

公立はこだて未来大学 2019 年度 システム情報科学実習
グループ報告書

Future University-Hakodate 2019 System Information Science Practice
Group Report

プロジェクト名
AI するディープラーニング

Project Name
AI loves Deep Learning

グループ名
グループ B
Group Name
Group B

プロジェクト番号/**Project No.**
16

プロジェクトリーダー/**Project Leader**
川村周也 Kawamura Shuya

グループリーダ/**Group Leader**
横濱拓樹 Yokohama Hiroki

グループメンバ/**Group Member**

横濱拓樹 Yokohama Hiroki
金澤大地 Kanazawa Daichi
寄田慧 Yorita Satoshi
神野健瑠 Kamino Takeru
安加賀元基 Yasukaga Motoki
小滝健太郎 Kotaki Kentaro
呉航平 Kure Kohei

指導教員
竹之内高志 寺沢憲吾 香取勇一 富永敦子 佐々木博昭 片桐恭弘

Advisor
Takenouchi Takashi Terasawa Kengo Katori Yuichi Tominaga Atsuko Sasaki Hiroaki
Katagiri Yasuhiro

提出日
2020 年 1 月 22 日
Date of Submission
January 22, 2020

概要

スマートフォンの普及に伴い、Web ブラウザやアプリケーションを介した「オススメ」を見ない日はない。特にディープラーニングを用いた「オススメ」システムは、Google や Amazon などの企業を中心に一際活気づいている。現在汎用的に使われているディープラーニングを用いた「オススメ」手法のひとつに、対象者がチェックしたデータと対象者以外がチェックしたデータの両方を用いて、そのパターンから人同士の類似性や商品間の共起性を解析するものがある。しかし、ジャンルにとらわれていたり、オススメ内に過去に読んだことのある作品が紛れ込んでいたりする。さらに対象をネット小説にまで限定すると、これに関するアプリケーションは電子書籍リーダーしかなく、オススメ機能は備わっていない。そこで、本グループはネット小説の本文を解析しオススメするシステムの開発を目的とする。私達はその方法として、感情分析を使いオススメする方法が有効であると考え、システムの検討、開発を始めた。そして、小説の本文、およびユーザが好む傾向にある小説を感情分析によって解析し、得られた感情の類似性によって小説をオススメするシステムを開発した。また、このシステムをより簡単に使えるようにするため、アプリケーション化することとした。最終的には、小説をオススメするためのモデルを完成させ、アプリケーション化することができた。

キーワード ディープラーニング, オススメ, アプリケーション

(文責: 横濱拓樹)

Abstract

With the spread of smartphones, there is no day when we do not see the “recommendation” via web browsers or applications. In particular, “recommendation” systems using deep learning are particularly popular among companies such as Google and Amazon. One example of the “recommendation” method using deep learning is the method that uses both the data checked by the target person and the data checked by other person, and analyzes similarity between people and co-occurrence between products from their patterns. However, there are works that have been read in the past that are tied to the genre or that have been read in the recommendations. Furthermore, if the target is limited to online novels, the application related to this is only an e-book reader and does not have a recommended function. According to this observation, we analyzed the text of novels and novels that users tend to like by emotion analysis, and adopted a method of recommending novels based on the similarity of the obtained emotions, and developed a system for that. We analyzed the text of novels and novels that users tend to like by emotion analysis, and developed a system that recommends novels based on the similarity of the obtained emotions. In addition, we decided to make this system an application to make it easier to use. Eventually, the model for recommending the novel was completed and turned into an application.

Keyword Deep learning, Recommendation, Application

(文責: 横濱拓樹)

目次

第 1 章	はじめに	1
1.1	背景	1
1.1.1	該当分野の現状・従来例	1
1.1.2	現状における問題点	1
1.2	本グループの目的	1
1.3	前期の課題設定	2
1.4	後期の課題設定	2
第 2 章	前期の活動内容	3
2.1	技術知識習得	3
2.2	テーマの再考	3
2.3	手法の検討	3
2.4	開発に使用するツール	4
2.4.1	Git/GitHub	4
2.4.2	Slack	4
2.4.3	Google Drive	4
2.4.4	Android Studio	4
2.4.5	Google Colaboratory	5
2.4.6	Jupyter Notebook	5
2.4.7	Visual Studio Code	5
2.4.8	Microsoft PowerPoint	6
2.5	開発準備	6
第 3 章	後期の活動内容	7
3.1	バックエンド	7
3.1.1	データ収集	7
3.1.2	感情分析	7
3.1.3	モデルの作成	7
3.2	フロントエンド	8
3.2.1	UI の検討	8
3.2.2	UI の作成	8
3.2.3	データベースの実装	8
3.2.4	モデルの組み込み	8
第 4 章	感情分析およびオスス手法の詳細	10
4.1	概要	10
4.2	感情分析	10
4.2.1	Word2vec	10

4.2.2	本文の感情分析	11
4.2.3	分析結果の規格化	11
4.3	判別用モデルの作成	12
4.3.1	判別用モデルの概要	12
4.3.2	データセットの作成	13
4.3.3	モデル構築	13
4.4	オススの手法	13
4.4.1	オスス作品の選出	13
4.4.2	オスス作品の決定	13
第 5 章	Android アプリケーションの詳細	15
5.1	概要	15
5.2	機能	15
5.2.1	ホーム画面	16
5.2.2	検索画面	17
5.2.3	マイページ画面	17
5.2.4	ネット小説の詳細画面	19
5.3	アプリの改善案	20
第 6 章	外部評価	21
6.1	中間発表	21
6.1.1	中間発表会の準備	21
6.1.2	評価結果	21
6.1.3	考察	22
6.2	最終発表会	22
6.2.1	最終発表会の準備	22
6.2.2	評価結果	23
6.2.3	考察	24
第 7 章	前期の振り返り	25
7.1	前期の各メンバーの役割と振り返り	25
7.2	前期全体の振り返り	30
第 8 章	後期の振り返り	31
8.1	後期の各メンバーの役割と振り返り	31
8.2	後期全体の振り返り	37
8.3	今後の展望	38
第 9 章	まとめ	39
	参考文献	40

第 1 章 はじめに

1.1 背景

1.1.1 該当分野の現状・従来例

現在、インターネット上にはさまざまなレコメンデーションサービスが存在する。例えば、Google の「Google 広告」や Amazon の「あなたへのおすすめ」などの Web サービスや、個人的に作られた Web サイトがある。Web サービスでは、今までに購入したものや閲覧記録などをもとにユーザが興味を持ちそうなものをピックアップしてオススメする。また、個人的に作られた Web サイトでは、そのサイトの作成者個人が気に入った商品をオススメしたり、ジャンルごとに分かれたものをもとに最も人気のものをオススメしたりする。

(文責: 安加賀元基)

1.1.2 現状における問題点

オススメの対象を一般的な小説に限定すると、これをオススメする Web サービスはいくつか存在する。例えば、「BookLive」や「LINE ノベル」などがある。しかし、ジャンルにとらわれている、オススメ内に過去に読んだことのある作品が存在するなどの問題がある。つまり、現状のオススメ手法ではユーザが本当に満足するオススメができていない。さらに対象をネット小説にまで限定すると、これに関するアプリケーションは電子書籍リーダーしかなく、オススメ機能は備わっていない。ネット小説とは、一般的にインターネット上の小説投稿サイトで公開されている小説であるが、ここでは「小説家になろう」 [1] という小説投稿サイトの小説とする。

(文責: 安加賀元基)

1.2 本グループの目的

本グループの目的は 2 つある。1 つ目は、ネット小説の内容を感情分析し、ネット小説をオススメするシステムを開発することである。内容を分析することで、従来のオススメよりもパーソナライズされたオススメが可能になるはずである。2 つ目は、この機能を Android アプリケーション（以下、アプリケーションと表記）として実装することである。アプリケーション化することによって、このシステムをより簡単に利用できるようになるはずである。ネット小説の感情分析およびネット小説のオススメにはディープラーニングを用いる。ディープラーニングとは機械学習の手法の 1 つで、ニューラルネットワークの中間層をより多くしたディープニューラルネットワークを用いた学習のことである。また、ここでの感情分析とは、ネット小説の本文の単語と感情を表す単語との関連性をそれぞれ数値化することである。感情分析は、既存のモデルである Google の「Word2vec」で行う。また、ネット小説のオススメは、CNN (Convolutional Neural Network) で行う。

1.3 前期の課題設定

本グループの目的は2つあった。1つ目は、ネット小説の内容を感情分析し、ネット小説をオススメするシステムを開発することである。2つ目は、この機能をアプリケーションとして実装することである。そこで、前期の課題はアプリケーションのプロトタイプを作成することとした。プロトタイプの機能は、小説の感情分析の結果を折線グラフで出力することである。

この課題に取り掛かる前にバックエンドとフロントエンドに班分けをし、タスクを明確にした。バックエンドのタスクは、ネット小説の本文のデータを収集することやデータ整形プログラムを作成すること、Word2vecを学習させることである。また、フロントエンドのタスクは、Android Studio および Kotlin の勉強をすることや UI および画面遷移を実装することである。

(文責: 安加賀元基)

1.4 後期の課題設定

後期の課題はアプリケーションを完成させることとした。完成したアプリケーションの機能は、プロトタイプの機能に加えて、アプリユーザにネット小説をオススメすることである。バックエンドのタスクは、ネット小説の本文のデータを収集することやデータ整形プログラムを作成すること、CNN の設計および学習させることである。また、フロントエンドのタスクは、UI および画面遷移を実装することや Realm Database を実装すること、CNN モデルを組み込むことである。

(文責: 安加賀元基)

第 2 章 前期の活動内容

この章では、前期の活動内容を時系列に沿って述べる。

2.1 技術知識習得

私達は当初、グループのテーマを「擬似人格」と定めた。この「擬似人格」とは、まるで人格を持っているように見える人工知能である。その後、ディープラーニングの手法や理論を学習するために『Python と Keras によるディープラーニング』を用いた。この書籍はいくつかのディープラーニングの実装例を Python でプログラミングしていく内容となっている。しかし、全てを読み理解する必要がないため、6 章まで進めることを決め学習に取り組んだ。1 章ではディープラーニングとはどのような手法なのかを学習し、2 章ではニューラルネットワークの基礎と実装を、3 章では 2 章の実装の続きと Keras の説明と環境構築を行った。さらに 4 章で機械学習の基礎を学び、5 章と 6 章で小さなテキストデータで CNN の実装や訓練を行った。学習はまず個人で行い、その中で理解できないことがあれば、理解しているメンバーに質問した。次に、理解を深めるためにプロジェクト学習の時間を使って、全ての章を講義形式で学習した。その際、講師役は分担して行った。

(文責: 寄田慧)

2.2 テーマの再考

グループ勉強会後は、各自で実際に擬似人格に関わる調査を行った。調査を踏まえて、実現可能な具体的なテーマをブレインストーミング方式で新たに決定した。そのテーマは「感情分析によるネット小説のオススメ」となった。また、このシステムの利便性を高めるため、Android でのアプリケーション化を目指すことになった。そこで、アプリケーション化を目指すにあたり、フロントエンドとバックエンドの 2 つの班に分かれた。班分け後は、それぞれの班で作業を行った。フロントエンド班はアプリケーションの開発を行い、バックエンド班はネット小説をオススメするシステムを構築を行うことになった。

(文責: 寄田慧)

2.3 手法の検討

このテーマを踏まえて、各自で感情分析の手法を調べ試行した。まずデータを取得するため、Web スクレイピングを用いてネット小説の文章を取得した。この Web スクレイピングで、「なろう API」が最も優れた手法であると判明した。また、試した手法は「Word2vec」と単語の辞書をベースとしたもの、IBM の API である「Watson Tone Analyzer」の 3 つである。その中で、「Watson Tone Analyzer」は日本語に対応したものがなく、これを利用しないことにした。

