知能情報実験 III（データマイニング班）
Twitter 上のテキスト文を対象とした「コロナで何が困っているのか」を見つける

グループの学籍番号 205759A, 205720E, 205763J, 195719J

提出日：2022 年 8 月 4 日

目次

1 **はじめに** 2

1.1 分担 2

1.2 概要 2

1.3 テーマ：Twitter 上のテキスト文を対象とした「コロナで何が困っているのか」を見つけるとは 2

2 **実験方法** 3

2.1 実験目的 3

2.2 データセット構築 3

2.3 モデル選定 4

2.4 パラメータ調整 4

3 **実験結果** 5

4 **考察** 8

5 **意図していた実験計画との違い** 8

6 **まとめ** 9

1　はじめに

1.1　分担

zoom を使って、ペアプロのようにグループの全員でモデルの適用までのコーディングを進め、１通りの考察まで行った。最後の 3 週は役割分担して作業を進めた。以下に最後の 6 週 (10~15 週目) の役割を示す。

205759A　比嘉和樹：コード、github リポジトリの整理。

205720E　内藤武 ：発表資料。モデルのドキュメント。

205763J　原田成斗：実験結果、考察。アクティビティ図の作成。

195719J　西村悠吾：レポートのデータセット構築。__init__.py(モジュール) の関数のドキュメント。

1.2　概要

グループ 3 は、Twitter 上のテキスト文を対象として「コロナで困っていること」を見つけることを対象問題として進めた。データセットとして、snscrape を用いて twitter 上から「コロナ」という単語が含まれた 25000 件のツイートを取得した。このツイートを oseti と呼ばれるライブラリ関数を用いてネガティブかポジティブかの判定を行い、ネガティブだけのツイートを抽出した。集めたネガティブなツイートを用いて、形態素分析した。その結果をモデルに適用した。モデルは、教師なし学習で文書のクラスタリングを行うため LDA(トピックモデル) を使用した。また、トピックモデルは pyLDAvis を用いてインタラクティブに可視化した。

1.3　テーマ：Twitter 上のテキスト文を対象とした「コロナで何が困っているのか」を見つけるとは

本グループでは Twitter 上のテキスト文を対象として「コロナで困っていること」を見つけることを対象問題として設定した。SNS を使ってコロナで発生している問題を可視化し、件数が多い項目を整理することで、その後の改善策を見出して今後に応用できる。それによって現在発生している問題に対処しつつ今後同様の問題が発生する際により効果的な対策ができるようになることが期待できる。

2　実験方法

2.1　実験目的

あるカテゴリに対するネガティブなツイートは、そのカテゴリによって「困っていること」だと判断する。したがって、今回のテーマである「コロナで困っていること」は、コロナというカテゴリについて述べているネガティブなツイート、すなわちコロナが含まれるツイートのうちネガティブなツイートをクラスタリングして得られた話題のことを指す。その得られた話題の中でもツイート数が多い話題は、より多くの人が困っている話題だと判断し、今回の実験ではそれを見つけることを目的とする。

