

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 6 月 9 日

## 目次

1	はじめに	2
1.1	概要	2
1.2	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	2
2.1	実験目的	2
2.2	データセット構築	2
2.3	モデル選定	3
2.4	パラメータ調整	3
3	実験結果	3
4	考察	7
4.1	追加実験	7
5	意図していた実験計画との違い	7
6	まとめ	7

# 1 はじめに

## 1.1 概要

本実験では、グループでテーマを決めて、半年間でデータ解析に取り組むグループワークである。グループワークを通して、機械学習や実験再現のためのドキュメント作成等を目指す。

## 1.2 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

コロナで何が困っているのかがわかることでその後の改善策を見出すことができ、今後に応用するために、Twitter 上のテキスト文を用いて解析する。

## 2.2 データセット構築

以下にデータセット構築におけるアクティビティ図を示す。？

`Twitterscraping()`：Python の `snsrape` ライブラリ・モジュールを使用して、ユーザー、ユーザープロフィール、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 [5]。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 [6]。

`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 [7]。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 [8]。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 [9]。

`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 [10]。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 [11]。

`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 [4]。

## 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。

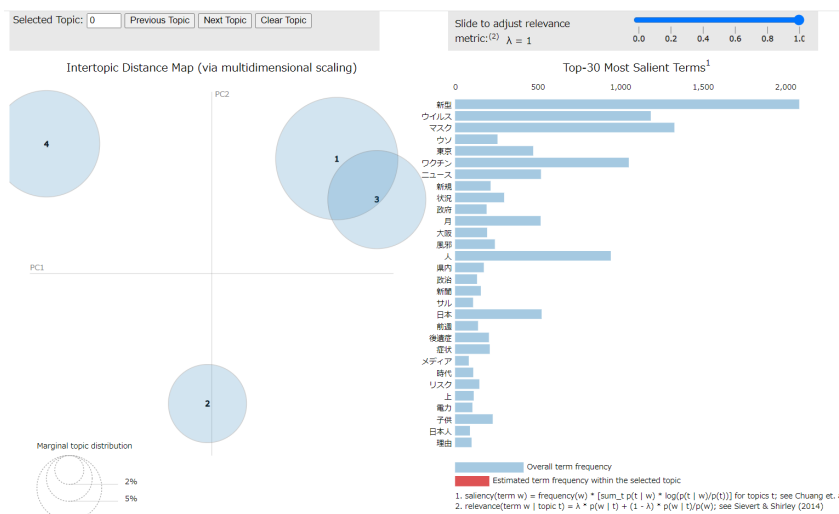


図1 トピックモデリングの出力結果

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 $\lambda$ 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。

