Datamining rep3 level3 205713B 朝比奈太郎 2022/06/30

Level3

ベクトル化してみよう

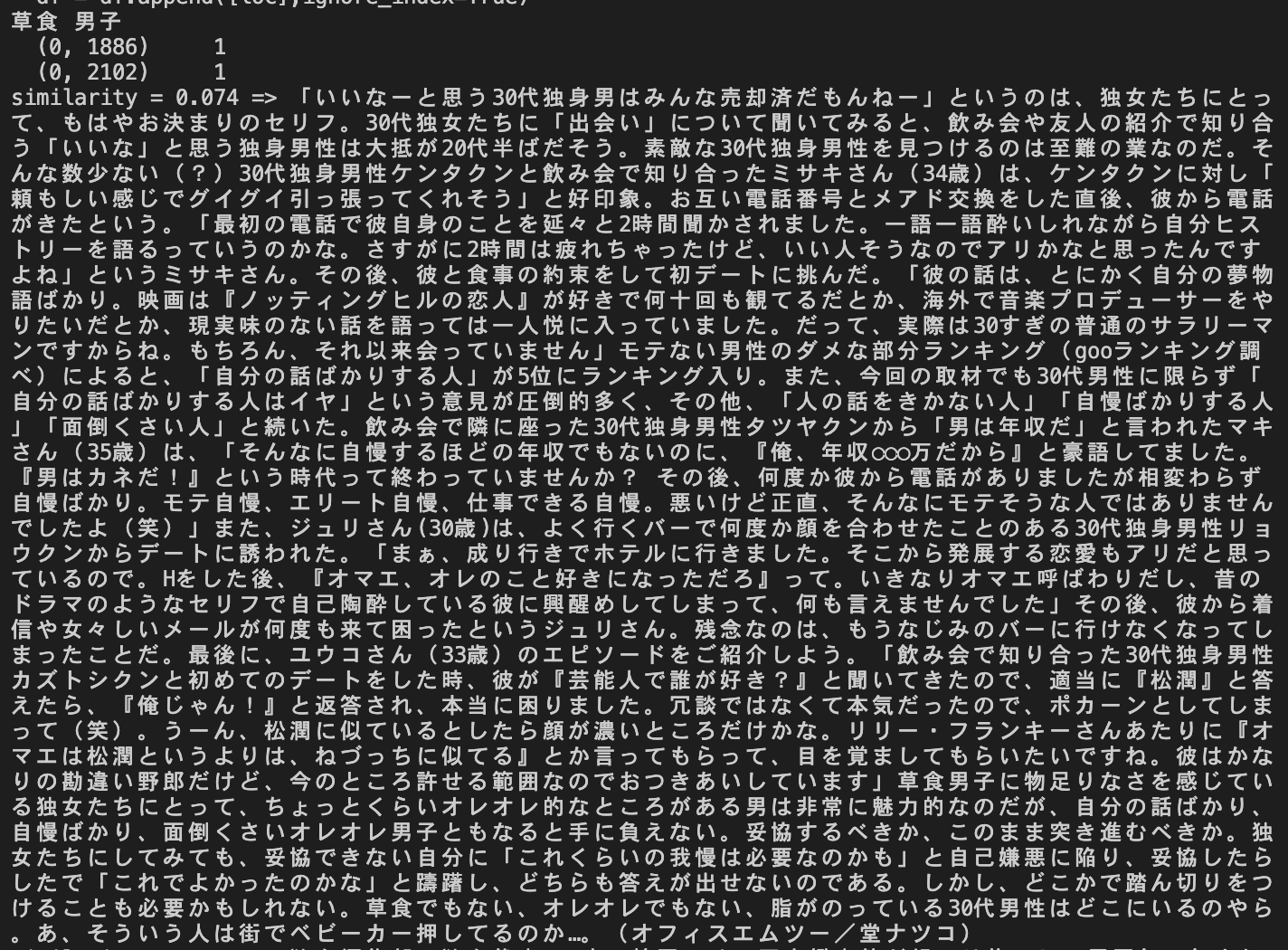
* (3) (2)で選んだ固有表現2件ならびに全文書についてBoW, TFIDF, word2vecいずれかでベクトル化し、最大類似文書上位3件を類似度とともに示せ。また類似文書について、類似度合いがどのぐらい妥当か主観で評価せよ。

