

感情分析を用いた SNS フィルターシステムの 開発

芝浦工業大学附属高校	東田繫洸
芝浦工業大学附属高校	須山莉緒
芝浦工業大学附属高校	須山玲央
芝浦工業大学附属高校	福田啓太
芝浦工業大学附属高校	良知航星
芝浦工業大学	佐々木毅
芝浦工業大学附属中学高等学校	山岡佳代
芝浦工業大学附属中学高等学校	横山浩司

目次

1. はじめに.....	1
1.1 概要	1
1.2 各機能の説明.....	2
1.3 開発環境	4
2. ソフトウェア	5
2.1 twitter_api(twitterAPI コンポーネント).....	5
2.2 sentiment_analysis(感情分析コンポーネント)	6
2.3 morphological_analysis (形態素解析コンポーネント).....	7
2.4 restricted_word (悪口判断コンポーネント).....	8
2.5 result_out (結果出力コンポーネント).....	9
2.6 youtube_comment (youtube コメント取得コンポーネント).....	10
2.7 instagram_scraping (instagram 投稿取得コンポーネント).....	11
3. 本システムの利用手順.....	13
3.1 各パッケージ、モデルの補足説明.....	13
3.2 torchvision, transformers, Mecab のインストール方法	14
3.3 コンポーネントの接続	15
4. 参考文献.....	17

1. はじめに

1.1 概要

SNS の普及は、世界的な情報インフラの発達などに貢献した反面、匿名性による誹謗中傷など、様々なデメリットが表面化していることが問題視されている。

そこで、我々は twitterAPI、感情分析、言語処理を用いて、twitter の指定したワードに関する最新のツイートから肯定的な物のみを選択して抽出するシステムを構築することで、これを解決しようと考えた。また、肯定的な意見のみでは意見が偏ってしまう問題が発生してしまうと考えたため、誹謗中傷にあたる言葉を表示せずに、否定的な意見を抽出するシステムも並行して構築し、検索する際に選択できるようにする。

1.2 各機能の説明

このシステムには、Twitter スクレイピング、形態素解析、感情分析、結果出力の 4 つの機能から成り立っている。以下の表に各システムの詳細と使用しているコンポーネントを示す。

Twitter スクレイピング	twitter_api コンポーネントを使用して、twitter から、スクレイピングを行い、結果を morphological_analysis コンポーネント、sentiment_analysis コンポーネントに送信する。
形態素解析	morphological_analysis コンポーネントを利用して、twitter_api コンポーネントから送信されてきた結果を以下の画像のように形態素解析し、restricted_word コンポーネントに送信する。
ニュアンスを含めた悪口伏字化	restricted_word コンポーネントを利用して以下の画像のようなニュアンスを含めた誹謗中傷の判断、悪口の伏字化を行う。 また、スクレイピング結果の分かち書きを行い、悪口の伏字化結果と共に result_out コンポーネントに送信する。
感情分析	sentiment_analysis コンポーネントを利用して、twitter_api コンポーネントから送信されてきたスクレイピング結果を-1 から 1 までのスコア、もしくは「ポジティブ」、「ネガティブ」の文字列を result_out コンポーネントに送信する。
結果出力	result_out コンポーネントを利用して、送信されてきた分かち書き結果と悪口の伏字化結果、感情データをもとにポジティブ、ネガティブのみの表示(コンフィグレーション変数で指定)を行う。

力された文章：単語ごとに分けられ、それぞれの品詞もわかりま
【単語】一般、*,*,*,*, 単語, タンゴ, タンゴ’,
【名詞】接尾, 一般、*,*,*,*, ごと, ゴト, ゴト’,
【助詞】格助詞, 一般、*,*,*, [に, ニ, ニ’,
【動詞】自立、*,*, 一段、未然形、分ける、ワケ、ワケ’,
【動詞】接尾、*,*, 一段、連用形、られる、ラレ、ラレ’,
【記号】読点、*,*,*,*, 、, 、,’
【名詞】副詞可能、*,*,*,*, それぞれ、ソレゾレ、ソレゾレ’,
【助詞】連体化、*,*,*,*, の、ノ、ノ’,
【品詞】一般、*,*,*,*, 品詞、ヒンシ、ヒンシ’,
【助詞】係助詞、*,*,*,*, も、モ、モ’,
【動詞】自立、*,*, 五段・ラ行、連用形、わかる、ワカリ、ワカリ’,
【動詞】*,*,*, 特殊・マス、基本形、ます、マス、マス’,
【記号】句点、*,*,*,*, .

