3.7 では，それぞれ提案手法，従来手法によるテストデータの 2018 年 6 月 10 日～2018 年 6 月 16 日の 1 週間の中の翌日の電力需要予測を示している。モデルに入力するデータの次元を削減したが，両者とも正確な予測ができている。図 3.8 の提案手法による学習損失のグラフで，左図は訓練データの損失グラフ，右図は評価データの損失グラフを表している。グラフから損失が収束に向かっていることから，電力需要予測の学習がうまくいっていることが確認できる。また，表 3.2 より，入力データの次元削減，モデルの改良により従来手法に比べて良い結果となり，学習時間も削減することができた。

従来手法と提案手法の学習損失を図 3.9 に示す。左図が訓練データによる損失の変化，右図が評価データによる損失の変化を表している。また，青線が従来手法，赤線が提案手法を表している。グラフから，提案手法の方が学習の収束が早いこ

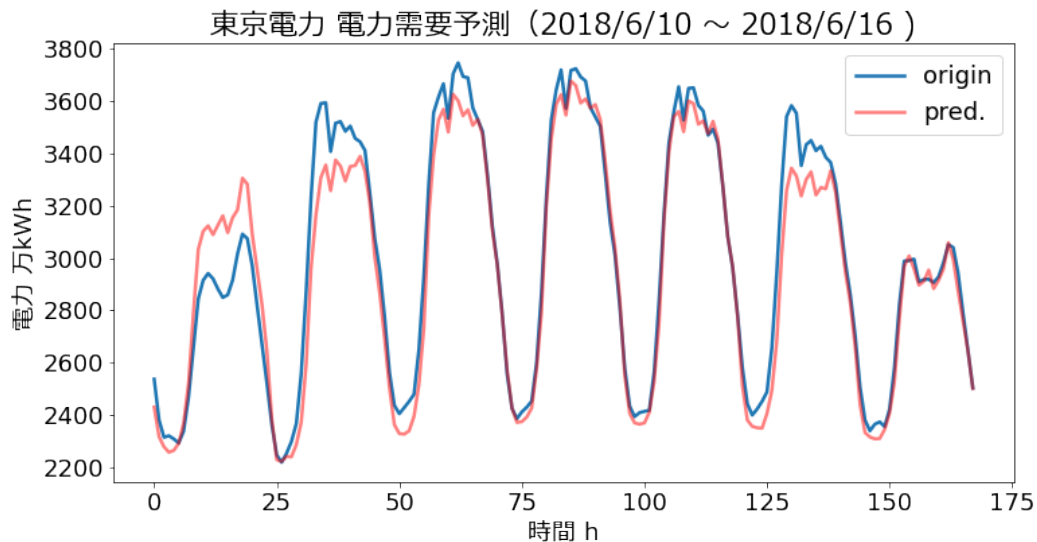


図 3.7 従来手法による翌日の電力需要予測の結果

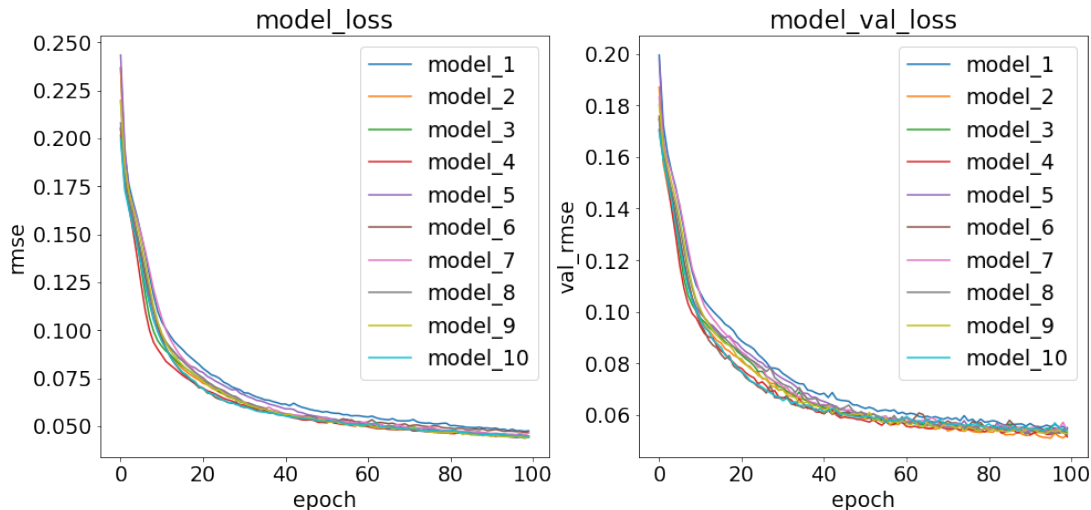


図 3.8 提案手法のアンサンブル法による各モデルの損失の変化：model\_\_番号はアンサンブル学習によるモデルを表しており、合計で 10 個のモデルを学習させている。左図は訓練データによる RMSE のグラフ、右図は評価データによる RMSE のグラフである。両者とも損失の収束に向かっていくことから、学習ができていることが確認できる。