2.4 開発に使用するツール

2.4.1 Git/GitHub

フロントエンド・バックエンド共にバージョン管理を行うために Git を使用した。Git は変更履歴を記録・追跡することができる。また、バージョン管理をしつつプロジェクトの成果物の共有を行うために GitHub を使用した。GitHub ではバージョン管理のほかにプロジェクトのタスク管理機能もあるため、これを利用し個々人のタスク管理を行った。

(文責: 金澤大地)

2.4.2 Slack

グループ内のコミュニケーションを円滑にするために Slack を使用した。Slack とは、チームコミュニケーションツールであり、グループチャット、1 対 1 のメッセージング、音声通話機能を備えている。Slack は、Web サービス、Windows、MacOS のデスクトップアプリおよび iOS、Android のスマートフォンアプリで提供されている。

Slack の利点として以下の点が挙げられる。第一に、チャンネルの存在により、用途別にチャットルームを作成できる。第二に、ファイル共有が容易である。第三に、検索機能が充実している。第四に、Google Drive や OneDrive、Github などの多種多様な外部サービスとの連携が容易にできる。

(文責: 呉航平)

2.4.3 Google Drive

バックエンドのデータを共有するために、Google Drive を使用した。また、報告書や発表資料の作成時には Google ドキュメントを使用した。Google ドキュメントは共同編集機能を備えており、複数人での同時編集が可能である。

(文責: 呉航平)

2.4.4 Android Studio

アプリケーション開発には Android Studio を使用した。Android Studio は、Android アプリケーション開発用の統合開発環境である。Android Studio は、柔軟なビルドシステムや高速なエミュレータ、管理の容易なライブラリなどがあり、Android アプリケーション開発をスムーズに行うことができる。また、レイアウト XML がテキストディタモードでリアルタイムでプレビューに反映されるため、アプリのレイアウトのトライアンドエラーをすぐに行うことができる。このシステムは開発言語の Kotlin を使用できる。Kotlin は、従来 Android アプリケーションに利用されていた Java 言語よりも簡潔かつ安全に開発することができるため、この言語を使用することにした。

2.4.5 Google Colaboratory

Python を扱う環境の 1 つとして Google Colaboratory を使用した。Google Colaboratory は、ブラウザから Python を記述し実行できる。また、ほかには基本的には環境構築が不要である。また、GPU への無料アクセスすることができる。これにより、機械学習をスムーズに行うことができる。

Google Colaboratory の利点として以下の点が挙げられる。第一に、基本的に環境構築が不要であるため、学習のための環境構築の手間はゼロになり、自分のパソコンに何もインストールする必要がないのでいきなり開発することができる。第二に、無料で GPU を使えるためにコストがかからないことである。

(文責: 寄田慧)

2.4.6 Jupyter Notebook

Python での開発環境の 1 つとして Jupyter Notebook を使用した。Jupyter Notebook は、ノートブックと呼ばれるファイルにプログラムや説明の文章、実行結果などをまとめて管理できる、データ分析用のツールである。また、オープンソースで公開されており、個人／商用問わず、誰でも無料で利用することができる。そして、ブラウザで動作するため、いろいろな OS から利用でき、チームメンバーとのプログラムの共有も容易である。

Jupyter Notebook の利点として以下の点が挙げられる。第一に、エクスポート機能である。Jupyter Notebook のノートブックは Ipynb 形式のファイルとして保存し、他の PC で開くこともできるが、Jupyter Notebook を利用していない人には PDF や HTML 形式でエクスポートとして共有することが可能である。また、Python のソースコード部分のみを実行可能な Py ファイルとして出力することもできる。第二に、Markdown に対応していることである。Markdown は、文書を記述するための軽量マークアップ言語のひとつであり、プレーンテキスト形式で手軽に書いた文書から HTML を生成するモノである。よって、Markdown 記法を使ってメモを取ることが切る。

(文責: 寄田慧)

2.4.7 Visual Studio Code

Python での開発環境の 1 つとして Visual Studio Code を使用した。Visual Studio Code は、Windows OS や .NET Framework の開発元である Microsoft 社が開発したオープンソースのテキストエディタである。プログラムの開発は IDE で行われるケースは多いが、機能が豊富な IDE は起動や動作が遅くなってしまうケースもよくある。Visual Studio Code は軽量のテキストエディタなので、プログラムの作成・修正を手軽に行うことができる。また、プログラムのデバッグを行うことができる。

(文責: 寄田慧)

2.4.8 Microsoft PowerPoint

ポスターやスライド制作に Microsoft PowerPoint を使用した。Microsoft PowerPoint とは、Microsoft のプレゼンテーションソフトで、プレゼンテーションの準備から本番までをトータルにサポートする。PowerPoint を使うと、文字や写真、グラフなどが入った見栄えのするスライドを簡単に作成できる。

Microsoft PowerPoint の利点として以下の点が挙げられる。第一に、表やグラフ、図表などを利用した視覚効果の高い資料を作成できる。また、資料の背景に使えるデザインも豊富に用意されている。第二に、対面式のプレゼンテーションだけでなく、プロジェクターを使って発表する本格的なプレゼンテーションも実施できる。実際に最終発表会では Microsoft PowerPoint をプロジェクターとして使用した。

(文責: 寄田慧)

2.5 開発準備

フロントエンドでは、Android Studio や Kotlin に対する知識が不足していたため、まず Android アプリケーション開発の勉強を行った。次に、プロトタイプに関する画面遷移図の作成と検討を行った。画面遷移図を個々人で作成し、フロントエンドのメンバーで検討を行い、夏季休暇中にプロトタイプの作成を進めた。

バックエンドでは、Python について知識が不足していたため、Web サイト [2] などを参考に勉強を行った。その後に、Word2vec をネット小説に最適化するための学習を行った。最適化をするために、Word2vec の公式サイト [3] [4] を参考にモデル作成時に必要なパラメータ調整を行った。

(文責: 金澤大地)

第 3 章 後期の活動内容

3.1 バックエンド

3.1.1 データ収集

ネット小説をオススメするシステムを構築するために、「Word2vec」に学習させるための単語と様々なジャンルのネット小説が必要であった。「Word2vec」に学習させるデータを収集するために Wikipedia を利用した。Wikipedia には日本語の記事をダンプ化させたものがあり、それをダウンロードして記事の本文データのみを取り出した。ネット小説を収集するために「なろう API」を使用した。「なろう API」とは、小説投稿サイトに掲載されている小説情報を簡単に取得できるものである。この「なろう API」を使用して、恋愛、ファンタジー、文芸、SF、その他、ノンジャンルの 6 ジャンルのデータをランキング上位 100 作品ずつネット小説の本文をスクレイピングした。スクレイピングとは、ウェブサイトから情報を取得し、その情報を加工して新たな情報を生成することである。このスクレイピングした小説は、1 話で終わる短編のものから 200 話以上ありアニメ化や書籍化までされた有名なものまであった。

(文責: 寄田慧)

3.1.2 感情分析

我々はデータ収集で得たネット小説の文章の感情分析を行った。最初に、データ収集した単語を「Word2vec」に学習させた。次に、スクレイピングしたネット小説の本文を単語に分けた。その後、本文を単語に分けたものを「Word2vec」を用いて小説の本文の単語と感情を表す「喜び」、「怒り」、「哀しみ」、「楽しみ」の 4 つの名詞 (4 感情) との距離をそれぞれ算出し、距離から単語と 4 感情との類似度を求めた。

(文責: 寄田慧)

3.1.3 モデルの作成

我々がネット小説をオススメするシステムとして採用したモデルは CNN である。我々はこの作成したモデルを学習させた。その学習データはネット小説の評価データである。ネット小説の評価データとは、各ユーザが小説に対して文章とストーリーをそれぞれ 1 から 5 ポイントの合計 10 ポイントの間で評価したものである。例えば、総合評価が 72924 ポイントの小説があったとすると、文章評価が 10416 ポイントあり、ストーリー評価が 10856 ポイントある。

(文責: 寄田慧)

3.2 フロントエンド

3.2.1 UI の検討

後期の初めにプロトタイプを作成し改良点の検討を行った。プロトタイプでは検索画面と詳細画面の 2 画面のみだったため、アプリケーション起動後に表示するホーム画面を作成することにした。次に、ネット小説の詳細画面でユーザーへのオススメ作品の選出に必要な好きな本と嫌いな本を設定できるようにした。また、この画面から「小説家になろう」の作品ページに移動できるように検討した。次に、ユーザーのグラフと好きな本と嫌いな本を表示するマイページを作成し、この画面で好きな本と嫌いな本に設定しているネット小説の確認と編集ができるようにした。最後に、ホーム画面には 3.1.3 で作成したモデルから選出されたオススメ作品を表示することにした。

(文責: 金澤大地)

3.2.2 UI の作成

まず、プロトタイプを参考にホーム画面を作成し検索画面に移動できるようにした。次に、ネット小説を好きな本と嫌いな本に登録できるようにした。また、ホーム画面から移動できるマイページに、ユーザーグラフの表示画面と好きな本と嫌いな本の確認、編集画面を作成した。そのあとに、3.1.4 で作成したモデルから選出されたオススメのネット小説を表示するためにホーム画面を改良した。

(文責: 金澤大地)

3.2.3 データベースの実装

感情分析したネット小説のデータを Android で使用するために、Realm Database に保存した。Realm Database は、Android で主に利用されている SQLite とは違い、オフライン上でもアクセスすることができ、より高速に結果を得ることができるため使用した。Realm Database では、スクレイピングした小説のデータや感情分析によって算出された感情データを保存している。これらのデータは、アプリケーションでネット小説を検索した際の検索結果やグラフを表示する際に感情データを使用する。

(文責: 金澤大地)

3.2.4 モデルの組み込み

3.1.3 で作成したモデルを Android アプリで使用するために ML Kit を使用した。ML Kit とは、モバイルアプリ向けに機械学習機能を組み込むことができる Firebase の SDK である。デフォルトでは、文字認識や画像ラベリング、顔認識などの機能が利用可能となっているが、本アプリケーションに組み込みたいモデルはデフォルトでは提供されていないため、3.1.3 で作成したモデルを利用した。

具体的には、まず 3.1.3 で作成した TensorFlow の機械学習モデルを TensorFlow Lite のモデ

AI loves Deep Learning

ルに変換した。次に，ML Kit カスタムモデル API を使用して，上記の事前トレーニング済みの TensorFlow Lite モデルをアプリにダウンロードした。最後に，ダウンロードしたモデルを使用して推論を実行した。

(文責: 呉航平)

第 4 章 感情分析およびオススメ手法の詳細

4.1 概要

本プロジェクトではネット小説を感情分析し、その結果を用いてネット小説をユーザにオススメする。3 章の 3.1 節では、この機能を実現するためにプロジェクト内で行った活動内容を記した。この章では 3.1 節で記した活動内容の詳細として、感情分析の開発手法やオススメ機能の開発手法、開発したオススメ手法を説明する。感情分析やオススメでは、ライブラリや計算式、2 つの機械学習のモデルを用いて開発を行った。そのため、それらの概要や活用方法なども説明する。

(文責: 小滝健太郎)

4.2 感情分析

4.2.1 Word2vec

Word2vec は Google の研究者である Tomas Mikolov 氏が提案した手法である。Word2vec は大量のテキストデータを解析することにより、それぞれの単語の意味をベクトルで表現することができる。単語をベクトル化することで単語間の意味がどの程度類似しているかを計算できたり、単語同士の意味を足し引きして該当する単語を導くことができたりする。本プロジェクトでは、前者を用いて感情分析を実装する。以下に、Word2vec モデルの入出力を図示する。