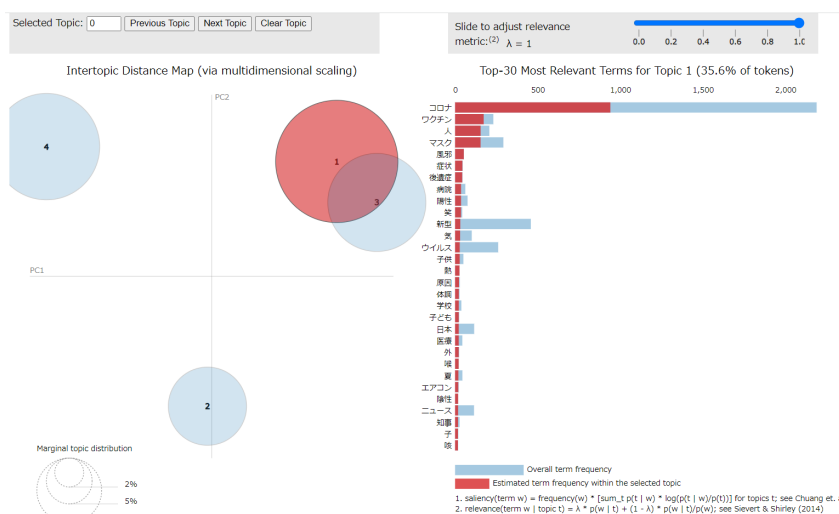


図2 クラス1の単語数

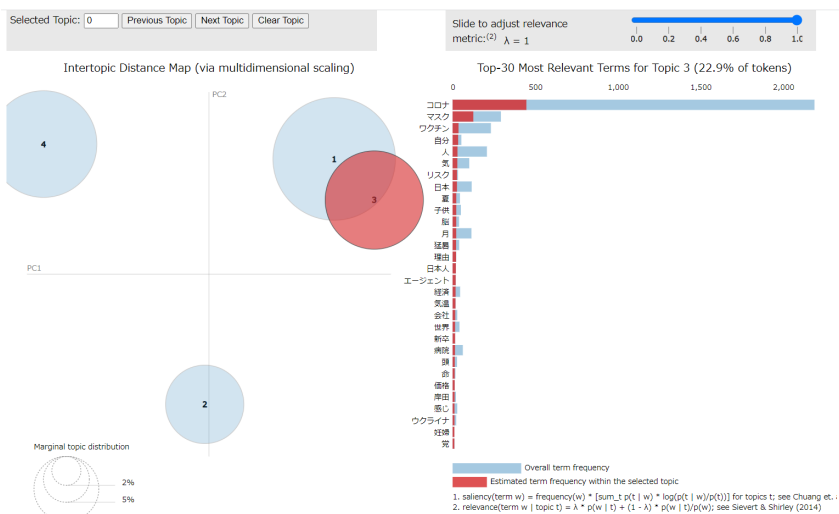


図3 クラス3の単語数

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。

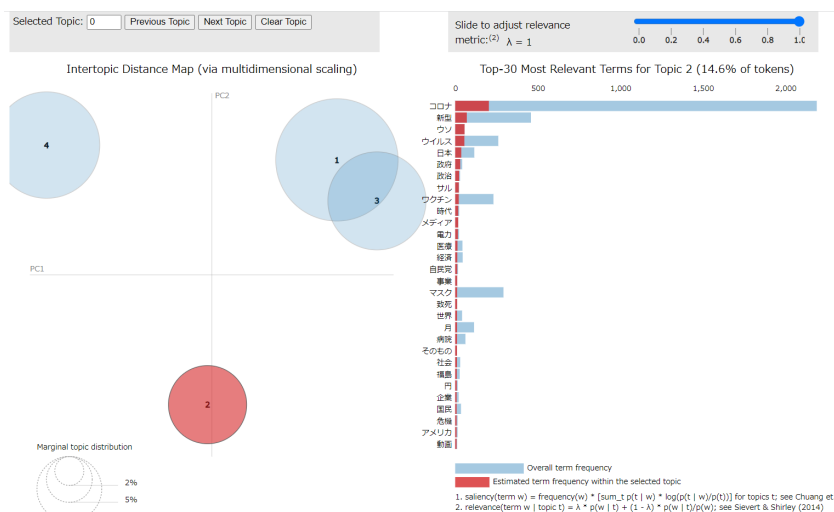


図4 クラス4の単語数

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。

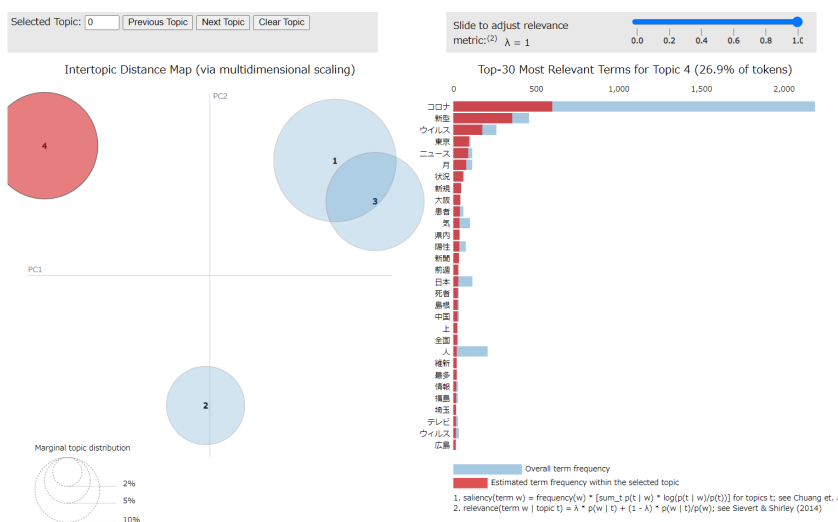


図5 クラス2の単語数

## 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える。したがって、コロナによって困っていることのうち、改善する優先度の高いトピックは、コロナ感染症状、ニュース、政治の順であると考ええる。

しかし、snsrape を用いて「コロナ」という単語を含めたツイートを取得する際、snsrape は単語だけでなく、ユーザ名やタグからも取得することに加え、同じ tweet にコロナという単語が複数回使われている可能性もある。そのため、一概にコロナという単語数の割合が多ければ優先度の高いトピックとはならない可能性もある。

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ぬ。

### 4.1 追加実験

(まだ実験していないので書いていません。)

## 5 意図していた実験計画との違い

## 6 まとめ

## 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] snsrape, <https://github.com/JustAnotherArchivist/snsrape>, 2022/07/28.
- [3] pandas documentation, [https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to\\_csv.html](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](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.