文章からワードクラウドを作成して頻出単語を視覚的に確認する

【重要】以下の欄をすべて埋めてから応募してください

氏名：青木　雄大

所属学部学科(専攻)：自然科学研究科　環境科学専攻

email：[redacted]

【考察記述欄】作業をすべて終えてから考察を自由に記述してください。実行結果の考察だけでなく、データを集めたりプログラムを実行するにあたって苦労した点や、こんなデータを収集、解析してみたい、プログラムを改造してこんな解析や予測もしてみたいというアイデアなどでも構いません。また、実際にプログラムを改造するなど特別に工夫したことがあれば記載してください。

－－－記述欄はじめ

1985 年と 2020 年の歌詞の傾向の違いを分析した。35 年間で歌詞の傾向がどのように変化したかを確かめた。傾向の分析後、歌詞を機械学習により AI に学習させ、その再現率を確かめた。

今回の分析では、1985 年と 2020 年の有線放送の年間ヒット曲のランキング上位 20 位の歌詞を分析の対象とした。

（参考ＵＲＬ：年間有線ヒット曲 1985 年 [https://entamedata.web.fc2.com/music2/usen_rank_1985.html]、

2020 年に有線放送でヒットした J-POP と演歌の曲って何？[https://entamedata.com/2020/12/05/2020%E5%B9%B4%E3%81%AB%E6%9C%89%E7%B7%9A%E6%94%BE%E9%80%81%E3%81%A7%E3%83%92%E3%83%83%E3%83%88%E3%81%97%E3%81%9Fj-pop%E3%81%A8%E6%BC%94%E6%AD%8C%E3%81%AE%E6%9B%B2%E3%81%A3%E3%81%A6%E4%BD%95%EF%BC%9F/]）

【ワードクラウド（単語）】

1985 年は「you」、「いい」、「love」、2020 年は「君」、「僕」が特に歌詞に用いられたことが分かった。このことから 35 年間で you から君という言葉を用いるように変化したことが分かる。また、いい、love は現在でも用いられているが、君、僕という言葉より少なくなった。一方、2020 年の僕という言葉は 1985 年にはなく、代わりに I という言葉が少ないが用いられていた。

この結果から、1985 年は相手に対しての気持ちを主語を使わずに伝えていて、2020 年は自分のことを強調したうえで気持ちを伝えるように変化したのではないかと考えられる。また、2020 年は特に僕という名詞が頻出していることから、35 年間の間に自分のことを伝える歌詞が増加したと考えらえる。

【ワードクラウド（名詞）】

1985 年は、「心」、「涙」、「愛」という言葉が多く用いられた。2020 年は、「夢」、「恋」、「心」、「夜」という言葉が多く用いられた。心という言葉は昔から多く使用されていたことが分かった。また、涙、

愛、恋という言葉も昔から多く使用されていた。涙という言葉が使われていることから、昔から失恋の曲が多いのではないかと思った。

また、1985 年はその人にしか思いつかないような特徴的な名詞が多くある（例えば、アモーレ）。このことから、1985 年はその人にしか書けない歌詞が多かったと考えらえる。一方で、現在は誰にでも書けそうな歌詞が増えたのでないかと考えられる。また、現在は、電話からスマートフォンに変化したため、ダイヤルという言葉はその時代に特徴的な言葉であると思った。

【ワードクラウド（形容詞）】

1985 年は「いい」という言葉が多く使用された。2020 年は、「いい」、「強い」、「よい」という言葉が多く使用された。このことから、いいという言葉は昔から使わており、使いやすい言葉であると思った。

1985 年はせつない、恋しいという言葉があるが、2020 年にはないため、35 年間で使わなくなった言葉もあることが分かった。

2020 年は美しい、優しいという言葉があるが、1985 年にはない。1985 年は特に「いい」という言葉が使われていたため、美しい、優しいという意味を含む「いい」という抽象的な言葉で表現することが多かったのではないかと考えられる。また、現在は「いい」という抽象的な言葉ではなく、より具体的な優しい、美しいなどの言葉を使用するように変化したと考えられる。

【機械学習による歌詞の再現】

1500 回の反復回数で機械学習をした。その結果、90%以上の精度となった。再現した歌詞を見ると 1985 年の方が 2020 年よりも歌詞を再現できた。1985 年は特徴的な言葉が多いため、ある単語の次にくる言葉を学習しやすかったと考えられる。一方で 2020 年は異なる歌詞が混ざった形の再現となった。2020 年は 1985 年に比べて普遍的で共通している歌詞が多く、学習回数をより増やさないと途中で歌詞が混ざってしますと思った。また、このことから、普遍的で共通しているものを学習させるためには、より多くの学習対象が必要で、多くの学習回数が必要であると思った。