ベクトル化をする際に用いた方法：Bag of Words

(2)で選んだ固有表現：草食男子, 婚カツ女子

実行結果

草食男子の場合(草食男子は”草食”, “男子”に分類されたため、両方の出現数を調べた)



草食 -> 2, 男子 -> 2(草食男子 -> 1)



草食 -> 1, 男子 -> 1(草食男子 -> 1)

背景パターン

中程度の精度で自動的に生成された説明

・類似度合い評価

similarity = 0.074の１番目の文書において、草食、男子というwordが使われていたことと、直接草食男子という表現が使われていたことより、全文章の中で一番類似度が高いことについて納得できた。しかし、文書が長いせいか、１番類似度の高い文章でもsimilarity = 0.074となったのは低く感じられた。

similarity = 0.045の２番目の文書においても１番目の文書の観点で行くと納得できる数値となった。

similarity = 0.000の３番目の文書において、草食, 男子の出現はそれぞれ0であったため、similarity = 0になるのは納得できた。また、文書を眺めてみても全く草食男子との関連のない文書であった。

婚カツ女子の場合



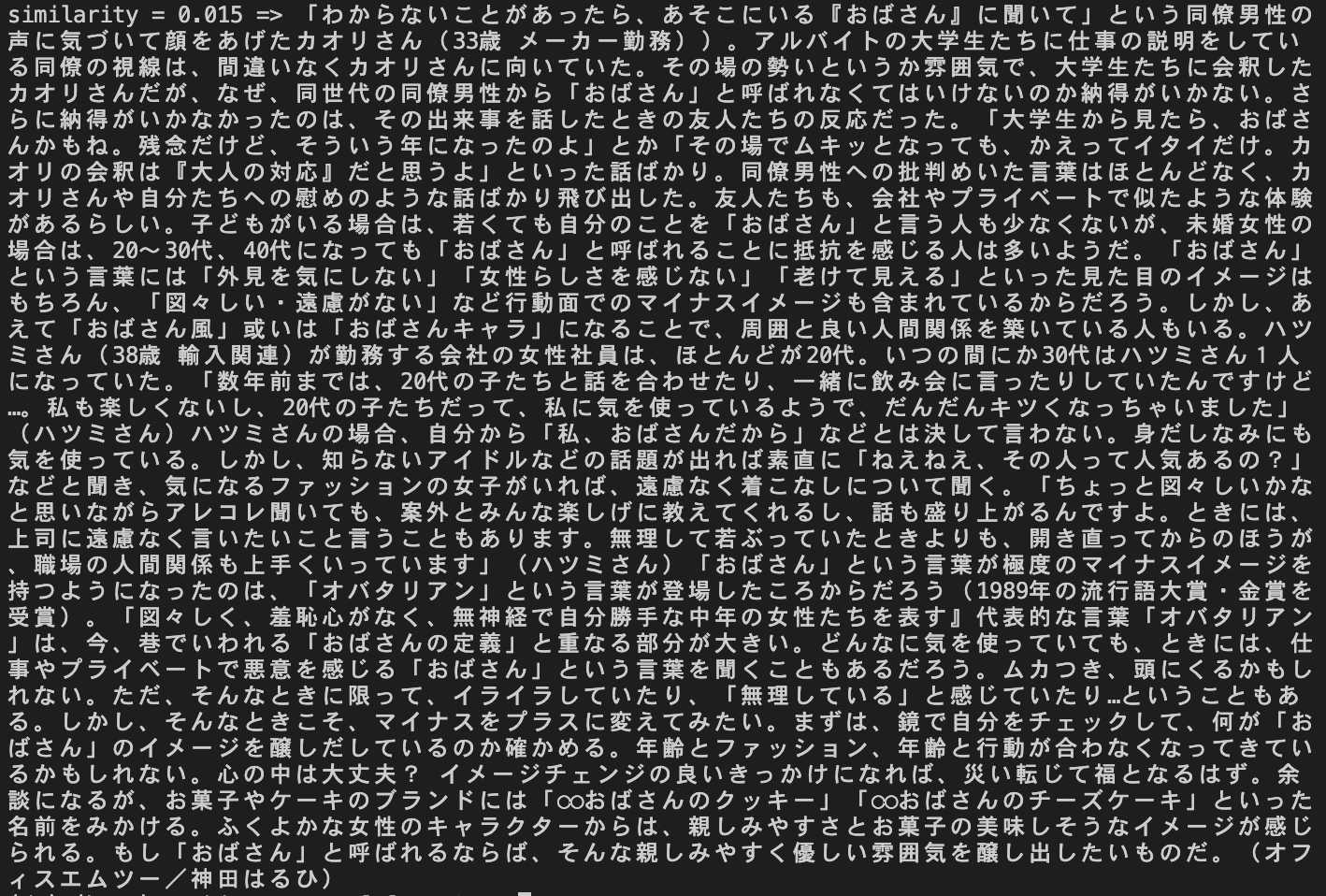
カツ、女子の2つに分けられていた。

背景パターン が含まれている画像

自動的に生成された説明

背景パターン

自動的に生成された説明



・類似度合い評価

上位からsimilarity = 0.167, 0.045 ,0.015となったが、草食男子同様、カツ、女子のword数が多い順であった。文書の内容も”婚カツ女子”に関連した順番で上位から並んでいることがわかった。