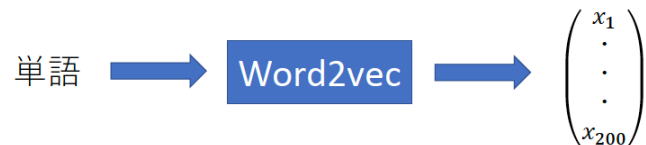


図 4.1 Word2vec の入出力

まず、Word2vec に単語を学習させるためのデータを Wikipedia から収集した。Wikipedia では、日本語の記事のデータがダンプ化されたものが用意されている。それをダウンロードしたあとに記事の本文データのみを取り出し、この約 7GB のテキストデータを Word2vec に学習させた。Word2vec では、学習をするときのパラメータを調節できる。[3] [4] 今回はその機能を活用して、モデルに学習させる単語の前後 10 単語を学習に用いた。

他にも、出現回数 15 回以下の単語を除外し 1 単語を 200 次元のベクトルで表現した。以上の設定を適用した後に Word2vec モデルの学習を行った結果、344,750 単語の学習に成功し、これを感情分析に用いた。

(文責: 小滝健太郎)

4.2.2 本文の感情分析

Word2vec では、2つの単語間の類似度を求められる。これで求められる類似度は、1.0 に近づくほど単語同士が類似しており、-1.0 に近づくほど単語同士が類似していないことを意味する。まずはこの Word2vec の機能を用いて、学習させた全ての単語と 4 感情との類似度を求めた。以下に単語と感情との類似度の計算結果の例を示す。

表 4.1 単語と 4 感情との類似度計算のイメージ

	涙	笑顔	激怒
喜び	0.1	0.9	-0.7
怒り	-0.2	0.1	0.8
哀しみ	0.8	-0.6	-0.5
楽しみ	0.2	0.7	0.3

その後、感情ごとに単語と類似度の組み合わせをリスト化し、これを類似度リストとした。小説の本文は、まず全ての話を結合して 1 つのネット小説に対して 1 つのテキストファイルにした。その後そのテキストを分かち書きしネット小説を単語リストに編集した。

次に、単語リスト内の単語を 1 つずつ 4 感情それぞれの類似度リストと照合することでネット小説の単語リストを類似度を表す数値リストに変換した。これが感情分析の具体的な流れである。ただし感情分析の際に、ネット小説に含まれる全単語のうち Word2vec が学習していない単語が 5% 程度あった。これは今回使用せず、残りの約 95% の単語を感情分析に用いた。

(文責: 小滝健太郎)

4.2.3 分析結果の規格化

感情分析を終えたあと、ネット小説ごとに作成された数値リストを編集し感情データとして扱えるようにした。しかし、感情分析をすることで完成した数値リストは小説の単語数によってサイズが異なるという問題がある。この問題を解決するために、まず数値リストを小説の進行度 10% ごとに 10 分割した。その後それぞれを感情ごとに合計し、1 作品に対し 40 個の数値データを作成した。

次に、その合計値を 0 から 1 に規格化するために以下の式を用いた。

$$n_{ij} = \frac{x_{ij} - x_{min}}{x_{max} - x_{min}}.$$

n_{ij} は数値リストを分割し合計することでできた 40 個の数値のうち 1 つを表している。ここで、 x_{ij} や n_{ij} における i は感情を表し、 j は進行度を表している。また、 x_{max} や x_{min} はそれぞれ 40 個の x_{ij} の最大値と最小値である。この式によって導かれる 40 個の n_{ij} が、0 から 1 に規格化されたデータとなる。これらの処理を行った結果、ネット小説の単語数によるサイズの変化を解決でき、これを感情データとした。

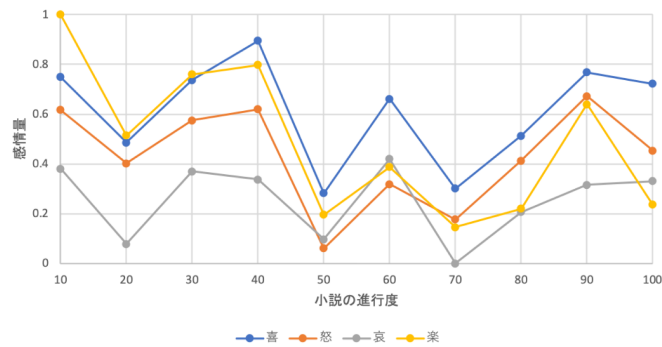


図 4.2 規格化された感情データの例

図 4.2 は感情データの例である．この例の場合，最初 10% は楽しみが強く，哀しみが弱く現れている場面であるとわかる．それに対して小説の結末では，喜びが強く楽しみが弱く現れている場面に切り替わっていることがわかる．それに加えて，この感情データを見ると，感情が全体的に強く現れている場面と弱く現れている場面があることがわかる．

その後オススの実装をするために，感情データの振る舞いを以下の式によって数値化し小説データを作成した．

$$y_i = \frac{1}{10} \sum_{j=1}^{10} (n_{ij} - 0.5)^2.$$

n_{ij} はネット小説の感情データやユーザの感情データを指している．この式を用いて感情ごとに振る舞いを規格化すると，1つのネット小説に対して1つのベクトル $Y = (y_1, y_2, y_3, y_4)$ が求められる．この計算の結果として導かれる Y が小説データやユーザデータとなる．

ユーザデータについても感情データから同じ方法で作成した．ユーザの感情データは，ユーザが好きな本に登録したネット小説の感情データが均等に作用するように，感情データのそれぞれの値を平均し求めた．

(文責: 小滝健太郎)

4.3 判別用モデルの作成

4.3.1 判別用モデルの概要

ユーザにネット小説を的確にオススめるために，オススの判別用モデルを作成した．入力 は小説データの元となる感情データとユーザデータの元となる感情データとした．また，出力はユーザに入力した小説をオススめすべきかそうでないかの判定とした．このモデルでは，実際に「小説家になろう」のユーザ（以下，なろうユーザ）がネット小説を評価した情報を学習させた．以下に判別用モデルの入出力を図示する．

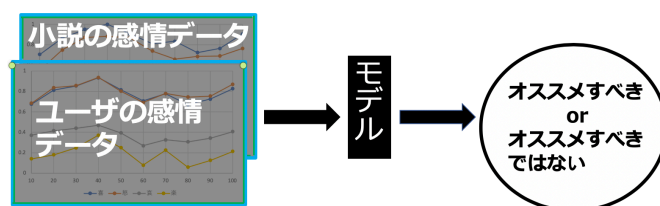


図 4.3 判別用モデルの入出力

4.3.2 データセットの作成

小説家になろうでは、ネット小説を文章とストーリーのそれぞれの観点から最大 5 ポイント、合計 10 ポイントで評価できる。このデータをなろうユーザ 100,000 人分収集した。データセットのユーザデータの元となる感情データは、ユーザが高い評価をつけたネット小説の感情データを平均して作成した。小説データの元となる感情データはデータベースに保存されているものとした。ラベル付けには評価ポイントを用いた。最大 10 点の評価ポイントで、10 点をつけたネット小説をオススメすべきと、それ以外をオススメすべきでないとラベル付けした。

(文責: 小滝健太郎)

4.3.3 モデル構築

判別用モデルでは、2 つの感情データを入力とする 2 分類問題である。また、入力の形式が画像の入力形式と同様に 3 次元の配列になっている。そのため、全結合層の他に畳み込み層を組み込んだ CNN を構築した。また、構築したモデルでの最適なハイパーパラメータを求めるためにグリッドサーチを行った。その結果、正解率が 70% 程度のモデルが完成した。

(文責: 小滝健太郎)

4.4 オススメの手法

4.4.1 オススメ作品の選出

まず、20 件のオススメ作品を候補として選出する。これは、ユーザデータと小説データを用いて計算をした結果を元にする。計算式は以下のものを用いた。

$$d = \sum_{i=1}^4 |y_i - y'_i|.$$

y_i は小説データ、 y'_i はユーザデータを表している。ここで、 y_i や y'_i は 4.2.3 節で求めた Y の要素である。この式を用いることでユーザデータと小説データの差が求められ、この差が小さいほどそれぞれのデータの元となる感情データが類似していることを意味する。この差が小さい順から 20 件リストアップしたものをオススメ作品の候補とした。

(文責: 小滝健太郎)

4.4.2 オススメ作品の決定

ユーザデータとリストアップされたネット小説を用いてオススメ作品を決定する。これには判別用モデルを用いる。このモデルを用いることで、20 件あるオススメの候補から判別用モデルの出力を元に最終的なユーザに対するオススメ作品を決定する。このモデルを介すことによって、単に感情分析の結果からオススメするのではなく、実際に小説家になろうに登録しているユーザの評価

AI loves Deep Learning

データを加味したオススめを実現できた.

(文責: 小滝健太郎)

第 5 章 Android アプリケーションの詳細

5.1 概要

私達が開発したアプリケーションは、ネット小説を感情分析した結果をもとにユーザにネット小説をオススメするアプリケーションである。このアプリケーションは、約 3500 件のネット小説のデータが登録されており、それぞれのネット小説の感情グラフを作品名や作者名、あらすじなどとともに見ることができる。また、好きな本や嫌いな本の登録を行うことで、あなたへのオススメやユーザグラフが表示されるようになる。

アプリの名前は、感情分析を使って本をオススメするアプリであるため、感情を意味する英単語である「emotion」と本を意味する「book」の 2 つの単語を組み合わせた「EmoBooks」とした。次節より、このアプリの主要機能や検討中の機能について詳しく記述する。

(文責: 呉航平)

5.2 機能

この節では、実装したアプリケーションの機能を画面ごとに説明する。このアプリケーションには大きく分けてホーム画面、検索画面、マイページ画面の 3 つの画面がある。これらは Bottom navigation で管理されており、それぞれの画面に遷移することができる。

次に、検索画面では検索を行うと、検索結果画面に遷移する。またマイページ画面には、ユーザグラフ画面、好きな本一覧画面、嫌いな本一覧画面の 3 つの画面が存在する。これら 3 つの画面はタブで管理されており、自由に遷移することができる。さらに、それぞれの画面からネット小説をタップするとその小説の詳細画面に遷移することができる。このアプリケーションの画面遷移図を図 5.1 に示す。

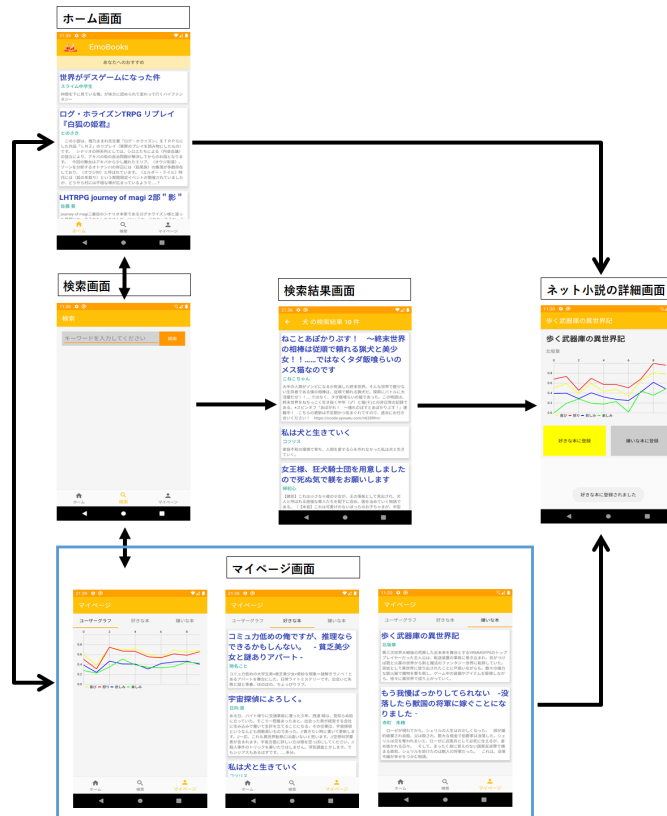


図 5.1 画面遷移図

(文責: 呉航平)

5.2.1 ホーム画面

ホーム画面には、登録した好きな本と嫌いな本に基づく、あなたへのオススメが最大20件表示される。あなたへのオススメを表示するには好きな小説を1冊以上登録する必要がある、好きな本や嫌いな本を変更すると、あなたへのオススメも変更される。ホーム画面を図5.2に示す。



