公立はこだて未来大学 2022 年度 システム情報科学実習
グループ報告書

Future University Hakodate 2022 Systems Information Science Practice
Group Report

プロジェクト名

未来へつなぐ新聞ビッグデータ

Project Name

Newspaper Big Data for the Future

プロジェクト番号/Project No.

19

プロジェクトリーダ/Project Leader

前田祥　Akira Maeda

グループリーダ/Group Leader

前田祥　Akira Maeda

グループメンバ/Group Member

遠藤晴人　Haruto Endo
川平覚士　Satoshi Kawahira
柴田公季　Kouki Shibata
高橋陽一　Yoichi Takahashi
辰己尚矢　Naoya Tatsumi
一入悠貴　Yuki Hitoshio
藤島海陸　Kairi Fujishima

指導教員

寺沢憲吾　美馬のゆり　角康之　坂井田瑠衣

Advisor

Kengo Terasawa　Noyuri Mima　Yasuyuki Sumi　Rui Sakaida

提出日

2023 年 1 月 18 日

Date of Submission

January 18, 2023

概要

本プロジェクトは、北海道新聞社から提供を受けた新聞ビッグデータを利用して新しい「何か」を生み出すことを目的としている。今年度は「何か」を「知的好奇心を刺激する Web アプリケーション」と設定し新聞のテキストを可視化し情報を提供する「View Picks」という Web アプリケーションを開発した。現在のマスコミュニケーションの手段として新聞以外に SNS やインターネットが挙げられる。しかし、これらのマスコミュニケーションの手段は、利用するユーザーに最適化された情報を多く提供するため、享受する知識が隔たってしまうという問題点がある。しかし、新聞やテレビ、ラジオといったマスコミュニケーションの手段では、街角の事件・事故やゴシップ、風俗、風潮、広告、連載漫画、小説など幅広いコンテンツの知識を享受することができる。その中でも新聞は、ジャンルを問わず自ずと求めていない知識を享受することが可能である。この新聞の特性に着目し、新聞に利用された言葉とインタラクティブに触れる体験を行うことが可能な Web アプリケーションの利用を通し、歴史の足跡を辿る行為で知的好奇心を刺激する。利用を通して、可視化された内容を読み取り気に留まった言葉を検索し新聞に記載された出来事へと繋げ新聞に対する興味関心を向上させることが狙いである。

キーワード　ビッグデータ, 自然言語処理, マスメディア

（※文責: 前田祥）

Abstract

The aim of this project is to create a new 'something' using the newspaper big data provided by the Hokkaido Shimbun. This year, the project set 'something' as 'a web application that stimulates intellectual curiosity' and developed a web application called 'View Picks', which visualises newspaper text and provides information. In addition to newspapers, SNS and the internet are currently used as means of mass communication. However, these means of mass communication have the problem that they provide a lot of information that is optimised for the user, so that the knowledge enjoyed by the user is separated. However, mass communication means such as newspapers, television and radio allow users to enjoy knowledge of a wide range of content, including street incidents and accidents, gossip, customs, trends, advertisements, serialised cartoons and novels. Among these, newspapers can enjoy knowledge that they do not naturally seek, regardless of genre. Focusing on this characteristic of newspapers, intellectual curiosity is stimulated by the act of tracing historical footsteps through the use of a web application that enables the user to interactively experience the words used in the newspaper. The aim is to improve interest in newspapers by reading the visualised contents, searching for words that catch the eye and connecting them to the events described in the newspaper.

Keyword Big Data, Natural Language Processing, Mass Media

（※文責: 前田祥）

目次

第 1 章　はじめに　1
1.1　本プロジェクトについて 1
1.2　本プロジェクトの目的 1

第 2 章　新聞の現状　2
2.1　新聞の社会的位置 2
2.2　新聞の利点 2
2.3　新聞の課題 3
2.3.1　問題点 3
2.3.2　改善すべき点 4

第 3 章　活動内容　6
3.1　前期 6
3.1.1　成果物案の提案 6
3.1.2　成果物案決定のためのグループワーク 6
3.1.3　成果物案の決定 7
3.1.4　成果物案の機能の案や必要技術の追求 7
3.1.5　中間発表へ向けての準備 8
3.1.6　中間発表会 8
3.1.7　中間発表会の振り返りと夏季休業中の活動予定の決定 8
3.2　後期 9
3.2.1　夏季休業中の活動の振り返り 9
3.2.2　成果物作成へ向けてのグループ作成 9
3.2.3　グループごとでの成果物作成 10
3.2.4　アプリケーションのプロトタイプ完成 10
3.2.5　最終発表に向けての準備 10
3.2.6　最終発表会 11
3.2.7　最終発表会の振り返りとアプリケーションの改良 11

第 4 章　開発　12
4.1　開発目的・目標 12
4.2　習得した技術・知識 12
4.2.1　自然言語処理 12
4.2.2　データの可視化: plotly 13
4.2.3　データの可視化: ワードクラウド 13
4.2.4　紙面データの補正 14
4.2.5　labelImg 15
4.2.6　YOLO 16

4.2.7 tf-idf 18
4.3 開発過程 19
4.3.1 データエンジニアリング班の開発過程 19
4.3.2 コンテンツ抽出班の開発過程 26
4.3.3 フロントエンド班の開発過程 29
4.3.4 バックエンド班の開発過程 30

第 5 章 成果 **34**
5.1 成果物の概要 34
5.2 成果物の各機能・目的 35
5.2.1 新聞記事データの様々な方法での可視化 35
5.2.2 その他機能について 38

第 6 章 まとめ **39**
6.1 目的達成度 39
6.2 振り返り 40
6.2.1 前田祥の振り返り 40
6.2.2 辰己尚矢の振り返り 41
6.2.3 藤島海陸の振り返り 44
6.2.4 遠藤晴人の振り返り 46
6.2.5 川平覚士の振り返り 48
6.2.6 一入悠貴の振り返り 49
6.2.7 高橋陽一の振り返り 52
6.2.8 柴田公季の振り返り 54
6.3 今後の課題 56
6.3.1 前期終了時点での課題 56
6.3.2 プロジェクト終了時点での課題 56
6.3.3 展望 57

謝辞 **58**

参考文献 **59**

第 1 章　はじめに

1.1　本プロジェクトについて

本プロジェクトは北海道新聞社のご協力を得て実現したプロジェクトである。北海道新聞のテキストデータ約 32 年分、北海道新聞の紙面画像データ約 88 年分をご提供頂いた。テキストデータには記事 4,498,758 件が含まれている。これらのデータを活用して新しい「何か」を生み出すことが本プロジェクトの大きな指針である。今年度は、新聞のビッグデータを可視化する Web アプリケーションを開発した。

（※文責: 前田祥）

1.2　本プロジェクトの目的

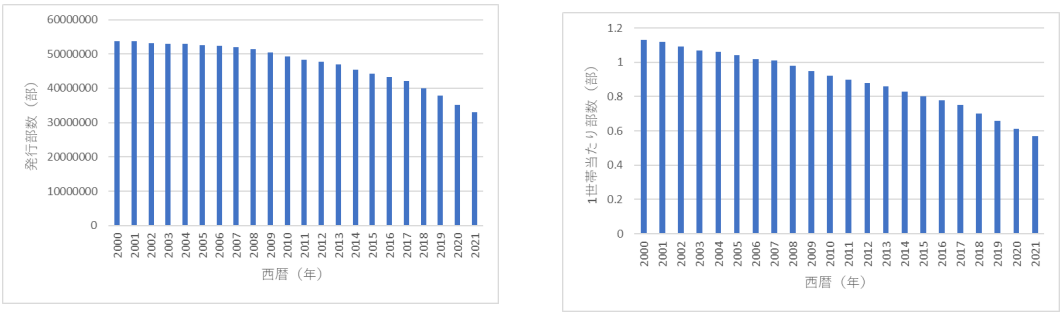
本プロジェクトの目的は、「新聞ビッグデータで知的好奇心を刺激する Web アプリケーションを生み出すこと」である。新聞には数あるメディアの中でも話題の多様性や信頼性、一覧性など多くの利点がある。それにもかかわらず近年は様々なニュースメディアが提供されていることもあり、購読者数は減少している。そこで本プロジェクトは、北海道新聞社から提供を受けた新聞ビッグデータを可視化する Web アプリケーションを開発した。ユーザーは Web アプリケーションの利用を通して多様な種類の情報に触れることができ知的好奇心が刺激され可視化されたデータについてより深く知ろうとすることになる。一連の流れを通して新聞に対する興味関心が向上することが期待される。

（※文責: 前田祥）

第 2 章　新聞の現状

2.1　新聞の社会的位置

4 大マスメディアのうちの一つである新聞は、古くから様々な人に利用されてきており、幅広い分野に関する情報を受け取ることができる情報源として、我々の生活に欠かせないものであった。しかし、インターネットの台頭や普及により、新聞の役割やそれらを取り巻く環境が大きく変化している。日本新聞協会 [1] によると、図 2.1a のように、2000 年における一般紙とスポーツ紙を合わせた発行部数が約 5400 万部であったのに対し、2021 年は約 3300 万部と、およそ 2100 万部減少していることが分かった。また、図 2.1b にあるように、1 世帯当たりの発行部数に着目すると、2000 年は 1.13 部であったのに対し、2022 年は 0.57 部と、約半数まで減少したことが分かる。このように、年々新聞の利用者が減少していることが分かる。その一方で、インターネットや SNS の勢いはとどまることがないため、新聞はマスメディアとしての立ち位置が脅かされるような状況にまで陥っている。



(a) 新聞の発行部数

(b) 新聞の 1 世帯あたり部数

図 2.1: 新聞の発行部数及び新聞の 1 世帯当たり部数

（※文責: 藤島海陸）

2.2　新聞の利点

新聞の利点として、次の 2 つを挙げる。一つ目の利点は、一覧性が高いことである。新聞は話題性の大小によって記事や見出しの大きさが異なっているため、見出し、もしくはリード文を読んでいくだけで、その日の重要なニュースや、記事の大まかな内容をつかみ取ることができる。それにより、テレビやインターネットで情報を受け取るよりも、比較的短い時間で社会全体の流れを俯瞰することができる。そして、新聞記事全体を一通り眺めていく中で、自分が今まで興味のなかった分野の記事に出会うことができるため、新たな知見を得たり、自分の興味をさらに引き出すきっかけにもなる。また、記事の本文については、実際に記者が現地に赴き、長い時間と手間をかけて取材をしているため、他のメディアでは真似できないほど内容の濃い記事を提供している。そのた

め、本文までじっくり見ていくことで、より深い部分の情報を得ることもできる。二つ目は、信頼性が高いことである。総務省の令和 3 年度版情報通信白書 [2] では、各メディアに対する信頼性について、図 2.2 のような調査結果が出ている。「信頼できる」と回答した人の割合は、新聞の 61.2 % が最も多く、次いで、テレビ（53.8 %）、ラジオ（50.9 %）の順に多くなっていた。この結果から、新聞は最も信頼されているメディアだということが言える。一方、本プロジェクトのメンバーのような若い世代が日常的に利用する SNS[3] や、その他インターネットを利用したメディアの信頼性については以下のような結果になった。SNS（15.3 %）、動画投稿・共有サイト（14.4 %）、ブログなどその他サイト（10.3 %）、掲示板やフォーラム（7.3 %）となっており、上記で挙げた新聞・テレビ・ラジオよりも信頼性が低いことが分かった。

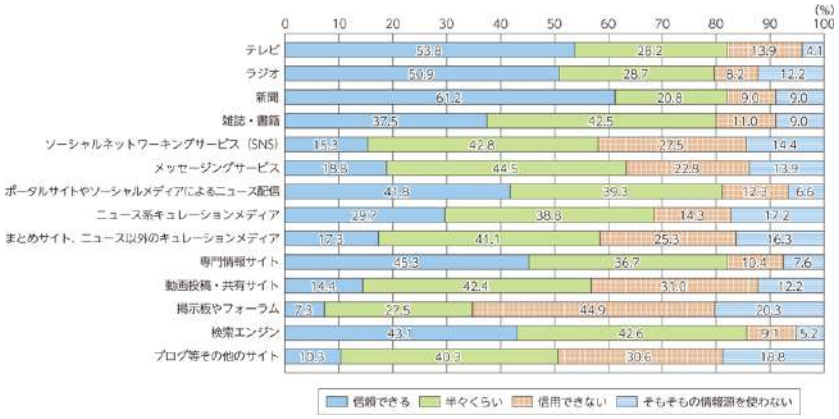


図 2.2: 各メディアに対する信頼性

（※文責: 藤島海陸）

2.3 新聞の課題

2.3.1 問題点

2.2 で新聞の利点について述べたが、この節では新聞の課題について次の 2 つを取り上げる。一つ目は、自分が見たい情報だけを効率よく得ることができないということである。SNS やインターネットの利用者は、困りごとを解決したり、自分が欲しいものを購入するため、さらには、興味のあるコンテンツを閲覧することで時間をつぶしたいなどといった理由から検索をする。検索を行うと、すぐに自分が求めている情報が大量に表示されるため、利用者は時間をかけずに自分が見たい情報だけ得ることができる。一方、新聞は、総合、政治、スポーツといった様々なカテゴリーが一つになっているため、自分にとって興味のない内容の記事も自然と目に入ってきてしまう。さらに、新聞自体に検索するような機能もないため、大量の情報の中から自分が欲しい情報だけを得るということが困難である。これらは確かに、自分が見たい情報だけを効率よく得るといった点では課題となってくる。しかしこれは、2.2 でも述べられているように、一覧性が高いという新聞の良さを生かすことで、今まで興味のなかった分野の記事に出会うことができたり、自分が知らなかった情報を新たに得ることができるともいえる。そのため、新聞は使い方によっては問題となりうるが、自分の目的や用途に合った使い方をすることで、新聞を読むことによる恩恵を最大限に受けることができると考えた。二つ目は、情報拡散能力が低いということである。テレビや SNS では、何か大きな事故が起きると、‘速報’ という形ですぐに情報が拡散されるため、時間をかけることな

く最新のニュースを取得することができる。また、SNS においても、その情報が正しいものなのかそうでないものなのかはさておき、一度流れた情報は一瞬で全世界に拡散され、多くの人の目に触れる。しかし、新聞の場合、その日起こったニュースを当日中に発行することは少ないため、多くの場合、次の日以降に情報を得ることになる。そのため、速報性が重要な情報はインターネットやテレビで受け取ってしまえば、わざわざ次の日以降に新聞を確認しなくても特に問題がないといった状況が出来上がってしまう。令和 3 年度版情報通信白書 [2] では、情報を得る目的ごとに「よく利用するメディア」について調査されていた。図 2.3 を見ると、「いち早く世の中のできごとや動きを知るメディア」については、テレビが 55.3 %と最も多く、続いてニュース配信（40.7 %）、検索エンジン（32.7 %）の順で多くなっていた。この結果から、22.9 %の割合であった新聞は、いち早く世の中のできごとや動きを知るメディアとしてはあまり利用されていないことが分かる。

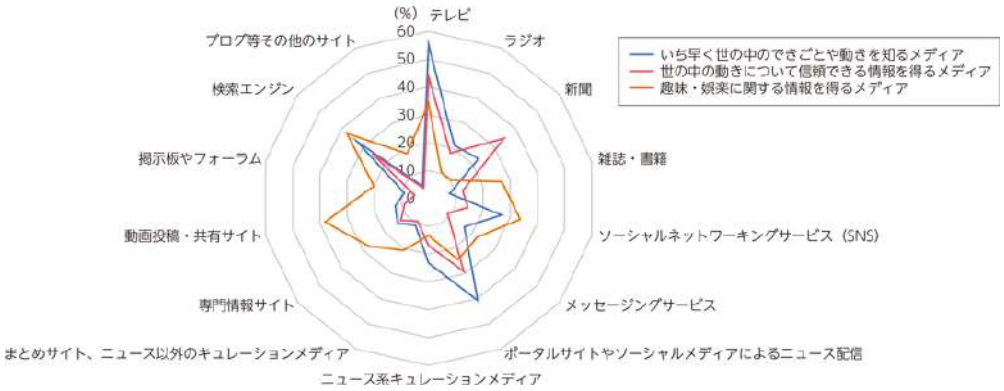


図 2.3: よく利用するメディア

（※文責: 藤島海陸）

2.3.2 改善すべき点

2.3.1 では、自分が見たい情報だけを効率よく得ることができないということ、そして、情報の速報性に乏しいという新聞に対する 2 つの課題を例として挙げた。新聞は、大量の情報があるゆえに自分が求めている情報だけを効率よく得ることができない一方、そのメリットである一覧性を生かすことで、興味のなかった分野の記事や自分が知らなかった情報を新たに得ることができるという改善策を述べた。そこで今回は、この改善策に対して、より具体的な取り組みを考えることで新聞の新しい使い方を提案する。具体的な取り組みとしては、一覧性が高い新聞と、情報拡散能力に優れた SNS を連携し、様々な形で可視化した新聞の情報を SNS のユーザーに対して提供するということである。新聞記事内に含まれるテキストデータや、広告などといった大量の情報が一目で分かりやすく可視化されていることにより、自分があまり興味のない情報でも受け取りやすいという利点がある。また、ユーザーにただ新聞記事を読んでもらうのではなく、様々な方法での可視化を通すことで、新聞に眠っている情報の価値や面白さを理解してもらうことができる。実際に読売新聞 [4] では、2017 年度から SNS と新聞を掛け合わせた「よみバズ」という取り組みをスタートしている。これは、読売新聞に掲載された広告がツイッター上で拡散している状況を計測し、どれだけの人に届いたかを推定することで、広告の効果を見える化するようなサービスである。また、2020 年度にはこれを進化させ、新聞広告に関するツイートを、最適化したターゲットへ絞って配信する「よみバズブースト」を開発した。この「よみバズブースト」には、広告効果を最大限引き出す目的がある。このように、新聞と SNS のそれぞれの良さを掛け合わせることで、ユーザーの

知的好奇心を刺激させたり、新聞自体への興味や関心も引き出せるのではないかと考える。

（※文責: 藤島海陸）

第 3 章　活動内容

3.1　前期

3.1.1　成果物案の提案

まず、私たちは 1 人 1 人がどのような成果物を作りたいか、このプロジェクトで何をしたいのか、実現不可能な夢物語でもよいという条件で各自のアイデアを発表した。これらのアイデアが実現可能であるかどうか、既存の似たサービスが存在するかなどの調査をした。次に考えたアイデアをフローチャート化し、システムの入出力を視覚的に分かりやすくなるようにして共有した。成果物案について、ここでは詳細は割愛し、タイトルのみを以下に示す。

- 新聞世界を VR で冒険＆迷路
- 記事切り取りアプリ
- 昔の記事をサイトで再現
- やさしい日本語にする
- 新聞ジェネレーター
- 新聞の文章を方言に変換する
- 新聞のテキストデータを抽出して観光に活かすプロダクト
- おすすめの食べ物を推薦するアプリ
- モザイクアート
- 新聞のクリッピングサービス
- テキストデータの数値を CSV ファイルに出力
- MMO 風の世界を新聞のデータを利用して作る
- 新聞のテキストマイニングで消費者の行動を予測・予言書を作る
- AR で新聞の情報を読み上げ

（※文責: 柴田公季）

3.1.2　成果物案決定のためのグループワーク

成果物案決定のために図 3.1 のようにブレインストーミングを使って新聞の良いところや悪いところを整理し、新聞にどのような機能があったら面白いかを考えた。これらを踏まえてそれぞれがやりたいことをその都度実装してグループ内で共有した。やりたいことが似ている者同士で 2 グループに分かれ、A グループと B グループとしてそれぞれで作業を進めた。A グループは新聞記事を分かりやすく読めるようにするアプリを作るという方向性で取り組んでいたが、教員から分かりやすく読めるようなアプリとビッグデータとの関連性の薄さを指摘された。B グループは新聞記事を使って面白いゲームのようなものを作るという方向性で取り組んでいたが、途中からゲームではなくデータの可視化をして分析を楽しめるようなツールにしようと方向性を変更した。最終的に 2 グループで話し合った結果、A グループはビッグデータとの関連性が薄いが、B グループが取り組んでいる成果物案はビッグデータとの関連性があり、面白い成果物を作ることができるのではな