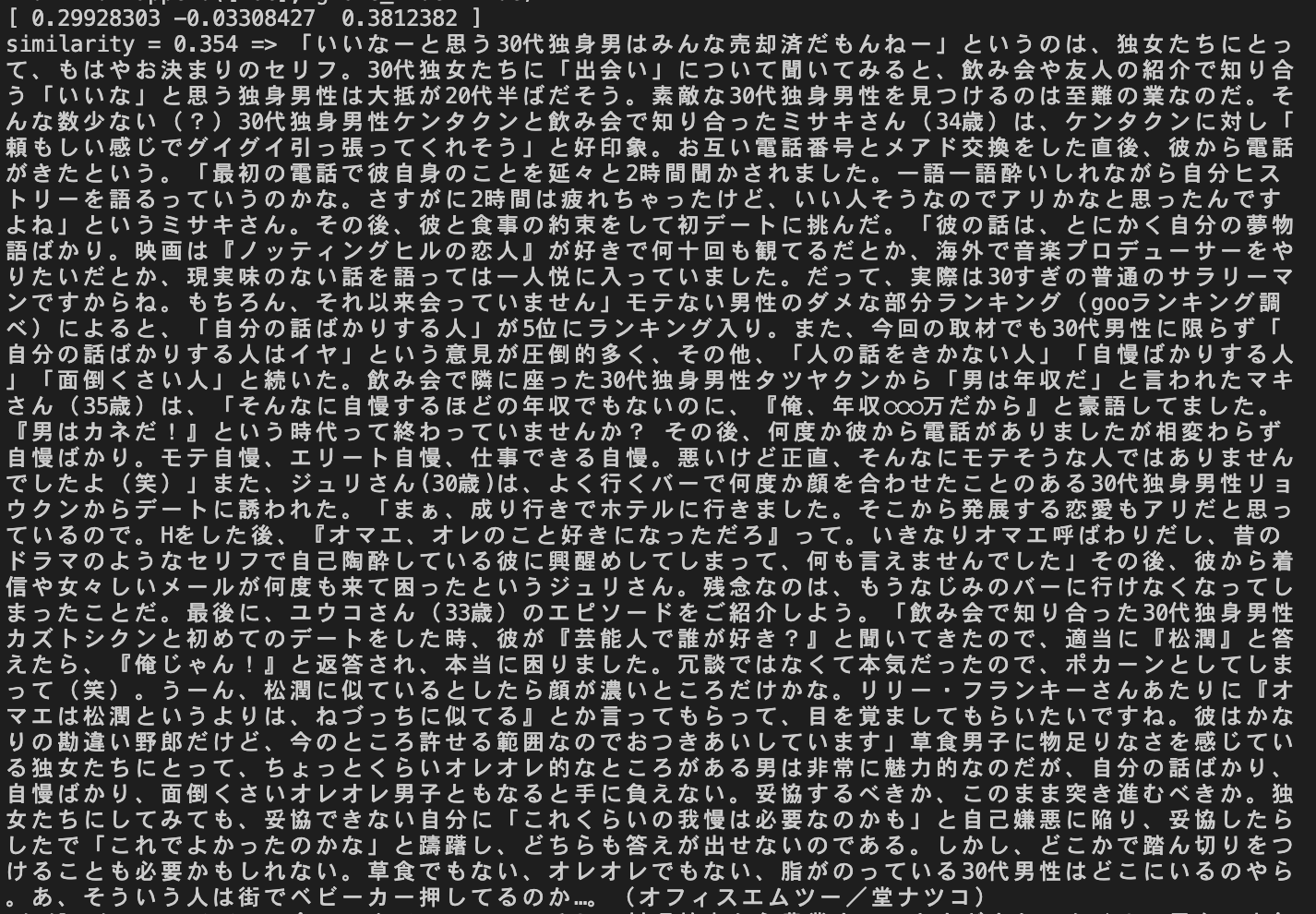
Option

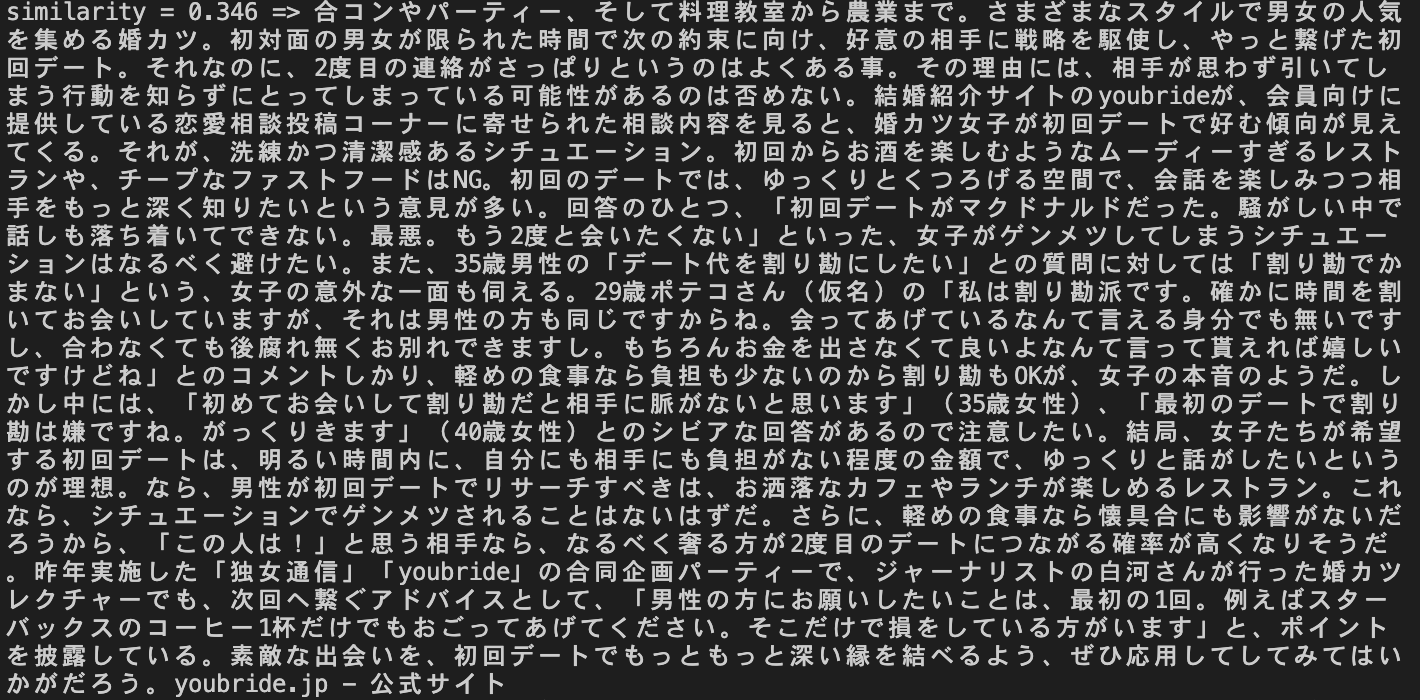
先ほどはDataFrameに対して、BoWを用いたが、word2vecを用いた場合とどう違いが出るのか気になったため、word2vecを用いて追加検証した。