図 5.2 ホーム画面

1. あなたへのオススメ
2. ホーム画面ボタン
3. 検索画面ボタン
4. マイページ画面ボタン

(文責: 呉航平)

5.2.2 検索画面

キーワードを入力することで、自分の好きなネット小説や気になるネット小説を探すことができる。検索方法には部分一致検索を使っており、検索結果としてキーワードを含むものを全て表示する (図 5.3)。

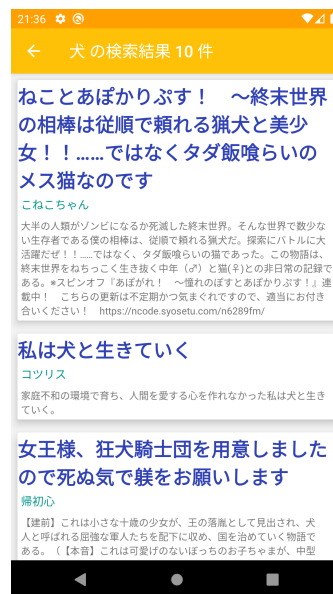


図 5.3 検索結果画面

(文責: 呉航平)

5.2.3 マイページ画面

マイページ画面では、ユーザグラフを見ることができる。ユーザグラフはユーザが好きな本に登録したネット小説をもとに算出したものである。ユーザグラフを見ることが、自分が好きなネット小説の傾向を把握することができる。マイページ画面を図 5.4 に示す。また、好きな本や嫌いな本に登録したネット小説が一覧として表示される。好きな本の一覧表示画面を図 5.5, 嫌いな本の一覧表示画面を図 5.6 に示す。

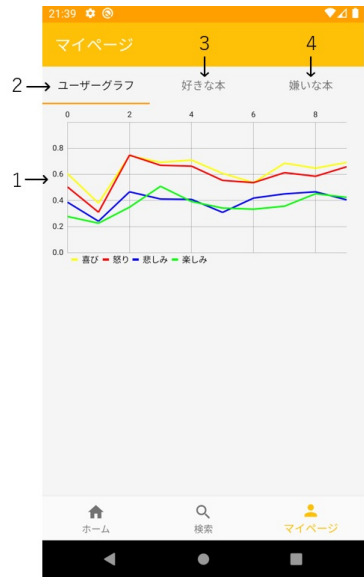


図 5.4 ユーザーグラフ表示画面

1. ユーザーグラフ
2. ユーザーグラフ表示画面
3. 好きな本の一覧表示画面
4. 嫌いな本の一覧表示画面

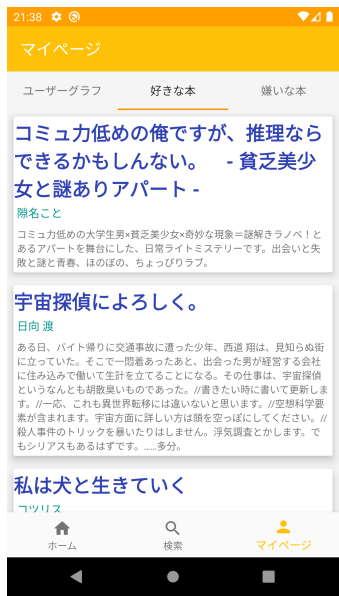


図 5.5 好きな本一覧表示画面

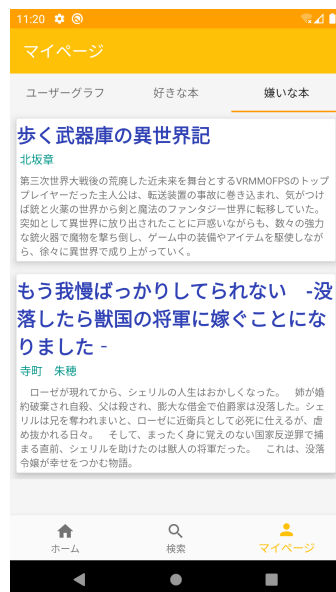


図 5.6 嫌いな本一覧表示画面

(文責: 呉航平)

5.2.4 ネット小説の詳細画面

ネット小説の詳細画面では、そのネット小説の感情グラフを見ることができる。また、評価機能もあり、そのネット小説を好きな本や嫌いな本に登録することができる。ここで登録した好きな本と嫌いな本に基づき、ホーム画面であなたへのオススメが表示される。ネット小説の詳細画面を図 5.7 に示す。

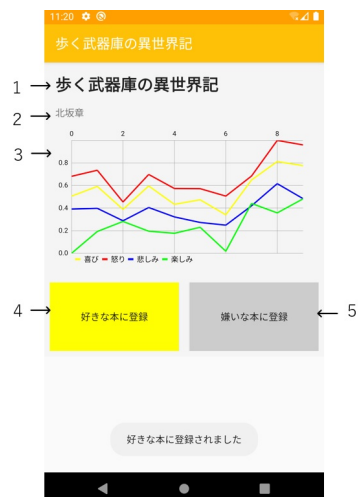


図 5.7 小説の詳細画面

1. タイトル
2. 作者名
3. 感情グラフ

感情グラフとは、ネット小説ごとに感情分析をした結果をグラフ化したものである。このグラフは縦軸に感情値、横軸に小説の進度を取り、「喜び」、「怒り」、「哀しみ」、「楽しみ」の4

感情それぞれの変化を表している。

4. 好きな本に登録するボタン
5. 嫌いな本に登録するボタン

(文責: 呉航平)

5.3 アプリの改善案

このアプリケーションの改善点や検討中の機能を以下に示す。

- コンテンツの充実

小説家になろうに掲載されている小説数は約 70 万件であるのに対し、このアプリに登録されている小説数は約 3600 件であり、このままではユーザのニーズに対応できないことが予想される。そのため、アプリ内の小説数を増やしてコンテンツを充実させる必要がある。

- チュートリアル作成

アプリの使い方の説明がないため、ユーザが困惑してしまうことが予想される。したがって、アプリ起動後にチュートリアル画面を作成する。

- 小説家になろうサイトとの連携

現在のアプリでは、気になったネット小説やあなたへのオススメに表示されているネット小説をすぐに読むことができない。これを解決するために、ネット小説の詳細画面に小説家になろうサイトのそのネット小説のページへ移動することができるボタンを追加する。

- 高度な検索機能の追加

現在のアプリは、キーワード検索にしか対応していない。ジャンルや作者ごとにネット小説を絞り込むカテゴリ検索を追加することで検索の効率と利便性を向上させる。

(文責: 呉航平)

第 6 章 外部評価

6.1 中間発表

6.1.1 中間発表会の準備

中間発表会の準備では、全体ポスターとグループポスター、全体スライドを作成した。全体ポスターと全体スライドはグループ A と一緒に作業分担し、グループポスターはグループ別で作業をした。ポスターとスライドの作成にあたり、ディープラーニングに関してやグループで何をするのかを初めて聞く人にも伝わりやすいように心掛けた。ポスターとスライドが出来れば、プロジェクトの担当教員に添削していただいた。担当教員からは気になる表現や伝わりにくい部分の指摘をいただいた。また、ポスターは簡潔にまとめ、スライドは見やすいように作成するようにとのアドバイスもいただいた。その後、ポスターとスライドの修正すべき部分の作業をした。また、スライドの発表に向けての原稿を作成し、制限時間内に発表できるようにリハーサルをした。リハーサルにあたって、時間や声を意識しながら発表を行った。

6.1.2 評価結果

プロジェクトの成果を聴衆にご覧いただくために中間発表会を行った。中間発表会は 2019 年 7 月 19 日 (金) に公立はこだて未来大学で行い、発表した内容を学生や教員に発表評価シートに記入していただいた。発表評価シートは、聴衆が発表技術と発表内容の項目を 1 点から 10 点で評価用紙である。また、それぞれの項目に評価したコメントの記述も可能である。コメントと評価の平均を以下に示す。ただし、評価の未回答は平均に関与しないものとする。

学生のコメント

- 質問に適切に答えていた
- スライドに統一感がない
- 感情分析が出来たら面白い
- 1 つの単語に複数の感情がある場合どうするのか
- 単語ごとで感情分析することが新しくて面白い
- 文字が重視されるべきスライドにもイラストがバックにあるのは微妙

教員のコメント

- 対象が興味深い
- 発表者の早口と資料を読むのは NG
- プレゼンが見やすかった
- 感情分析による小説の評価がユーザが求める小説とどのくらいマッチするかが疑問
- 技術的な詳細を述べてもらえると良かった
- 既存の技術とそれに対する新規性をもう一段説明してもらえると良かった

表 6.1 中間発表会の平均評価点数

評価	発表技術	発表内容
未回答	1	3
1	0	0
2	0	0
3	1	0
4	2	0
5	3	1
6	2	2
7	7	3
8	9	14
9	0	3
10	0	0
合計	25	25
平均	6.6	7.8

6.1.3 考察

発表技術の平均評価点数が 6.6 であり，発表内容の平均評価点数が 7.8 であった．発表技術の平均が低い原因は聴衆にとっては発表者の声が聞き取りにくい点が挙げられる．各ブースで発表が行われており，聴衆に声が通りにくい点のため発表の仕方を考えるべきであった．具体的には，発表者と聴衆を近づけて発表を行うことで声が聞き取りやすくなったと考えられる．また，早口で発表していた点も聞き取りにくい点であると考えられる．発表内容の平均が高かった理由は感情分析でネット小説をオススメすることが聴衆にとって興味深い点であると挙げられる．この内容は感情を基にオススメをすることが斬新なアイデアなためだと考えられる．他にも技術的なコメントがあり，これからの活動で参考になる．

（文責: 神野健瑠）

6.2 最終発表会

6.2.1 最終発表会の準備

最終発表会の準備では，全体ポスターとグループポスター，全体スライドとグループスライドを作成した．全体ポスターと全体スライドはプロジェクトやグループの背景に関して記述し，グループポスターとグループスライドはオススメ手法の詳細を記述した．最終発表会のポスターとスライドは聴衆に対して，伝わりやすく表現することを心掛けた．ポスターの完成版をメールで送り，スライドの完成版をリハーサルで，それぞれの添削をいただいた．担当教員から技術的なことを詳細に記述し，初めて聞く人にも伝わるようにとのアドバイスをいただいた．そのアドバイスを基にポスターとスライドの修正作業に取り組んだが，聴衆に伝わりやすく表現を考えることが困難であった．そのため，グループで残って伝わりやすい表現方法を議論した．

6.2.2 評価結果

評価に関しては中間発表会で行った評価方法と同様である。最終発表会の学生のコメント，教員のコメント，一般のコメントを以下に示す。また，学生と教員，一般の方から評価した平均も以下に示す。ただし，評価の未回答は平均に関与しないものとする。

学生のコメント

- 発表を分かれて同時に行っているので少し聞きづらく感じた
- 単語をベクトルにして感情との度合いを表すところが面白い
- 外部からユーザデータを集める計画があれば良かった
- 感情データからパターンを作成するのは面白い
- ユーザの評価だけでなく作品の感情データがおすすめに反映されるシステムは良い

教員のコメント

- スライドはわかりやすかったがモデルの詳細を知りたかった
- 結果の評価をきちんと行った方が良かったと思った
- 扱っている情報がほぼ記号的意味を持っているものなので，ニューラルネットワークを使う課題としては適しているのか疑問
- A と B グループの発表がぶつかり聞きにくい
- 自分たちのやっていることが理解できている

一般のコメント

- ネット小説からシステムをつくる発想が面白かった
- わかりやすくなるような順番のスライドで良かった
- プロジェクトを達成するまでの目標や手順がはっきりしていてわかりやすかった
- ネット小説以外の本もあればいい
- マイクが必要