いかという考えにまとまった。その後、データを可視化する際の表示方法や作成する意味などをグループ全体で話し合い成果物案を決定した。

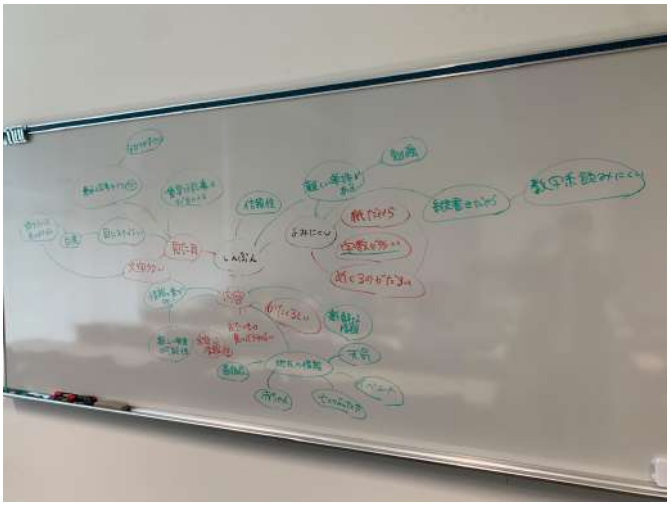


図 3.1: ブレインストーミングの様子

（※文責: 柴田公季）

3.1.3 成果物案の決定

大量のデータの中を年代やカテゴリ、細かい日付まで遡り、その日に何が起きていたのかを様々な可視化の表示方法で楽しむツールを作成することに決定した。年代は 1878 年から 2020 年までの北海道新聞の記事データを使用する。スポーツ、政治、経済、健康など多くのカテゴリで表示できるようにする。前期末に考えられていた表示方法はワードクラウドやバーチャートレースなどがある。ワードクラウドとは、文章中で出現頻度が高い単語を複数選び出し、その頻度に応じた大きさで図示する手法である。バーチャートレースとは、棒グラフ（Bar Chart）で作られたランキングが年代の推移により変化していく動くランキンググラフのことである。前期末の時点では頻出単語と重要単語のみを分析するように作られているが、後期以降は記事の内容がポジティブかネガティブか判別できる感情分析などの技術にも取り組んでいくことを計画した。記事データは 120 年分あるが、そのうちの 88 年分は画像データでありテキストデータは 32 年分しか存在しない。そのため、後期は画像認識技術を身につけ 88 年分のデータを取り扱うことができるようにする必要がある。決定した成果物案は、得られた分析結果から記事だけでは確認しにくい現象や関係性、変化などを一目見れば分かる形にすることで、新しい発見や考察が生まれることが面白味である。また、関心のある単語から新聞の興味に繋げることも可能である。最終的には新聞に興味を持ってもらうことを目的としている。

（※文責: 柴田公季）

3.1.4 成果物案の機能の案や必要技術の追求

データの可視化を行う際に「年代」、「日付」、「分類」、「分析」、「結果」の五つの項目を指定できる機能を実装することとした。「分類」とは記事のジャンルのことであり、例えば、総合、社会、ス

ポーツ、経済などである。「分析」とは分析するためにテキストデータから抽出するデータのことであり、例えば、頻出単語、重要単語などである。「結果」とは最終的なデータの表示方法のことであり、例えば、単語、グラフ、ワードクラウドなどである。

アプリケーションを制作するにあたって必要な技術は自然言語処理であるとの結論に至った。なぜなら自然言語処理を用いると頻出単語、重要単語などを抽出することができるからである。膨大なテキストデータを処理し、扱いやすい形に加工するためには絶対に必要な技術である。また、データの整理と可視化のために Python を用いる必要がある。データの整理については csv ファイルを読み込み、記事データを年月日を指定して検索する際に Python を用いる。データの可視化については、Python のライブラリを用いることでワードクラウドを生成し、画像として出力することができる。

（※文責: 高橋陽一）

3.1.5　中間発表へ向けての準備

中間発表に向けて大きく四つの準備をした。第一にポスターである。ポスターには新聞の現状、本プロジェクトで使用するデータのイメージ、成果物のイメージ、想定される効果、今後の課題を記述した。第二に発表用のスライド資料である。スライド資料は中間発表の際にポスターだけでは記述しきれなかった事柄を説明するため、また動きのあるデモンストレーションをするために作成した。第三に Python のソースコードである。Python でワードクラウドがどのように作られているのかという内部の処理の様子をデモンストレーションするために Python のプログラムを用意した。第四にアプリケーションの動作イメージである。中間発表の時点で想定しているアプリケーションの動作を見せることで聴講者に興味を持ってもらうために作成した。

（※文責: 高橋陽一）

3.1.6　中間発表会

中間発表会ではスライドを用いた説明とアプリケーションのイメージ、Python を用いたワードクラウドのデモンストレーションを行った。まずスライドを用いてプロジェクトの背景や目的を説明した。次に成果物としてアプリケーションの制作を予定していることを述べ、アプリケーションのイメージをスライドのアニメーションで表現した。次にワードクラウドのデモンストレーションを行った。一般に広く知られている出来事が表示されるワードクラウドで説明した後、聴衆に誕生日を尋ね、その誕生日のワードクラウドを Python で表示した。このデモンストレーションによって、本プロジェクトで想定している成果物がインタラクティブであることをアピールした。その後技術的な課題と今後の活動予定を説明した。

（※文責: 高橋陽一）

3.1.7　中間発表会の振り返りと夏季休業中の活動予定の決定

中間発表会の外部評価では今後のアプリケーション開発の参考になる意見が多数あった。「ワードクラウドの単語をクリックしても何も起こらないのか」、「事件や出来事の構造を概観し、表現で

きるようなアウトプットもできるのではないか」、「ある話題が 10 年単位でどのように変化したのか見比べることができると面白い」などの意見があった。一方で、新聞の要素が少ないという指摘もあった。「新聞というものの理解や解像度が表面的すぎる、浅すぎると思った」、「技術的な新規性はどこにあるのか」、「小説や漫画でも同じことができそうだが新聞でやる意味はあるのか」などの意見である。これらの外部評価の検討を通して、ワードクラウド以外の機能の追加が必要であるという課題、新聞というものの理解が浅すぎるという課題が明らかになった。

夏季休業中の目標として、「新聞やメディアに対する理解を深めること」と「アプリケーション開発に必要な知識・技術を習得すること」の 2 つを設定した。教員から薦められた書籍や情報ライブラリーの書籍を用いて、各自で活動することを確認した。さらに教員の提案で書籍、論文、Web サイトにそれぞれポイントを設定し、読んだ文献のポイントを合計してノルマを達成することを条件にした。この条件は夏季休業中に全メンバーが平等に活動することを目的としている。

（※文責: 高橋陽一）

3.2 後期

3.2.1 夏季休業中の活動の振り返り

後期の活動では、最初に夏季休業中に各自が行った活動を報告し合った。多くのメンバーが書籍を読んで技術や知識を身に着けていた。各自、前期末の時点で決定していた可視化のテーマに合わせて、活動していたことを報告した。大きく分けて、可視化に関する技術を勉強したメンバー、自然言語処理に関する技術を勉強したメンバー、新聞やメディアについての知識を深めたメンバーがいた。可視化に関する技術では、ワードクラウドやサークルパッキング、共起ネットワーク図など可視化の手法には多くの種類があることを確認した。書籍は、データ分析者のための Python データビジュアライゼーション入門 : コードと連動してわかる可視化手法 [5]、R によるテキストマイニング入門（第 2 版）[6] などであった。自然言語処理に関する技術では、tf-idf や関連語・頻出単語の抽出に関する技術を身に着けたことを確認した。書籍は、Python ではじめる機械学習 [7]、Python で動かして学ぶ 自然言語処理入門 [8] などであった。新聞やメディアに関する知識は教員から薦められた書籍を読んで今後のアプリケーションの開発方針決定に役立つ知識を身に着けたことを確認した。書籍は、ガールズメディアスタディ [9]、ニュースの多様性とは何か [10] などであった。

（※文責: 高橋陽一）

3.2.2 成果物作成へ向けてのグループ作成

夏季休業中の活動の報告を受けて、グループに分かれて開発を行うとよいのではないかという意見があった。そこで、まずはアプリケーションのフロントエンド班、バックエンド班、自然言語処理班に分かれた。しかしアプリケーション開発をできるだけのフロントエンド・バックエンドの知識を身に着けているメンバーが少なく、それ以外のメンバーはフロントエンドに関する技術の勉強をするか手が空いてしまうという状況になってしまった。また、教員から手が空いているメンバーは画像データに目を通して、画像データを使って何かできそうなことはないか模索してみてはどうか、という助言を受け、新たに班を分割しなおすことにした。その結果、アプリケーションのフロ

ントエンドを担当するデザイン班 1 名、アプリケーションのバックエンドを担当するバックエンド班 1 名、自然言語処理を担当するデータエンジニアリング班 3 名、画像データを使った作業を担当するコンテンツ抽出班 3 名というグループに分かれた。ただし、自分のグループの作業だけを行うのではなく手が空いていれば、ほかの班の手伝いをするなどの工夫をした。

（※文責: 高橋陽一）

3.2.3 グループごとでの成果物作成

グループごとの活動では、活動時間のうち、最初と最後の 15 分ほどでグループごとに活動予定と活動報告を毎回行った。これによって他のグループの進捗具合やほかのグループの活動について気になった点の指摘や助言を行うことができ、活動の効率が向上した。また機能ごとにグループに分かれて作業することで、自然言語処理や画像などの各グループのテーマに集中して作業することができ、全員で同じ機能の開発を行うよりも開発の時間を短縮できた。デザイン班ではアプリケーションのデザインを担当した。バックエンド班ではアプリケーションの API の作成を担当した。データエンジニアリング班ではテキストデータを自然言語処理することによってアプリケーションで利用可能な形式に変換する作業を担当した。コンテンツ抽出班では機械学習を用いて画像データから 4 コマ漫画と天気図を抽出する作業を担当した。

（※文責: 高橋陽一）

3.2.4 アプリケーションのプロトタイプ完成

アプリケーションのプロトタイプはデータエンジニアリング班のプログラムの実行結果をバックエンド班に渡し、フロントエンド班で表示することで完成した。ただし、コンテンツ抽出班の画像に関してはアプリケーション内でどのように使用するかのアイデアがまとまらなかったため、成果発表会の時点では画像データから抽出した画像は使用しないこととした。

（※文責: 高橋陽一）

3.2.5 最終発表に向けての準備

最終発表に向けてアプリケーション開発と同時進行で発表用スライドとポスターの制作を行った。スライドは中間発表会で使用したスライドをもとに修正と後期の活動内容を書き足すことでスライド作成の時間を削減した。これによってアプリケーション開発に専念することができた。ポスターはプロジェクトのテーマに合わせ新聞紙面を参考にデザインすることにした。新聞紙面のデザインを参考にするにあたって、日本語のポスターと英語のポスターを分けて、それぞれ日本の新聞、英字新聞のデザインを参考に制作することにした。完成した第一版は新聞紙面に寄せてはいるが違和感があるという指摘を受けた。そして教員から使った技術の説明ではなく、プロジェクトで行ったことがどうすごいのか、活動内容について斬新な点などを大きく掲載するように助言を受けた。また、見出しの部分は単語を書くのではなく、新聞の小見出しのように短い文章で本文の内容を端的に表すようにすべきと助言を受けた。これらの助言を受けて修正した第二版は非常に新聞に近いデザインとなった。

（※文責: 高橋陽一）

3.2.6　最終発表会

最終発表会ではスライドを用いた説明とデモンストレーションを組み合わせて発表を行った。スライドでは新聞の現状や課題など、プロジェクトの背景を説明してからアプリケーションの内容や機能を説明した。次にデモンストレーションを行いアプリケーションが動作している様子を聴衆に示した。デモンストレーションでは聴衆の中からひとり指名し、その人の誕生日のワードクラウドを表示することでアプリケーションの面白さを伝えるとともに、興味を持ってもらえるように工夫した。その後、各グループごとに技術的な内容と今後のアプリケーションの課題を説明した。最後に質疑応答とデモ機を使って聴衆が実際にアプリケーションを体験できるようにした。

（※文責: 高橋陽一）

3.2.7　最終発表会の振り返りとアプリケーションの改良

最終発表会では様々な質問や意見があった。最も多かった質問・意見はワードクラウドの機能についてである。最終発表会の時点では指定した年月日のワードクラウドを表示するだけであった。しかし、ワードクラウドの単語をクリックすると記事を見ることができるといいとか、単語にルビを付けてみてはどうかという意見があった。また、想定している利用者の世代や利用現場はどのようなものかという質問があり、アプリケーション開発において想定しているユーザーや場面を考える機会が少なかったことがわかった。秋葉原の学外発表会に向けてワードクラウドの機能の改良、サークルパッキングの機能の追加、コンテンツ抽出班が用意した４コマ漫画と天気図の利用など、最終発表会の結果をもとにアプリケーションの開発を続ける予定である。

（※文責: 高橋陽一）

第 4 章　開発

4.1　開発目的・目標

本プロジェクトでは、新聞ビッグデータを扱ったアプリケーションを通じて、新聞に眠る情報の価値を引き出し、 マスコミュニケーションの手段としての新聞ではなく教育への活用や知的好奇心を刺激する方向へと導き、新聞の新しい使い方を提案する可能性を示唆することを目的としている。そのため開発目的としては、

- ただ新聞記事を読むという方法以外で、新聞に含まれている情報を利用者に提供する
- 情報について、様々な方法での可視化を通して利用者に新聞に秘められていた情報の面白さを理解してもらう
- 知らなかった情報を知ることができる機会を作り、知的好奇心を刺激できるようなものを作る

というものが挙げられる。また開発目標としては、最終発表会でいただいた意見を参考に、開発目的を達成できているかどうか判断し、全員が目的が達成できていると納得できるものを開発することとした。

（※文責: 一入悠貴）

4.2　習得した技術・知識

4.2.1　自然言語処理

自然言語処理（Natural Language Processing）とは、自然言語を機械で処理し、内容を抽出、解析する処理技術のことである。処理に使用する形態素解析エンジンとして MeCab、Janome、juman などがよく知られているが、本プロジェクトでは、学習する際に参考にできる情報量などの観点から、MeCab を採用した。

自然言語処理によって行ったことは、新聞に出てくる固有名詞の頻出度・重要度の解析である。まず読み込んだ新聞テキストデータを分かち書きし、分かち書きされた語それぞれについて品詞を分析して固有名詞のみを抜き出す。これらの動作を読み込んだテキストデータ全体に対して行い、各単語の出現回数や情報量などを、csv ファイル形式で出力させた。なお、この csv ファイルは後述のデータの可視化の際に使用している。

MeCab を使用するにあたって、問題となったのは辞書である。例えば、「北海道日本ハムファイターズ」という単語を形態素解析したとき、デフォルトの辞書を使用すると「北海道」「日本」「ハム」「ファイターズ」のように、４つの独立した単語として認識してしまう。この問題を解決するために、本プロジェクトでは mecab-unidic-NEologd というシステム辞書を導入した。mecab-unidic-NEologd とは、形態素解析エンジンである MeCab と共に使う単語分かち書き辞書であり、週 2 回以上更新更新され、新語・固有表現に強く、語彙数が多く、オープンソース・ソフトウェアとして公開されているものである。この辞書の導入により分かち書きの精度が向上し、よ

り質の高い分析結果を出すことが可能になった。

（※文責: 遠藤晴人）

4.2.2　データの可視化: plotly

新聞テキストデータから抽出した頻出単語分析の際に、データの変遷が可視化されるような表現方法があると面白いという意見があり、そこから着想した。グラフ下部にあるスライダーで日や月、年などを可変にし、スライダーを動かすことでデータそのものも年代に沿って変化するというグラフを作成した（図 4.1）。このグラフの作成には plotly[*1] というオープンソースライブラリを使用した。このライブラリの活用により、折れ線グラフや棒グラフ、散布図など、様々な表現でのインタラクティブなグラフが作成可能となり、想定した表現方法の実現に大きく近づいた。しかし、現状は NLP によって作成された csv ファイルをそのまま扱えるわけではなく、plotly で扱いやすいように csv ファイルを加工したのちグラフ化を行っているため、これを解決しなければ実用性は低いだろう。

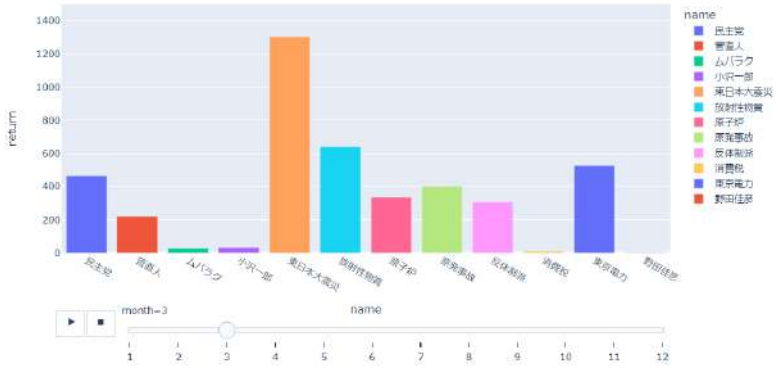


図 4.1: plotly で作成したグラフ

（※文責: 辰己尚矢）

4.2.3　データの可視化: ワードクラウド

WordCloud（図 4.2）とは、「文章中で出現頻度が高い単語を複数選び出し、その頻度に応じた大きさで単語を図示する手法」と定義されるものである。頻出している多くの単語を表示させるため一覧性が高い、単語ごとに文字の大きさが変わるので重要な単語が分かりやすい、などの理由からこの可視化方法を選択した。

出力方法として前期の段階では、Python の標準ライブラリである collections の Counter クラスを用いて自然言語処理によって指定した範囲の新聞における名詞の出現回数を数え、WordCloud のライブラリと作成した辞書を用いることで png 型式にて画像を出力させた。また WordCloud を出力させる際の設定として、画像サイズを 1000 × 600、使用語数を 60 語、背景色を白、文字に使用するカラーマップを tab10、フォントを HG 創英角ゴシック UB とした。

後期の活動では、Python で日付・単語・登場回数の書かれた json を作成し、javascript で json

[*1] Plotly. Plotly Open Source Graphing Library for Python https://plotly.com/python/

図 4.2: WordCloud

データを取得し出力させる方法へと変更し、実行速度の高速化を図った。また、画面や文字のサイズなどを鑑みて使用語数を 100 語程度に増やし、新聞らしさを出すために背景を新聞調とした。

（※文責: 遠藤晴人）

4.2.4 紙面データの補正

今回使用している新聞紙面の画像データには傾きや欠損箇所が存在している。そのままテキストデータ化といった処理を行った場合、処理結果に影響を与えることが考えられる。従って、傾き補正および欠損箇所の修正を行うこととした。よって傾き補正のために習得した技術について紹介する。　今回、傾き修正にはテンプレートマッチングを用いて傾きを取得、修正する方法を用いた。新聞には必ず上部にヘッダーとして新聞紙名が記されている。その新聞紙名が記されている部分を PhotoShop などの画像編集ソフトを用いて「北海」と「新聞」に分けて抽出し図 4.3 のような画像を得た。これらの総数 n の画像に対して Python を用いて、すべての画像の中で最もサイズが大きいものに合わせる拡大を行った後、すべての画像の画素値の平均を取る処理を行った。その結果、図 4.4 のような結果を得た。この結果をテンプレートとしてマッチングを行い、ヘッダー部分の「北海」と「新聞」部分のそれぞれの座標を取得した。それらの座標から座標差を求め、傾いている角度を計算し、補正を行った [11]。

図 4.3: 紙面からヘッダーの一部を切り出したもの

図 4.4: 作成したテンプレート

（※文責: 川平覚士）

4.2.5 labelImg

labelImg とは、物体検出を行うためのデータセットを作成するアノテーションツールである。本プロジェクトでは、新聞紙面の画像データから特定のコンテンツの抽出を行うために、labelImg を使用してラベリングを行った。4 コマ漫画や天気情報であると判断した箇所を範囲指定し、アノテーションを YOLO 形式で保存した。4 コマ漫画であれば “MANGA”、天気情報であれば “Weather”、広告であれば advertisement を省略し “ads” としてクラス分けを行った。図 4.5 は実際にラベリングを行っている画面であり、緑色で範囲指定されているものが 4 コマ漫画の箇所であり、青色で範囲指定されているものが天気情報の箇所である。4 コマ漫画は、漫画以外の情報が入らないように範囲指定した。天気情報は、明確な枠組みが存在していなかったため天気情報のコーナーであると判断した箇所を範囲指定した。

