付録A　スコア算出プログラムについて

３．１（３）項で述べたスコアを算出することができるプログラムを図A．１に示す。なお、図A．１において、左側の列は説明のための行番号であり、プログラムの一部ではないことに注意する。本プログラムは、近年最もよく用いられるコンピュータ言語の 1 つである Python で記述されている。また、本プログラムはDWPI抄録を処理するものである。その他のデータを処理する場合については、一部を書き換える必要がある。書き換えるべき内容については詳細を後述する。

以下、本プログラムを実行するための環境整備と、プログラムの使い方、最後に、DWPI 以外のデータへの対応等を含めプログラムの詳細な解説を行う。

A．１　環境整備

図A．１のプログラムを実行するためには、以下の手順で、Python プログラムを実行する環境と各種ライブラリを事前にインストールする必要がある。

１．anaconda のインストール
２．nltk の形態素解析機能をインストール
３．nagisa のインストール

インストール手順の詳細を以下に示す。

（1）anaconda のインストール

anaconda は、Python でデータサイエンスや機械学習、自然言語処理を行うために必要なソフトウェアが一括でインストールできるソフトウェア（ディストリビューション）である。

anaconda の Web サイト[1)]からこれをダウンロードし、インストールする。ここで、Python 3.X 版と Python 2.7版が選択できるが、3.X版(2019年5月26日時点ではXは7）を選択する。

Anaconda の使い方については説明を省略する。

（2）nltk のインストール

次に英語の形態素解析（文章を単語に切り分けて、単語の品詞を推定する処理）や単語の組み合わせを抽出する処理を実行するための機能を追加する。コンソール(anaconda コンソール)を開き図A．２のａのコマンドを実行する。

（3）nagisa のインストール

最後に、日本語の形態素解析を実行するためのライブラリを追加する。先と同様にコンソールを開き図A．２のｂのコマンドを実行する。ここで、Windows 環境では、エラーが発生することがある。その場合、Build Tools for Visual Studio 2019[2)] をインストールする。

以上の手順でプログラムを実行するための準備が整う。

（4）補足：プロキシが存在する場合

インターネット接続時にプロキシサーバを通す必要がある場合、図A．２のコマンドではインストールできない場合がある。その場合は、図A．３のコマンドを用いてインストールを行う。なお、ｂ’において、パスワードに記号が含まれる場合は URI エンコーディング（記号を%nn（n は数字）に置き換える）する。

A．２　使い方

上記環境が整備されたコンピュータならば、図A．１のプログラムを実行可能である。具体的には、プログラムを任意のディレクトリに保存後、環境整備時と同様にコンソールを開き、図A．２のｃのコマンドを実行する。

コマンドにおいて、データファイルとは、処理対象とするcsvファイル名である。処理対象のcsvファイルは、1行目にヘッダ、2行目以降に1行1公報の形でテキストが格納されたcsvである。入力ファイルの例を図A．４に示す。図A．４のように、本プログラムでは、処理対象以外の列（公報番号やタイトルなど）の列が含まれていてもよい。すなわち、特許データベースからエクスポートしたcsvファイルをそのまま入力可能である。なお、図A．１のプログラムでは、入力csvファイルの文字コードとしてShift_JIS（正確にはWindowsが用いているShift_JIS）を想定している。

また、出力もcsv形式であり、各行にキーワードとそのスコアが出力される。

A．２　プログラム解説

次に、DWPI以外の形式のデータへの対応等を行う際に重要な部分についてプログラムの詳細を解説する。図A．１のプログラムは以下のように構成される。

１．必要なライブラリの準備（1～6行目）
２．設定（8～19行目）
３．キーワードのカウント（24～47行目）
４．スコアの算出（49～54行目）
５．出力（56～58行目）