表 6.2 中間発表会の平均評価点数

評価	発表技術	発表内容
未回答	1	2
1	0	0
2	0	1
3	0	2
4	2	1
5	1	1
6	12	0
7	13	16
8	18	20
9	4	6
10	4	9
合計	55	55
平均	7.3	7.5

コメントにはポジティブな意見や改善すべき点を挙げていただいた。この意見を基にどのように改善すれば良かったのかを考察する。

6.2.3 考察

発表技術の平均評価点数は 7.3 であり、発表内容の平均評価点数は 7.5 であった。発表技術の平均が中間発表会のとときと比較して良かった。その理由はスライドが見やすかったことであり、聴衆のコメントにも記述されている。しかし、グループ同士の発表のときに近すぎたために一部のコメントからは発表が被っていて聞き取りづらいと意見があり、発表の仕方を改善するべきであった。発表内容の平均は中間発表会のとときと比較して悪かった。その原因はアプリを利用して評価実験を行うことが出来なかった点であると推察される。また、聴衆にとってどのくらいの精度で自分に合ったオススメしてくれるのかが想像がつきにくいことも考えられる。今後の課題はユーザにアプリの評価実験を行うことでモデルの精度を向上していくことである。

(文責: 神野健瑠)

第 7 章 前期の振り返り

7.1 前期の各メンバーの役割と振り返り

- 安加賀元基

前期ではまず、勉強会で RNN (Recurrent Neural Network) や LSTM (Long short-term memory) について解説した。説明のために自分で理解することや説明方法を考えることが大変であった。しかし、メンバーにはある程度理解してもらえたと思う。

次に、感情分析の手法検討では、IBM の API である「Watson Tone Analyzer」の試行を担当した。API を使用することが初めてであったため、これに関連する Web サイトで調べながら試行した。最終的にこれは使用しないことになったが、個人的には API を利用する良い経験になったと思う。

次に、Android Studio と Kotlin の使い方を「作って楽しむプログラミング Android アプリ超入門」[5] を利用して勉強した。勉強の内容は、この書籍で紹介されている「おみくじアプリ」の作成である。このアプリケーションの作成で学べることは、アプリケーションの画面を視覚的に把握しながら Kotlin でコーディングしていくことである。Kotlin は Java の後継のような言語で、Java との完全な互換性を持つ。しかし、Java よりもコーディングがシンプルな点や null セーフである点に非常に使いやすい印象を受けた。このアプリケーションを作ったことによって、アプリケーションファイルの階層がどのようになっているのか知ることができた。さらに、Kotlin に関してもある程度は何も見ずにコーディングできるようになった。

その後、アプリケーションの画面遷移および UI の設計を行った。この際、フロントエンドのメンバーと一緒に機能を洗い出し、必要な UI を決め、画面遷移の階層構造を明確にした。これを行ったことによって、アプリケーションの機能が明確になった。タスクを決める際に、画面の作成者を決めるだけになったので、これを行っておいて非常に良かったと感じることができた。

前期を振り返ってみると、勉強や話し合いでほとんど終わったなという印象を受ける。勉強内容は新鮮なものばかりであったが淡白に感じた。さらに、話し合いもメンバー同士で意見が割れることが多く、なかなか決着しなかった。しかし、この活動が後期に大きく影響したのでやってよかったと思う。

(文責: 安加賀元基)

- 呉航平

前期では、まずディープラーニングの基礎的な手法や理論を学習するためにグループ全体で勉強会を行った。しかし、私はそれまでディープラーニングに対する知識がほとんどなかったため、理解に苦しんだ。そのため、勉強会以外でも参考書を使って自習をすることで、ディープラーニングに対する理解を深めた。

次に、テーマ決めを行った。アイデア出しの際には、予め調べてきた先行研究や情報をもとに意見を出した。最初のうちはなかなか発言機会が少なかったが、徐々に意見を出せるようになったと思う。しかし、結果的に何度もテーマを変更することになり、テーマ決めにか

なり多くの時間を割いてしまった。この原因は2つ考えられる。1つ目は、全体的に知識が不足していたことだ。一度決めたテーマ内容が現実的に実行することが難しいとわかったためテーマの再考を行うこととなってしまった。2つ目は、活発な議論が出来なかったことだ。これらの反省を生かして、次からは事前にテーマに関する確実な知識を付けた上で、積極的に話し合いに参加していきたい。

テーマ決め後は、私は Android アプリケーションを作成するフロントエンド担当となった。まず、Android アプリケーション開発は初めてであったため、技術書を使って Android Studio と kotlin の勉強を行った。授業時間外に多くの時間を割くことで、新規の技術習得に努めた。次に、画面遷移や UI の設計を行った。具体的には、アプリに必要な機能を、フロントエンドのメンバーで洗い出し、UI の検討を行った。その結果、アプリケーションのプロトタイプを、検索画面と検索結果画面と詳細画面を備えたものとすることにした。

また、本プロジェクトでは、コミュニケーションツールである Slack、ソースコードのバージョン管理システムである Git、Git のホスティングサービスである GitHub などを使用した。私はチーム開発は本プロジェクトが初めてであったため、チーム開発を効率的に進めるためのツールやサービスについての知識がなかった。Git と GitHub についてはグループ内で使用目的や使用方法についての勉強会を行うことで、使い方を理解しようと努めた。

前期では、テーマ決めやアイデア出しを通して、チーム開発におけるコミュニケーションの取り方を実践的に学ぶことができた、しかし、メンバー間による認識の相違が起きることが多かったため、抽象的な表現をつかわないことや情報共有をこまめに行うことが重要であると感じた。また、全体的に時間が足りなくなることが多かったため、スケジュール管理をしっかりと行い、効率的に作業を行うことや授業時間外の時間を活用することが必要である。

(文責: 呉航平)

- 横濱拓樹

前期では主に、テーマ決め人口知能やディープラーニングに関する勉強会、開発環境の設定、中間発表などを行った。

まずはテーマ決めに向けて、今の人工知能にできることを調べ案を出した。プロジェクト全体で意見を出し合い、一度大きなテーマの枠組みとして「擬似人格」としてグループに分かれた。その後グループに分かれそれぞれで勉強会を行った。

グループでディープラーニングについて勉強を行った際には、早めに決められた参考書を進め、担当以外の所も終わらせた。おかげで他の人が担当していた部分でも、足りない所を自分が補足しながら勉強会を進めていくことができた。そのため、全体で進行を合わせながら勉強を行え全体の進捗を把握しながらディープラーニングについて学ぶことができた。また私は、個人的にもディープラーニングの学習を行ったことで、基礎的な事をしっかりと抑えることができた。CNN や LSTM, RNN などの様々な種類の手法を一通り実装し試してみたり、それぞれの手法がどのような場面で有効なのかを抑えたりすることができ、後期にディープラーニングを実装する時に役に立つと思う。

次に、プロジェクトで用意されているパソコンの環境の整理を行った。ディープラーニングは計算にかなりの時間を使うため、ディープラーニングの計算に強い GPU を使うことにし、そのための準備を行った。また、使う OS も Ubuntu としたためその勉強もした [6]。普段は Windows を使っているため、UNIX 系の OS には慣れておらず大変だった。また、

GPU を使えるようにするためにドライバをインストールしたり、パスを通すなどの作業も大変だった。さらに、Python は沢山のバージョンがあり中には互換性がないものや、ライブラリのバージョンなども細かく気を付けなければいけないため、仮想環境も用意した。

そして、テーマについて再度話し合いを行い、最終的に進めていくテーマの決定をした。そしてテーマが現在の「感情分析による小説のオススメ」に決定した。このテーマ決めは思っていた以上に時間が掛かり決めるのにも苦労した。今後何かアイデア出しをしてテーマなどを決める際にはアイデアの出し方や、絞り方などの勉強をして臨みたい。確定した後は感情分析の手法やデータセットなどを調べ、実際に簡易的な感情分析を行ってみた。

最後に中間発表に向けてプロジェクトでやってきたこと、これからやっていくことをまとめ、スライド、ポスターの制作を行った。思っていた以上に先生方の指摘が多く改善の余地が沢山あることが分かり勉強になった。

(文責: 横濱拓樹)

- 金澤大地

本プロジェクトでは、最初にテーマ決めを行った。テーマごとにグループで分かれた後にディープラーニングについて、勉強会を行った。勉強会が終わった後に、テーマについて具体的に行っていくことを話し合ったが、テーマ内容が現実的に実行することが難しかったためテーマの再考を行った。十分な知識が事前になかったことによりテーマの再決定に時間がかかってしまった。このことから、事前に勉強したうえでテーマの決定を行うべきだったと考え、同じ機会があった場合に活かしたいと思う。

テーマ決定後にアプリケーション側を作成するフロントエンドを担当することになった。Kotlin や Android Studio はこれまで利用したことがなかったため、技術書 [7] [8]などを参考にこれらの学習から行った。しかし、初めて使う言語と開発環境だったため学習に時間がかかり、肝心のアプリケーション開発を行うまでに時間がかかってしまった。勉強の後に、画面遷移の検討と UI の検討を行った。画面遷移図をフロントエンドのメンバーで作成し、グループで評価し検討を行った。次に、UI の検討を行った。検討の結果、プロトタイプは検索画面と検索結果画面、詳細画面を作成することになった。

また、前期に Git と GitHub の勉強会を行った。Git などに詳しいメンバーがいたので、グループ全体で勉強会を行った。

本プロジェクトでは、Slack を使い情報共有を行っていた。しかし、Slack は初めて使うツールだったため、グループのメンバーへの反応が遅くなってしまうことがあった。後期になると操作に慣れこのことから、使用するツールを十分に利用できるように、事前に勉強することが必要だと感じた。

中間成果発表の際には、発表用原稿の推敲などを行った。発表会では前期までの作業内容と後期の目標を説明した。また、報告書の作成では一部の文章を作成し、文章の推敲を行った。しかし、文章作成に時間がかかり提出時間間際に提出していたため、最終提出では余裕をもって作業をしていく必要があると考えた。

(文責: 金澤大地)

- 小滝健太郎

まず最初に、ディープラーニングへの理解を深めるために参考書 [9] を用いて勉強会を行った。私はそれまでディープラーニングに対する知識が全くなかったため、非常に理解に

苦しんだ。参考書の一部の解説を担当する機会もあったが、自身の理解が足りずに満足はいく説明ができなかった。しかし、この勉強会では各メンバーの解説に加えて、理解をしたメンバーによる補足説明があった。そのため、自分では理解できなかったものでもその説明である程度理解ができたため、ディープラーニングの良い勉強となった。

その後にテーマ決めを行った。テーマ決めでは、なかなか意見がまとまらずに多くの時間を費やしてしまった。しかし、この機会にメンバーの意見を共有することができたため、非常に有意義な時間となった。ここでは、積極的に意見したり、意見することを促したりすることで円滑に議論できるように行動できた。

議論では、最終段階で私ともう一人のメンバーのテーマ案が残った。しかし、私はこのような場面での議論の進行役はそれまでほとんど経験したことがなかった。そのため、自分の意見を主張しすぎない点と、もう一方の意見の利点もしっかりと聞き公平に議論を進めることがとても難しかった。それらのことに注意して議論を進めたつもりではあったが、自分の案を推す場面も多かった。そのため、今後そのような機会があればより公平な議論ができるように注意したい。改善すべき点はあったが、議論を進行したりその中で意見したりする貴重な経験ができた。

(文責: 小滝健太郎)

● 寄田 慧

はじめに、本プロジェクトで行ったことはグループテーマを検討した。そのために、人工知能やディープラーニングについての基礎的な勉強を行った。私はそれまでディープラーニングに対する知識が全くなかったため、非常に理解に苦しんだ。勉強会の中でニューラルネットワークの解説役を受け持ったが、ニューラルネットワークの基本と実例を説明したがうまく説明できなかった部分が多々あった。