とが確認できる。また、エポック数が増えていくにつれて、右図は左図に比べてノイズが混じっているようなグラフとなっている（RMSE がスムーズに減少していない）。これは、評価データの RMSE において、徐々に最適解に近づくにつれて、局所解に陥っているのではないかと考えられる。

また、電力データの次元数削減による RMSE の変化についても調査を行った。

その結果を図 3.10 に示す。横軸が電力データの次元数，縦軸が RMSE である。電力データの次元数を減らしても，RMSE の変化があまり見られないのが確認できる。この結果から，元の 24 次元の電力データを次元圧縮しても翌日の電力需要予測ができています。これは図 3.5 より，第 4 主成分までで累計寄与度が 99% 以上であったため，次元数を減らしても翌日の電力需要予測ができたと考えられる。

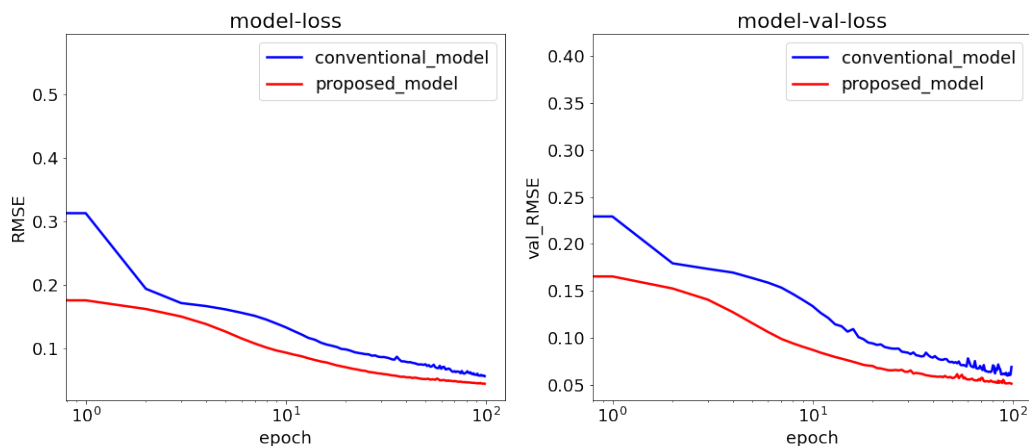


図 3.9 エポック数を対数にしたときの従来手法と提案手法の学習損失：左図が訓練データによる損失の変化，右図が評価データによる損失の変化を表している。青線が従来手法，赤線が提案手法を表している。また，横軸のエポック数は対数を取っている。グラフから，提案手法の方が学習の収束が早いことが確認できる。また，エポック数が増えていくにつれて，右図は左図に比べてノイズが混じっているようなグラフとなっている（RMSE がスムーズに減少していない）。これは，評価データの RMSE において，徐々に最適解につれて，局所解に陥っているのではないかと考えられる。