今回は、歌詞の最初の単語をランダムに指定したうえで歌詞を再現した。そのため、自分で再現する歌詞の単語を指定することで途中からでも歌詞を再現することができ、単語から歌詞を探すことが可能であると思う。

今回の分析を通して、機械学習をする上で、学習させる歌詞の数が重要で、学習させる歌詞が少ないと機械学習をするプログラムが動作しないことが分かった。反復回数や学習させる歌詞の数を考えるのが大変だった。

【まとめ】

歌詞の分析により、1985 年と 2020 年の違いが分かった。現在では使用しなくなった言葉やその時代を強調する言葉があることが分かった。また、現在は昔に比べて具体的な言葉の使用が増えたと思った。頻出単語が分かったため、その時代風の歌詞を作ることも可能であると思った。

今回は LSTM を用いて AI に学習させた歌詞の再現率を確かめた。その結果、高い再現率となり、単語から歌詞を探すことが可能であると思った。抽象的な言葉より、具体的な言葉の方が機械学習をしやすいのではないかと思った。

今後、1980 年、1990 年、1995 年などの歌詞も調べることで、どのように歌詞の傾向が変化していったかをより詳細に理解することができると思うので、様々な年代の歌詞を分析したい。また、マルコフ連鎖などを用いて、新たな歌詞を自動的に生成するプログラムを作りたいと思う。また、学習させる歌詞の数を増やすことで、新たな歌詞を生成する際、より様々な歌詞を自動的に生成できると思う。

記述欄おわり---

(ここからプログラムが始まります)

最近良く見かけるワードクラウドは、文章の中から単語を出現頻度に応じた大きさで可視化する手法です。MATLAB のテキストマイニング機能を利用して文章データから簡単にワードクラウドを作成できます。

身の回りにあるテキストデータを使ってワードクラウドを作成してみてください。よく使われている単語を視覚で確認することで、思わずおおっ！と声が出そうになる驚きや気づきがあるかもしれません。

事前準備:

- テキストファイルを用意してください。英語でも日本語でも大丈夫です。自分が毎日つけている日記やメモ書き、ブログを開設していたらブログ記事でも良いでしょう。インターネット上で公開されているパブリックドメインの文章、演説や小説などでは話し手や書き手の癖が垣間見えたりするかもしれません。ファイル形式がテキスト形式ではない場合は、Windows メモ帳を開いて、該当の文章をすべてコピーしてメモ帳に張り付けてから、テキスト形式(.txt)のファイルとして保存してください。

プログラムの実行:

下のステップ 1 を読んでプログラムを書き換えてから、このプログラムを実行してください。

Tip: メニューに実行ボタンが見当たらない時は、[ライブエディター]タブを選択してください。

プログラム

ステップ 1: テキストファイルの読み込み

以下の作業を済ませてからプログラムを実行してください。

- 事前準備で用意したテキストファイルをこのプログラムと同じ場所に保存してください。
- MATLAB の「現在のフォルダー」をこのプログラムとテキストファイルが保存されているフォルダーに変更してください。
- 下のプログラムの filename = の後のファイル名を実際に読み込むファイル名に書き換えてください。

同梱のサンプルファイル"hakusho.txt"は令和 2 年度版情報通信白書「はじめに」の本文です。

出典：「令和 2 年度版情報通信白書」(総務省) 「はじめに」　https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/r02/pdf/n1000000.pdf

令和 2 年度版情報通信白書の二次利用について(二次利用可能)：https://www.soumu.go.jp/main_content/000700124.pdf

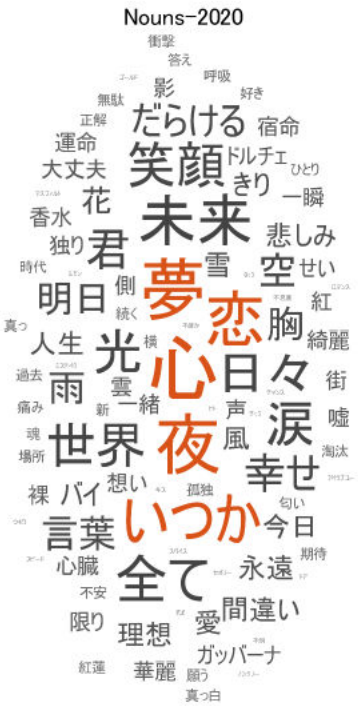
```
% 初期化します
clear

% 実際に読み込むファイル名に書き換えてください
filename1985 = ['song_1985.txt'];
filename2020 = ['song_2020.txt'];

% 上で指定したファイルを文字列として変数 text に格納します
text1985 = extractFileText(filename1985);
text2020 = extractFileText(filename2020);
%
documents1985 = tokenizedDocument(text1985);
documents2020 = tokenizedDocument(text2020);
%
tdetails1985 = tokenDetails(documents1985);
tdetails2020 = tokenDetails(documents2020);
%head(tdetails)
```