勉強会後は、自身でテーマに関わる自然言語処理についての調査を行い、それをもとにグループテーマを決定した。しかし、何度もテーマを変えることとなった。テーマ決めの際に自分の意見をうまく表現でできなかったことが悔やまれる。また、テーマを変更するたびに開発環境の整備を行う必要があったためかなりの時間を費やしてしまった。

次に、チーム開発を効率的に進めるためのサービスやツールに関する知識を深めた。私は本プロジェクトが始まるまで開発を効率的に進めるためのツールについての知識がなかった。本プロジェクトでは、コミュニケーションツールとして Slack やソースコードの管理システムとして Git や GitHub を使用した。Git, GitHub については勉強会を行うことで、使い方や利便性を理解することができた。しかし、実際に本プロジェクトでは Git, GitHub を有効活用することができなかった。

その後、中間成果発表会に向けた準備を行った。その中でも、私は発表用原稿の推敲を行った。発表会では背景、該当分野の現状や従来例、現状における問題点を説明した。しかし、発表会では発表する部分を細かく分けてたために発表に対する評価はあまり高くなかった。後期にある最終発表会では、誰か 1 人に全ての説明をさせるべきだと考えた。

発表会が終わった後は報告書の作成では一部の文章を作成した。しかし、文章作成に時間がかかり提出時間間際に提出していたため、十分に推敲を繰り返すことができなかった。また、文章作成の担当箇所をバラバラにしてしまったために章ごとの文のつながりがあまりない報告書が完成してしまった。これにより、さらに文章の推敲の時間が増えてしまった。

前期の反省のまとめとしては、プロジェクト学習が初めてだったためにテーマ決めの段階

から問題が起きてしまった。その影響が後の作業にも影響を及ぼしてしまった。また、コミュニケーションツールとして Slack を多用したが、ソースコードの管理システムとして Git や GitHub がほとんど使われなかったことも問題であった。前期で出てきた問題点を後期の活動に活かしていきたい。

(文責: 寄田慧)

- 神野健瑠

まず最初に勉強会を行った。内容はディープラーニングに関することで参考書を用いて説明した。その参考書の説明は各章で代表者がプロジェクトの時間に説明を行う。私は参考書の一部を解説する担当となったが、自身のディープラーニングの勉強不足が原因で伝わりやすい発表にはならなかった。改善点は相手に伝わりやすい表現をできなかったことと理解不足だったことだと感じた。

次にテーマ決めを行った。テーマについて話し合いを行い、アイデアが出たものの調査を行った。しかし、現実的ではないアイデアや時間がかかりすぎるアイデアが話し合いで見つかったため、テーマ決めの再考を行った。その結果、時間を予定より多く費やしてしまったので、残れるメンバーと一緒に夜遅くまでアイデアを出し合った。その後は再度グループで話し合い、2つの案が残り、どちらが良いか1人ずつ意見を言った。最終的に多数決で決まったが、皆が納得できる内容では無かった。この点の一つにテーマを絞ることはやりたいことを出来ない人もいるのでこの議論を通して、とても難しいと感じた。

テーマが決まると、グループでソースコードを共有・編集できるように GitHub を使用した。使用にあたり、初めて使う人がいたため勉強会を行い、理解を深めた。私も初めての GitHub だったため、グループメンバーに操作を聞きながら使用した。

また、進捗報告をできるように Slack というコミュニケーションツールを使用した。このツールも初めて使用したため円滑にグループでコミュニケーションを取るのに時間がかかった。

その後は、中間発表会に向けて準備を行った。私は完成したポスターとスライドのチェックをした。また、発表原稿の一部を作成した。ポスターとスライドを完成した後は、グループメンバーで発表を分担し、リハーサルを行った。リハーサルにあたってハッキリと喋り、制限時間内に発表できるようにグループで心掛けた。

中間発表会が終わった後は報告書の一部を作成した。しかし、誤字脱字や違和感のある文章が多く、グループメンバーに大きな迷惑をかけてしまった。最終的に文章を修正するのに時間がかかったため提出する際にギリギリになった。この原因は自分にあり、グループメンバーに負担をかけてしまったことが反省点である。

前期では、プロジェクトの担当教員がリードするのではなく自分たちで企画・開発した。この点は自分の印象とは大きく異なることだった。話し合いに関しても主に学生同士で意見を出し合った。しかし、話し合いは多くの時間を費やした。その結果、全体的に時間がたりなくなることが多かった。このことからスケジュール管理を行うことで計画的に時間を用いるべきだったと感じた。また、このプロジェクトにおいてディープラーニングや GitHub などを初めて学ぶことで、様々な知識を習得できた。

(文責: 神野健瑠)

7.2 前期全体の振り返り

前期は主に、ディープラーニングに関する基礎的な勉強とテーマ決め、使う手法とプラットフォームの検討、開発の準備を行った。

まず勉強では、1冊参考書を決め全体でそれを進めることで、足並みを揃えながらディープラーニングについて学習することができた。ただ、全体では基礎的なことしかやらなかったため、今後も各自で勉強していく必要があった。

全体での勉強のあとは、実際にプロジェクト学習で行うテーマを決定した。まず各自で、擬似人格について調査し、実現可能なテーマを新たに検討した。その結果、テーマは「感情分析によるネット小説のオススメ」となった。また、このシステムをより簡単に使えるようにするため、モバイルでのアプリケーション化を目指すことになった。テーマ決めに多くの時間を使ってしまい、この後の作業にも影響してしまった。なぜそんなに時間がかかってしまったのかを考え今後に生かしたい。この後、データの収集や整形、ディープラーニングのモデルの構築などを行うバックエンド班とアプリケーションのUIや機能の部分を作るフロントエンド班に分かれて作業を行った。

バックエンドではまず、システムを開発していくための環境を開発した。ディープラーニングにはGPUを使用するため誰もやったことのない環境であり苦労した。その後、各自で感情分析の手法を調べ、試行してみた。まずデータを取得するため、Webスクレイピングを用いて「小説家になろう」の文章データを取得した。また、試した手法は「Word2vec」と単語の辞書をベースとしたもの、IBMのAPIであるWatson Tone Analyzerの3つである。その中で、Watson Tone Analyzerは日本語に対応したものがなく、私達が求めたい感情とマッチするものも少ないことから、これを利用しないことにした。

フロントエンドでは、プロトタイプのUIや機能を考え、それに必要な技術の勉強や手法の調査・試行を行った。色々な開発環境も検討し、実際に環境をインストールしどれを使っていくか話合った。その結果Android StudioとKotlinを使用し開発していくことになった。

前期の問題点としてはフロントエンド、バックエンドともに技術力が足りないことが一番に挙げられる。そのため、各々の技術力の向上が必要であるので、フロントエンドはAndroid Studioを、バックエンドは感情分析のモデルの作成を勉強する。後期からはサービスの評価方法として、ユーザがオススメしたものを実際に読んだかを採用することにし、そのための準備を行う必要がある。また、オススメ手法の工夫としてユーザが1つ前に読んだ作品と最近読んだ作品のグラフの平均をとり、ユーザの好みに柔軟に対応することを考えているが、評価が良くなる別の方法も検討する。さらに、Word2vecと辞書ベースの2つのモデルを試し、使用するモデルを決定する必要もある。

夏季休暇の期間中は各自問題点で挙げた勉強と、Word2vecの学習、プロトタイプ版のフロントエンドの制作を始めた。Word2vecの学習を進めたことで実際に小説の本文を感情分析できるようになっていった。

(文責: 横濱拓樹)

第 8 章 後期の振り返り

8.1 後期の各メンバーの役割と振り返り

- 安加賀元基

後期にはまず、SQL データベースを構築した。SQL データベースを扱う知識は全くなく、使い方そのものもわからなかった。そのため、まず「ゼロからはじめるデータベース操作」[10]を利用して自主学習した。学習は「PostgreSQL」を使い、データベースの構築やテーブルの作成、クエリの実行など一通りの操作を学んだ。その後、この書籍と SQL に関する Web サイトを参考に実際に「PostgreSQL」を使ってデータベースを構築した。この際、バックエンドとどのようなスキーマにするのかを話し合い、スキーマ決定後データベースの正規化を行った。データベース構築後は、データの挿入を行った。約 5000 件のデータを挿入しなければならなかったため、JSON ファイルを読み込んで一度に挿入する工夫をした。JSON ファイルの作成は、スキーマをバックエンドのメンバーに指示し、バックエンドに任せた。

次に SQL データベースに保存したデータを Android アプリケーションで扱うために、サーバサイドのプログラムを書いた。このとき、サーバは私のパソコンとした。まず、データベースから指定したデータを取り出し、インターネットブラウザ上で表示されるようにした。次に、同じ LAN 内であれば、インターネットブラウザを使えるデバイスからアクセスできるようにした。しかし、この作業が終わったときに Realm Database の存在に気づいた。Realm Database とは、Android や iOS などのスマートフォンの OS で動作するアプリケーションに搭載できるデータベースである。このデータベースを利用するとサーバとデバイス間の通信が不要になる。さらに、SQL データベースと同様にデータの読み込み操作や書き込み操作が可能である。Realm Database を利用することにしたので、SQL データベースやサーバに関する技術は使わないことになった。しかし、Realm Database の操作は SQL データベースの操作に似ているので、これらの作業で得た経験は役に立ったと思う。個人的にも SQL データベースやサーバサイドなどを扱う技術面で大きく成長するきっかけになったので、このタスクを担当して良かったと思う。

次に、Realm Database に関するプログラムを Kotlin で書いた。Realm Database の使い方が全くわからなかったので、「初めての Android プログラミング」[8] や Realm Database の公式リファレンス [11] を利用して学習した。初めはメンバーと同時に取り組んでいたが、データベースからデータを取り出せるようになってから私一人で取り組んだ。最も反省すべき点は、Realm Database のテーブル作成を SQL データベースのテーブル作成と混同してしまったことである。SQL の場合テーブルを作るときにコマンドが必要であったが、Realm ではクラスとしてテーブルを作るため、コマンドが必要ない。つまり、Realm Database は SQL データベースとは DDL (Data Definition Language) に関する操作が根本的に違う。さらに、これ以外の DML (Data Manipulation Language) や DCL (Data Control Language) に関する操作はほとんど同じであるため、操作を混同してしまった。これで大幅に時間をロスしてしまったが、その後の操作は SQL データベースと同様であったため、作業はスムーズに進められた。個人的にはこのタスクを通して、Android アプリケーション

で使うデータベース技術がどのような仕組みになっているのかを学ぶことができた。モバイルデバイスの OS で動作するアプリケーションの開発は、なかなか体験することができないと思うので、非常に貴重な体験ができたと思う。

次に、アプリケーションのソースコードのリファクタリングを行った。メンバーと私が書いたソースコードの冗長な部分をなるべく簡潔に書き直した。具体的には、計算量を減らすことやメモリの使用を抑えることを意識してソースコードを書き直した。

次に、アプリケーションに CNN のモデルを組みこむことを試みた。最終発表会当日までメンバーが担当していたが、エラーが消えなかったので手伝う形で取り組んだ。エラーは、モデルのファイルの読み込みに失敗していることとモデルからの出力を得られないことであった。モデルのファイル読み込みはハードコーディングすることで解決した。また、出力が得られないことに関してはモデルへの入力形式を訂正することで解決した。しかし、モデルからの出力を得られた時点で最終発表会が始まってしまったため、モデルの組み込みは間に合わなかった。これに関してはあと 30 分で完成していたと思うので、非常に悔しかった。他にもアプリケーションのロゴを作成したり、発表会のポスターや報告書の文章を訂正したり、話し合いに積極的に参加したりした。アプリケーションのロゴ作成は「ロゴメーカー」というソフトで作成した。デザインを勉強したことがなかったため苦労したが、デザインを勉強するいい機会になったと思う。発表会のポスターや報告書の文章の訂正に関しては、私自身の教養の向上につながったと思う。話し合いでは、意見を積極的に出し、メンバーの意見をよく聞けたと思う。メンバーに指摘されることで、自分では気づくことができなかったことにたくさん気づけた。また、私がメンバーに指摘するときは、メンバーの意見をただ否定するだけではなく代替案を出すことを心がけた。