図 4.5: 実際に labelImg を使用したラベリング

（※文責: 辰己尚矢）

4.2.6 YOLO

YOLO（You Only Look Once）とは、高速な物体検出アルゴリズムの一つである。この YOLO は End-to-end の物体検出手法であり、従来の物体検出手法と比べて高速かつ高精度な検出をすることが可能となっている。また、YOLO では信頼度スコア（Confidence score）という要素を使用しており、このスコアの値から、検出結果の精度がどのくらい高いのか判断することができることも YOLO の大きな利点となっている。

YOLO を使用すると図 4.6 からわかる通り、特定の物体を検出することができる。この画像ではクルマや自転車、信号機などを検出することができていることがわかる。

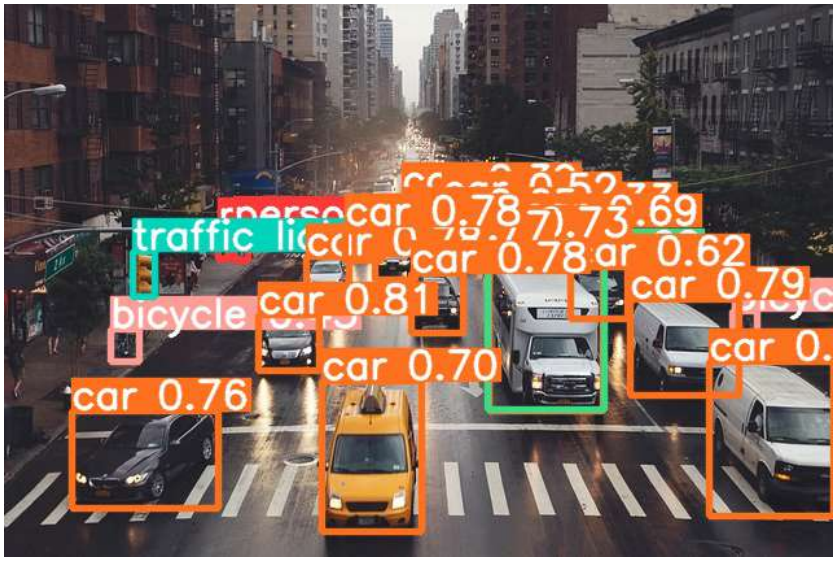


図 4.6: YOLO の使用例

本プロジェクトではこの技術を新聞紙の画像データから 4 コマ漫画等のオブジェクトを抽出するために使用することとした。

YOLO では検出したい物体についてまず学習させないと、その物体を検出することができないため、私たちはまず学習させることにした。これは 4.2.5 で記述した labelImg を使ってラベリングされた学習用のデータセットを使い、学習を行った。これらの学習では、寺沢研究室の卒業生である北清敦也さんの卒業論文、特に 4 章等を参考にしたり、Web サイトを参考にし、寺沢先生に機械学習の結果のグラフなどを見てもらいアドバイスをいただきながら学習を行っていった。図 4.7 は結果のグラフのうちの 1 つである。

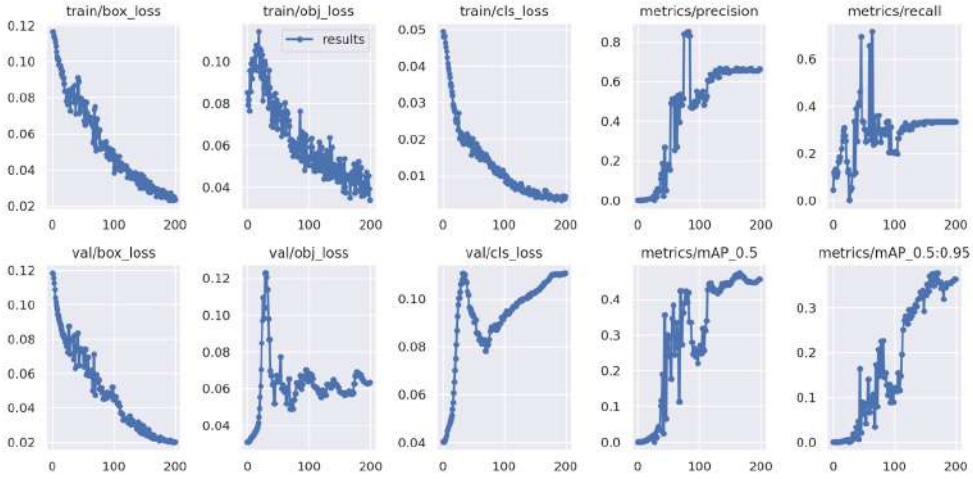


図 4.7: 学習結果のグラフ

そのように YOLO での学習を進めていき、最終的には一定以上の精度で新聞の画像データから、４コマ漫画と天気コーナーの検出に成功した。図 4.8 はそれらを検出することに成功した結果である。左上に４コマ漫画、左下に天気コーナーを検出することに成功したことがわかる。

図 4.8: 検出結果

また、検出していた部分を保存する抽出については、始めは実行した際のコンソールログから、画像のどの部分が検出されたのかに関する座標を獲得し、元の画像からその座標を指定することで解決しようとしていた。しかし、YOLO では「–save-crop」のコマンドを使用することにより、検出結果の画像を保存する際に、検出したものを自動で保存することが可能なことがわかった。私たちはそのコマンドを使用することにより、大量の４コマ漫画や天気コーナーの画像を抽出することに成功した。また図 4.9 については抽出に成功した４コマ漫画の画像となっている。

図 4.9: 抽出した 4 コマ漫画

（※文責: 一入悠貴）

4.2.7 tf-idf

tf-idf は、term frequency–inverse document frequency の略であり、文書の集まりの中で各文書のトピックを判断するのに有効な単語を特定することができる [12]。tf（term frequency：単語の出現頻度）値は文書内でのある単語の出現頻度を表す。すなわち、出現回数が多いほど tf 値は大きくなり、出現回数が低いほど tf 値は小さくなる。idf（inverse document frequency：単語の逆文書頻度）値は全文書中である単語を含む文書がどれくらい少ない頻度で存在するかを表す。すなわち、単語が他の文章にも多く出現しているほど idf 値は小さくなり、単語が他の文章にあまり出現していないほど idf 値は大きくなる。この tf 値と idf 値をかけ合わせたものが tf-idf 値である。この tf-idf を用いることで文書の特徴語を抽出することができる。実際に簡単な具体例を以下に示す。

- 文書１：「リンゴ　バナナ　ブドウ　なら　リンゴ　が　好き」
- 文書２：「リンゴ　メロン　スイカ　なら　メロン　が　好き」
- 文書３：「リンゴ　スイカ　なら　リンゴ　が　好き」

という 3 つの文書があるとする。「リンゴ」という単語は全ての文書に登場しているため、idf 値が小さくなる。したがって tf-idf 値も小さくなるということが分かり、文書を特徴づけるような役割がない単語であると言える。逆に「メロン」という単語は文書 2 にしか存在しないため、idf 値は大きくなり、さらに文書 2 に複数回存在しているため tf 値も大きくなる。したがって tf-idf 値は大きくなることが分かり、文書を特徴づけるような役割がある単語であると言える。

（※文責: 柴田公季）

4.3 開発過程

4.3.1 データエンジニアリング班の開発過程

4.3.1.1 開発環境

データエンジニアリング班では、自然言語処理を行うための開発環境として、Google Colaboratory を使用した。前期はワードクラウドを表現するだけであったため、個人のローカル環境で開発するだけでも問題無く活動できていた。しかし、後期から新たに可視化方法を追加し、データエンジニアリング班のそれぞれのメンバーが開発を同時進行で進めていく必要があったため、環境構築が不要でプログラムの共有がしやすい Google Colaboratory を使用するのが最適であると判断した。Google Colaboratory には、無償版、そして有償版の Pro と Pro + がある。無償版は GPU が自動割り当てであり、最大実行可能時間が 12 時間でバックグラウンド実行ができないが、通常の処理を行う分には問題がないので、データエンジニアリング班では基本的に無償版の Colab を使用することにした。しかし、無償版の最大実行可能時間である 12 時間を超えてしまうような大量のデータを処理する際は、Colab Pro + を利用している角康之先生に自分たちのプログラムを代わりに実行してもらうことで、この問題を解決することができた。

また開発初期は、北海道新聞社様から頂いた CSV ファイル形式のテキストデータを、メンバー個人でダウンロードして使っていたが、何度もダウンロードする手間があること、そしてダウンロードしたファイルをローカルで保存しようとすると、かなりの容量を必要してしまうなどの問題があった。これらの問題を解決するために、Google drive 内で共有ドライブを作成し、そこでテキストデータとソースコードを管理していくことにした。これにより、データエンジニアリング班内でデータの共有が簡単になり、効率よく開発を進めることができるようになった。さらに、昨年度、新聞ビッグデータプロジェクトのメンバーであった方から、去年の成果物を作成する際に使用したデータセットやソースコードを共有アイテムとして頂いたので、それも同時に自分たちの共有ドライブに載せておいた。昨年度培われた知識や技術をいつでも参考にできる状態にしたことで、開発途中で詰まってしまったり、分からないことがあった際に、時間をかけずスムーズに解決することができた。その結果、さらに高品質でパワーアップした成果物を作ることができたと考える。

また前期にワードクラウドを作成していた時は、一つの PC 内で、自然言語処理からワードクラウドの表示まで一括で行っていたため、プログラムで出力した結果は、コンソール内に表示させておくだけでデモンストレーションができるため、ファイルに出力する必要はなかった。しかし、後期から、各班ごとに分かれての活動が新たに始まったので、JavaScript を使用するフロントエンド班やバックエンド班が利用できる形にするため、出力結果を json 形式のファイルに変換し、参照しやすい場所に置いておく必要があった。そこで、バージョン管理システムである GitHub を使うことにした。GitHub では、データエンジニアリング班のリポジトリを作成し、その中に逐一出力結果やソースコードを載せることで管理した。加えて、Issues に現状の課題を挙げてメンバー間で共有することで、優先的に解決しなければならない問題から先に取り組むことができた。

（※文責: 藤島海陸）

4.3.1.2 可視化方法の考察

考察のための準備

前期はワードクラウドしか可視化方法がなかったため、どのような可視化があればさらにさらに

成果物の価値を高めることができるのかを考えた。そこで、夏季休業中にプロジェクトメンバー全員が複数の論文や本を読み、知見を広げた。そうすることで後期の活動が始まると同時にメンバーで話し合うことで、前期には生まれなかったような考察や発想が生まれた。その中で関連語と特徴語を可視化できるようにするのが良いのではないかと話し合った。本プロジェクトでは最終成果物として、新聞のテキストデータを可視化し、知的好奇心を刺激する web アプリケーションをつくることを目標として活動している。そのためワードクラウドのみでは知的好奇心を刺激しているとは言える状態ではなかったため、関連語や特徴語を可視化できるようにすることでさらに web アプリケーションの価値が高まるのではないかと考えた。

関連語の可視化

関連語を可視化することで、自分の興味を持った単語に関連する単語にも興味を持つことができ、知見を広げていくことができると考えた。例えば、「自民党」という単語を中心として考えると、その関連語として「民主党」や「公明党」など、自民党以外の日本の政党の名前が出てくるようになる。さらに「民主党」という単語から関連して「対立候補」や「衆院選」、「大統領選」などの単語が出てくるようになる。この場合だと、自民党以外の単語も同時に知ることができるため、政治に興味を持てるようになるかもしれない。そういった意味で関連語はワードクラウドとは違った可視化の方法で知的好奇心を刺激できるのではないかと考えた。

特徴語の可視化

特徴語を可視化することで、頻出単語からつくられたワードクラウドでは可視化されていないような単語を可視化することができると考えた。例えば、ある新聞記事において頻出単語で「北海道」や「大統領」という単語が圧倒的な頻度で出現している一方で、特徴語を抽出すると「パレスチナ」や「サハリン」といった単語が上位に出現するようになる。このように、特徴語は頻出単語とは全く異なるということがわかる。特徴語の可視化はワードクラウドや、頻出単語のワードクラウドに表示されている単語と被った単語に色をつける可視化の仕方などを想定している。こうすることで、頻出単語とは異なった可視化表現をすることでき、知的好奇心を刺激できるのではないかと考えた。

（※文責: 柴田公季）

4.3.1.3　データの整形・高速化

北海道新聞社様から頂いた CSV ファイル形式のテキストデータには、【訂正あり】、【続報注意】などといったノイズが含まれているものがあった。また、一つのファイルに含まれている記事の期間がバラバラで統一されていなかった。例えば、1988 年は一つのファイルに 7 月〜12 月の半年分の記事データが含まれていたのに対し、1989 年は 1 月〜12 月まで一年分の記事データが含まれていた。このようにノイズが入っていたり、テキストの形式が統一されていなかったことで、元データをすぐにプログラムに落とし込んで開発を進めることができなかった。そこでまず初めに記事データから ‘12 月’、‘31 日’ といった日付や、‘総合’、‘スポーツ’ といった記事属性などの分析に必要な情報のみを抜き出すことで、Python 上で扱いやすいデータに整形をする作業を行った。さらに、北海道新聞の休刊日である 1 月 2 日や、うるう年の 2 月 29 日の記事は除くことで、想定外のエラーが発生することを防いだ。

また、頂いた元データには約 5,000,000 個の記事が格納されており、ファイルサイズも 8.4GB

とかなり大きい状態であった。そのため、ダウンロードしたものを Python で処理しようとすると実行にかなりの時間を要し、効率よく開発を進めることが出来ていなかった。そこで、処理や計算を高速化させ、より効率的な開発を図るため、Python の標準ライブラリである numpy や glob モジュールを用いた。numpy は記事内に出現した単語間のコサイン類似度を求める際に活用し、glob モジュールは、引数に指定されたパターンにマッチする記事のファイルパス名を取得することに活用できた。さらに、関連語を出力する際に、比較的データサイズが小さい日本語の学習済み Word2vec モデルを使用した。これらの手法を用いた結果、計算の高速化とデータサイズの縮小化に成功した。具体的なデータサイズを見ていくと、図 4.10 のように、日ごとの名詞の出現回数が格納されているワードクラウド用のデータでは 700MB、そして記事内に出現した単語の類似度が格納されている関連語用のデータでは 70MB まで、サイズを軽量化することができた。どちらのデータも 90 %以上の軽量化ができたため、データエンジニアリング班での利用はもちろん、バックエンド班の処理時間短縮にも貢献することができた。

	情報	サイズ
元データ	約 5,000,000 個の記事データ(.csv)	8.4 GB
ワードクラウド用に処理したデータ	日ごとの名詞の出現回数 (.json)	700 MB
関連語の表示用に処理したデータ	記事内に出現した単語の関連語 (.json)	70 MB

図 4.10: 軽量化したデータの例

（※文責: 藤島海陸）

4.3.1.4　頻出単語の抽出

データエンジニアリング班では、約 30 年分の新聞記事を使用して日付ごとの頻出単語の抽出を行った。本プロジェクトでは、ユーザに提示する可視化方法として WordCloud での可視化が決定しており、これを実装するためにはテキストデータ中の全単語について「日付」「単語」「その日の単語の登場回数」の 3 つのデータが含まれる json データが必要であった。これを作成するため、全単語について日付ごとに登場回数を数え上げるプログラムを書いた。プログラムの詳細について以下に記述する。なお、この作業がデータエンジニアリング班として最初の作業であったため、データの取り込み、整理方法などについてもここで紹介する。

プログラムでは、まず新聞のテキストデータが入った CSV ファイルを読み込んだ。この CSV ファイルは、「記事情報」「訂正フラグ」「見出し」「見出し・本文・その他」という 4 つのラベル付けされた情報からなる。「記事情報」には主に年月日や記事属性 (総合や社会、生活・暮らしなど)、記事の字数、朝刊か夕刊か、などが書かれており、「訂正フラグ」には訂正された記事の情報、「見出し」には記事の見出し、「見出し・本文・その他」には見出しと本文が書かれている。これらの情報を取捨選択して、プログラムでは年月日・見出し・本文・記事属性のみを読み込んだ。

つぎに、各日付ごとに自然言語処理を行って、その日に出てきた全名詞を重複ありでリストに格納し、その後 Python の標準ライブラリである collections の Counter クラスを用いて各単語がその日に何回出てきたかを調べた。この際、WordCloud のデータとして不適切である、一般的な単語（「日」や「こと」、「もの」、「１」～「９」など）が多く出てきてしまう問題が露呈した。これら

のノイズをできるだけ少なくするため、データエンジニアリング班では Twitter のトレンドワードのアルゴリズムを参考にし、3 文字以上の単語のみ取得することとした。

最後にフロントエンド班の指定する形に書き換えることで、（図 4.11）のような json データを作成することができた。このデータは WordCloud の作成に必要とする情報のみが書かれたコンパクトなものとなっており、元の CSV ファイルから毎回情報を読み取って WordCloud を作成すると 2,30 秒程かかるところ、この json データを利用すると 1,2 秒で作成することができる。


```
[
    {"date":"1989-01-01","word":"北海道","count":16},
    {"date":"1989-01-01","word":"書記官","count":13},
    {"date":"1989-01-01","word":"書記長","count":9},
    {"date":"1989-01-01","word":"テスト","count":9},
    {"date":"1989-01-01","word":"生まれ","count":8},
    {"date":"1989-01-01","word":"政治改革","count":8},
    {"date":"1989-01-01","word":"スケート","count":8},
    {"date":"1989-01-01","word":"アメリカ","count":8},
    {"date":"1989-01-01","word":"わが国","count":8},
    {"date":"1989-01-01","word":"大統領","count":7},
    {"date":"1989-01-01","word":"ゴルバチョフ","count":7},
    {"date":"1989-01-01","word":"ところ","count":7},
    {"date":"1989-01-01","word":"明らか","count":7},
    {"date":"1989-01-01","word":"フロン","count":7},
    {"date":"1989-01-01","word":"アイヌ","count":7},
    {"date":"1989-01-01","word":"二十一世紀","count":6},
    {"date":"1989-01-01","word":"アジア","count":6},
    {"date":"1989-01-01","word":"ヘクタール","count":6},
    {"date":"1989-01-01","word":"試験場","count":6},
    {"date":"1989-01-01","word":"イスラマバード","count":6},
    {"date":"1989-01-01","word":"北方領土問題","count":5},
```


図 4.11: 作成した json 形式のデータ

（※文責: 遠藤晴人）

4.3.1.5 関連語の抽出

データエンジニアリング班のメンバーである遠藤が作成したプログラムにより、記事内に出現した頻出単語を抽出することができるようになった。そこで、それらを用いることで本プロジェクトの成果物の、‘インタラクティブに可視化し知的好奇心を刺激する’ という目的を達成できるような可視化方法がないかチーム内で検討した。感情分析判定や記事の多様性を表現するなどといったさまざまな案が挙げられたが、最終的に ‘関連語’、つまり、ある頻出単語に対して最もコサイン類似度が高い単語を抽出することに決定した。

関連語を抽出することが最適だと判断したのには、大きく二つの理由がある。一つ目は、成果物の目的を達成するための可視化方法としてベストだと考えたからである。ある特定の頻出単語に対し、JavaScript の D3.js というライブラリを適用することで、単語間の関係性をインタラクティブに表現できる可視化方法の一つが関連語である。また、関連語を特定の頻出単語の周りに配置することで、利用者の「この単語に関連している言葉は何だろう？」という疑問や「この関連語は聞いたことがないから調べよう！」などといった知的好奇心を掻き立てることができると考えた。2 つ目の理由としては、関連語の抽出が他の可視化方法に比べて、実装しやすかったからである。関連語抽出の大まかな手順としては、まず初めに全期間分の頻出単語を求める。その後、word2vec を用いることで、求めた全期間分の頻出単語と、ある日の頻出単語の上位 10 個とのコサイン類似度を計算するという 2 ステップである。しかしながら、もし、上記で挙げた記事の多様性を表現しようとした場合、以下の 6 つの手順が必要となる。