上記構成のうち、ポイントとなる部分の詳細を述べる。

（1）各種設定（8～19行目）

8～19行目には様々な設定（定数）が記述されている。各設定項目を表A．１に示す。表A．１の各項目を書き換えることで、DWPI抄録以外のデータへ対応が可能となる。

表A．１において、TARGETとはスコアの対象とする列の名前であり、COMPARISIONは、TARGETと比較される列の名前である。図A．１の例では、「抄録 - DWPI 用途」と、「抄録 - DWPI 新規性」または「抄録 - DWPI 優位性」を比較して、用途らしさを表すスコアを算出する。

また、N_GRAMはキーワードを構成する語の最大数を表す。

また、POSTAGはキーワードを構成する語の品詞を表し、STOP_WORD はスコア算出の対象外とする語を表す。ここで、POSTAGとSTOP_WORDについては、正規表現で記載されている。詳細は省略するが、正規表現とは、文字列パターンの記述方法である。図A．１の場合、POSTAGはCD、NN、またはFWで始まる品詞を表し、STOP_WORDは、数字のみで構成されるキーワード、または「e.g」、「i.e」、「etc」、「claimed」のいずれかを含むキーワードを除外している。なお、POSTAGとSTOP_WORDの書き換え例では、re.compileを省略しており、実際は re.compile(？)の「？」部分を表の文字列に置き換える。

（2）キーワードのカウント（24～47行目）

プログラムの大半を占める24行目から47行目は、各列に現れるキーワードの頻度をカウントする処理である。25～30行目は「csvファイルの各行の各処理対象の列（スコア算出対象＋比較対象）に対して処理を行う」ことを意味し、31行目～32行目で言語の判定を行っている。本プログラムでは簡便に「アルファベット、数字、カンマ、ピリオドがテキストの９割以上を占めたら英語であり、それ以外は日本語」という判

定方法を採用している。その後、事前にインストールした各ライブラリを利用してテキストの形態素解析処理を行っている（英語は34行目、日本語は37,38行目）。形態素解析を行うことで、文字列を語の配列に変換している。最後に、語の配列から1～N_GRAM個で構成されるキーワードの候補を抽出し（41 行目）、その候補がすべて対象品詞（POSTAG）に該当するかをチェックし（42 行目）、そのチェックを通過した候補をキーワードとして追加している。なお、本プログラムでは、キーワードをすべて小文字に変換し、かつ、語の境界の左右の文字がアルファベットであるときのみ境界にスペースを挿入している。

なお、先にも述べたように、本プログラムではShift_JISのcsvファイルを入力として想定している。もし、Shift_JIS以外のcsvファイルを扱いたい場合は、25行目のcp932の部分をcsvファイルの文字コードに書き換える。

（2）スコアの算出（49～54行目）

キーワードの頻度のカウントが終了した後、49行目から54行目でキーワードのスコアを算出している。この際、長さが1文字のキーワードや、キーワードがSTOP_WORDとマッチした場合は、そのキーワードをスコア算出の対象から除外している。