同じ単語でも別の意味をもつ文章の例と、形態素解析結果	
<p>「ゴミ野郎」の場合</p> <p>'ゴミ' '名詞,一般,*,*,*,ゴミ,ゴミ,ゴミ';</p> <p>'野郎' '名詞,一般,*,*,*,野郎,ヤロウ,ヤロー'</p>	<p>誹謗中傷</p> <p>誹謗中傷 ではない</p>
<p>「ゴミを拾う」の場合</p> <p>'ゴミ' '名詞,一般,*,*,*,ゴミ,ゴミ,ゴミ';</p> <p>'を' '助詞,格助詞,一般,*,*,を,ヲ,ヲ';</p> <p>'拾う' '動詞,自立,*,*,五段・ワ行促音便,基本形,拾う,ヒロウ,ヒロウ';</p>	

ニュアンスを含めた悪口判断の例↑

1.3 開発環境

本コンポーネントの開発環境を下記の表に示す。

OS	Windows 10
開発環境	Visual Studio Code 1.72.2
RT ミドルウェア	OpenRTM-aist-2.0.0- RELEASE(Python 版) OpenRTM-aist-1.2.0- RELEASE(Python 版)
Python	Python 3.7.5

2. ソフトウェア

2.1 twitter_api(twitterAPI コンポーネント)

・説明

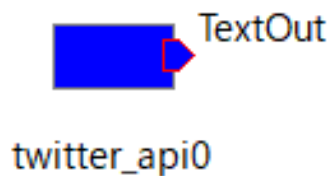
twitterAPI を利用して twitter からスクレイピングを行うコンポーネント。スクレイピング結果をリストにまとめて送信する。

・補足説明

APIKEY の取得方法は以下のリンクを参考

https://toxublog.com/blog/twitter_api_apply/

・画像



・データポート

以下の表にデータポートの説明を示す。

データポート	ポート名	データ型	説明
Outport	TextOut	TimedWStringSeq	twitterAPI のスクレイピング結果をリスト型で送信する。

・出力データの概要

['ツイート 1','ツイート 2','ツイート 3','ツイート 4']

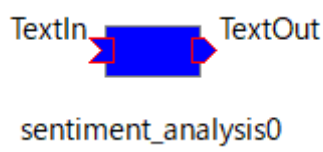
↑のように検索ワードに関するツイートをリスト型で出力

2.2 sentiment_analysis(感情分析コンポーネント)

・説明

BERT(google の自然言語処理 AI モデル)を前提とした bert-base-japanese-sentiment を利用した感情分析を行うコンポーネント。

・画像



・データポート

以下の表にデータポートの説明を示す。

データポート	ポート名	データ型	説明
Inport	TextIn	TimedWStringSeq	他コンポーネントから送られてきたリスト型データを受信する。
Outport	TextOut	TimedWStringSeq	リスト型データで送信されてきた文章を-1~1 の値で感情分析し、リスト型で送信する。

・出力データの概要

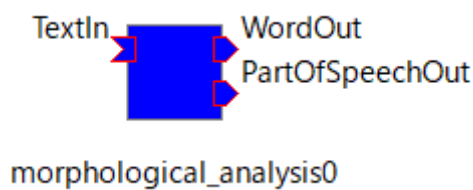
onActivated で”label”を指定した場合
[‘感情(ポジティブ or ネガティブ)’,’感情(ポジティブ or ネガティブ)’]のように

onActivated で”score”を指定した場合
[‘感情値 1’,’感情値 2’,’感情値 3’,’感情値 4’]のように

出力

2.3 morphological_analysis (形態素解析コンポーネント)

- ・ 説明
Mecab を利用した形態素解析を行うコンポーネント。
- ・ 画像



- ・ データポート
以下の表にデータポートの説明を示す。

データポート	ポート名	データ型	説明
Inport	TextIn	TimedWStringSeq	他コンポーネントから送信されてきたリスト型データを受信する。
Outport	WordOut	TimedWStringSeq	リスト型で送信されてきた文字列を分かち書きしてリスト型で送信する。
Outport	PartOfSpeechOut	TimedWStringSeq	リスト型で送信されてきた文字列を形態素解析して品詞をリスト型で送信する。