後期を振り返ってみると、開発作業にほとんどの時間を使っていることがわかる。開発作業は慣れていなかったのも、かなり非効率なことを行っていたと思う。今になってそう感じるが、この経験ができたからこそ今の自分があり、後期の活動が成功したと実感できる。これらを総合的に振り返ってみると、様々な経験ができたと思う。フロントエンドに所属しながら、バックエンドのタスクもできたことに非常に満足している。ただし、ディープラーニングのモデルや Python を使ったタスクにあまり関われなかったことが心残りである。最後に、このプロジェクト活動で得られた知識や経験を使って、今後も様々なことを学んでいきたいと思う。

(文責: 安加賀元基)

- 呉航平

後期では、前期に作成したプロトタイプをもとに本格的なアプリケーションの開発を開始した。そのためにまずフロントエンドにおける各人の役割を明確にした。その結果、私は基本的な UI や画面遷移の作成を担当することになった。次から、アプリケーション開発のために私が行ったことやその他の活動を具体的に述べる。

はじめに、プロトタイプの機能に加えて必要な機能の洗い出しと画面遷移の検討をした。その際に、既存の読書管理アプリなどを調査した。

次に、各画面の UI を作成した。UI 作成やデザインの経験はなかったため非常に苦労したが、なるべく使いやすくシンプルな UI を作成するように心がけた。

最後に、アプリケーションに CNN のモデルを組み込もうと試みた。モデルの組み込みにはモバイルアプリ向けに機械学習機能を組み込むことができる ML Kit を活用した。今回

アプリに組み込みたいモデルは ML Kit のデフォルトでは提供されていないため、独自のカスタムモデルを利用することとなった。そのため、参考文献やサンプルコードが少なく非常に苦労した。多くの時間をかけて、モデルの組み込みを試みたが、結果的に時間が足りなくなってしまう、最終発表会までにモデルの組み込みは間に合わなかった。

最終発表会に向けては、ポスターや発表原稿の修正を行い、最終発表会本番では、アプリケーションのデモンストレーションを担当した。しかし、アプリケーションの開発が思うように進まなかったことで、十分な発表練習が行えず、満足のいく発表ができなかった。

私は、本格的なアプリケーション開発は今回が初めてであったため、言語を学びながら開発方法を学ぶということになり、非常に苦労した。授業時間以外の時間を活用することでその遅れを取り戻そうとしたが、結果的にアプリケーションを完成させることができなかった。

また反省点として GitHub に精通していなかったため、GitHub を使わなかったことが挙げられる。そのため、ソースコードの共有に時間がかかってしまい、余計な時間がかかってしまった。これから日常的にこれらのツールを積極的に使うことで、今後のチーム開発の際には活用できるようにしたい。

さらに、前期の活動が思うように進まず、後期のタスクの比重が重くなってしまった。今後は、活動のはじめにタスク管理やスケジュール管理を徹底する必要があると感じた。

以上より、今回のプロジェクト学習では非常に多くの経験と学びを得ることができた。この経験や学びを今後の活動に活かしていきたい。

(文責: 呉航平)

● 横濱拓樹

私はこのプロジェクトを通して様々なことを学んだ。前期、後期と1年間を通して学んだことを挙げていく。

まず、チームでシステムを開発した経験がほとんど無かったため、チーム開発をどのように行っていくのかの検討が大変だった。これは、前期ではテーマ決めや方針ぎめ、後期では開発の進めかたやコードの共有に苦労した。全体の連絡には Slack を用い、コードなどのファイルの共有には GitHub や Google Drive を用いたがほとんどメンバー全員が今まで使っていたツールとは違っており、使いこなすまでに時間が掛かってしまった。それでも後期に入るころには、Slack や Google Drive は使いこなせるようになったが、GitHub はなかなか使いこなすのは大変だった。今後また新しいチームで開発を行うときには、始めの段階でしっかりと勉強会を行いたい。

また、テーマ決めや役割分担、タスクの振り分けなども大変だった。プロジェクトマネジメントなどの勉強もほとんどしたことが無かったため、どのようにタスクを分解し、それぞれのタスクがどれくらい時間が掛かるのかの見積もりを出すことが難しかった。今後は、プロジェクトマネジメントのようなプログラム以外の勉強もしていく必要があると感じた。

そして、ディープラーニングについて学ぶことができた。まず始めに全体で行った勉強会で基本的なことを勉強したうえで、後期では実際にそれを実際に作ることで前期で学んだことを応用出来るようになった。

バックエンドに分かれた後では、システムの開発を行っていった。そこでは主にネットワークを通してスクレイピング、データの整形を行った。夏季休暇中に Web ページのから必要な情報を入手したり、API を用いた、JSON データの扱いの学習をしたことで [12]、後期から始まった開発はスムーズに始められた。課題としては大量のデータを扱うことがこれ

まで無かったため、HOST側やネットワークのエラーに戸惑ってしまった。まだ大量のデータの扱いには慣れていないのでこれからも勉強していきたい。また、スクレイピングはとても時間が掛かるのだが、始めの方はプログラムに例外処理を入れていなかったため度々途中でプログラムが止まってしまうそのたびに状態を確認しなくてはならず余計に時間が掛かってしまった。後半からはプログラムに例外処理を入れることでほとんどプログラムが途中で止まることなくスクレイピングを終わらせることができた。今までは、状況によってプログラムが動かなくなるようなことは無く、例外処理は慣れていなかったが最終的にはしっかりと例外処理のあるプログラムを組むことができるようになった。

そして、中間成果発表や最終成果発表を通しポスター制作やスライド制作、それらを用いた発表の技術も身に着けることが出来た。今までも少しはポスターやスライドを使った発表をしたことがあったが、先生方のチェックを通すことで多くの改善点が見つかりそれらを修正しながら色々なことが学べた。分かりやすい日本語の表現、構成の仕方、図や式の使い方などを学ぶことができた。

これからの研究や開発などにもプロジェクトで学んだことを活かしていきたい。

(文責: 横濱拓樹)

- 金澤大地

後期の初めには、プロトタイプを完成させた。プロトタイプをもとにUIの検討を行った。検討の結果、プロトタイプの画面に加えて新たに、選出されたオススメの本を表示するホーム画面とユーザの好きな本と嫌いな本の確認ができるほか、ユーザ自身の読んできた本の感情データから算出したグラフを表示するマイページを追加することにした。

検討後、私はアプリケーションで使用するデータを保存しておくためにデータベースを組み込むこととなった。フロントエンドのメンバーがSQLデータベースを構築していたため、アプリケーション側から通信するための勉強を行った。しかし、Realm Databaseが私達のアプリケーションに適していたため、Realm Databaseについての勉強を行った。勉強後、フロントエンドのメンバーと共に構築を行っていたが、当初モデル側で作成する予定だったネット小説やユーザの感情グラフをアプリケーション側で作成することにしたため、こちらの開発を行うことになった。

私は、ネット小説やユーザの感情データをもとにグラフを作成し表示するコードを書いていた。これは、MPAndroidChartという機能を使い作成した。この機能を開発した本人のページ [13] を参考にグラフの作成を行った。しかし、慣れていない言語であったことや参考資料などが少なかったためグラフ作成を行うコードを完成させるまでに時間がかかってしまった。これは、事前に学習すべき範囲が狭かったと考え、今後の課題として改善していかなければならないと思う。

最終成果発表会では、前半に発表を行った。しかし、アプリケーションの開発が思うように進まず発表用原稿の推敲や発表練習を満足に行うことができなかった。その結果発表の際に内容を伝えることに苦労した。このことから、自分のタスクの管理が甘かったと考えた。

前期と後期を通して自分自身の時間管理が甘かった。夏季休暇中の作業内容が少なく後期の作業量が多くなっていたため、タスク管理を徹底して行う必要があると考え、今後の活動に活かしていきたいと思う。

(文責: 金澤大地)

● 小滝健太郎

夏季休業から後期にかけてはバックエンドの開発に取り掛かった。私はそこで、GitHub を用いたバックエンドのタスク管理と感情分析のプログラムを作成した。GitHub はそれまで使用したことがなかったため、今後役立つ良い経験ができた。感情分析には Word2vec を用いたが、それ以外はほとんど基礎的な Python のコーディングだった。web サイトを参考に感情分析を実装した。[14] ほとんどをサンプルコードなどを見ずにプログラミングしていたため、それまで少ししか使っていなかった Python の良い経験となった。結果として、その先のプログラミングが大きく捗ったり他のメンバーへのアドバイスもできた。

感情分析のプログラムが完成したあとは、ネット小説のスクレイピングを行った。今回はなろう API を用いて小説の情報をスクレイピングし、BeautifulSoup を用いて小説本文のスクレイピングを行った。BeautifulSoup は Python で提供されているライブラリの 1 つで、HTML のタグを指定するとその中のテキストを抜き出すことができるようになるライブラリである。本文のスクレイピングのコードは他のメンバーが書いたが、自分で使う時に行った改善でそのコードを解説した。このライブラリも Word2vec と同様に使ったことのないものだったため、良い経験となった。

スクレイピングと同時に、オススメに使う判別用モデルの作成を行った。ディープラーニングについての勉強は勉強会と Word2vec の使用しかしていないため、1 からモデルを作成することは容易ではなかった。入出力を何にするかを決定したあとは、Keras のホームページ [15] や Qiita など、色々なサイトを熟読した。その結果、CNN のモデルの構築を計画した。CNN がどのようなものかは最初に行った勉強会の時に学習していたが、活性化関数や損失関数の選定、データセットへのラベリングや完成したモデルの使用方法など、自分が知識として持っていないものが多かった。

CNN のモデルを構築するにあたり、様々な web サイトを参考にした。まず、勉強会でのわずかな知識と、実際にモデルを作成する手順を解説している web サイトを用いて、そのサイトに掲載されているモデルを作成した。[16] ここで作成したモデルは、判別用モデルと同じく 2 分類問題であったため、入力を判別用モデルに合う形に編集した。判別用モデルの入力は普通の画像を入力とする場合と比較するとかなり小さい。そのため、モデルの層を減らすだけでなく、畳み込み層の隠れ層も減らした。このように、参考にしたモデルを変更し調節していくことで判別用モデルを作成した。モデルの活性化関数や損失関数は、参考にしているサイトが同じ分類問題であるため同様のものを用いた。

また、モデルの構築が進んできたタイミングでデータセットを作成し、モデルの学習をした。データセットには小説家になろうに登録しているユーザの情報を使用した。小説家になろうではブックマーク機能や小説に対する評価点を付ける機能がある。ここでは評価点を付ける機能に着目し、ユーザごとに、データベースに登録したネット小説の評価点をスクレイピングし、そのデータを整形することでデータセットとした。最初の精度は 60% 程度であった。

その後もディープラーニングについての学習を進めていくうちに、グリッドサーチを知った。早速、調べながらグリッドサーチをするためのプログラムを作成し、現状のモデルでの精度を最大限引き出した。グリッドサーチの結果、精度が 70% まで向上した。しかし、損失が大きいまま精度が向上したことで、実際に使う時の精度は完成度の良いものとは言えなかった。原因はデータセットにもあるがモデルの構築も CNN ではないものを試してみるべきだとわかった。

モデル構築の一連の流れを経験できたことは非常に将来役立つものとなった。結果的には成功とは言えないが、この失敗を良い参考としてこのモデルの精度をより向上させたり、他のモデルの構築をする機会に活用したい。

(文責: 小滝健太郎)