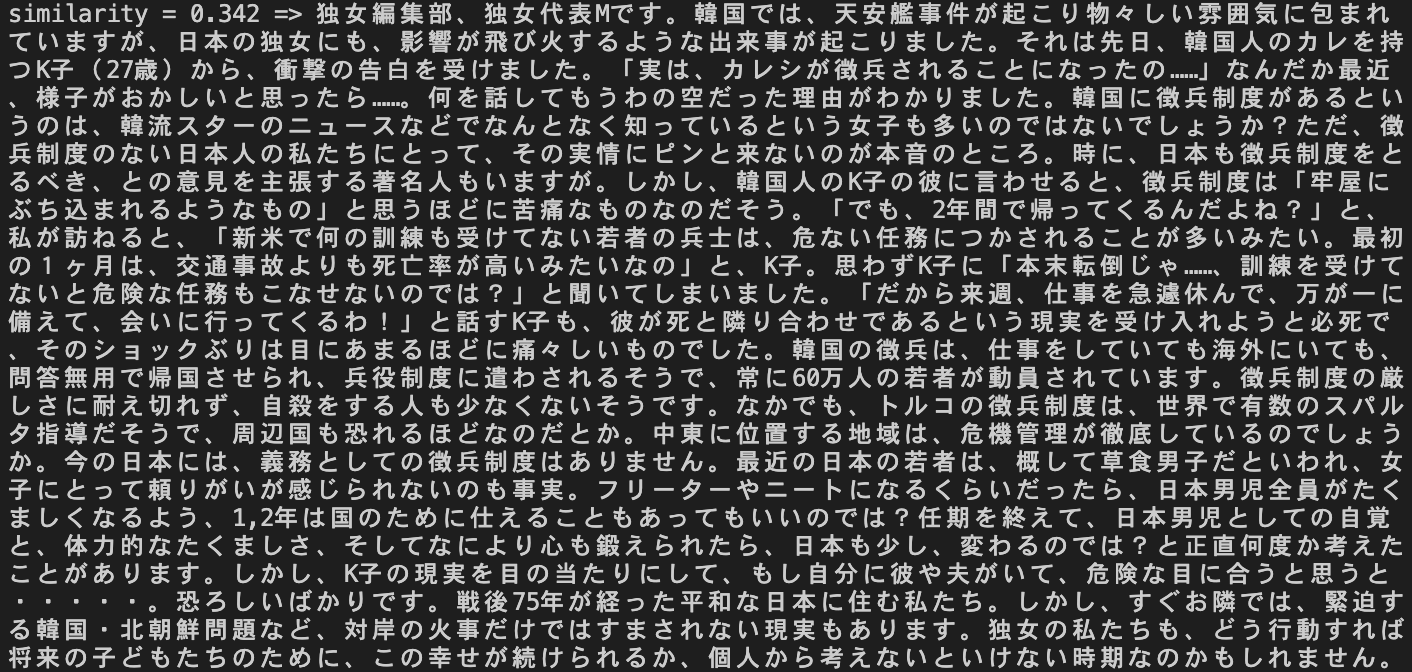
ベクトル化をする際に用いた方法：Word2vec

(2)で選んだ固有表現：草食男子, 婚カツ女子

草食男子の場合