- ・ 出力データの概要
WordOut からは、以下のように出力
[‘単語 1’,’単語 2’,’単語 3’,’単語 4’] 例:[‘今’,’は’,’一時’,’です’,]

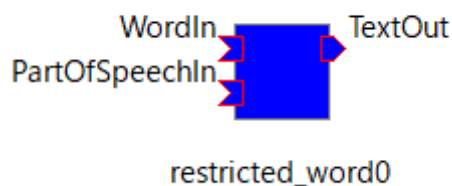
PartOfSpeechOut からは、以下のように出力
[‘品詞データ 1(例:名詞,非自立,一般,*,*,*)’,’品詞データ 2’,’品詞データ 3’]

2.4 restricted_word (悪口判断コンポーネント)

・説明

形態素解析結果を元に悪口をニュアンスを含めて伏字化するコンポーネント。

・画像



・データポート

以下の表にデータポートの説明を示す。

データポート	ポート名	データ型	説明
Inport	WordIn	TimedWStringSeq	リストに格納した分かち書き結果を取得する。
Inport	PartOfSpeechIn	TimedWStringSeq	リストに格納した形態素解析結果を取得する。
Outport	TextOut	TimedShoatSeq	分かち書きリストにおける悪口の位置をリスト型で送信。

・出力データの概要

[[‘悪口の位置 1’,‘悪口の位置 2’],[‘悪口の位置 3’]]

例:[[‘3’,‘7’],[‘17’]] ※←一つ目の文章の3つ目と7つ目の単語、二つ目の文章の17つ目の単語が悪口であるという意味

2.5 result_out (結果出力コンポーネント)

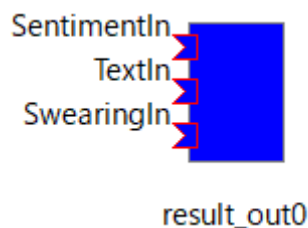
・説明

悪口の伏字化データ、感情分析データを元に最終結果を出力するコンポーネント

・補足説明

出力する感情(ポジティブかネガティブ)はコンフィグレーション変数で変更可能。(初期はポジティブ)

・画像



・データポート

以下の表にデータポートの説明を示す。

データポート	ポート名	データ型	説明
Inport	SentimentIn	TimedWStringSeq	リスト型の感情分析結果を取得する。
Inport	TextIn	TimedWStringSeq	リスト型の分かち書きリストを取得する。
Inport	SwearingIn	TimedShortSeq	リスト型で分かち書きリストにおける悪口の位置を取得する。

・コンフィグレーション変数

変数名	データ型	説明
Emotion_selection	int	ポジティブ、ネガティブのみを表示する際の感情を指定する変数。 ポジティブを指定したい場合は 0 ネガティブを指定したい場合は 1 初期値は 0

2.6 youtube_comment (youtube コメント取得コンポーネント)

・説明

YoutubeAPI を利用して youtube のコメントを取得するコンポーネント。

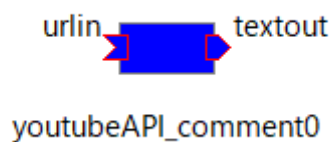
・補足説明

URL を格納したリストを入力すると、コメントをリスト型で出力する。

APIKEY の取得方法は以下を参照

<https://qiita.com/shinkai/items/10a400c25de270cb02e4>

・画像



・データポート

以下の表にデータポートの説明を示す。

データポート	ポート名	データ型	説明
Inport	UrlIn	TimedWStringSeq	リスト型で URL を取得する。
Outport	TextOut	TimedWStringSeq	リスト型でコメントを出力する。

・出力データの概要

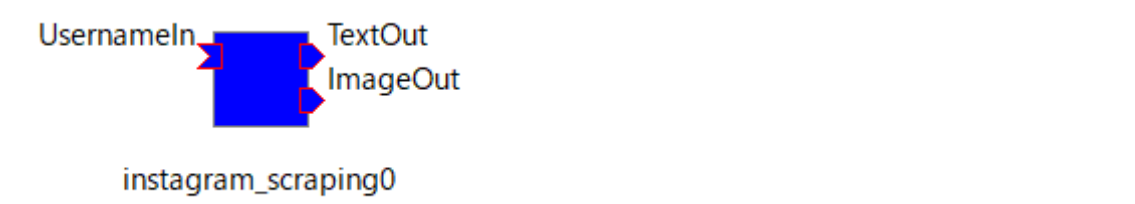
["コメント 1","コメント 2","コメント 3","コメント 4"]