(1) 単語間の距離から意味的一貫性を計算するためのテキスト（コーパス）を用意する。
(2) 用意したコーパスに対し、形態素解析を行うことで、テキストを単語ごとに分割する分かち書き処理をする。
(3) 分かち書きされたテキストに対して R の word2Vectors パッケージを用いて word2vec によるベクトル化を行い、テキスト中に出現する単語間の cos 類似度を測定する。
(4) 分析対象となるテキストデータにおいて、特定の単語が何回出現したかを示す文書単語行列を作成する。
(5) 作成した文書単語行列に対して、R の lda パッケージでトピックモデルを実行することで、最も意味的一貫性の値が高くなるトピック数を求める。
(6) 求められたトピック数で新聞記事を機械的に分類し、分類された各トピックの比率に基づいて多様性指標を計算する。

これらの処理を実際に行うとなると、(1) では単語間の距離を適切に測定できるほどの十分さサイズを備え、かつ分析対象のデータに含まれる単語を網羅するようなコーパスを自分たちで作成しなければならなくなる。初期の頃は、実際に自分たちでコーパスを作ることも検討したが、昨年新聞ビッグデータプロジェクトの一員だった方から、「自分たちでコーパスを作るのは卒業研究以上のレベルであるため、作成することはかなり難しい」という指摘を頂いたため、結果的に断念する形となった。

次に、抽出された関連語に対し、フロントエンド側でどのような可視化表現を使用するかによって、出力される json ファイルの書き方が異なってくるため、フロントエンド班の前田と相談をした。可視化表現は JavaScript のライブラリである D3.js にあるものの中から選択することになったが、図 4.12 のような、円の中に円を詰め込むことで、階層的なデータを効果的に可視化することができるサークルパッキングという手法を取り入れることにした。そこで、json ファイルの中身もサークルパッキング用の書き方にすることができた。

また、出力された関連語は図 4.13 のようになる。例えば、2020 年 12 月 9 日の頻出単語が自民党だったとき、自民党という単語に対して cos 類似度が高い単語は上位から公明党、民主党、幹事長の 3 つである。よって、1 段目に、自民党に対する関連語として上記の 3 つが挙げられることになる。また、2 段目にそれぞれの関連語に対しての関連語も同時に求めている。公明党の関連語としては、上位から麻生派、衆院議員、与野党の 3 つが挙げられている。さらに、2 段目の関連語には、どれくらい意味的に近い単語かというものを表す、類似度を追加した。なぜ、一段目の関連語には類似度を追加しなかったかというと、もし追加してしまった場合、サークルパッキングで出力できる形ではなくなり、フロントエンド側で処理を行えなくなってしまうからである。このように階層的な表現ができたことで、インタラクティブな可視化を実現し、読者の知的好奇心を刺激できるものになったと考える。

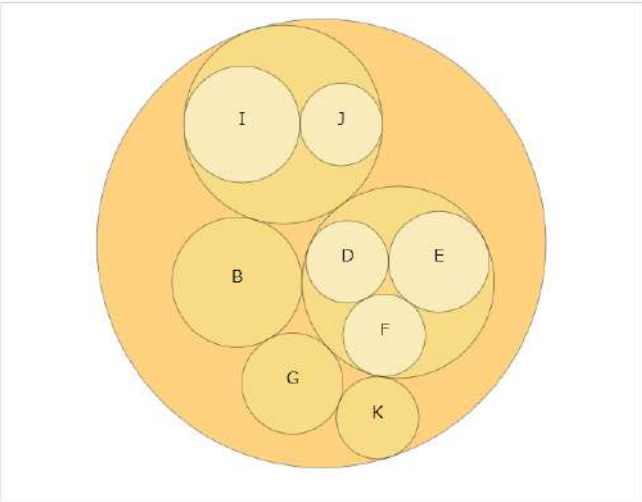


図 4.12: サークルパッキング

```
data = {
    "name": "自民党",
    "date":2020-12-09,
    "children": [
      {
        "name": "公明党",
        "children": [
          { "name": "麻生派", "value": 0.86042386},
          { "name": "衆院議員", "value": 0.8553953},
          { "name": "与野党", "value": 0.8384836}
        ]
      },
      {
        "name": "民主党",
        "children": [
          { "name": "対立候補", "value": 0.84413654},
          { "name": "大統領選", "value": 0.825132},
          { "name": "衆院選", "value": 0.81695104}
        ]
      },
      {
        "name": "幹事長",
        "children": [
          { "name": "石破茂", "value": 0.9113241},
          { "name": "菅直人", "value": 0.90914536},
          { "name": "村山富市", "value": 0.90231854}
        ]
      },
```

図 4.13: 出力された関連語

（※文責: 藤島海陸）

4.3.1.6　特徴語の抽出

本プロジェクトでは tf-idf 値を用いて特徴語の抽出に取り組んだ。tf-idf を使うことで頻出単語の抽出では現れないような単語や、他の新聞記事には書かれていないような特別なキーワードの抽出が期待できると考えた。また、昨年のプロジェクトでも tf-idf を使用していたという情報を頂き、先輩と話し合いながら今年も取り入れたほうが良いという結論に至ったため、今年も取り組むことにした。ここからは GoogleColaboratory を使ってどのように実装したかを説明していく。本プロジェクトでは tf-idf を求めるために scikit-learn を使った。scikit-learn は、Python で実装されたオープンソースの機械学習ライブラリである。分類、回帰、クラスタリングなどの非常に多くのアルゴリズムを実装していて、機械学習モデルを作成する上で必要不可欠なライブラリである。使い方も比較的シンプルでプログラミングがあまり得意でないメンバーでも簡単に扱うことができた。わかち書きとは、文章において語の区切りに空白を入れて記述することである。scikit-learn を使うためにはテキストをわかち書きする必要があったため、1 記事ごとにわかち書きしたデータを与え、tf-idf を求めた。わかち書きをするための準備として、まず上記のデータの整形・高速化で示したように記事データから分析に必要な情報の記事データだけを抜き取った。抜き取った記事データをオープンソースの形態素解析エンジンである MeCab を用いて形態素解析を行った。辞書には ipadic に含まれていない単語も採録している mecab-unidic-NEologd を用いた。MeCab.Tagger クラスのインスタンスを生成するときに「-Owakati」のオプションを指定し、parse メソッドに記事データを与えることでわかち書きをすることができた。このようにしてわかち書きしたデータを scikit.learn の sklearn.feature_extraction.text.TfidfVectorizer を用いて解析し、tf-idf を求めた。出力結果を図 4.14 に示す。はじめは ipadic を辞書として使っていたためカタカナ語ばかりが特徴語として抽出されたが、mecab-unidic-NEologd を使うことでカタカナ語だけでなく漢字が使われた複雑な単語なども抽出することに成功した。また、特徴語をさらに絞り込むために tf-idf の値を 0.3 以上にし、単語の文字列の長さを 3 文字以上にした。文字列の長さを 3 文字以上にした理由は Twitter のトレンドの文字列の長さが 3 文字以上という設定から着想を得た。tf-idf は 0 以上 1 以下の値で出力されるため、ワードクラウドをつくる際には tf-idf の値を数 10 倍にする必要がある。このようにすることで、当初期待していたような、頻出単語の抽出では現れないような単語や、他の新聞記事には書かれていないような特別なキーワードの抽出ができたと考える。

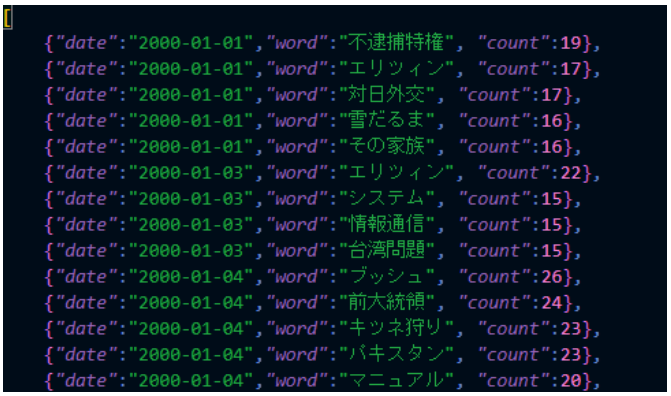


図 4.14: tf-idf の出力結果の例

（※文責: 柴田公季）

4.3.2　コンテンツ抽出班の開発過程

4.3.2.1　コンテンツ抽出班概要

前期の課題として挙げられていた 88 年分の画像データの活用と画像認識技術の向上を目指して発足した。新聞の画像データのテキストデータ化を進めるうえで、画像の欠損やレイアウト処理の難航によって、テキストデータとしての活用ではなくコンテンツ抽出での画像データの活用が提案された。

（※文責: 辰己尚矢）

4.3.2.2　活動内容考案

次にコンテンツ抽出班では具体的な活動内容をきめるために動き始めた。

まず始めに、画像データから何かを抽出するということは、その画像データについて詳しく知っていなければならないと考え、コンテンツ抽出班の３人で画像データを隅から隅まで確認していくこととした。その過程で、古すぎて扱うことが難しそうな記事データを見つけたり、新聞紙の今と昔で違うレイアウトを発見したりなどいくつも知らなかった事を発見していった。

その後、それらの発見や気づきを元にどのような活動を行っていくのかを提案する段階に移った。その段階では、まず始めにこれらの画像データから文字認識の技術を使用して、今あるテキストデータ以外に更に他の年代のテキストデータを増やすという前期に取り組んでいた内容を引き継ぐ案が出た。しかし、この分野に詳しい寺沢先生に相談したところ、この古い新聞の画像データから、文字認識をすることはとても難しく、あまり画像認識に詳しくない学生がいきなり始めて残り数か月のプロジェクトで手を付けて達成することができる難易度ではないと教えていただき、この案は没となった。

その次の案では、テキストデータ化が難しいということから、画像データの中から何かコンテンツを抽出しようという考えに至った。その中でも何を抽出していこうか話し合った結果、４コマ漫画と天気、広告を切り抜こうという案になった。

その後も、新聞の横文字が右からなのが左からになったり、広告の雰囲気やレイアウトが変わるタイミングを発見したりしたため、これを元に何か良い案ができないか考えたが、コンテンツの抽出を超えるような案が出なかったため、４コマ漫画などを抽出する案で決定した。

（※文責: 一入悠貴）

4.3.2.3　必要技術模索

画像認識の学習を進めるうえでこの分野に詳しい寺沢先生へ相談したところ、座標指定して切り抜く方法があるという提案を受けた。その提案を踏まえて、一般的な 4 コマ漫画は紙面の左上に配置されていることが多いため、該当部分を座標指定し切り抜くことができるのではないかと考えたが、88 年分の画像データを 3 人で分担し確認したところ、紙面中央や右側に配置されている 4 コマ漫画も多く存在していることが分かった。座標指定で切り抜きを行ったとしても十分なデータを得られるほど紙面左側の 4 コマ漫画は存在していたが、画像認識の学習という点において物足りなさを感じ、機械学習で画像認識を行いたいという思いから、再度寺沢先生に相談した。そこで、寺沢先生から YOLO というツールとそれを使用して卒業研究を行った先輩方の卒業論文を紹介して

いただき、それらを参考資料として学習を進めた [13]。

（※文責: 辰己尚矢）

4.3.2.4　YOLO の環境構築・学習

YOLO でコンテンツ抽出を行うことが決定したため、まずは YOLO の環境構築を行うことにした。YOLO の環境構築について調べていたところ、YOLO の環境は anaconda などを使うローカルの環境と、Google Colaboratory を使うものの 2 種類があった。そこで私たちは情報やデータの共有が容易であることや角先生が本プロジェクト用に Google Colab Pro+ を契約してくださっていたことなどから、Google Colaboratory での環境構築を進めることに決定した。

その後実際に構築してみた YOLO の環境で、Web サイトにあったサンプルを動かしながら、YOLO についての学習を行っていった。その学習が一段落ついた段階で、次は大量にある新聞記事データを活かすために、このデータを使って YOLO を動かす方法について調べた。その過程で、これらのデータを使うためには、学習用のデータを使って YOLO で学習させる必要があること、それらの学習用のデータは自分たちでラベリングをし、作成しなければならないことがわかった。そこで私たちは、寺沢研究室の卒業生である北清敦也さんの卒業論文の 3.3.2 ラベリングや様々な Web サイトなどを参考にし、labelImg を使ってラベリングを行い、学習用のデータセットを作成することにした。そして、そのためにも labelImg の環境構築を調べながら行った。

（※文責: 一入悠貴）

4.3.2.5　YOLO を使用してコンテンツ抽出

初めに、3 か月分の紙面画像データの中から 4 コマ漫画の記載されている紙面のみを手作業で分別した。そこで得られた約 90 日分の画像データに対して、labelImg を用いてラベリングを行い、4 コマ漫画の学習用データセットを作成した。その学習用データを使用し、YOLO に学習を行わせた。そしてその学習後に、1 ヶ月分の紙面画像データを使用し、検出を行った。その結果、４コマ漫画以外のものを４コマ漫画と判断してしまうことが若干見受けられたが、Confidence score が 80 を超えていたものは精度よく４コマ漫画のみを検出することに成功していたため、Confidence score が 60 以上のもののみ検出結果として保存するように設定したところ比較的精度良く検出するようにできた。 最初の段階で 4 コマ漫画の認識精度が良かったため、他のコンテンツである天気情報と広告の抽出も加えて行うことにした。前回と同様に、紙面画像データの中から 4 コマ漫画と天気情報、広告の 3 つのコンテンツが同時に記載されている紙面のみを手作業で分別した。labelImg を用いて学習用データセットの作成を進めたが、抽出するコンテンツの数が増え、想定以上に時間がかかってしまうと判断し、一度約 30 日分のデータセットで機械学習を行うこととした。しかし、前回と同様に学習させてから検出を行ったものの、今回は精度が良いといえるような結果にはならなかった。まず４コマ漫画については前回行ったときよりも検出できていない物や間違って検出してしまっているものが多く、また Confidence score も低いものが多い結果となっていた。次に天気コーナーについてはそもそも検出できていないことが多く、検出できていたとしても Confidence score がとても低くなってしまっていた。広告については他の 2 つに比べて良く検出できていることが多かったが、４コマ漫画を広告として検出してしまっていたり、天気コーナーを広告として検出してしまうことも少なからずあった。このように誤検出が多かったり、Confidence score が低かったりなどして決して精度が高いとは言えない結果となってしまってい

た（図 4.15）。 精度が悪かった原因として、ラベル数に偏りがあったことが挙げられる。当時の北海道新聞では、紙面一面に対してのコンテンツ量の比率が 1（漫画）: 1（天気）: 3（広告）であり、そのすべてにラベル付けを行ってしまっていたことから、このような結果になってしまったと考えられる。この結果を踏まえて、広告のラベル数を 4 コマ漫画や天気情報と同じ数にし、ラベル数の比率が一律になるようにしたうえで学習用データセットを作成した。しかし、今回も前回同様に誤検出が多かったり、そもそも検出しないことが多かったりなど、精度が高いとは言えない結果となってしまった。ラベル数の比率を一律にしても、うまくいかなかった要因として、広告に固有の特徴が存在しないことによる誤認識が考えられる。4 コマ漫画であれば特有の縦長の矩形やコマ割り、天気情報であれば特有のタイトルや天気図などが影響し判別しやすい。しかし、広告の場合企業によって大きさも内容も違うため、新聞上のコンテンツという広い捉え方で見たときに特徴があるとは言えない。あるとすれば矩形であるということだけである。以上を踏まえて精度が上がらないこと、本プロジェクトで作成するアプリケーションでの活用方法が見出せないことから広告のコンテンツ抽出を断念することにした。その後、４コマ漫画と天気コーナーのみで学習を行った。そして、どちらも誤検出が少なく、検出ができていないことも少ない、また、Confidence score も高い結果が多かったため検出精度が悪くないものが多かった結果となった。ここでうまくいったものを抽出するという話になり、抽出する方法について調べ始めた。結果としては、4.2.6 でも記述した通り、–save-crop というコマンドを使うことで抽出に成功した。これを踏まえて、現段階のデータ量では精度高く抽出が可能であると判断し、データ量を増やすために年代を広げる作業を進めることとした。年代を広げるために古い紙面画像を取り扱うことから、レイアウトが特殊な 4 コマ漫画や 6 コマ漫画、8 コマ漫画などの今のままでは認識できないコンテンツが多く現れた。これらのコンテンツを抽出するために、6 コマ漫画は “MANGA6”、8 コマ漫画は “MANGA8” のようにそれらの特徴を踏まえたラベリングを行い、特殊な例専用の学習用データを作成した。それらの学習用データを使用し、YOLO での学習と検出を行ったところ、6 コマ漫画や 8 コマ漫画についても問題なく抽出を行うことができた。また、今までは抽出できていなかったような 4 コマ漫画や一部広告入りの天気コーナーについても高精度で検出を行うことができるようになった。

図 4.15: 漫画と広告の二重認識

（※文責: 辰己尚矢）

4.3.2.6　抽出したデータの整理

YOLO の学習済みデータを使って多くの年代のデータから４コマ漫画と天気図を抽出した。この時点では４コマ漫画と天気図としてそれぞれ YOLO が検出した状態でフォルダーに分けられている。このままではアプリケーションで利用しづらい形式であったため、手作業で抽出したデータの整理を行った。４コマ漫画は４コマ漫画ではないのにもかかわらず４コマ漫画であると誤検出されたデータを取り除いた。次に作品ごとにフォルダーに分類した。最後に作品ごとにデータを「作品名＋通し番号」という名前に変更した。ただし通し番号は掲載年月日の古い順に割り当てた時系列順である。天気図は誤検出されたデータが４コマ漫画に比べて多く、天気図によって分類する必要もなかったため、天気図のみを別のフォルダーに分類した。その後データを「tenki ＋通し番号」という名前に変更した。ただし通し番号は４コマ漫画と同様に時系列順である。このようにデータを整理することによってアプリケーションで４コマ漫画と天気図を使用する際に使い勝手が良くなる。例えば、４コマ漫画を作品ごとに表示したり、４コマ漫画や天気図を連続して表示したりする場合に通し番号があると扱いやすくなると考えられる。

（※文責: 高橋陽一）

4.3.3　フロントエンド班の開発過程

4.3.3.1　フロントエンドの開発概要

デザインとフロントエンドの開発は前田祥が一貫して担当した。これにより、デザインとフロントエンドの開発の認識に齟齬が出ず工数を意識したデザインを行い、時間内で目標であったアプリケーションの機能を実装することができた。

（※文責: 前田祥）

4.3.3.2　Web アプリケーションのデザイン

アプリケーションのデザインはコンポーネントを意識したデザインにした。これにより、ユーザーが使用する際にどの機能がどこにあるかが明確になり操作性が向上した。また、OOUI（Object Oriented User Interface）と呼ばれるオブジェクト指向 UI を導入した。これにより、少ない手順で目的の操作に切り替えることでユーザーの操作感に自由な印象を与えることができた。結果としてサイト内のアーキテクチャ構造が複雑化せず、はじめて利用するユーザーでもわかりやすいデザインとなった。また、当初は新聞風のデザインでの作成を想定していたが、工数が多くなり発表までに間に合わないと判断した為途中からコンポーネントを意識した工数の低いデザイン案を採用した。この判断により、発表日までに目標であった機能を開発することができた。

（※文責: 前田祥）

4.3.3.3　採用した開発手法

基本的に１人の開発にはなるが GitHub を用いてソースコードを管理しながら開発を行った。これによりバグが起こった際でもあらかじめ開発用のブランチを切ることでソースコードの保守性を高めた。開発のタスクやフィードバックを Notion に記録することでやるべきことを明確にして

開発に取り組むことができた。GitHub に慣れていないメンバーも居た為、GitHub の issue 機能を使うのではなく簡単に操作可能な Notion を採用した。これにより予定よりフロントエンド以外の開発を担当しているメンバーとのやりとりが促進され、予定されていた工数よりも早く実装を行うことができた。また、使用したライブラリは英語の文献が多かったため英語での情報検索に努めた。