2.2　データセット構築

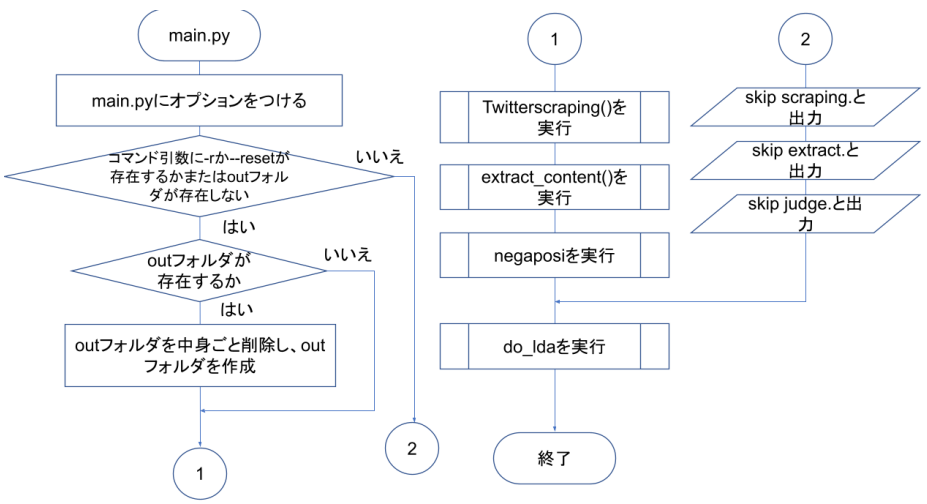

図 1　トピックモデリングの出力結果

以下にデータセット構築に必要なモジュールの中で定義した関数について箇条書きで説明する。
Twitterscraping()：Python の snscrape ライブラリ・モジュールを使用して、ユーザー、ユーザープロファイル、ハッシュタグ、検索、ツイート（単一またはスレッド）、リスト投稿、トレンドから「コロナ」を含むテキストを取得する。件数 (行数) は 25,000 件である。そして、それをデータフレームに変換する [2]。その後、df.to_csv() でテキストをカンマを区切り値とした csv ファイルに書き出す [3]。
scraping_results.csv：カンマを区切り値とした csv ファイル。UTF-8 Unicode テキストで書かれており、非常に長い行 (25000 件) を持つ。スクレイピングした結果が保存されている。
extract_content()：pandas.read_csv() で scraping_results.csv をデータフレームとして読み込む

[4]。このデータフレームのうち、content カラムの列だけで新たなデータフレームを作り、そのデータフレームを df.to_csv() で csv ファイルに書き出す。
contents.csv：カンマを区切り値とした csv ファイル。UTF-8 Unicode テキストで書かれており、非常に長い行 (25000 件) を持つ。スクレイピングした結果のうち、ツイートとリツイートのテキストが保存されている。
judge(text)：引数で渡されたテキストを、日本語評価極性辞書を利用した Python 用 Sentiment Analysis ライブラリ oseti を使ってネガポジ判定 () を行う [5]。評価極性のスコアが 1 に近いほどポジティブで、 -1 に近いほどネガティブとなる。oseti を使ってネガポジ判定をしているサイトを参考にコードを書いた [6]。
negaposi()：pandas.read_csv() で contents.csv ファイルをデータフレームとして読み込む。そして、データフレームに保存されているテキストを引数として、judge(text) を実行し、その戻り値を新たなカラム negaposi に保存する。その後、negaposi カラムの値が 0 よりも大きい行を消し、新たなデータフレームを作る。そのデータフレームを df.to_csv() で csv ファイルに書き出す。
negative.csv：カンマを区切り値とした csv ファイル。UTF-8 Unicode テキストで書かれている。ツイートとリツイートのテキストのうち、ネガティブと判定されたテキストが保存されている。
parse(tweet_temp)：形態素解析エンジンの Mecab を使って、形態素解析をする [7]。まず、negative.csv の content カラムに保存されている値の１つ (1 ツイート文) のすべての単語について辞書を通して形態素解析し、単語、品詞、品詞細分類、活用型、活用形、原形、読み、発音の情報を取り出して 2 次元リスト化する。情報がない部分には*を代入する。Mecab を使ってネガポジ判定をしているサイトを参考にコードを書いた [8]。
parse_to_df(tweet_temp)：negative.csv の content カラムに保存されている値の１つ (1 ツイート文) を parse 関数で形態素解析する。その後、カラムを形態素の種類ごとに設定したデータフレームを作成し、parse 関数の結果を該当するカラムに保存する。作成したデータフレームを戻り値として返す。

2.3 モデル選定

実験目的に沿って、教師なし学習で文書のクラスタリングを行うため LDA(トピックモデル) を使用した。また、トピックモデルは pyLDAvis を用いてインタラクティブに可視化できる部分も LDA を選んだ理由である。

2.4 パラメータ調整

クラスター数 (カテゴリ数) の最適数を見つけるため、クラスター数３～５０の結果をメンバーで確認し、トピック名をつけやすいクラスター数を探した。また、その他のパラメータは参考にしたサイトの値をそのまま使用した [9]。

3　実験結果

pyLDAvis を使用してトピックモデリングの内容を html に出力した結果は以下の通りである。(リンク)