ステップ 2: ワードクラウドの作成

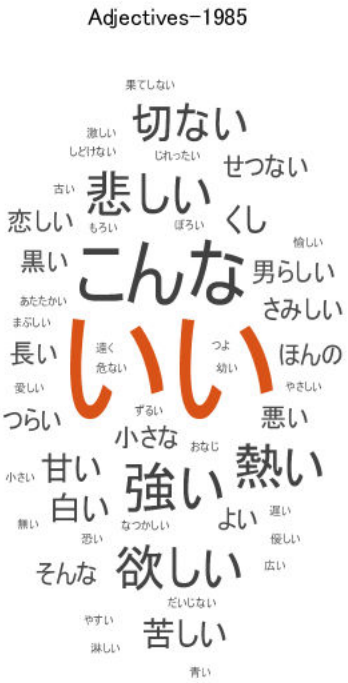
MATLAB では文章データから自動的に単語を抽出、カウントしてワードクラウドを作成してくれます。

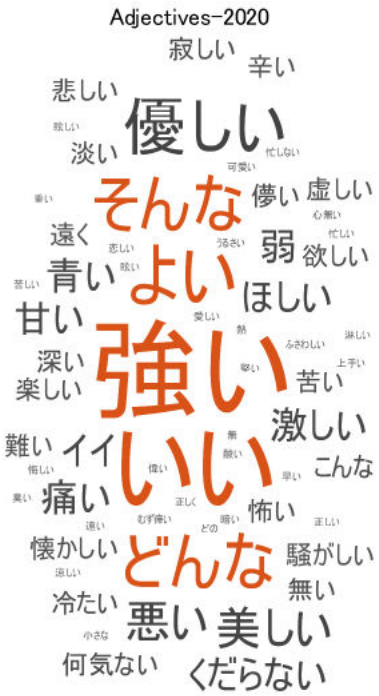
```
% ワードクラウドを作成する
% 1985
f1 = figure;
subplot(1,2,1)
wordcloud(text1985);
title("Word-1985")

% 2020
subplot(1,2,2)
wordcloud(text2020)
title("Word-2020")
```


```
% 名詞の wordcloud
% 1985
idx = tdetails1985.PartOfSpeech == "noun";
tokens = tdetails1985.Token(idx);
f2 = figure;
subplot(1,2,1)
wordcloud(tokens);
title("Nouns-1985")

% 2020
idx = tdetails2020.PartOfSpeech == "noun";
tokens = tdetails2020.Token(idx);
subplot(1,2,2)
wordcloud(tokens);
title("Nouns-2020")
```




```
% 形容詞の wordcloud
% 1985
idx = tdetails1985.PartOfSpeech == "adjective";
tokens = tdetails1985.Token(idx);
f3 = figure;
subplot(1,2,1)
wordcloud(tokens);
title("Adjectives-1985")

% 2020
idx = tdetails2020.PartOfSpeech == "adjective";
tokens = tdetails2020.Token(idx);
subplot(1,2,2)
wordcloud(tokens);
title("Adjectives-2020")
```