↑のようにリストでコメントを出力

2.7instagram_scraping (instagram 投稿取得コンポーネント)

- ・説明
- Instkiader を利用して instagram のスクレイピングをするコンポーネント
- ・補足説明
- 取得したいユーザーのユーザーネームを入力すると、コンフィグレーション変数で指定した形式(画像かテキスト)で保存し、絶対パスをリスト型で出力する。

- ・画像



- ・データポート
- 以下の表にデータポートの説明を示す。

データポート	ポート名	データ型	説明
Inport	UsernameIn	TimedWString	スクレイピングしたいユーザーのユーザーネームを取得する。
Outport	TextOut	TimedWStringSeq	リスト型で投稿のテキストを出力する。
Outport	ImageOut	TimedWStringSeq	リスト型で画像の絶対パスを出力する。

・コンフィグレーション変数

変数名	データ型	説明
Datatype	float	取得したいデータの種類(画像 or テキスト)を指定する。 テキストを指定したい場合は"text" 画像を指定したい場合は"image" 初期値は"image"

・出力データの概要

TextOut の場合

["投稿のテキスト 1"," 投稿のテキスト 2"," 投稿のテキスト 3"," 投稿のテキスト 4"]

ImageOut の場合

["保存した画像の絶対パス 1"," 保存した画像の絶対パス 2"]

3. 本システムの利用手順

3.1 各パッケージ、モデルの補足説明

BERT・・・google の自然言語処理モデル

torchvision・・・PyTorch プロジェクトに含まれているパッケージ。コンピュータビジョンのための一般的なデータセット、モデルアーキテクチャ、および一般的な画像変換から構成されている。

transformers・・・2017 年に発表された自然言語処理（NLP）分野で用いられる深層学習モデル。

Mecab・・・オープンソースで利用できる形態素解析ソフトウェア

3.2 torchvision, transformers, Mecab のインストール方法

- ・「torchvision」のインストール方法

以下のコマンドを実行する

```
pip install torch torchvision
```

```
C:\Users\Z19096>pip install torch torchvision
```

- ・「transformers」のインストール方法

以下のコマンドを実行する

```
pip install transformers[ja]
```

```
C:\Users\Z19096>pip install transformers[ja]
```

- ・「mecub」のインストール方法

以下のコマンドを実行する

```
pip install MeCab
```

```
C:\Users\Z19096>pip install MeCab
```

- ・その他スクレイピングコンポーネントの利用方法

YoutubeAPI コンポーネント

以下のコマンドを実行する

```
pip install google-api-python-client
```


InstaramScraping コンポーネント

以下のコマンドを実行する

```
pip install instaloder
```

[2]以下の図のようにコンポーネントを接続する






The screenshot shows a Windows command prompt window with the title bar "C:\Windows\py.exe". The command prompt displays a list of Japanese words and their corresponding numerical values in a list format:

```
死, コミ, こみ, アホ, あほ, 馬鹿, バカ, はか, テフ, うさい, きもい, キモイ, クソ, 糞, 嫌い
```

The output shows the following lists:

- [0, 26, 0, 26, 0, 39]
- [1, 0, 1, 3]
- [3, 19]



C:\Windows\py.exe

[0.8498095870018005, 0.91440749168396, -0.5542663335800171, -0.819185197353363]

[illegible]

<https://atmarkit.itmedia.co.jp/ait/articles/2102/05/news027.html>

<https://www.fenet.jp/dotnet/column/language/7805>

<https://resanaplaza.com/2022/06/12/%E3%80%90%E5%AE%9F%E8%B7%B5%E3%80%91python%E3%81%A8bert%E3%81%A7%E6%84%9F%E6%83%85%E5%88%86%E6%9E%90%E3%81%97%E3%82%88%E3%81%86%E3%82%88%E3%BC%81>

https://norari-kurari-way.com/python_twitter_api_intro