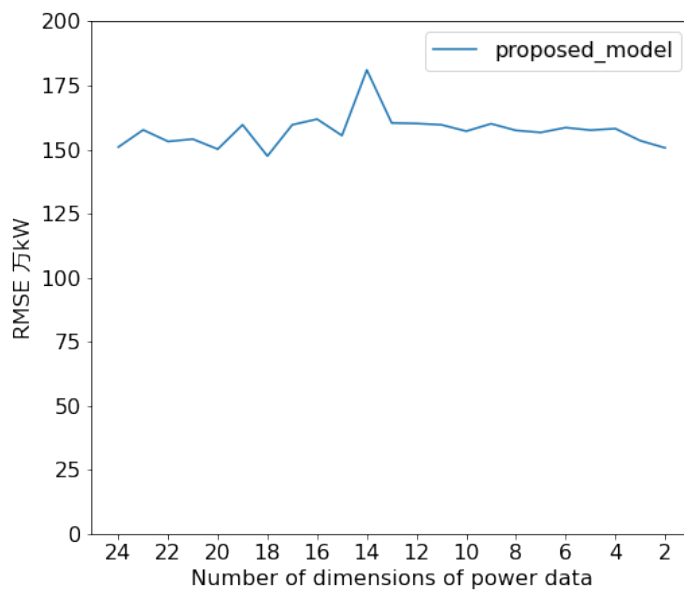


図 3.10 電力データの次元数による RMSE の変化：電力データの次元数を減らしても RMSE の変化があまり見られないのが確認できる。この結果から、元の 24 次元の電力データを次元圧縮しても翌日の電力需要予測ができています。これは図 3.5 より、第 4 主成分までで累計寄与度が 99% 以上であったため、次元数を減らしても翌日の電力需要予測ができたと考えられる。

## 第 4 章

# 結論

本研究では，すごい外国語ペラペラばりぐっどくんとドキュメント翻訳ばりぐっどくんの 2 つの LINE アプリケーション開発と，ML システムとして実用化するための電力需要予測モデルの開発を行った。すごい外国語ペラペラばりぐっどくんについては，音声分割と形態素解析を行い，句点を付与を行い，日本語の長文翻訳を実現したが，ユーザに要求することが多くなり，アプリとして複雑なものとなってしまった。ドキュメント翻訳ばりぐっどくんについては，ドキュメント翻訳だけでなく，Google Drive に自動保存も実装でき，スマホでもドキュメント翻訳が可能にすることができた。電力需要予測については，従来手法に比べては低い性能の電力需要予測モデルとなってしまったが，モデルの入力次元削減と学習コストの削減を行ったことで，モデルの複雑さを改善することができ，ML システムとして実用化できるまでモデルの改良を行うことができた。今後の課題・展望としては，すごい外国語ペラペラばりぐっどくんについては，よりユーザに使いやすい LINE アプリにする必要がある。電力需要予測モデルについては，他の電力データで評価を行い，今回のモデルが実用的か調査する必要があると考えられる。また，東芝グループが研究として発表したスパースモデリングによる気象地点選択を考慮した電力需要予測 [9] と組み合わせることで，より強固な電力需要予測を構築することができると考えられる。

# 謝辞

本研究に取り組むにあたり、多くの方から多大なご助力を賜りました。

まず、私の指導教員である九州工業大学情報工学研究院物理情報工学研究系小田部荘司教授に御礼申し上げます。小田部教授からは、研究に関する知識を次々にご教授いただきました。特に、研究面では、電力需要予測で結果の評価を行うときに、多くのアドバイスをいただきました。また、研究面だけでなく、カンファレンス・学会・アプリコンテストのイベントなど、数々のサポートをいただきました。深く感謝を申し上げます。