最後に、スコアが高い順にキーワードを並び替え、キーワードとスコアを出力するのが図A.１のプログラムの流れとなる。

注 記

1) Anaconda
https://www.anaconda.com/distribution/
（参照日 2019年5月26日）

2) Build Tools for Visual Studio 2019
https://visualstudio.microsoft.com/ja/downloads/
（参照日 2019年5月26日）

```python
import sys
import csv
import re
from collections import Counter
import nltk
import nagisa

# 計測対象の列名
TARGET = '抄録 - DWPI 用途'
# 比較対象の列名
COMPARISION = ['抄録 - DWPI 新規性', '抄録 - DWPI 優位性']
# 処理対象の列名
ALL_COLUMNS = [TARGET] + COMPARISION
# n-gram の最大値
N_GRAM = 3
# n-gram の対象とする品詞
POSTAG = re.compile(r'^(CD|NN|FW)')
# 除外語
STOP_WORD = re.compile(r'^\d+$|e\.g|i\.e|etc|claimed')

# 語の出現頻度カウンター
df = {col: Counter() for col in ALL_COLUMNS}

# キーワードの出現頻度のカウント
with open(sys.argv[1], 'r', encoding = 'cp932') as csv_file:
    for row in csv.DictReader(csv_file):
        for col in ALL_COLUMNS:
            if not (col in row) or len(row[col]) == 0:
                continue
            txt = row[col]
            a_txt = re.sub(r'[^\w\s\,\.]', '', txt, flags = re.ASCII)
            if len(a_txt) / len(txt) >= 0.9:
                # 英語の形態素解析
                words = nltk.pos_tag(nltk.word_tokenize(txt))
            else:
                # 日本語の形態素解析(nltk の形式に合わせる)
                words = nagisa.tagging(txt)
                words = list(zip(words.words, words.postags))
            # 単語 n-gram の取得
            ngrams = set()
            for ng in nltk.everygrams(words, 1, min([len(words), N_GRAM])):
                if all((POSTAG.search(w[1]) for w in ng)):
                    # 文字列化
                    ng_str = ' '.join([w[0] for w in ng]).lower()
                    ng_str = re.sub(r'([^a-z])\s([^a-z])', r'\1\2', ng_str)
                    ngrams.add(ng_str)
            df[col].update(ngrams)

# キーワードのスコア計算
r_df = {}
for w, f in df[TARGET].items():
    if len(w) <= 1 or STOP_WORD.search(w):
        continue
    r_df[w] = f / (max([df[c][w] for c in COMPARISION]) + 1)

# 出力
for w, f in sorted(r_df.items(), key=lambda x:-x[1]):
    w = w.replace('"', '""')
    print(f'"{w}",{f}')
```

図A. 1　スコア算出プログラム（DWPI 版）

ａ．**nltk** の形態素解析機能のインストール

```
> python -c "import nltk; nltk.download(['punkt', 'averaged_perceptron_tagger'])"
```

ｂ．**nagisa** のインストール

```
> pip install nagisa
```

ｃ．プログラムの実行方法

```
> python analyze.py データファイル > 出力ファイル
```

図A．２　各種コマンド

ａ’．**nltk** の形態素解析機能のインストール（改行せずに入力）

```
> python -c "import nltk; nltk.set_proxy('http://ユーザ名:パスワード@サーバ:ポート');
                                    nltk.download(['punkt', 'averaged_perceptron_tagger'])"
```

ｂ’．**nagisa** のインストール

```
> pip install nagisa --proxy="http://ユーザ名:パスワード@サーバ:ポート"
```

図A．３　各種コマンド（プロキシがある場合）

	A	B	C	D	E	F
1	公報番号	タイトル - DWPI	抄録 - DWPI 新規性	抄録 - DWPI 用途	抄録 - DWPI 優位性	…
2	JP2010527805A	Bernoulli gripper …	The gripper has a clamping …	Used for contactless admission …	The damper runs about the …	…
3	JP03138660U	Bipedal movement	A lower drive device moves …	Bipedal movement robot.	The fall of robot during curve …	…
4	JP04088332B1	Industrial robot …	The robot has impact sensor …	Industrial robot for welding.	The control characteristics …	…
5	JP03140558U	Rehabilitation…	The apparatus has a moving …	Rehabilitation apparatus for …	The apparatus can be stopped …	…
6	⋮	⋮	⋮	⋮	⋮	

図A．４　csv ファイルの例（…は省略を表す）

表A．１　設定（定数）

行数	定数名	意味	書き換え例
9	TARGET	スコアを計算したい列	**'課題'**
11	COMPARISION	TARGET と比較する列（複数可）	**['技術分野', '手段']**
15	N_GRAM	キーワードを構成する語の最大数	**4**
17	POSTAG	キーワードを構成する語の品詞	**r'^(接頭辞|名詞|接尾辞)'**
19	STOP_WORD	スコア算出の対象外とする語	**r'特許|出願|特開|特表|'**