まず出力された Intertopic Distance Map の分布の見方から説明する。

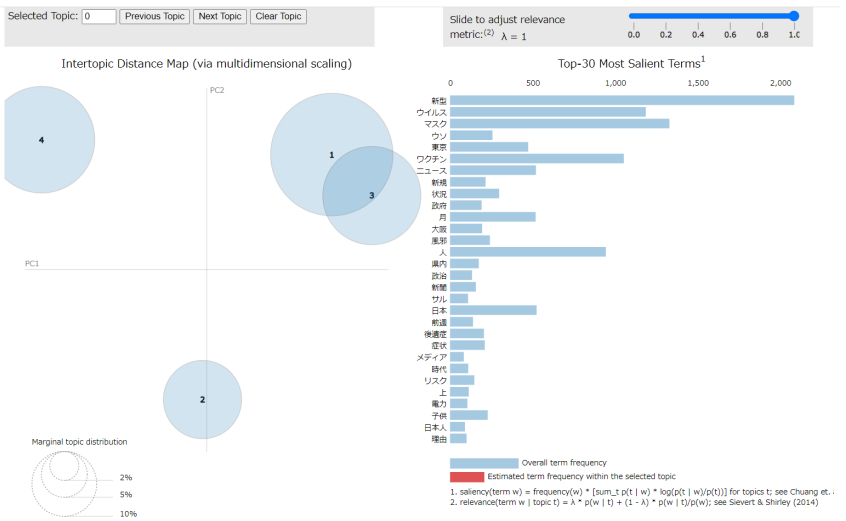

図 2　トピックモデリングの出力結果

図 1 の左側には各トピックを多次元尺度法 (MDS) で 2 次元に落として配置したものが表示され、円の大きさは全体の単語に対する、各トピックに所属する単語の割合を表している。また、右側には選択したトピックにおける頻出単語、同トピック内での出現頻度が大きい単語とその頻度が表示される。右上にあるパラメータ λ を変えると出現頻度、同トピック内での出現頻度の割合の比率を変えて順位を変えられる。右側の単語にマウスを重ねると各トピック内における出現頻度が左側の円の大きさとして表示される。

図１によるとクラス１と３は似たトピックで、２と４はそれぞれ離れたトピックであることがわかる。また、全てのツイートに含まれている「コロナ」という単語を除くと、図２，３より、１と３のトピックで多く含まれる単語は「ワクチン」、「マスク」、「人」であり、また、「風邪」「症状」や「後遺症」という単語が１のトピックのみに含まれている。