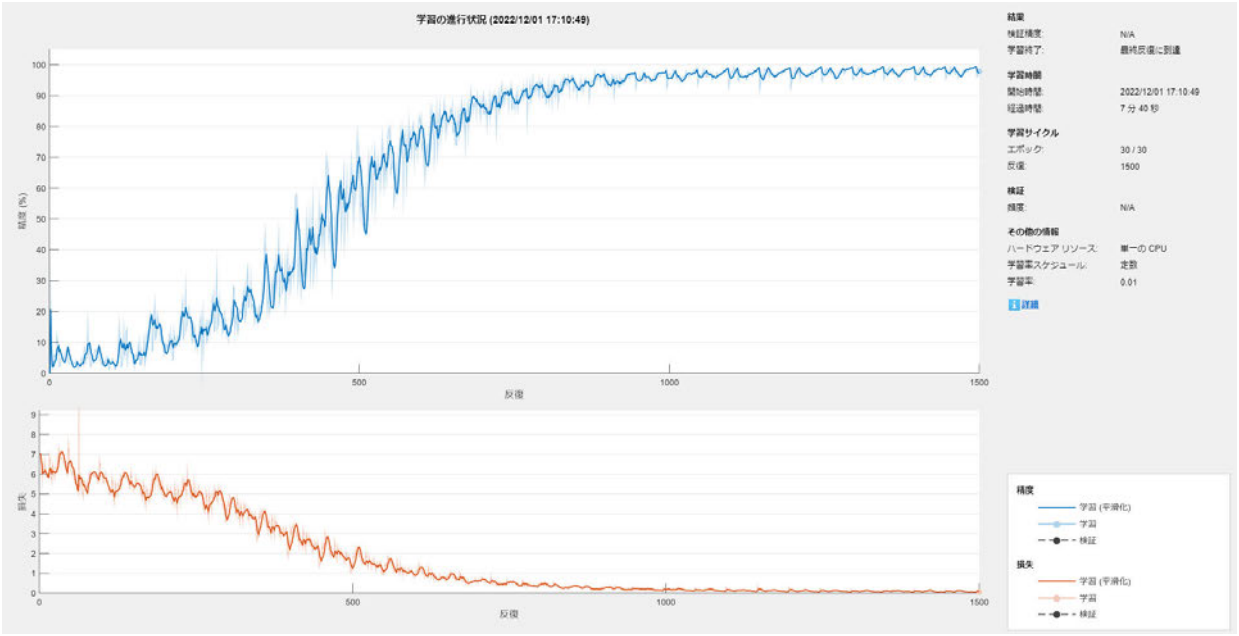




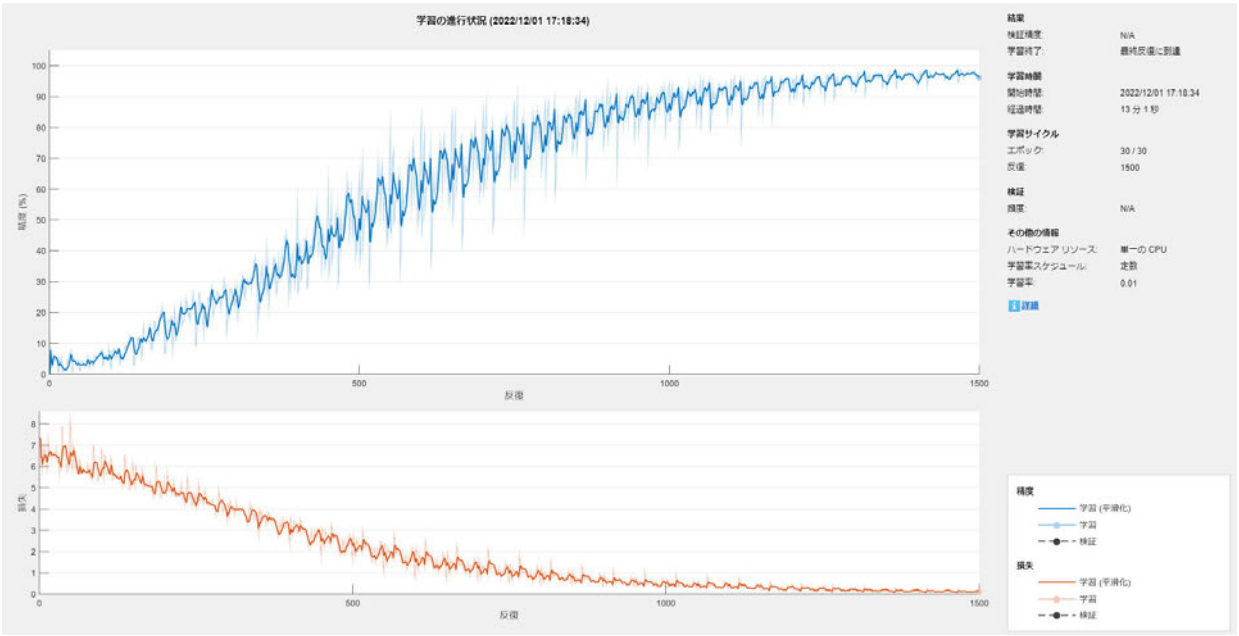
機械学習による歌詞の再現

```
% 学習データ
textData1985_ai = readlines("song_1985_ai.txt"); % 1つの歌詞につき10個の学習データとする。
textData2020_ai = readlines("song_2020_ai.txt"); % 1つの歌詞につき10個の学習データとする。
textData1985 = text_song_sort(textData1985_ai); % 今回、200個(歌詞20個×10)の学習データとした。
textData2020 = text_song_sort(textData2020_ai); % 今回、200個(歌詞20個×10)の学習データとした。
```

```
% 機械学習
% 1985
[net1985,ds1985] = net_song(textData1985);
```




```
% 2020
[net2020,ds2020] = net_song(textData2020);
```




```
% 歌詞生成
% 1985
song_1985_ai = generate_song(net1985,ds1985)
```

```
song_1985_ai =
" あなた を さがし て のばし た 指先 が 踊り の 渦 に まかれ て く 人ごみ に 押さ れ て リオ の 街 は カーニバル 銀 の 紙吹雪
```

```
% 2020
song_2020_ai = generate_song(net2020,ds2020)
```

```
song_2020_ai =
" 時には 誰 か とともに 胸 に 残り 離れ ない 苦い レモン の 匂い 雨 が 降り 止む まで は 帰れ ない 切り 分け た 果実 の 片方
```

```
save('net1985','net1985','-mat')
save('ds1985','ds1985','-mat')
save('net2020','net2020','-mat')
save('ds2020','ds2020','-mat')
```