（※文責: 前田祥）

4.3.3.4　開発言語の選定

Web アプリケーションを選択した理由として「API との接続が容易なこと」、「インタラクションを加えやすい」、「アクセスが容易」が挙げられる。Web アプリケーションを開発する上で、前田１人でフロントエンドを担当することとなった場合は他のメンバーの助けを借りたい場面も出てくる可能性がある。その時に JavaScript のフレームワークを使用して開発を進めていると学習コストが高くなる。よって、学習難易度が低い、HTML や CSS を開発に採用した Web アプリケーションを開発することとした。バックエンドから API を呼び出し取得する JSON データを操作してインタラクティブに操作可能な UI を開発する予定であった。その為、JavaScript で DOM を操作し可視化した物を表示するライブラリとして D3.js を選定した。D3.js はデータに基づいてドキュメントを操作する為の JavaScript ライブラリである。D3 は、HTML、SVG、CSS を使用してデータに生命を吹き込むように DOM を操作することが可能である。D3 は Web 標準を重視し、強力な視覚化コンポーネントと DOM 操作へのデータ駆動型アプローチを組み合わせることで、独自のフレームワークに縛られることなく、モダンブラウザの全機能を利用できるようになる。データ駆動型とは、データを元に次のアクションを決めたり、意思決定を行ったりすることである。以上より、D3.js の拡張性の高さからデザイン通りのインタラクションの再現が可能となった。

（※文責: 前田祥）

4.3.4　バックエンド班の開発過程

このプロジェクトの制作物が Web システムになるということは、夏休み明け早々に決まっていたため、バックエンドの構築作業とバックエンドに載せる API の開発を始めた。

（※文責: 川平覚士）

4.3.4.1　機材の選定

まずは、何にバックエンドを載せるのかということを決めるところから着手した。当初は AWS 上にバックエンドを載せることを考えていた。そうすると、インターネットへの公開が容易になり、発表会の時のデモンストレーションがやりやすくなる。AWS 使用のためには、大学との話し合いが必要であり、さらにシステム構成図が必要であるとのことだったので、その作成を行った。図 4.16 にそのときのシステム構成図を示す。

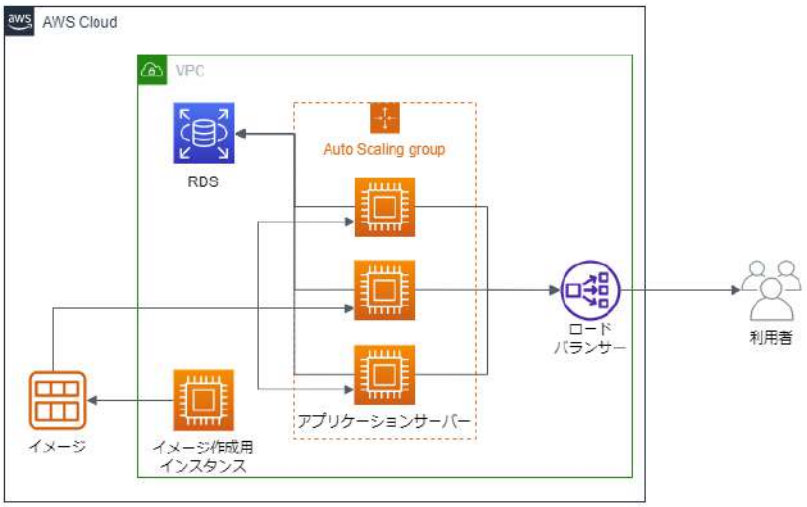


図 4.16: システム構成案

このときは、とにかく発表会の時に落ちることを防ぐことを最重要とし、同時に低コスト化を図ろうとしていた。そのために、サーバーを AutoScallingGroup 配下においている。こうすることによって、通常時は性能の低く、低コストなインスタンスのみを、そして、アクセス増加時にはその台数を増やし、システム全体の性能を向上させるという予定であった。

この構成図を持参し、大学との話し合いを行ったのだが、次のような指摘がされた。

- RDS は常時稼働状態となるためコストがかかりすぎる
- AutoScalling は不要ではないのか
- 講義用仮想マシンではダメなのか

当時は、開発がスタートした直後であったため、詳細な仕様が決まっておらず、これらの指摘に対する回答をすることができなかった。よって、詳細な仕様が決まり次第再び話し合いの場を開くということになった。しかし、AWS の利用が決定後、事務処理のために 4 週間は必要であるとのことであった。開発状況から、４週間前までの使用確定は不可能であることは明らかであったため、AWS の使用は断念することになった。最終的には、インターネットへの公開を諦め、角先生の研究室にあった Linux マシン上の Docker を用いて本番環境を構築することとなった。

（※文責: 川平覚士）

4.3.4.2　ソフトウェアの選定

Web アプリケーション業界には Linux、Apache、MySQL、PHP の頭文字をとった LAMP[14] という王道的な構成が存在する。しかし、開発期間が短いこと、機能拡張が容易である必要があることなどといった理由から LAMP は無視することとした。その結果、Web サーバー兼リバースプロキシには Nginx、アプリケーションサーバーには Nginx Unit、リレーショナルデータベースには MySQL を、そして API を実装するための言語には Python を用いた。

（※文責: 川平覚士）

4.3.4.3 Nginx を用いた理由

Apache には C10K 問題 [15] というクライアント数が 1 万を超えるとレスポンス性能が大きく低下する問題が存在する。今回のプロジェクトの制作物に対して 1 万以上のクライアントが発生することは考えにくいが、防げる問題を防ぐというのはとても重要なことである。

Nginx にはこの問題が存在しないため、Apache ではなく Nginx を使用することによって C10K 問題の対策になる。また、Nginx は静的サイトのレスポンス性能が Apache よりも高く、フロントエンドを静的サイトとして制作した本プロジェクトに最適であると考えられる。よって今回は、Apache ではなく Nginx を採用した。

（※文責: 川平覚士）

4.3.4.4 Nginx Unit を用いた理由

Nginx 単体では、静的サイトの配信しかすることができない。よって Python で作成した API を配信するためにはアプリケーションサーバーが必要である。

Python とともに使用するアプリケーションサーバーとして uWSGI が挙げられるが、今回は同じ F5 社開発ということで Nginx Unit を用いた。

（※文責: 川平覚士）

4.3.4.5 できあがったもの

今回、予定していたすべての機能を実装することはできなかったが、ある程度の機能を実装することはできた。その機能を紹介する。

ワードクラウド用の機能

ワードクラウドをフロントに表示するためには、図 4.17 のような json 形式のデータが求められた。


```
[
    {"date":"1994-07-01","word":"自民党","count":80},
    {"date":"1994-07-01","word":"社会党","count":79},
    {"date":"1994-07-01","word":"委員長","count":42},
    {"date":"1994-07-01","word":"課長補佐","count":34},
    {"date":"1994-07-01","word":"事務所","count":30}
]
```


図 4.17: ワードクラウドを表示するための json データの一例

データエンジニアリング班が新聞の記事データを元に生成した単語の頻出数のデータを元に、指定された期間に応じて図 4.17 に示す形式のデータを返す。また、図 4.18 のように頻出数のデータが存在している日付を一覧で返す機能も作成した。これは、元データとなる北海道新聞が時々休刊日を設けており、データが欠損していることの対策のための機能である。

```
[
        {"date":"1994-07-01"},
        {"date":"1994-07-02"},
        {"date":"1994-07-03"},
        {"date":"1994-07-04"},
        {"date":"1994-07-05"},
]
```


図 4.18: ワードクラウド用の解析データに基づいた日付の一覧の例

実装することができた機能は、ワードクラウド用の機能のみであるが、完成度と快適性を追求して様々な工夫を行った。まずは、データベース内ですべてのデータ操作を行うことである。今回、API はすべて Python で実装している。Python は C 言語や Go 言語と違い、インタプリタ方式の言語であるため、動作は低速である [16]。そのため、データベースからすべてのデータを取得し、データの加工を行うとなると実行時間が長期化しユーザーの快適性が損なわれてしまうし、何よりデータベースを用いる意味がなくなってしまう。よって、データの期間選択といった処理をすべてデータベース上で行うこととした。そして、レスポンスのキャッシュをサーバーに設定した。今回は、どのユーザーが同じリクエストを送っても、すべて同じレスポンスを返す仕様であるため、一々レスポンスを生成するのは単純な無駄である。よって今回は、Redis を導入した。Redis とはオンメモリディクショナリサーバーである [17]。どういうことかというと、メモリ上に Python でいうところの辞書型を蓄積し続けるソフトウェアである。メモリ上に蓄積するということは電源を落とすと消滅してしまうが、ストレージに保存することもできる。よって、データベース的な使い方をすることがあるようだが、今回はストレージへの保存は一切せずに純粋なオンメモリキャッシュとして用いた。Redis 導入と導入前を比較すると最高で 10 秒程度レスポンスが早くなっており、導入の効果はあったと考えられる。

（※文責: 川平覚士）

第 5 章　成果

5.1　成果物の概要

本プロジェクトでは、「新聞ビッグデータから『何か』を生み出す」というテーマに基づき、新聞のテキストデータを可視化し知的好奇心を刺激する “View Picks” という Web アプリケーションを開発した。この “View Picks” は、新聞のテキストデータから新聞内における情報の強弱や関係性をインタラクティブに可視化し知的好奇心を刺激する体験を提供するものである。本プロジェクトの開発目的であった「ただ新聞記事を読むという方法以外で、新聞に含まれている情報を利用者に提供する。」や「情報について、様々な方法での可視化を通して利用者に新聞に秘められていた情報の面白さを理解してもらう。」「知らなかった情報を知ることができる機会を作り、知的好奇心を刺激できるようなものを作る。」を達成するために作成したものであり、新聞に眠る情報の価値を引き出し、 マスコミュニケーションの手段としての新聞ではなく教育への活用や知的好奇心を刺激する方向へと導くなどの目的に通じる Web アプリケーションである。

また、この Web アプリケーションの構成は図 5.1 の通りとなっている。この図では View Picks を使ったときのリクエストやレスポンスの流れやデータの流れ、フロントエンド班、バックエンド班、データエンジニアリング班の担当したところなどがわかるものとなっている。View Picks ではデータエンジニアリング班が整形したデータをもとに、それらのデータをバックエンド班が Web アプリケーションで使いやすいようにし、フロントエンド班がそれらのデータを可視化するなどして、Web アプリケーションを作成した。

図 5.2 は Web アプリケーションのホーム画面、図 5.3 は Web アプリケーションの可視化の画面である。Web アプリケーションのホーム画面には、各可視化のイメージ画像と可視化の概要、各可視化へのリンクを表示している。Web アプリケーションの可視化の画面では、日付の区間を選択できる UI と可視化されたグラフィックを表示する領域が表示されている。日付の区間を選択することでその日付に対応した新聞のテキストデータが可視化される仕組みになっている。

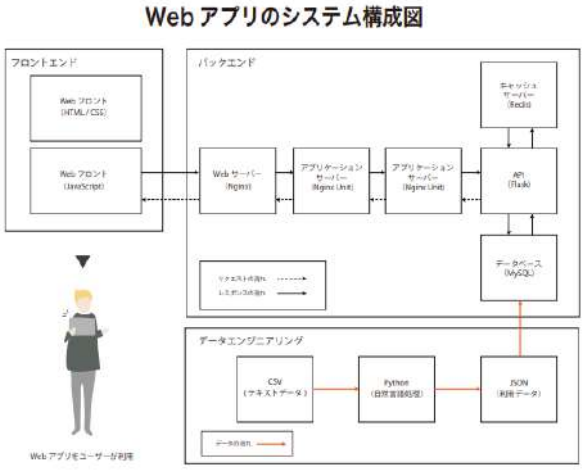


図 5.1: システム構成図

図 5.2: Web アプリケーションのホーム画面

図 5.3: Web アプリケーションの可視化画面

（※文責: 一入悠貴）

5.2 成果物の各機能・目的

5.2.1 新聞記事データの様々な方法での可視化

Web アプリケーションで表示することができる可視化方法は 3 種類である。

（※文責: 前田祥）

5.2.1.1 WordCloud（ワードクラウド）

一つ目の可視化方法は図 5.4 に示す Word Cloud（ワードクラウド）である。ワードクラウドとは、文章中で出現頻度が高い単語を複数選び出し、その頻度に応じた大きさで図示する手法である。ウェブページやブログなどに頻出する単語を自動的に並べることなどを指す。文字の大きさだけでなく、色、字体、向きに変化をつけることで、文章の内容をひと目で印象づけることができる。今回の Web アプリケーションでは、北海道新聞に頻出する単語をワードクラウドで表示する方法を取った。この可視化方法を行うことで、指定した日付に話題になった単語や話題にならなかった単語を知ることができ、注目されていた時事を視覚的に分かりやすく知ることができる。記事のジャンルで色分けを行い、頻出度合いで大きさを変化させる予定であった。頻出度合いは実装すること

ができたが、ジャンル別での色分けは実装に至らなかった。

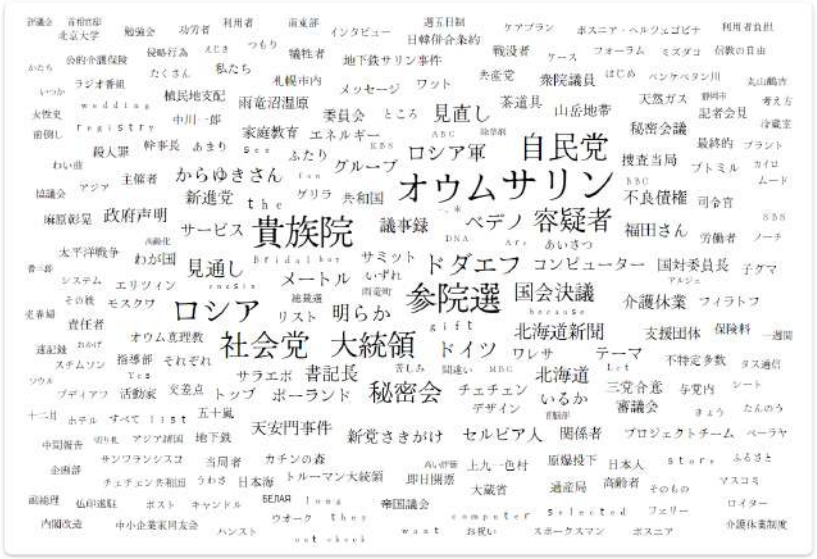


図 5.4: 1995 年 6 月 5 日から 2000 年 4 月 9 日でのワードクラウド

（※文責: 前田祥）

5.2.1.2 Circle Packing（サークルパッキング）

二つ目の可視化方法は図 5.5 に示す Circle Packing（サークルパッキング）である。サークルパッキングとは、与えられた表面上に様々な大きさの円を重なりが生じないように充填する可視化方法である。開発した Web アプリケーションでは新聞中に出現する言葉の関連度に着目した。例えば、新聞中で「北海道」という単語と関連した「洞爺湖」という言葉があった場合、「北海道」に関する円の中に「洞爺湖」に関連する円が内包されている。今回開発した Web アプリケーションでは円をクリックすることによって表示する階層の移動を行うことができ、新聞中の単語感の内包関係を知りやすくすることができる。図 5.6 は「洞爺湖」という言葉に紐づけられた円クリックし、階層が一段階深くなった後の Circle Packing である。このことより「洞爺湖」という言葉には「駒ヶ岳」、「海洋温度差発言」という二つの言葉が関連していることが分かる。ジャンル別に色分けを行う予定であったが実装には至らなかった。

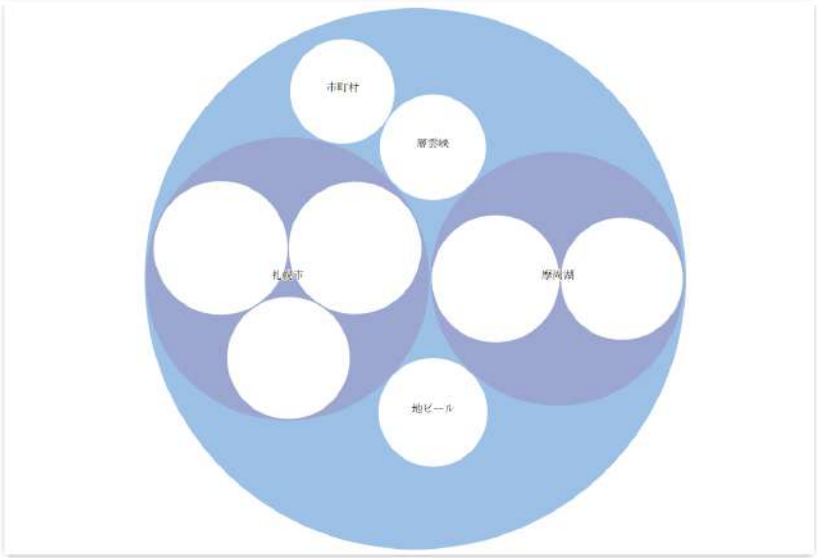


図 5.5: Circle Packing

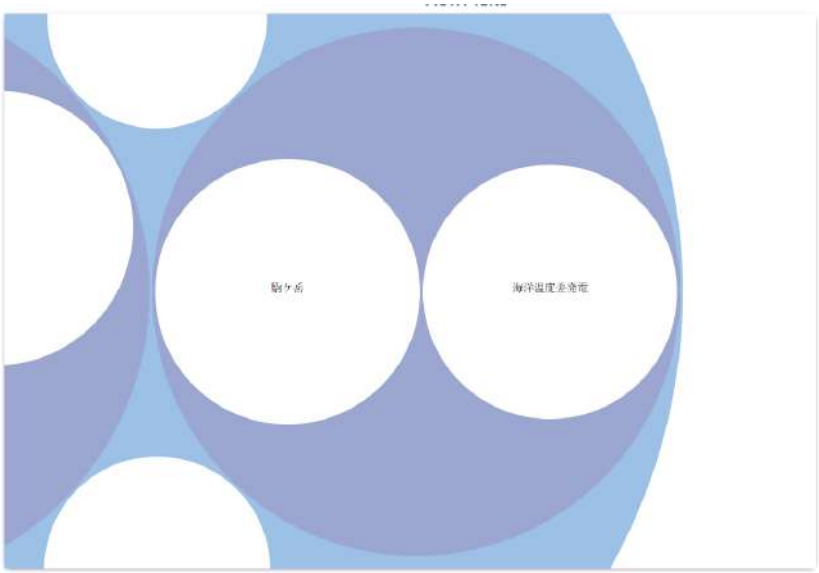


図 5.6: 洞爺湖をクリックし表示される階層が一段階深くなった Circle Packing

（※文責: 前田祥）

5.2.1.3 WordCloud（ワードクラウド）

三つ目の可視化方法は、図 5.7 で示す Tree（ツリー）である。ツリーとは、階層構造を可視化する方法である。今回開発した Web アプリケーションでは、単語感の関連度をツリーで可視化した。サークルパッキングと異なる点は、単純な線のみで単語の関係性が表されていることである。サークルパッキングでは円の色などで単語のジャンルなどを表すことができる。しかし、ツリーでは、ジャンルごとの色分けが難しい代わりに関連度が見やすいというメリットがある。サークルパッキングやツリーのような階層構造を表した可視化を通して新聞中に出現した単語を関連付けることで Web アプリケーション中で話題の派生に繋がることが予想される。

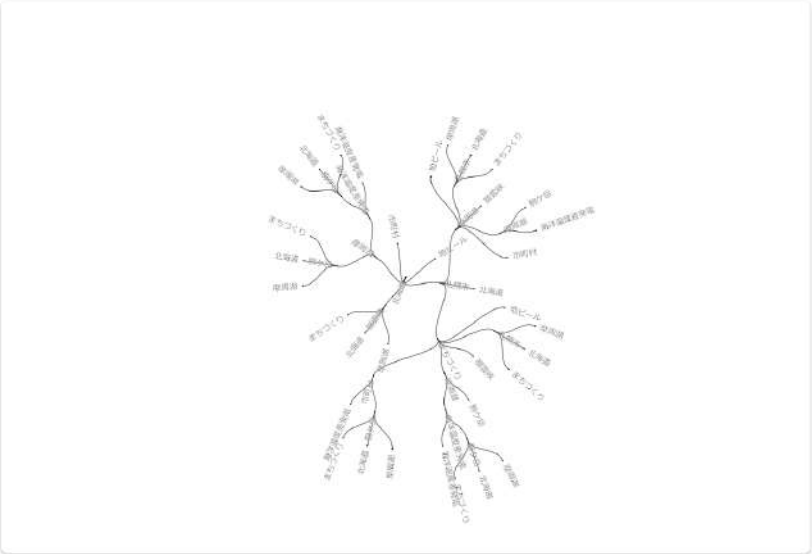


図 5.7: Tree

（※文責: 前田祥）

5.2.2 その他機能について

その他の機能としてはワードクラウドで出現した単語をクリックすると関連した記事の内容が表示されるなどの機能を想定していたが実装が間に合わなかった。また、日付の区間指定が現在はサーバーのスペックを考慮し 5 年と設定してあるが今後はサーバーを強化し 30 年分の区間を指定できるようにすることが目標の 1 つである。教育機関での使用を視野に入れているため、ワードクラウドで出現した単語にホバーすると漢字にルビが振られるなどの機能も想定している。