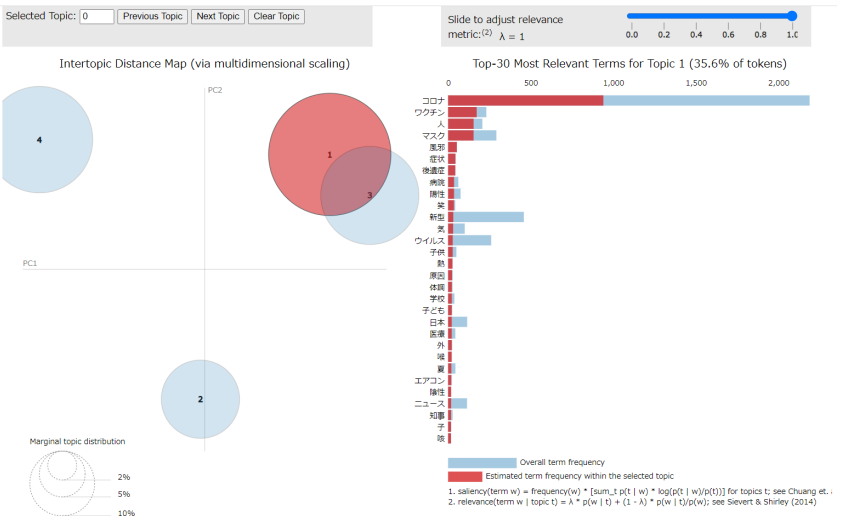

図３　クラス１の単語数

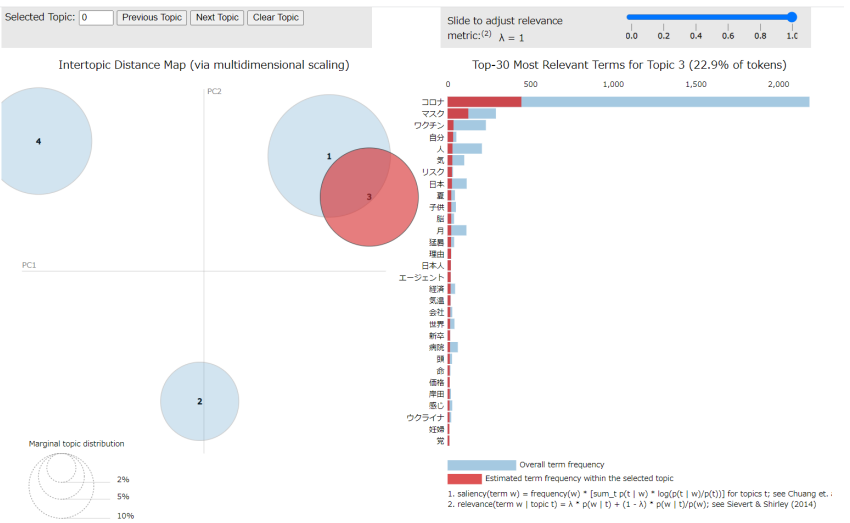

図４　クラス３の単語数

図４を見ると、２のトピックで多い単語は「新型」「ウソ」「日本」であり、さらに、「政府」や「政治」という単語が多い。

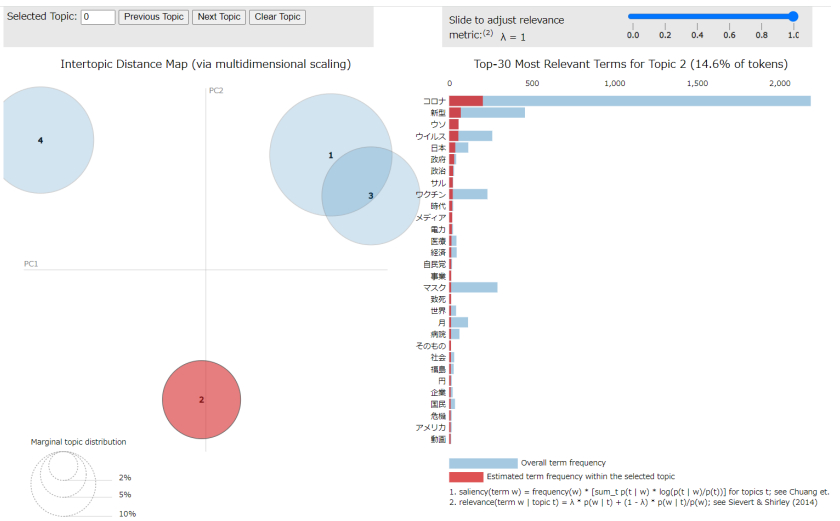

図５　クラス４の単語数

また、図５より４のトピックで多い単語は「新型」「ウイルス」「東京」であり、地名や「ニュース」、「新聞」といった単語が４のトピックに多いことがわかる。

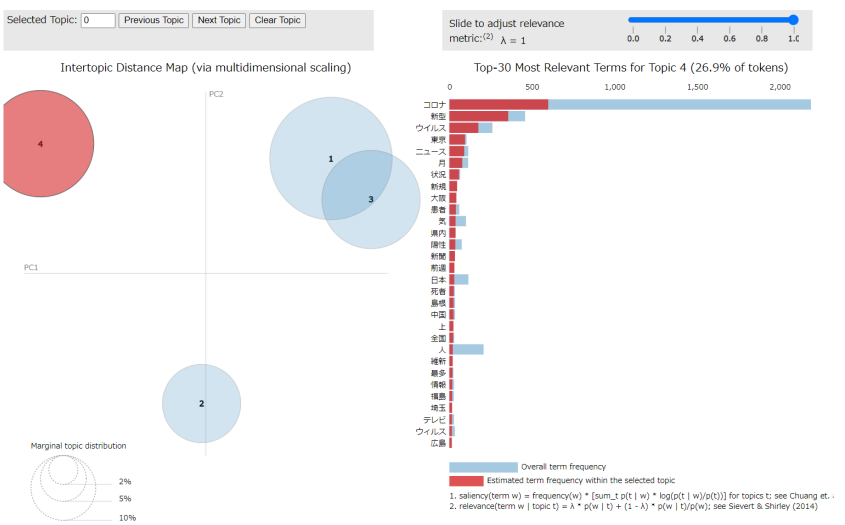

図６　クラス２の単語数

4　考察

実験結果より、トピック 1,3 は「コロナ感染症状」、トピック 2 は「政治」、トピック 4 は「メディア」の 3 つのトピックに分けられると考えた。さらに、今回データセットの構築を行う際「コロナ」という単語を含めたツイートを取得していたため、トピックごとに占めるコロナという単語数の割合が多いトピックが今回の実験目的に即していると考える。したがって、コロナによって困っていることのうち、より多くの人が困っているトピックは、コロナ感染症状、ニュース、政治の順であると考える。

また、今回の実験は参考資料をもとに行っていたため、LDA のパラメータを理解し、調整するなどを行えていない。今後、より今回の実験に合わせたモデルの構築を行うことができればより良い結果が出るのではないかと考える。また、データの数としてネガティブとして判定されたツイート数は約 1 万件だった。今回の実験としてこのデータ数が最適なのかということも考えなければならない。