- 寄田 慧

夏期休業から後期にかけては、私は主に Python を使用したネット小説をスクレイピングするプログラムの作成とその完成したプログラムを用いてネット小説のスクレイピングを担当した。その中でも、プログラムの作成では Ncode のリストを作成して、それを順次スクレイピングし、編集本文は小説情報がまとまった JSON ファイルの一番後ろに追加する部分を担当した。このプログラムの作成には Google Colaboratory を使用した。なぜかというところこのプログラムを制作する際には構築が不要であり、GPU を無料でアクセスすることができるため使い勝手がとても良いからだ。このプログラムの仕組みは、Ncode から作品本文ページの URL を見つけ出し、そこから作品本文をスクレイピングしてその本文情報を JSON ファイルの名前で保存するものだった。このプログラムを制作しての反省は、今まで Python をあまり使ったことがなかったのでプログラムの一部を制作するのにもグループメンバーの力を借りる必要があった。これにより、プログラム完成まで時間がかかってしまいプロジェクトメンバーに迷惑をかけてしまった。

ネット小説をスクレイピングするプログラムの完成後、ネット小説のファンタジーのジャンルをスクレイピングした。スクレイピングするにあたり使用した環境は Jupyter Notebook である。なぜ以前と同じように Google Colaboratory を使用しなかったかというと、Google Colaboratory では時間が経過するとスクレイピングしたファイルが消去されてしまう。ネット小説をスクレイピングすることは時間がかかるために使用には不適切だと考えたからだ。しかし、開発環境の問題からプログラムがうまく動かない問題があり、ネット小説をスクレイピングに遅れてしまった。

スクレイピングが終了した後、折れ線グラフをスムージングして曲線にするプログラムの作成に移った。スムージングとは、文字や図形を表示したり、印刷したりする際に、なるべく綺麗に見せるための手法のことである。このプログラムを制作するにあたり使用したのが Google Colaboratory を使用した。このプログラムの作成はスムーズにいき、数字を入力すればグラフを表示で来たがほかのタスクに変更された。

この後にネット小説のその他のジャンルのスクレイピングを行うことになった。この時は、スムーズにスクレイピングを行うことができた。

これらのタスクが終了した後には私はポスター制作を行った。これは普段から使い慣れている「Microsoft PowerPoint」での作業となった。しかし、ポスター制作の際のテンプレートに対する無知が露わになり、ほかのメンバーに頼ったことを非常に後悔している。しかし、完成したポスターは自身の納得するものになった。

ポスター制作が終了した後には、私は最終発表会のスライド作成に加わった。その中でもスライドの原稿作りに注力したが、スライドの作成が遅れてしまったために発表練習を行えなかったことが心残りである。

最終発表会では、スライドの発表を担当した。声を大きく出すことができたが発表会ということで緊張していたのか、スライドを説明するスピードが想定よりも早くなってしまった。この原因としては、前日までスライド制作や原稿作成を行っていたために、発表練習を

全くすることができなかったことが考えられる。

本プロジェクトでは、自身の力不足を感じるが多かった。これらの反省を生かして今後の研究や開発に役立てたい。

(文責: 寄田慧)

- 神野健瑠

後期では、バックエンドの作業に取り組むにあたり、ネット小説の本文をスクレイピングした。スクレイピングは実際の記事を参考にプログラミングをした。プログラムの完成後はジャンルごとにグループメンバーと分担でスクレイピングを行った。私は2つのジャンルをスクレイピングしたので多くの時間を費やした。この原因は自身のスクレイピングをする際、効率的に時間を使えなかったことが反省点である。

その後、スクレイピングしたファイルを Google Drive を用いてグループメンバーで共有した。一度 GitHub で共有しようとしたがデータの容量が大きかったために Google Drive での編集となった。結果的に Google Drive でのコードの共有がメインとなってしまったために GitHub を使う頻度が減った。GitHub を使いこなせなかった点は悔いが残った。

次に、Python を用いてプログラミングを行った。Python に関しては初めて使った言語のため、自身のわからない部分はグループメンバーに協力を必要とした。このことから、自身の知識と勉強不足であり、事前に Python の勉強をするべきだったと感じた。

また、グループメンバーから振り分けられたタスクの調査も行った。主に Python に関する情報を web で検索し、参考にした。

スクレイピングやプログラミングを終えると次は最終発表会に向けてスライドの作成を行った。スライド作成では、初めて聞く人に伝わりやすく表現することを意識した。そのため、スライドが長くなりリハーサルの際に制限時間内での発表が問題であった。だから、グループメンバーの協力を得て、スライドの流れや表現を改善することで完成したが、自身の力不足を感じた。しかし、スライドをまとめ、担当教員に添削を受けて自身の気づかない点が多くあった。このことはこれからスライドを作成するうえで、とてもためになる内容だった。

最終発表会では、グループスライドの発表を行った。スライドをグループ別で発表する際に発表するスペースが狭かったために声が被ってしまった。聴衆者からのフィードバックにも、声が被っていて聞き取り辛かったと意見があったことから事前に発表を工夫するべきだったと感じた。

最後に、このプロジェクトで学んだグループ活動やコミュニケーションなどを今後の活動に活かしたい。

(文責: 神野健瑠)

8.2 後期全体の振り返り

後期に入ってからには本格的に開発にとりかかった。

バックエンドは、「なろう API」を使いながら小説本文のスクレイピングを始めた。スクレイピングをするためにデータの整形をしたり、どの情報が必要なのかといった情報の取捨選択をしたりすることに時間がかかった。また、スクレイピングをしている時にネットワー

クが悪かったり、サーバー側の問題でエラーが起こったりなど、今まで経験したこの無い問題が発生し大変だった。時間はかかったが欲しい情報を整理し、取得することができた。次に、それらのデータと Ward2vec を用い小説の感情分析をおこなった。そして、感情分析をしたデータを使いユーザに小説をオススメするモデルの検討、開発を行った。そしてオススメは、ユーザが評価した小説からユーザの感情データを定義し、ユーザの感情データと小説の感情データを比較することで、オススメすることとした。最後に、開発したモデルを評価する方法の検討し実施した。評価方法は小説家になろうのサイト内にある各ユーザが評価した小説のデータを利用した。

フロントエンドでは、後期に入ってすぐに夏季休暇から作り始めていたプロトタイプを完成させ、UI の検討を行った。プロトタイプから UI に関しての改良点を挙げ、それに沿って UI の改良を行った。次にバックエンドが分析したデータなどを保存、読み込みができるようにデータベースの実装を行った。本プロジェクトでは Realm Database を使用した。Realm Database を使用することでオフライン上でもアクセスすることができ、より高速に結果を得ることができるようになった。この Realm Database は今まで扱ってきた MySQL のようなデータベースとは仕様が異なるため学習が大変だったが、しっかりと実装することができた。最後に、バックエンドで開発したオススメのモデルをアプリケーションへ組み込んだ。

最終成果発表会ではエミュレータを使い実際のアプリの流れを紹介することができた。実際の発表も中間発表の時よりもいい発表ができた。しかし、課題もありバックエンドの進行の問題で、最終成果発表までにバックエンドで開発したオススメのモデルを組み込むことはできなかった。

(文責: 横濱拓樹)

8.3 今後の展望

今後の展望としては、オススメのモデルを最後まで作り上げアプリケーションに組み込むこと、オススメのシステムの評価実験を細かく行い改良を行うことが挙げられる。

オススメのシステムの改良としては、ニューラルネットワークの層の数やネットワークの種類など、試していないものが沢山あるので、さらにいくつかのモデルを作り試してみることが必要である。そして、そのモデルをアプリケーションに組み込みアプリケーションの評価も行う必要もある。

そして、まだ成果発表の場が残っているので、そこに向けてこれらの課題に取り組んでいきたい。

(文責: 横濱拓樹)

第 9 章 まとめ

まず、初めの私達のグループのテーマを「擬似人格」と定め、グループ全体でディープラーニングについて勉強を行った。共通の参考書 [9] を用いて勉強を進めたため、全体で進捗を合わせながらディープラーニングについて学ぶことができた。その後、テーマを「感情分析によるネット小説のオススメ」に変更し、アプリケーションのフロントエンド側を作成するフロントエンド班とバックエンド班に分かれた。

フロントエンド班では、開発していくプラットフォームの決定のために実際に環境をインストールするなどし、最終的に Android Studio に決定した。その後、アプリケーション開発の勉強を勉強をし、前期までにプロトタイプの画面遷移図の作成、検討を行った。そして、夏季休暇中にプロトタイプの作成を進めた。

バックエンド班では、感情分析について調べ手法の検討を行った。また同時に、Web スクレイピングについても調べ試しにネット小説の文章を少し取得してみた。その後、感情分析の手法を「Word2vec」に決定し、夏季休暇中に、ディープラーニングを実装するための勉強を行った。

夏季休暇中にフロントエンド班、バックエンド班それぞれ基本的な勉強を終わらせ、後期に入り本格的にシステム、アプリケーションの開発を行った。

フロントエンド班はまず、UI の検討、作成を行った。夏季休暇中に作っていたプロトタイプからの改良点を挙げ、使いやすい UI の検討を行った。次に、データベースの実装を行った。感情分析の結果をアプリケーションで使用するために、どのデータベースのを使用するかを検討し最終的に Realm Database でデータベースの実装を行った。最後に、バックエンド班が作成した小説をオススメするためのモデルをアプリケーションに組み込んだ。

バックエンド班はまず、データの収集を行った。Word2vec のネットワークの学習を進めるために Wikipedia の単語を収集し、実際に感情分析をするネット小説のデータも収集した。その後集めた小説を Word2vec で分析し、小説データを作成した。最後に作成した小説データを使って小説をオススメするためのモデルを作成した。

通年の活動を通して、ディープラーニングの知識やアプリケーション開発の知識を身に付けることができた。また、その知識を活かし 1 つのアプリケーションを作りきることができた。

(文責: 横濱拓樹)

参考文献

- [1] 小説家になろう [online]<https://syosetu.com>
- [2] Qiita[online]<https://qiita.com/tags/python>
- [3] gensim,「models.word2vec-Word2vec embeddings」,[online]<https://radimrehurek.com/gensim/models/word2vec.html#gensim.models.word2vec.Word2Vec>2020 年 1 月 8 日アクセス
- [4] Qiita,「Word2Vec の使い方」,[online]<https://qiita.com/kenta1984/items/93b64768494f971edf86> 2020 年 1 月 8 日アクセス
- [5] 高江 賢. 作って楽しむプログラミング Android アプリ超入門 Android Studio 3.3 & Kotlin 1.3 で学ぶはじめてのスマホアプリ作成. 日経 BP 社. 2019
- [6] Ubuntu スタートアップバイブル小林準著. – マイナビ出版, 2018.
- [7] 齋藤 新三. 基礎&応用力をしっかりと育成! Android アプリ開発の教科書 Kotlin 対応 なんちゃって開発者にならないための実践ハンズオン. 翔泳社. 2019
- [8] 金田 浩明. 初めての Android プログラミング 第 4 版. SB クリエイティブ. 2019
- [9] Francois Chollet. Python と Keras によるディープラーニング. マイナビ出版. 2018
- [10] ゼロからはじめるデータベース操作 ミック. SQL 第 2 版ゼロからはじめるデータベース操作. 翔泳社. 2019
- [11] Realm Database の公式リファレンス <https://realm.io/support> 2020 年 1 月 10 日アクセス
- [12] Python ではじめるデータラングリング: データの入手, 準備, 分析, プレゼンテーション Jacqueline Kazil, Katharine Jarmul 著; 長尾高弘訳. – オライリー・ジャパン, 2017
- [13] [online]<https://github.com/PhilJay/MPAndroidChart>2020 年 1 月 10 日アクセス
- [14] Qiita,「word2vec で 吉 川 英 治 本 の 感 情 分 析 を し て み た 」,[online]<https://qiita.com/TakahiroYamamoto/items/8efaae49ea4ff4f7eeec>2020 年 1 月 10 日アクセス
- [15] Keras, [online]<https://keras.io>, 2020 年 1 月 15 日アクセス
- [16] Qiita,「Python と Keras で 画 像 認 識 CNN 構 築 」, [online]<https://qiita.com/mainvoidllll/items/db991dc30d3ddced6250>, 2020 年 1 月 15 日アクセス