（※文責: 前田祥）

第 6 章　まとめ

6.1　目的達成度

本プロジェクトは、アプリケーションを作成し、それが新聞に眠る情報の価値を引き出すものであり、マスコミュニケーションの手段としての新聞ではなく教育への活用や知的好奇心を刺激する方向へと導き、新聞の新しい使い方を提案する可能性を示唆することを目的としていた。

そして私たちはその目的を達成できたと考えている。View Picks を使い、情報を様々な方法で可視化したことにより、普通に新聞の紙面を読むことで得られる情報とはまた違った情報を手に入れることができたためである。また、様々な方法で可視化を通じて、知的好奇心をくすぐることにも成功しており、これは、発表会のときにデモ機を使って発表を聞いてくれた人に実際にアプリを動かしてもらったときのリアクションからも明らかであったと判断した。

さらには、そのリアクションから、子ども達にも同じように知的好奇心を刺激し、昔の情報にも興味を持ってもらうことも可能であると考えており、それらのことから、歴史的な観点や情報的な観点からの教育への利用もできそうだと感じた。これらの点から私たちは本プロジェクトの目的を達成できたと考えている。

しかし達成できたといっても、完璧に目的を達成できたというわけではなく、まだまだ改善できるような点も多く感じた。例えば、アプリの可視化表現の種類である。今現在実装できているものは３つであり、これらの機能だけでも、発表会では確かな手ごたえを感じた。しかし、その３つでは興味を持つことができない人がいることや、これらでも興味を持つことができたが、他の可視化表現の方がより知的好奇心を重資される人がいることも考えられるため、これ以上に可視化表現の種類を増加させていかなければならない。

また、可視化表現機能の深掘りも必要だと感じた。例えば可視化表現機能の１つであるワードクラウドであれば、今の機能としては、指定した期間に多く登場していた単語を知ることができる。このことから、それらの時代にどの様なことが起きていたり、流行っていたりしたのかを推測することができる。しかし、今の機能ではそこまでしかすることができなく、推測した事象が実際にあったのかを確かめることができないため、そこから先はこのアプリのユーザに調べるかどうかを委ねることしかできないのである。この問題は、刺激した知的好奇心の行き場を失わせてしまい、わからないならいいやといった思考にさせてしまうことで情報への興味を失わせてしまう可能性があると考えられる。そこでこの問題点を解決する方法の１つとして、ワードクラウドで表示された単語を指定することで、その単語が使われている記事データを表示するというものが提案されたが、その機能などはまだ実装できていない。このように各種機能をより良いものに発展させるため可視化表現機能の深掘りも必要だと考えた。

そのためこれらのことから、現段階の本プロジェクトの目的達成度としては、完璧とは言えないものの、当初の目的は達成したという結論に至った。

（※文責: 一入悠貴）

6.2 振り返り

6.2.1 前田祥の振り返り

メディアへの理解

メディアの未来 [18] を読み、メディアとは何なのかということについて理解を深めた。夏季長期休暇を通して再度、読み直しを行い更なる理解に繋げる予定である。

プロジェクトリーダ

プロジェクトリーダーになり、全体の進捗管理や率先してメンバーを引っ張って行くことを経験した。所属や性格特性が異なる個人をまとめあげることが、苦労した点の 1 つである。一番苦労したことは全体の役割り振り分けである。各々のスキルや特性を見極め、配分することは非常に難しい。なぜなら、マネジメント担当者が技術的な知識に精通して居なければ工数の想定判断が難しいからである。また、コミュニケーションをとれる手段を構築することの重要性を再認識した。メンバー内で意見がある場合はその場で意見できるように場を構築することがプロジェクト内での生産性を向上させる重要なポイントであったと前期が終わった段階で感じたことである。後期からは各々の役割が明確化し、それぞれのタスクに奮闘した。全体での認識を合わせる為に、プロジェクトの終わりには進捗を共有する場を積極的に設けた。その結果、皆の進捗を把握しながらお互いが開発や作業に取り組むことができ開発が円滑化した。

ライティング技術

前期には、角康之教授が主催のワークショップを通して分かりやすいポスターとわかりにくいポスターの違いを学習した。 事前に知識がない分野のポスターを見る限り専門用語が多く内容がわからない物が大半であった。このことから読者の UX を意識した書き方を行うようにした。具体的には専門用語には知識がなくとも伝わる様に図や解説文を添付したものである。最終成果発表会では、ポスターの作り方を工夫し新聞のエディトリアルデザインを踏襲した。その際に小見出しのつけ方などを聴衆が目に入りやすいような言葉にし、興味を引くようにした。

論文の読み方

プロジェクトリーダーになり役割分担の決定を行う上で自己の学習が不可欠であった。そこで自然言語処理や情報の可視化、新聞の歴史などを調べていく中で論文に触れる機会が数多くあった。論文を読むに当たって、全体の要約と結論に目を通した後に実験内容や調査内容を見ることによって内容を理解する時間が短縮された。 また、論文内で出てきた未知の単語の意味を直ぐに調べることで身になる理解を得ることができた。

プロトタイプの開発

チームでの開発を行う際にスクラム開発を導入した。小さなプロトタイプを何度も開発し、ライブラリーや API が正常に動作しているかを確認しながら開発を進めた。スクラム開発を採用した結果、バグがある際に直ぐに修正に移ることができ、開発工数の削減に繋がった。また、スプリントは Notion で管理し、バージョンは GitHub で管理した。このおかげで口頭でのコミュニケーションに加え各自が自宅で開発する際も円滑にオンライン上でコミュニケーションを行いながら開発を進めることができた。

プレゼンテーション技術

練習風景を動画撮影し、見返すことで自身の至らない部分の早期発見に繋がると共に、メタ的に認知できることから改善が行いやすくなった。プレゼンテーションでは、自身の技術向上だけでなくメンバーの発表の様子なども観察し互いに建設的なフィードバックを出すことでお互いの発表技術向上に力を入れた。その結果、前期の中間発表と比べ、発表技術の点数が向上していた。

Web アプリケーションの開発

北海道新聞に含まれる 30 年分の膨大なテキストデータを自然減処理で形態素解析し API に活用した。また、API を格納する場合、サーバーが必要になる。私は Web アプリケーションのフロントエンド部分を担当した為、API を活用する必要があった。バックエンド担当と認識をすり合わせどのような API が必要かの議論を交わし開発を協同で進めた。その結果、認識の齟齬が無く開発を進めることができた。また、API の開発をバックエンド担当に頼む前に使用するライブラリである D3.js の仕様を確認するなど開発途中で工数が増えないように事前の準備を丁寧に行った。しかし、GitHub で一元的に管理したほうが情報の整理が行いやすかったという失敗もあったので今後は GitHub を活用してスプリントを管理するなど意識しようと思えた。

（※文責: 前田祥）

6.2.2 辰己尚矢の振り返り

前期の活動全体

プロジェクトを通して各メンバーとの積極的なコミュニケーションに努めることができた。これによりメンバー間のコミュニケーションも活発になり、活動時間を有意義に過ごせたと感じている。発表スライドやアプリケーションデモの作成では、一貫性をもって作成することを心がけた。新聞を扱っているため、全体的な色味を無彩色で統一した。これを発表スライド、アプリケーションデモともに適用することで閲覧者の認知的負荷の軽減を目指した。成果物の作成では、データの可視化におけるインタラクティブなグラフを作成した。進捗をメンバーに逐一報告することで、客観的な改善案を取り入れることができたため、グラフの作成はスムーズに達成することができた。しかしデータのリスト化に関する知識が乏しかったため、現時点では実用的なグラフとは言えない。今後は機能の実用化を目指してアプリの改善を行いたいと考えている。

サブリーダー

私は本プロジェクトにおいてサブリーダーというリーダーを補佐する役割を担った。前期では 4 人グループのまとめ役を担い、活発な議論になることを狙って、積極的なアイデア出しに努めた。この際に考案されたアイデアが今プロジェクトで取り組んでいるアプリケーションの元手となっている。しかし、別グループとしてアイデア出しをしていた 4 人に、作成するアプリケーションのイメージが伝わりきっていないと感じることがあった。後期では反省点として改善していきたい。

中間発表に向けた発表スライドとアプリケーションデモの作成

中間発表における発表スライドとアプリケーションデモを作成した。これらの作成には Figma[*2] という Web デザインツールを使用した。Figma を使用した理由は、発表スライドとアプリケーションデモを同時に作成でき、発表スライドにアプリケーションデモを組み込むことができるからだ。

発表スライド作成では、新聞を扱っているため全体的な色味を無彩色で統一した。スライド上部にプロジェクト名を記載し、その左下にスライドの見出しと本文、その右に図や画像などを配置した（図 6.1）。このフォーマットを軸としてスライドを構成し、統一することで閲覧者の認知的負荷の軽減を目指した。内容としては、初めにプロジェクトの概要からやりたいことの概要、具体例まで説明し、その直後にアプリケーションデモを行うように構成した。これにより、スライドだけでは理解しきれなかった部分をアプリケーションデモで補うことができ、やりたいことが伝わりやすくなると考えたためだ。その後は、やりたいことと既存のサービスとの比較を行い、現在行っている技術的な取り組みと今後の課題を説明して発表終了とした。このように構成した理由としては、アプリケーションデモによってやりたいことを明確にしてから、既存のサービスと比較することで違いを理解しやすいと考えたためである。加えて、現在行っている技術的な取り組みと今後の課題を並べることで、今後の課題の内容が何についての内容なのかを理解しやすくなると考えたためである。

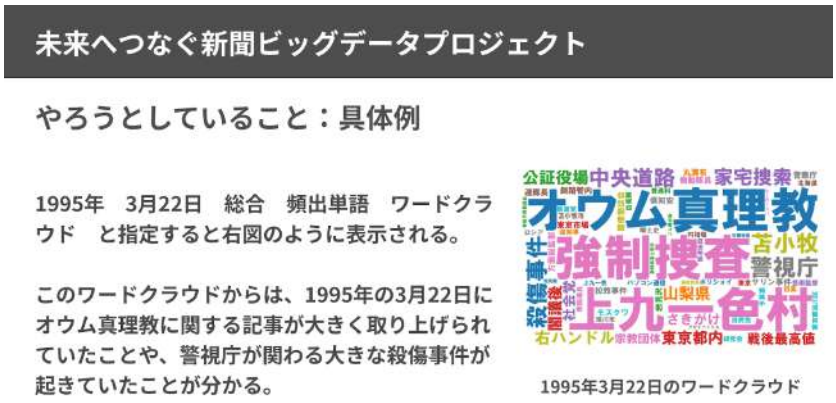


図 6.1: 発表スライド

アプリケーションデモ作成では、見た目でのわかりやすさややりたいことが伝わりやすいかどうかを重要視した。こちらも発表スライド同様、新聞を扱っているという理由から全体的な色味を無彩色で統一した（図 6.2）。これにより画面左部のアイコンが視認しやすくなり、認知的負荷の軽減につながったと考えられる。アプリケーション UI に関しては、Scratch[*3]というプログラミング教育で使用される、ビジュアルプログラミング言語を参考にした。想定されるアプリケーションの動きを伝えるという点に着目した際に、小学生でも扱えるようなビジュアル的に簡略化された UI を目指すことで、認知的負荷を軽減し内容の理解への誘導を目指すことができると考えたためである。

[*2] Figma. The Collaborative Interface Design Tool https://www.figma.com
[*3] Scratch. 世界最大の子ども向けコーディングコミュニティ https://scratch.mit.edu

図 6.2: アプリケーションデモ画面

後期の活動全体

前期での反省を踏まえて、機能の実用化に向けた勉強を夏休み期間に行った。Python の知識・技術向上のため、Python 実践データ加工/可視化 100 本ノックという本を参考に学習を進めた。加えて、グラフと可視化に関する論文を 2 本、新聞情報の信頼性を分析する論文を 1 本読むことで学習の進め方と方向性を定めることができた。後期活動開始後はフロントエンド班としての活動を予定していたが、画像データの活用に興味が沸いたためコンテンツ抽出班として新たに活動を進めた。画像認識に関する知識が全くない状態でありながらも、自分を含めた 3 人の班員とお互いに意見交換を行いながら学習を進められたため、画像認識に対する困難な印象が大きくやわらぎ、学習を楽しめるようになった。上記の活動に加えて、最終発表に向けた発表スライド作成を担当した。中間発表で使用したスライドをもとに作成を進め、中間発表スライドと同様に全体的な色味を無彩色で統一した。仮提出の度に受ける指摘をスライドに反映しつつ、先生だけではなく班員にも意見を求めることでよりよいスライド作成に努めることができた。

コンテンツ抽出班での活動

labelImg を用いたラベリングでは、抽出するコンテンツごとの適切なクラス付けや丁寧な範囲指定、作業の高速化を意識して作業を進めた。範囲指定が原因となって起こる誤認識を可能な限り減らすため、抽出するコンテンツとは関係のない情報が含まれないように丁寧な作業を意識した。しかし、作業を丁寧に進める影響でラベリング待ちの時間が生まれてしまい、効率が悪いと判断したため高速化に向けてショートカットキーの活用や設定の見直しを行った。これにより丁寧な作業を維持したうえで高速化することに成功した。

最終発表に向けたスライドの作成

新聞を扱っているため、中間発表スライドと同様に全体的な色味を無彩色で統一した。第 1 回仮提出では文字が多いという指摘を受け、班員からの意見を積極的に取り入れながら改善に努めた。イラストなどを活用して話す内容の可視化を行ったり、話す文章をそのままスライドに記載するのではなく要約した文を箇条書きで並べるように工夫した。これによってスライド全体がすっきりとした印象になり、仮提出前と比べて情報量を減少させることができた。第 2 回仮提出では、イラス

トの使い方やスライド上での視線誘導に関して、美馬のゆり先生と先生の研究室の方に直接指導を受けた。使用するイラストが統一されていることや見てほしい場所を強調表示することなど、実際に先生が作成したスライドを見ながら指導を受けた。それらを踏まえて、イラストの統一と視線誘導を意識したスライドを作成し、見るべき場所と話している内容がわかりやすいスライドを作成することができた。最終発表後の評価シートでスライドが新聞風で面白いというコメントがあり、テーマとしてこだわって良かったと感じた。

（※文責: 辰己尚矢）

6.2.3 藤島海陸の振り返り

前期の活動

私は最初、新聞記事内の文章から固有名詞を抽出し、その取り出した言葉を使って観光アプリや方言変換システム、そして言葉遊びといったものを作りたいと考えていた。そのため、まず初めに、固有表現の抽出を行うことに力を入れた。IREX や「関根の拡張固有表現階層」で定義されている固有表現の種類について調べ、その固有表現を抽出する方法をパターン化して考えた。そして次に、固有表現抽出を行うために必要なスキルを習得するため、学内の図書室で自然言語処理に関する本を 2 冊借りて読んだ。その本に書かれていることを実行するためには、形態素解析を行うためのツールであ “Mecab” のインストールが必須であったため、環境構築をしたうえで、python 上に Mecab をインストールをした。Mecab で形態素解析ができるようになった後は、さまざまな文章を実際に読み込ませて形態素解析を行っていった。また、実際にアプリケーションとして単語を活用していくためには、“わかち書き” という作業が必要だということが分かったので、わかち書きを自動でしてくれるような関数を作成した。形態素解析が一通りできるようになった後は、ワードクラウドやアスキーアートなどといった、抽出した単語をどう扱うのか、という面について考え始めた。ワードクラウドについては、中間発表のデモンストレーションに使用することができたが、アスキーアートについては、細部まで表現することが難しかったり、その記事を特徴づけるような単語に関連した画像を自動で引っ張ってくることが技術的に困難であったため、中間発表までに実装することはできなかった。次に、アスキーアートを作成するときに行き詰った部分である、特徴づけるような単語を抽出する技術について考え始めた。特徴づけるような単語、つまり、“重要” を記事内から取り出すためには、tf-idf 分析を行う必要があることが分かった。この分析を行うことができれば、新聞記事データから流行語を予測する機能を作れるのではないかと考えたため、そのための手順や作成する際の問題点を考察した。

夏期休暇中の活動

夏季休暇開始段階では、頻出単語をワードクラウドで表現するという 1 つの機能しかなかった。多機能化できていない主な原因としては、新聞に対する理解が浅いことや、そもそも新聞ビッグデータを生かせるような機能にどのようなものがあるか分かっていないことがあると考えた。そこで夏期休暇中の個人活動は、追加機能を考えて、必要な技術を学習するといった目的で進めた。具体的に行ったことは以下の 3 つである。

(1) 新聞に関する本を 2 冊読む。
(2) 新聞に関する論文を 1 つ読む。
(3) インターンシップで学んだ内容を本プロジェクトに生かせないかを考える。

（1）では、「新聞の嘘を見抜く」という本を読んだ。これを読んだことで、断定調の報道に注意することや、世論調査の質問内容や結果に注意するなどといった新聞の嘘を見抜く上でのポイントを抑えることができたが、実際にこれを新聞記事データに反映させて嘘を抜き出すことは難しいと考えた。

さらに（1）では、「ニュースの多様性とは何か」という本も読んだ。これを読んだことで、そもそもニュースにおける多様性とは何かといったことや、実際のニュースの多様度を測った事例を学ぶことができた。しかし、4.3.1.5 でも述べられているように、かなり複雑で困難だという結論に達した。

また、（2）では、ブロック紙である北海道新聞の良さを何か引き出すことはできないかということで、「地方紙の現状と課題」という論文を読んだ。若者の新聞離れを抑えるために、全国紙との差別化を行っている新聞社の具体的な取り組みについて学ぶことができた。これにより、我々の成果物もまだ世に出ていない新規性のあるものを作成することで、差別化し、若者の興味を引けるのではないかと考えた。

そして最後にインターンシップで学んだ内容を本プロジェクトに生かせないかを考えた。私はインターンシップでウォーターフォール型のソフトウェア開発を行ったが、Web アプリを作成する本プロジェクトでは、そもそもウォーターフォール型では取り組んでいなかったため、直接取り入れられることはなかったように思う。

以上のことから、学んだことを実際に開発へと反映させることはできなかったが、新聞が置かれている現状を学ぶことができた。また、新聞記事の嘘を見抜いたり、多様性を表現することを取り入れることをあきらめたが、時間があれば実際に取り組んでみたいと思った。

後期の活動

後期では、データエンジニアリング班のスクラムマスターとして活動し、データの整形や軽量化、そして自然言語処理など、北海道新聞社様から頂いたテキストデータの処理をメインに担当した。また、スクラムマスターということもあり、開発だけでなく、データエンジニアリング班の他のメンバーの進捗確認やサポートといったマネージメントも行った。開発を行う中で自分が主に担当したのは、関連語を抽出するプログラムを作成することであった。関連語に関する具体的な説明は 4.3.1.5 でされているが、開発段階でさまざまな課題があったため、それらを乗り越えるために自分は主に 4 つの工夫を行った。

一つ目は、形態素解析を行うための辞書として、mecab-unidic-NEologd を用いたことである。形態素解析を行うためには、辞書を作成、もしくはインターネット上にある辞書を使用する必要があるが、自作することはかなり困難であると判断したため、後者を採用することにした。インターネット上にある辞書を使用するとしても、mecab-ipadic や macab-naist-jdic、そして UniDic-mecab など様々な辞書が存在するため、自分たちの目的や用途に合った辞書を選定し、利用していく必要があった。その中で私は mecab-unidic-NEologd を使うことにした。その理由は、我々のプロジェクトが使用する新聞には、かなりの語彙数があり、様々な固有名詞もたくさん使われているため、これらに最も対応できる辞書が上記のものであったからである。[19] によると、Unidic は、文字ベースの誤り率が Ipadic の半分程度、かつ未知語の数が Ipadic の 1/5〜1/4、さらには、ブログ・文学作品・新聞といった全てのジャンルで辞書の解析精度が約 98 %維持していたという。これらのことから私は形態素解析を行うための辞書として、mecab-unidic-NEologd を用いるという工夫を行った。