5　意図していた実験計画との違い

以下に、本グループでのガントチャートを示す。

タスク名称	開始日	完了日	担当者
グループワーク			
アイデア出し	5/19	5/19	全員
テーマ決め	5/26	5/26	全員
データ構築	6/2	6/2	全員
モデル適用	6/9	6/16	全員
レポート作成			
init .py(モジュール) の関数のドキュメント	6/30	7/21	西村
レポートのデータセット構築	6/30	7/21	西村
発表資料	6/30	7/21	内藤
モデルのドキュメント	7/14	7/21	内藤
コード、githubリポジトリの整理	6/30	7/21	比嘉
実験結果、考察	6/30	7/28	原田
アクティビティ図の作成	7/21	7/28	原田
発表資料、レポート仕上げ	7/28	7/30	全員

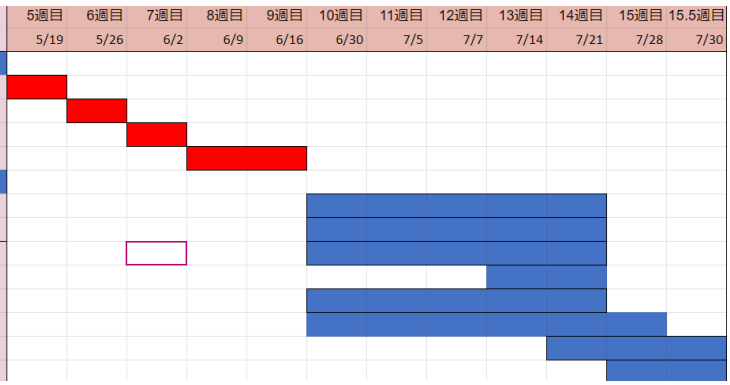

図 7　ガントチャート

図のようにまず、我々は全員でアイデア出しからモデルの構築までを行った。その際、テーマ決めが終わった段階で、データ構築からモデルの適用までの大まかな流れを決めた。そうした行動もあってか、我々が考えていたよりも早くモデルの完成までは終わった。
そこから役割分担してレポート作成や成果物の整理などを行ったのだが、予想以上に時間がかかった。その理由として、モデル完成までは大まかな流れを決めていたため、ある程度終わりが見えて

いた状態なのだが、その後の作業では大まかな流れを決めずに行っていたためゴールが見えずらい状況になっていた可能性が考えられる。その結果、授業時間内では終わることができず、授業時間外に集まって作業を行う結果になった。今後こうした事態を防ぐためには、作業を始める前に、一度ゴールを決め、それに向かって各々が作業をしていけばより効率的に作業を終えることができたのではないかと考える。

6 まとめ

目標としていた「コロナで困っていることの抽出」について、実験用のプログラム作成には成功したものの抽出した内容は具体性に乏しく、達成には至らなかった。pyLDAvis による結果の可視化の重要性が理解できた。Mecab と gensim_models を利用して簡単にトピックモデリングができ、日本語極性辞書の oseti によるネガポジ判定は人の目にも精度が高かったため、優秀なライブラリが多いという python の長所を実感した。

参考文献

[1] レポート作成の手引き　レポートの基本的形式に関するガイド, https://www.kanazawa-u.ac.jp/wp-content/uploads/2015/01/tebiki2.pdf, 2022/07/28.

[2] snscrape, https://github.com/JustAnotherArchivist/snscrape, 2022/07/28.

[3] pandas documentation, https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to_csv.html, 2022/07/28.

[4] pandas documentation, https://pandas.pydata.org/docs/reference/api/pandas.read_csv.html#pandas.read_csv, 2022/07/28.

[5] 日本語評価極性辞書を利用した Python 用 Sentiment Analysis ライブラリ oseti を公開しました, https://qiita.com/yukinoi/items/46aa016d83bb0e64f598, 2022/07/28.

[6] oseti による日本語の感情分析, https://note.com/npaka/n/n3c7722d2e4bc, 2022/07/28

[7] MeCab: Yet Another Part-of-Speech and Morphological Analyzer, https://taku910.github.io/mecab/, 2022/07/28.

[8] Mecab（形態素解析）で遊んでみた！, https://qiita.com/str32/items/c1ced3b017ff5cacc4dc, 2022/07/28.

[9] Shingo の数学ノート, http://mathshingo.chillout.jp/blog27.html, 2022/07/28.