機械学習に用いた関数

歌詞を整理する（順番に並べる）関数

```
function textData = text_song_sort(textData_ai)
a = size(textData_ai,1);
c = 1;
for b = 1:a
    if textData_ai(b,1) == ""
        if c == 1
            textData1(c,1) = join(textData_ai(1:b-1,1));
        else
            textData1(c,1) = join(textData_ai(x:b-1,1));
        end
    c = c + 1;
    x = b+1;
    end
    if b == a
        textData1(c,1) = join(textData_ai(x:b,1));
    end
end
clear textData_ai
textData = textData1;
textData(textData == "") = [];
end
```

機械学習の関数

```
function [net,ds] = net_song(textData)
documents = tokenizedDocument(textData);
ds = documentGenerationDatastore(documents);
ds = sort(ds);
inputSize = 1;
embeddingDimension = 100;
numWords = numel(ds.Encoding.Vocabulary);
numClasses = numWords + 1;

layers = [
    sequenceInputLayer(inputSize)
    wordEmbeddingLayer(embeddingDimension,numWords)
    lstmLayer(100)
    dropoutLayer(0.2)
    fullyConnectedLayer(numClasses)
    softmaxLayer
    classificationLayer];
```

```
options = trainingOptions('adam', ...
    'MaxEpochs',30, ...
    'InitialLearnRate',0.01, ...
    'MiniBatchSize',4, ...
    'Shuffle','never', ...
    'Plots','training-progress', ...
    'Verbose',false);

net = trainNetwork(ds,layers,options);
end
```

歌詞を生成する関数

```
function song_ai = generate_song(net,ds)
enc = ds.Encoding;
wordIndex = word2ind(enc,"startOfText");
%p = randperm(length(enc.Vocabulary));
%wordIndex = p(1,1);
vocabulary = string(net.Layers(end).Classes);
song_ai = "";
maxLength = 400;
while strlength(song_ai) < maxLength
    % Predict the next word scores.
    [net,wordScores] = predictAndUpdateState(net,wordIndex,'ExecutionEnvironment','cpu');
    % Sample the next word.
    newWord = datasample(vocabulary,1,'Weights',wordScores);
    % Stop predicting at the end of text.

    if newWord == "EndOfText"
        newWord = "";
    elseif newWord =='startOfText'
        newWord = "";
        %break
    end

   %{
    if newWord == "EndOfText"
        break
    end
   %}
    % Add the word to the generated text.
    song_ai = song_ai + " " + newWord;

    % Find the word index for the next input.
    wordIndex = word2ind(enc,newWord);
end
punctuationCharacters = ["." "," "’" ")" ":" "?" "!"];
song_ai = replace(song_ai," " + punctuationCharacters,punctuationCharacters);
punctuationCharacters = ["(" "‘"];
song_ai = replace(song_ai,punctuationCharacters + " ",punctuationCharacters);
net = resetState(net);
end
```

考察:

実行結果に対して、冒頭の考察記述欄に考察を自由に記述してみましょう。

保存(レポート化):

ここまでの作業お疲れさまでした。最後に作業の結果を残しておきましょう。

このプログラムは実行結果とともにレポート出力できます。

[ライブエディター]タブの[エクスポート]ボタンを押して、PDF を選んでエクスポートしてください。

保存された PDF 形式のファイルをコンテスト窓口に送付してください。