二つ目は、単語の類似度を求める際に使用する日本語の学習済み word2vec モデルとして、白ヤ

ギコーポレーションのモデルを用いたことである [20]。この word2vec モデルも、上記で説明した辞書と同じように、インターネット上に様々なものが存在する。主なものとしては、エンティティベクトル、白ヤギ、chiVe、fastText などが挙げられる。各モデルにはそれぞれメリット・デメリットがあったため、それらをしっかり考慮したうえで、私は白ヤギコーポレーションのモデルを用いた。白ヤギコーポレーションのモデルには、未知語が多く、精度が低めであるが、ベクトルの次元数が他のモデルより小さいという特徴がある。過去約 30 年分の大量のテキストデータを扱っている本プロジェクトにおいて、他の word2vec モデルはサイズが大きすぎるため、処理にかなりの時間を要してしまう。しかし、白ヤギはモデルサイズがほかのものよりも小さく、データ処理に時間がかかりにくいというメリットがあったため、これを採用するという工夫を行った。

三つ目は、ファイルパスの指定に glob モジュールを用いたことである。テキストデータを Google Colaboratory 上で使用する際には、マウントされた Google Drive 上から CSV ファイルを持ってきて、そのファイルパスをプログラムに書く必要があったが、開発初期はそのパスを全部記載していたため、非常に冗長で時間がかかっていた。そこで、昨年新聞ビッグデータプロジェクトのメンバーであった方に解決策をお聞きしたところ、glob モジュールの存在を知った。その先輩も昨年プログラムを作成する際に使用していたとのことだったため、提供していただいたソースコードを参考にしながら、新たに取り入れた。その結果、かなり短いコードで済むようになったので、スッキリさせることができた。

四つ目は、コサイン類似度を求める際に、numpy モジュールの numpy.dot() と、numpy.norm() を用いたことである。コサイン類似度を求める方法としては、model.wv.similarity(word1,word2) で求める方法と、numpy モジュールで求めるという 2 パターンがあると考えていた。そのため、初期は前者を採用していたが、実行にかなりの時間がかかってしまい、時には自分の Google Colaboratory がフリーズしてしまうということもあった。そこで、解決策として numpy モジュールを使用することにしたが、結果的に全体の処理時間を 1/3 程度まで短縮することができたため、よりスムーズな開発を行えるようになった。　これらの工夫を行ったことにより、抱えていた問題を解決できた。また結果的に、より質の高い成果物を作成することに貢献できたと考える。

（※文責: 藤島海陸）

6.2.4　遠藤晴人の振り返り

前期の活動

前期では、グループの目的や成果物の詳細を決めることや、新聞の意味や意義を考えることが議論の中心であった。「新聞のビッグデータを使用し、何らかのサービスを作成する」という大まかな方針のもとアイデア出しを行い、成果物として作成するものを決めていった。各アイデアにおいて、良いと感じた箇所や、既存のサービスがないか、ユーザの興味を惹けるか、といった部分などを素直に発言できる環境になっていたことで、より洗練されたアイデアが出るようになったと感じた。アプリの概観が決定してからは、北海道新聞社から頂いたテキストデータをどのように使えるのかに興味を持ち、開発を進めていった。元の csv ファイル形式だとデータが使いにくいように感じたため、テキストデータをプログラム内でリスト化するなどの前処理を行い、リストのインデックスや要素名による管理ができるようにした。その後、Wordcloud を作るという案が出たため、WordCloud を作成するためのプログラムを完成させ、その後記事の日付指定機能や、より精度の高い辞書の導入などを行った。またそれらを複合させ、中間発表会でのデモンストレーション

で使用する、各評価担当者の誕生日の記事で Wordcloud を作成できるプログラムを開発した。

夏季休業中の活動

前期での活動では、プロジェクトとしての成果物が JupyterNotebook 内で動作する WordCloud の表示プログラムのみだった。この時の最終目標は、ユーザの好みの可視化方法で新聞を見られる Web アプリケーションを作ることだったので、このままでは後期の作業量が膨大になると考え、夏季休業中もプロジェクトに関連する作業を進めた。

私が夏季休業中に行ったことは 2 つあり、1 つ目は、新聞に関する知識を深めることであった。中間発表の際、新聞に対する理解が足りていないのではないかという指摘を受け、またその自覚もあったため、夏季休業中の時間を利用して新聞についての知見を得て、後期の活動に活かすために取り組んだ。具体的には、「2050 年のメディア」という本を読んだ。この本は、読売新聞・日経新聞・Yahoo の三社を中心的に取り上げて 2000 年代初頭からのメディア動向の歴史が分かる本で、新聞の発行部数がどんどん少なくなる中ネットメディアが勢いを増してくる様子を、様々な立場の人の視点から記した内容であった。

2 つ目は、アプリケーション開発の準備・練習をすることであった。夏季休業に入った段階では、ローカル環境で動くプログラムしかできておらず、アプリケーション開発のためのフレームワークは決まっていなかった。そこで、アプリケーション開発に慣れておこうと考え、練習として Django を用いた Python での開発を行った。学習には「Python ではじめる機械学習」、「Python Django3 超入門」などの参考書を用いて、Python のプログラムや実行結果を Web アプリケーションに落とし込む方法などを学んだ。結果的に成果物には Django は使用しなかったが、Web アプリケーションについての理解を深めるいい機会になったと感じた。

後期の活動

後期では、中間発表で指摘された部分を改善するため、まずアプリケーションの練り直しを行った。今後のアプリケーションの方向性として、ユーザの好みの可視化方法で新聞がみられる Web アプリケーションを作成することとなった。この時点では、範囲指定できる WordCloud と単語の登場回数の折れ線グラフの実装を目標とした。フロントエンド班、バックエンド班、データエンジニアリング班、画像処理班と班分けをし、各グループで作業を行った。私はデータエンジニアリング班として、大きく分けて 3 つの活動に取り組んだ。

1 つ目は、前期に作成したプログラムの改良である。前期では、中間発表に間に合わせるため愚直なプログラムを多く書いており、実行時間・可読性ともに問題があった。後期では、変数名の見直しやコメントの追加、実行時間を短くするアルゴリズムの実装、機能の関数化などを用いて、他のメンバーが理解でき応用できるプログラムに書き換えた。この作業でデータエンジニアリング班で使用するプログラムの基礎が完成し、今後の機能追加がしやすくなったと考える。

2 つ目は、単語の頻出度リストを json 型式で出力させたことだ。詳細については 4.3.1.4 にて説明したが、WordCloud 作成の実行時間を短くするために、膨大なデータ量の CSV ファイルの中から必要な情報のみを抽出し、json 型式のデータへと置き換えた。

3 つ目は、同班メンバーのサポートをしたことだ。前述した基礎となるプログラムをメンバーに改良・応用して貰う際、変数名やコメントだけではプログラムの動きや入っている情報が分かりにくい場合があった。そのため、まずプログラム全体の動きを伝え、適宜分からない部分を教えに行く、という形で対応した。また、私の担当した頻出度リストの作成の作業が前期の活動と重なっていたこともあり、早い段階で終わってしまったため、他メンバーのプログラムエラーの解決や一部

機能の実装などを行った。

（※文責: 遠藤晴人）

6.2.5　川平覚士の振り返り

活動全体について

新聞の過去の紙面データおよびそのテキストデータを北海道新聞社から提供していただき、それを元に何かを作り出すのがこのプロジェクトである。しかし、100 社以上の新聞社が国内には存在しており、当然それぞれの新聞社は自社のデータを活用しようと様々な動きを見せている。アイデア出しをしても、前例を少し調べただけでもうすでに似たようなサービスが新聞社自身などによって提供されている。安易に、記事閲覧サービスを作成しようものなら新聞各社が提供しているサービスを上回るものが作成できないのは明らかである。そこで、我々は新聞記事一つ一つに着目するのではなく、ジャンルと日付に着目したアプローチを取ることとなった。

前期の間主に何をしていたか

前期の間、私は主にブレインストーミングに参加したり、後半は特に紙面データのテキストデータ化（以下、OCR）に取り組んでいた。私は過去に OCR に取り組んだことがなかったので、OCR とはどのような処理を行うのかについて調べる所から始まった。

OCR の処理はまずレイアウト解析から始まる。当然、私もレイアウト解析から取り組み始めたのだが、作業が思うように進まない。理由としてはまず、前例がうまく見つけられなかったのでどのようなアプローチを取れば良いのかが分からない。そして、紙面データにノイズが多いことがあげられる。このうち比較的対処が簡単なものが、紙面データのノイズである。よって、私はノイズ除去に取り組むことにした。

紙面データからノイズを取り除くための作業の一環として、紙面データの傾き補正を行った。そのとき得られた知識について記述しているため「4.2.4 紙面データの補正」を参照していただきたい。

後期の間主に何をしていたか

後期の間、私はバックエンドの構築に取り組むこととなった。当初の予定では、AWS 上にバックエンドを構築する予定であったが、大学との調整のために構成を発表 4 週間前に確定させなければならなかった。これはあまりにも現実的ではなかったため断念することとなった。では、バックエンドを何に載せるのかという問題が発生する。今回はこのプロジェクトの担当教員である角康之先生の研究室にある Linux マシンを使うことができた。よって、バックエンドはこの Linux マシンに載せることとなった。

バックエンドを何に載せるのかが決まったため、API の開発を開始した。当初の予定では GO 言語で実装する予定であった。しかし、私は GO 言語をあまり使ったことがなかったため、開発可能期間中に開発を完了させることが不可能であると考えられたため、Python を用いて実装することとなった。

開発開始当初は、Python に Flask という Web フレームワークと MySQL Connector という MySQL 用のドライバを用いて実装していた。しかし、これらを用いる方法だとソースに SQL を直に書かざるを得ない状態になってしまう。この状態は SQL インジェクションに対して脆弱であ

る。よって、OR マッパーである SQLAlchemy の Flask 用に調整されたものである SQLAlchemy-Flask を用いることとした。OR マッパーを導入した当初は、SQL を直接書いてデータベースに渡すよりもただただ面倒になるだけでは?と考えていた。しかし、開発が進むにつれ、OR マッパーによってデータがオブジェクト指向的に使えることに利便性を感じ始めた。よって、この先データベースを用いるようなものを作る時には積極的に OR マッパーを使おうと思う。

また、今回は初めて Redis を用いた。Redis というのか過去に何回か名前を聞いていたし、どのような物かはなんとなく知っていた。しかし、実際に必要とするような場面がなく今まで使うことがなかった。今回は、データが重すぎてキャッシュが必要ということで、いい機会だと思い使うことにした。Redis を導入した結果、思っていた以上に API の動作が高速になった。そして、導入のために導入前から書き換えたコードは 10 行以内に収まる程度だったため、手軽に導入できることがわかった。

OR マッパーや Redis 以外にも、Docker を初めて本格的に使用した。Docker 上のバックエンドを載せることは、本番発表の 3 日前に決まったことであった。よって、対応できる自信がなかったが、仮想ネットワークを建ててコンテナさえ建ててしまえば、通常の Linux マシンと同じようにして設定することができたため、なんとか対応することができた。今までは、Dockerfile の異様な複雑性から Docker に対しては「何故流行ったのかよくわからないただただ不便なソフトウェア」という考えだったが、今回の件を通じて「仮想環境としては便利なもの」という認識に変わった。よって、今後も機会があれば使おうと思う。

まとめ

一年間の活動を軽くまとめると、前期は新聞記事の OCR、そして後期はひたすらバックエンド構築となる。残念ながら前期に取り組んでいたことから成果を上げることはできなかったが、後期には様々なことをすることができた。AWS を使用するために、大学や管理している業者の方と話したことはよい経験だったと思う。また、今までに使うことのなかったソフトウェアや技術を用いたことは、よい経験でもあるし、なにより技術の幅が広がった。こういった点から、とても得るものが多かったプロジェクト学習だったと思う。

（※文責: 川平覚士）

6.2.6　一入悠貴の振り返り

前期終了時点での振り返り

前期の活動としては、まずは Zoom を通してプロジェクト内での自己紹介を行った。個人的にはこのプロジェクトで行った自己紹介はとても収穫のあるものであった。美馬のゆり先生が行ってくれた指摘のおかげで改善点がわかりやすく、ここで培った自己紹介の技術は今後も生きてくると考えられる。これは実際にバイト先での自己紹介の時などにも活用できており、プロジェクト学習で学んだことを他の場面でも生かすことができたいい例だと考えている。

次にメンバー全員で各々が何に取り組みたくて本プロジェクトに参加したのかの情報共有を行った。この情報共有のおかげでそれぞれ他のメンバーのやりたいことについて理解しやすく、その後に行った成果物案の提案もスムーズに良い意見が多く出たのだと考えられる。また、この成果物案の提案については前期で行った取り組みの中でも特に力を入れていたところである。はじめの数回についてはメンバーが個人で様々な意見を出し合い、多くの視点から面白い提案を行うことができ

た。またこの時に、新聞の利点と欠点についてもメンバー間で確認を行った。この利点と欠点の確認について、利点を活かすにしろ、欠点を克服するにしろ、成果物の案を出すうえでとても参考になった。この利点と欠点を確認するという行動は、このプロジェクト以外の取り組みの時にも非常に役に立ちそうな技術であり、今後様々な活動で使っていきたい。

個人での成果物案の提案が一段落したところで、プロジェクトのメンバーを２つのグループに分け、さらに発展した成果物案の提案を行った。この時に情報デザインコースの二人が行ってくれていた、アイデアを発展させるためのホワイトボードの使い方などはとても参考になった。その後は、成果物案の中から良い案を決定し、その案を実現するために必要な技術などの学習を行った。この時、私は自然言語処理について学習した。形態素解析や感情分析などについての学習を行ったが、どちらについても具体的な成果を上げることはできなかった。そのため、後期には具体的な成果が挙げられるよう努力していく。

その他には、自然言語処理を行ったデータの可視化方法について追及するために、Python での様々な可視化方法について調べたり、アプリケーション開発を行う上で、どの形態のアプリケーションを開発するのが良いのかについて調べた。これらの調べた内容をメンバーに伝えるときの資料のまとめ方についてはまだまだ改善できそうなことがあったため、個人的な後期の課題としてメンバーに何か情報を伝えるときの方法の改善を行おうと考えた。そして前期最後の活動として中間発表に臨んだ。聞き取りやすいよう、声の大きさや速さに気を付けて発表を行ったが、焦ってしまったときに修正がききにくかったり、発表時に間違えて発表用の資料を消してしまうアクシデントを起こしてしまったりしたので、それらの問題点を改善していくために対策を講じていくことも個人的な後期の課題として挙げられた。

夏季休業中の活動の振り返り

夏季休業中には Web アプリケーション開発のフロントエンドに関する学習を行った。まずは HTML や CSS、JavaScript に関する勉強をオンライン学習サービスを利用して学習した。その後インターンシップに行き、短期での Web アプリケーション開発実習に挑んだ。ここでは Vue.js を使った開発に取り組んだ。ここでは技術的な学習を行うこともできたが、それ以上に Web アプリケーション開発に取り組むうえで技術的な事以外のことを学ぶことができたことが大きかったと考えている。ここでは、Web アプリケーションを利用するユーザーの気持ちを考えることの大切さや難しさ、一緒に開発を行う人と連携を取ることの大切さについて学ぶことができた。また、技術的な事には Vue.js についての学習を本などを使って行ったり、マテリアルデザインとは何なのかについてなど、自分の知らなかった知識についても学ぶことができた。これらのことを実践的に学ぶことができたことはプロジェクト学習に活かすという観点でもとても良かったと考えている。そのため、これらの学んだことを後期の活動に活かして行くことが大切だと考えた。

後期活動の振り返り

後期一番初めの活動としては、夏季休業中に取り組んだ内容について発表会を行ない、プロジェクトメンバーとの情報の共有を行った。この活動では他の人がどのようなことを行っていたのか理解するために、他の人の発表を聞き、どのような内容であったか情報をまとめる良い練習の機会となり、また、自分が発表する上で、自分が行った活動を綺麗な形にまとめ、かつそれを発表する練習するための良い機会ともなった。

次には後期で行う Web アプリケーション開発の班分けを行い、活動を行った。自分はコンテンツ抽出班として活動した。まず始めは画像処理班という名前であったため、どのように画像を処理

し、Web アプリケーション開発に活かそうかアイデア出しを行った。ここでは寺沢先生からのアドバイスを意識しながら話し合いを行い、前期よりも積極的にアイデアを出し、話し合いを活発に行うことができたように感じている。その次にはコンテンツ抽出に取り組んだ。この活動では、取り組もうとしていることについて調べる技術について、前期よりもうまくなっているように感じた。また、この活動で実際に行っていてとても大切だと感じたことがある。それはその分野について詳しい人に相談を行うことの大切さである。私は画像処理に関する活動に取り組んでいたため寺沢先生に相談を行っていたが、これがとても大切であったと感じた。班のメンバーと様々なことについて話し合いを行いいくつかのアイデアを出したりしていたが、それらの案について先生に相談を行ったところ、それらの意見の難しいところや代替案、それらの活動を行う上で参考にしたらいい資料を見せてもらえるなど、詳しい人としての視点で様々なアドバイスをもらえた。自分たちで意見を出すこともとても大切だし、頼りすぎることもよくないが、詳しい人に相談することで自分たちでは思いつかなかったような考えに気づかせてくれたり、知らなかったことについて知ることができたりなど詳しい人への相談は活動を行う上でとても重要なことだと知ることができた。また、コンテンツ抽出の活動を行っているうえで良かった点がいくつかあった。まずは班のメンバーとの協力である。前期よりも積極的にメンバーと相談したり、協力することができ、複数のメンバーでの協力活動を行う良い練習をすることができたと考えている。次は技術的な話である。先生に相談をしたり、班のメンバーと協力しながらではあったが、自分で何か調べ、学習しながら、技術的な取り組みに取り組むことができた。これは前期ではあまりできておらず、改善しようと考えていた点であったためとてもよかった。最後に情報を共有することの大切さに再度気づき、実際に班のメンバーと共有することができたことである。これは班のメンバーとの協力にとても関係してくるところであるが、Google drive などを使い、YOLO に関する情報などを班のメンバーとよく共有できたと考えている。情報の共有の大切さには前期終了段階や、夏季休業中にも気づいてはいたが、本プロジェクトの OB である日置さんに「それ俺も見れる？」と聞かれたときに、情報の共有はしていたが、プログラムのソースコードや実行結果をメンバーが気軽に見ることができない情報であったことに気づき、そこで情報の共有の仕方について再度考え直すことができ、その大切さにもう一度気づくことができた。そこからは先ほども述べた通り、班のメンバーとよく情報の共有が行えていたと考えている。

後期最後の活動としては、プロジェクトの最終発表会を行った。この発表会では、前期の中間発表と比べて、お客さんの方を向きながら大きな声で発表することができたため成長することができたと思う。また、前期の発表会の時は発表直前に発表用の資料を間違って見れない状態にしてしまうアクシデントを発生させてしまったが、今回はそのようなミスはなく、問題なく発表することができたと考えている。まだ、あまり上手な発表であったとはいえないが、発表の技術についてはこのプロジェクト学習で学んだことを活かしながら今後も向上させていきたいと考えている。

まとめ

今回のプロジェクト学習で学ぶことができたことのまとめとしては、やはり、複数人で開発を行うことの難しさに気づくことができ、それを行う上で大切なことについて知ることができたことが一番大きな成果であったと考えている。これはプロジェクト学習だからこそ学ぶことができたことであり、この学んだことは今後の大学生活や、社会に出た後も大切になってくるとても有意義なものであることがその理由である。

また、知らない人たちに向けて発表をすることができたことも良かったことであった。なかなかこのような機会はなく、人に自分の考えや自分のしてきたことを伝えることはとても大切なことで