次に、株式会社西海クリエイティブ・カンパニーの宮里賢史さん、星野美緒さん、株式会社 Youlib の兵頭悠生さん、小田部教授のご子息の小田部荘志さん、長崎大学の池原陽大さん、日本大学の大友祥基さん、同じ九州工業大学情報工学部の西野颯真さんからは、開発したアプリのマーケティング面や技術面、デザイン面でサポートしていただきました。毎週のようにミーティングを繰り返し、アプリのリリースや進捗状況、今後のビジネスについての相談をいただきました。また、私はアプリ開発が初めてということで、アプリに関する知識を伝授していただきました。心から感謝を申し上げます。

次に、小田部研究室メンバーの辻本天翔さん、有田拳さん、石井皓也さんからは公私ともに助言をいただくなど大変世話になりました。研究に関すること以外にも、日常でかわすコミュニケーションは、私にとって大きな支えとなりました。改めてここに感謝の意を表します。

最後に、私をこれまで育ててくださり、ここまでの学生生活にたくさんの助言や支援をしてくださった両親に多大な感謝を申し上げます。



# 研究業績

## カンファレンス

1. 橋口駿亮, 「LINE Bot の音声翻訳」, 2021 年 7 月 30 日, SAI ZEN SEN for tech 2021 SUMMER

## 国内学会

1. 橋口駿亮, 小田部荘司, 宮里賢史, 星野美緒, 小田部荘志, 山口学, 兵頭悠生, 「日本語の複数文に対応した LINE Bot の音声翻訳」, 2021 年 9 月 25 日, 2021 年度 (第 74 回) 電気・情報関係学会九州支部連合大会

## アプリコンテスト

1. 橋口駿亮, 「ドキュメント翻訳ばりぐっどくん」, 2021 年 11 月 20 日, e-ZUKA スマートフォンアプリコンテスト 2021

## 参考文献

- [1] 新型コロナウイルス陽性者数 (チャーター便帰国者を除く) と PCR 検査,  
<https://www.mhlw.go.jp/content/10906000/000624007.pdf>
- [2] D. Sculley et al., Hidden Technical Debt in Machine Learning Systems, 7-12th  
December 2015, NIPS:1486
- [3] Diederik P. Kingma and Jimmy Lei Ba, Adam: A Method for Stochastic Opti-  
mization, 30 Jan 2017, arXiv:1412.6980.
- [4] Z-H.Zhou et.al., Ensembling neural networks:Many could be better than all,  
2002, Artif.Intell, 137, pp.239-263.
- [5] 松尾雄司ら. 人工ニューラルネットワークを用いた短期電力需要予測 -国内 10  
地域を対象とした検討-, 2018, <http://eneken.ieej.or.jp/data/8106.pdf>
- [6] 松尾雄司ら. 人工ニューラルネットワークを用いた電力需要予測の影響要  
因評価 -学習条件による影響-, 2019, Journal of Japan Society of Energy and  
Resources, 40, 5, pp.144-153.
- [7] 過去の電力使用実績データのダウンロード - でんき予報.  
<https://www.tepco.co.jp/forecast/html/download-j.html>
- [8] 気象庁—過去の気象データ・ダウンロード.  
<https://www.data.jma.go.jp/gmd/risk/obsdl/>
- [9] 気象予測データと機械学習を用いた高精度な電力需要予測手法, September  
2019, 東芝レビュー, 74, 5

# 引用

1. GCP アイコン, <https://techblog.istyle.co.jp/archives/7050>
2. AWS アイコン, <https://aws.amazon.com/jp/console/>
3. Oauth2.0 アイコン, <https://www.xlsoft.com/jp/products/kong/hub.html>
4. Google Translate API アイコン, <https://cloud-ace.jp/gcp/>
5. Cloud Functions アイコン, <https://blog.apar.jp/web/10611/>
6. Firestore アイコン,  
<https://medium.com/google-cloud-jp/firestore1-a62405a7cd82>
7. Google Cloud Storage アイコン, <https://blog.apar.jp/web/7745/>