あるため、その技術について学ぶことができてよかった。

（※文責: 一入悠貴）

6.2.7　高橋陽一の振り返り

前期の活動の振り返り

アイデア出しの段階ではなるべくほかのメンバーと異なるアイデアを考えることを意識した。その結果ビッグデータの分析を楽しむという既存のサービスにはないサービスを思いつくことができた。グループに分かれて活動したときは自分の考えたアイデアを数名のアイデアで実践した。その結果、想定していたより難しいことがわかったのでグループでの活動の利点を生かせた活動だったといえる。しかし、前期の活動ではいくつかの課題があった。第一に、技術習得のとき手が空いてしまった時があった。まだ成果物案がまとまっておらず、どのような技術を学ぶとよいかわからなかったからである。加えて誰がどの技術を学ぶかといった情報の共有ができていなかったことも原因である。もう少し成果物案が早くまとまりその後の活動の方針や役割分担が明確になっていれば、中間発表の時点でもう少し具体的なプロトタイプを制作できたと考えられる。したがって後期ではプロジェクト内での情報共有を積極的に行い、時間を無駄にしないように行動する。第二に、発表の方法である。中間発表では発表練習の時間があまり確保できず、本番ではパソコンの画面を見ながら話すことになってしまった。その結果声が聞き取りづらいと指摘された。また、発表の分担方法も適切ではなかったといえる。中間発表では発表箇所ごとにメンバーで役割分担した。しかしその結果、デモンストレーションの際、一人でパソコンを操作しながら話したり、スライドを見て指示しながら話したりすることになってしまった。デモンストレーションの際には操作役と説明役に分担するといった工夫をするべきだった。したがって、後期の成果発表会では発表の準備を十分行い堂々と発表できるようにする。さらに夏季休業中に自然言語処理の勉強をして後期の活動に役立てられるようにしたい。

後期の活動の振り返り

夏季休業中は自然言語処理に関する書籍を読んで技術的な知識を深めることを目標にした。具体的には Python を使った自然言語処理と web アプリケーションの基礎に関する書籍と、R を使った自然言語処理と可視化手法に関する書籍を読んで勉強した。web アプリケーションの基礎知識とサークルパッキングや共起ネットワーク図などワードクラウド以外の可視化手法を知ることができたという点で、夏季休業中の活動は後期の活動に役立ったといえる。

後期はまずグループに分かれて開発を行う方針になり、デザイン班に割り当てられた。しかしグループ活動を行う中で開発に javascript フレームワークの知識が必要であることがわかり、javascript の知識や技術を身に着ける段階から始めなければならず、なかなか開発に取り掛かることができなかった。そこで教員からの助言により手が空いている人は画像データを使ってアイデアを考えてみるのはどうかという話になり、コンテンツ抽出班に立候補した。コンテンツ抽出班では、まずグループ内で画像データから何が得られるかを考えた結果、4 コマ漫画や広告、天気図などテキストデータには存在せず画像データにしかないものを抽出しようという結論に至った。しかし紙面のどこにコンテンツがあるかは画像データごとに異なっていて、当初予定していた画像データ内の座標を指定して切り抜くという手法が使えないという問題が発生した。そこで教員に助言を求めたところ YOLO と LabelImg というアルゴリズムとソフトウェアを薦められた。そこでコン

テンツ抽出班では LabelImg を使って紙面データと紙面データ内のコンテンツの座標を組にした訓練データを作成し、YOLO に学習させるという作業を行った。最初のうちは検出精度が低く、コンテンツを検出できないことが多かったが、原因を調査すると訓練データの不足や学習回数の不足であるという原因が明らかになった。そのため訓練データを増やしたり、学習回数を増やすことで 4 コマ漫画と天気図は 90 %以上の精度で検出できるようになった。また画像データを調べていくうちに 6 コマ漫画や 8 コマ漫画が存在していることが明らかになり、4 コマ漫画とは形状が異なるため 6 コマ漫画や 8 コマ漫画も訓練データに含めることにした。しかし、6 コマ漫画や 8 コマ漫画を学習させると 4 コマ漫画の検出の精度が低下するだけでなく、6 コマ漫画や 8 コマ漫画もあまり検出できないことがわかった。したがって YOLO の学習の仕組みや複数の種類の形状を持つコンテンツの抽出方法について理解することが必要だと考えた。

グループ内では主に学習済みの YOLO を使用して実際の画像データからコンテンツを 4 コマ漫画と天気図を抽出し、抽出したデータを整理するという作業を行った。個人作業であったが、アプリケーションで使えるようにデータの名前を整理したり、不要なデータを除去したりと工夫した。

後期の活動で良かった点と反省点

後期の活動でよかったことはグループワークが前期よりも効果的に行えたことである。グループ内で担当作業を分担しつつ、必要に応じてグループメンバー全員で問題に取り組んだり、それぞれの担当作業についてお互いにアドバイスし合ったりできた。例えば検出精度を上げるためにどのような訓練データにすればいいかをグループメンバー全員で考えたり、YOLO で検出したコンテンツを YOLO 上で抽出するための方法を調べて、訓練データ作成担当のメンバーに教えたりした。

一方で後期の活動では課題もあった。第一に、スケジュール管理である。後期の活動ではすぐにアプリケーションの開発に取り掛からずに開発方針の検討や技術習得の続きなどでアプリケーション開発の時間が短くなってしまったといえる。また成果発表会直前には発表スライドやポスターの制作も行わなければならなかったため、予定していた機能である 4 コマ漫画と天気図の組み込みや tf-idf などの機能の実装が間に合わなった。これらの 4 コマ漫画と天気図、tf-idf のデータは用意できていたにもかかわらず、アプリケーションでの利用には至らなかったため非常にもったいないと感じた。この問題は成果発表会の日から逆算して長期的な活動のスケジュールを立てるということを後期の活動の初期にやっておくことで回避できた問題だと考えられる。第二に、グループごとの情報共有である。自分が担当していたコンテンツ抽出班では 4 コマ漫画と天気図を抽出してアプリケーションで利用可能なデータにするという方針は早期に決まっていた。しかし、そのデータをアプリケーション内でどのように利用するのかという議論をほかのグループと行う機会が少なかったため、4 コマ漫画と天気図の利用方法が定まらないままデータだけ用意してアプリケーションには実装されなかった。スクラム開発は機能ごとに開発を行うため、グループ内での議論ばかりが活発になり全体での情報共有が疎かになってしまう可能性がある。したがってスクラム開発を行う際には、よりいっそう全体での情報共有や議論が重要だと感じた。

（※文責: 高橋陽一）

6.2.8 柴田公季の振り返り

6.2.8.1 前期の振り返り

プロジェクトの初期段階では、プロジェクトメンバーと積極的にコミュニケーションを取り、ブレインストーミングでは多くのアイデアを出すように努めた。メンバーとコミュニケーションを取ることで、多様性のある複数の視点を得ることができ、お互いのアイデアを探求することができた。また、グループでアイデア出しをすることで仲間意識が育ち、強い当事者意識を生むことができた。成果物案の提案では、どのような成果物があれば面白いかをメンバーと考えた。考えた案が既存のサービスであったり、そのサービスの意義はあるのかなどの様々な問題点があり、最終的な成果物案を決めるのに多くの時間が取られてしまった。そのため、あらかじめ新聞の主な既存のサービスをグループ内で共有し、新聞への理解をよりしなければならなかったと感じる。中間報告会では、角教授が主催したワークショップでの経験などを活かし、良い発表ができた。しかし、発表の際に常に前を向くことを意識したが、準備が足りなかったため何度も発表内容の文章を見てしまった。最終報告会ではより時間をかけて準備をするようにしたい。技術的な点においては、他のプロジェクトメンバーに頼っていた部分や、成果物案の決定に時間がとられてしまったため、あまり貢献することができなかった。情報ライブラリーで自然言語処理やテキストマイニングの本を借りて勉強したり、自然言語処理百本ノックというサイトを使って勉強したが、まだ完璧には技術を身につけていない。後期からは本格的に作業が始まるので、プロジェクトに貢献できるように夏休みの間にしっかりと技術を身につけていきたい。

6.2.8.2 後期の振り返り

夏季休業中の振り返り

まず前期の課題として、個人的には本プロジェクトで作ろうとしている成果物に対しての技術がまだついていなかったため、夏季休業中に技術力をつけることを目標とした。具体的には情報ライブラリーでテキストマイニングの本を借りて勉強したり、オンライン学習プラットフォームである udemy を使って勉強した。勉強していく中で、いままで理解していなかった基礎的な部分を理解することができ、Google Colaboratory で形態素解析などの作業を理解しながら進めることができた。また、夏季休業中に複数の論文や本を読むという課題がプロジェクトから課されたため、4 つの論文と 1 つの本を読んだ。前期は可視化の方法がワードクラウドのみであり、さらに完成度の高い成果物をつくるためにプロジェクトメンバーの知見を前期よりも広げる必要があったためである。後期の最初の活動時間では各々が読んできた論文や本を発表しあうことで、知識の幅を広げることができた。

データエンジニアリング班での活動

後期の具体的な活動としては成果物を作るために三つの班に分かれ、それぞれで作業を進めた。私はデータエンジニアリング班に所属し、Google Colaboratory を使って記事データを形態素解析し、json ファイルに出力する作業に取り組んだ。データエンジニアリング班の中でも私は tf-idf を使って特徴語を抽出するという作業を進めた。tf-idf の出力には様々なものがあり、どのようにして特徴語を抽出すればよいのか分からなかったが、去年のプロジェクトの先輩とコミュニケーションをとることで、scikit.learn の sklearn.feature_extraction.text.TfidfVectorizer を使うこと

で特徴語を抽出することができるということが分かった。これを使って tf-idf を求めるためには記事データを分かち書きする必要があったため、分かち書きに取り組んだ。分かち書きをするためにまず必要な記事データを抜き出すという作業に取り組んだ。新聞記事から記事データのみを抜き出し、それらを 1 つの記事ごとに分けるという作業が難しかったが、他のメンバーが共有してくれたソースコードを応用することで解決することができた。1 つの記事ごとに分かち書きしたデータを与え、tf-idf を求めた。はじめは辞書のパスを上手く指定できず、想定していたような特徴語は抽出されなかったが、メンバーと協力することで辞書のパスを上手く指定することができ、想定していたような特徴語を抽出することに成功した。json ファイルへ出力するために、特徴語とそれに対応する tf-idf 値を辞書型にした。こうすることで多少強引ではあるが、for 文を回し比較的簡単に json ファイルへ書き込むことができるようになった。また、新聞記事データが入っている csv ファイルを読み込む際、他のメンバーが取り組んでくれた glob モジュールを使うことで簡単に新聞記事データを読み込むことができるようになった。これらのように、データエンジニアリング班の活動を通して、私はプロジェクトにおいて「進捗管理」と「コミュニケーション」の 2 つが何よりも重要であると感じた。メンバーの作業状況をしっかりと把握し、誰が何をしていて、何ができていないのかを全員が理解し、メンバー間で作業を調整するような進捗管理をすることで作業をスムーズに進めることができたように思う。例えば、データエンジニアリング班ではプログラミングが得意な人が中心となり、自分に対して課題を提示してくれたり、作業を見直してくれたりするなどの進捗管理をしてくれたおかげで、自分はスムーズに作業を進めることができた。また、先輩や教員とコミュニケーションを取ることでも作業はスムーズに進めることができた。コミュニケーションについては、個人的にプログラミングがあまり得意ではないため、積極的にしなければならなかった。自分の担当である tf-idf で行き詰ったときは先輩に何度も聞くことで課題を解決することができた。これらのことから、「進捗管理」と「コミュニケーション」の 2 つは個人的に重要であると感じた。

最終発表

最終発表では前回の反省を生かし、当日までに何度も練習した。1 人が発表するのではなく全員で分担して発表するため、ぎりぎりまでミスがないか詳細まで確認してから望むことができた。資料を見ながら話すのではなく、スクリーンを指さしながら、聴衆の目を見て発表するように心がけた。中間発表とは違い何度も練習することができたため、聴衆からの評価も良く、中間発表の反省を活かすことができた発表になったと思う。担当教員からも高評価を受けることができた。

今後の課題

今後の課題としては、私が担当した tf-idf を json ファイルへ出力することができたがまだ web アプリケーションには実装することができていないので、フロントエンド班と話し合いながらどのような可視化をするか決める必要がある。現段階では、頻出単語のワードクラウドとは別に tf-idf のワードクラウドを作成するか、頻出単語で作成したワードクラウドに表示されている単語のうち、tf-idf で抽出した単語と被っている単語に色を付けるという可視化の仕方を考えている。

（※文責: 柴田公季）

6.3 今後の課題

6.3.1 前期終了時点での課題

本プロジェクトは Web アプリケーションを最終的な成果物にしようとしている。しかし、前期終了時点でアプリケーションのプロトタイプしかできていないので後期の活動では実際にアプリケーションをビルドしバックエンドとフロントエンドの両方を開発することを想定している。また、夏休み中にメンバーは自主的に Web アプリケーションの構築について学ぶ予定である。

中間発表にて、「新聞への理解が表面的で浅い」というご意見を頂いた。発表時には主観的な物の見方でしか新聞を語っていなかった為、今後夏休みを通して「メディアの未来 著 ジャック・アタリ」を読み新聞の社会的な役割を再度学びなおす予定である。

自然言語処理を用いて処理したテキストデータに感情分析を行ったことを中間発表で報告しました。その際に想定している Web アプリケーションと感情分析との関連性が分からないとご指摘を受けた。今後の課題は Web アプリケーション内に感情分析の結果を関連付けたコンテンツを配置し、より充実した可視化アプリケーションにしていくことを予定している。

（※文責: 前田祥）

6.3.2 プロジェクト終了時点での課題

プロジェクト終了時点で当初予定していたアプリケーションの機能のうちいくつかが実装できなかったことが課題だといえる。大きく分けて 2 つの課題がある。

第一に画像データから抽出したコンテンツの利用である。画像データから抽出した４コマ漫画と天気図をアプリケーション内でどのように利用するかという議論が不十分であったためアプリケーションでは利用しなかった。一時、ワードクラウドを読み込んでいる間の待ち時間に表示してはどうかという意見があったが、その時点でアプリケーションのプロトタイプが完成していなかったり、バックエンド班との調整が不十分だったりしたため実現しなかった。また４コマ漫画に関してはその時代の社会情勢を風刺しているのではないかと考え、同年代のワードクラウドやサークルパッキングと共に表示することでアプリケーションの利用者に新聞の内容と４コマ漫画の内容の関連に気づけるようなきっかけを与えるようにすると、アプリケーションの目的と合致するのではないかという提案もあった。しかし、議論が深まらず実装には至らなかった。結果として、成果発表会で４コマ漫画はアプリケーションには使われていないのかという質問が多く寄せられたため、成果発表会までに何らかの形でアプリケーション内で利用すべきだったといえる。

第二に実装済みの機能の拡張である。ワードクラウドでは期間指定の範囲を現在の５年間から３０年間に拡張したり、ワードクラウドで出現した単語をクリックすると単語に関連した記事を表示したり、ワードクラウド上の単語にルビを付けたりするといった機能の拡張を予定している。これは成果発表会で挙がった、「ワードクラウドの説明やデモを見て、一般的な機能レベルにとどまっている感じがした」という意見を参考にした改良案である。このほかにも成果発表会では多くの意見・質問が挙がったため、これらの意見・質問をもとにアプリケーションを改良する必要がある。

（※文責: 高橋陽一）

6.3.3 展望

成果発表会を終えて、ユーザの好みに合わせた可視化方法で表示するアプリケーションを完成させることはできた。ただ、サークルパッキングの実装が間に合っていないことや、tf-idf の結果が反映されていないこと、４コマ漫画などのコンテンツ抽出班の成果をまだアプリに反映できてないことなど、プロジェクト終了時点では様々な課題が残った。また、本プロジェクトの目的は、「教育への活用や知的好奇心を刺激する方向へと導き、新聞の新しい使い方を提案する」アプリケーションの作成であるが、成果発表会では作成したアプリケーションの対象者や目的が不明瞭だとの声もいただいた。今後の活動として、秋葉原でのプロジェクト学習成果発表会がある。アプリケーションの方向性を今一度精査し、機能の追加など現状の課題を解決して、これらに臨みたい。

（※文責: 遠藤晴人）

謝辞

本プロジェクトでは北海道新聞の過去 33 年分の新聞記事データを活用した。プロジェクトの構想やシステム試作において北海道新聞社の三浦辰治氏に多大なるご協力を頂戴したので感謝申し上げます。

（※文責: 前田祥）

参考文献

[1] 日本新聞協会（2022）, 新聞の発行部数と世帯数の推移. https://www.pressnet.or.jp/data/circulation/circulation01.php (2023/01/04 アクセス)

[2] 総務省（2021）, 令和 3 年度版 情報通信白書. pp.218-219, https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/r03/pdf/01honpen.pdf

[3] 株式会社 NTT ドコモ モバイル社会研究所 （2022）, モバイル社会白書 2022 年版. p.87, https://www.moba-ken.jp/whitepaper/wp22/pdf/wp22_all.pdf

[4] 読売新聞, デジタルで広がる新聞の未来. https://saiyou.yomiuri.co.jp/ism/future

[5] 小久保奈都弥（2020）, データ分析者のための Python データビジュアライゼーション入門 コードと連動してわかる可視化手法. 翔泳社

[6] 石田基広（2017）, R によるテキストマイニング入門（第 2 版）. 森北出版

[7] C. Muller, Sarah Guido 著, 中田秀基 訳（2017）, Python ではじめる機械学習. オライリージャパン

[8] 柳井孝介 庄司美沙 （2019）, Python で動かして学ぶ 自然言語処理入門. 翔泳社

[9] 田中東子 竹田恵子 上村陽子 中條千晴 中村香住 東園子 有國明弘 渡辺明日香 村上潔 梁・永山聡子（2021）, ガールズメディアスタディ. 北樹出版

[10] 千葉涼（2021） ニュースの多様性とは何か データ分析で問い直すジャーナリズムのあり方. 勁草書房, pp.157-160

[11] 薬剤師のプログラミング学習記録 Python で画像の傾きを修正する. https://www.yakupro.info/entry/programming-img-rotate 2023 年 1 月 6 日閲覧

[12] Rajaraman, A.; Ullman, J.D. “Data Mining”. Mining of Massive Datasets. pp. 1–17, 2011. http://113.161.98.146/jspui/bitstream/123456789/110/1/50.%20Anand%20Rajaraman%2C%20Jeffrey%20David%20Ullman%20-%20Mining%20of%20Massive%20Datasets%20%20-Cambridge %20University%20Press%20%282011%29.pdf

[13] 北清敦也（2022）, 深層学習を用いた新聞画像からの広告挿絵の検出と日付ごとの整理. 公立はこだて未来大学寺沢研究室卒業論文

[14] クラウド・データセンター用語集 用語集—LAMP. https://www.idcf.jp/words/lamp.html 2023 年 1 月 6 日閲覧

[15] さくらのナレッジ いまさら聞けない Node.js. https://knowledge.sakura.ad.jp/24148/ 2023 年 1 月 6 日閲覧

[16] 物理の空き地 Python C++ 比較. https://physics-mek.com/2022/05/13/cpython-%E8%A8%88%E7%AE%97%E9%80%9F%E5%BA%A6%E6%AF%94%E8%BC%83/ 2023 年 1 月 6 日閲覧

[17] AWS Redis—AWS. https://aws.amazon.com/jp/redis/ 2023 年 1 月 6 日閲覧

[18] Jacques Attali（2021 年）, メディアの未来.

[19] 小木曽智信, 小椋秀樹, 小磯花絵, 宮内佐夜香, 渡部涼子, 伝康晴（2010） 形態素解析辞書のベンチマークテスト —IPAdic・NAIST-jdic・UniDic のジャンル別精度比較—. 言語処理学会 第 16 回年次大会 発表論文集, 326-329,

https://www.anlp.jp/proceedings/annual_meeting/2010/pdf_dir/PA1-14.pdf

[20] AIAL 最先端情報吸収研究所（2017）, word2vec の学習済み日本語モデルを公開します. https://aial.shiroyagi.co.jp/2017/02/japanese-word2vec-model-builder/ （2023/01/06 アクセス）

[21] YOLO v5 で物体検出と学習をする方法 Google Colab で動作 — ゆっくりキカイガクシュウ. （laid-back-scientist.com）,（参照 2022-10-21）