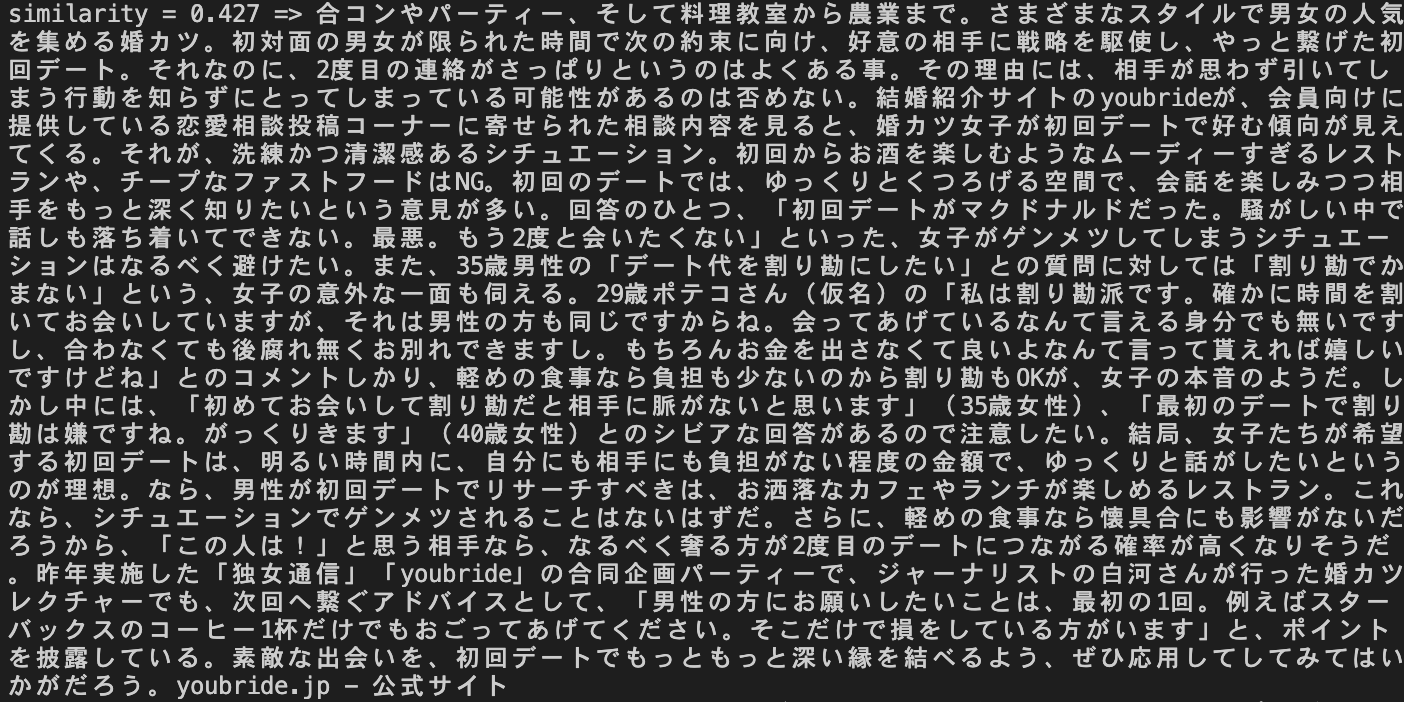


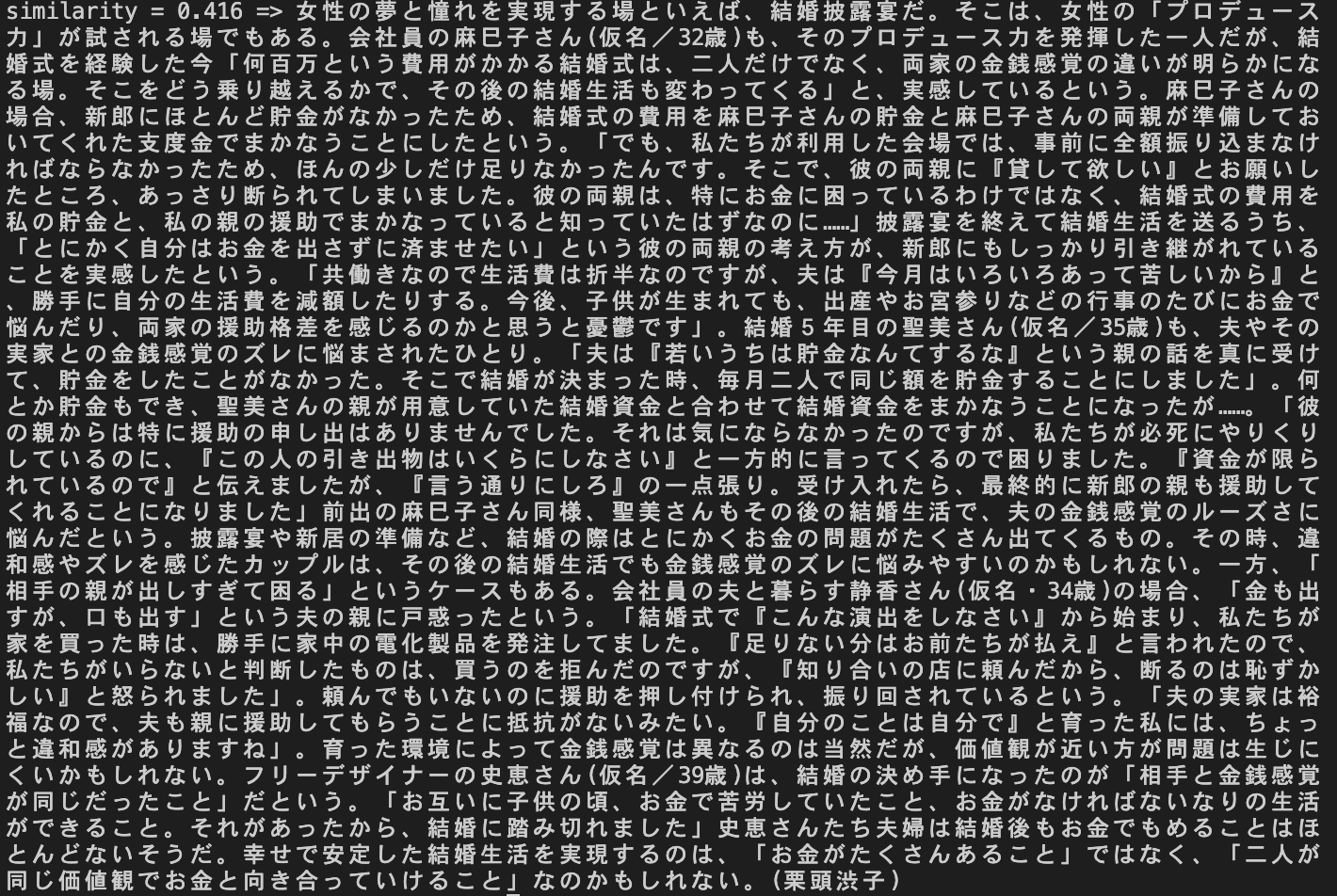


婚カツ女子の場合

背景パターン

自動的に生成された説明





“草食男子”、”婚カツ女子”の2つの固有表現を用いてBoWとの比較

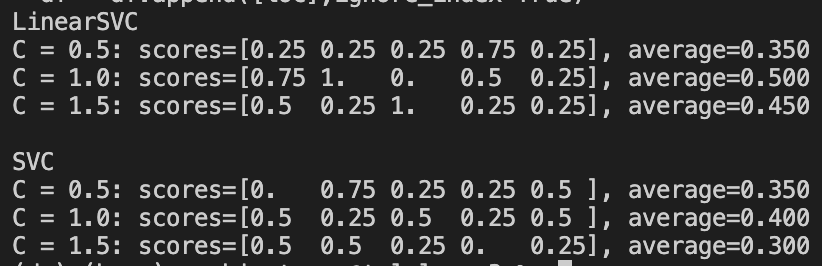
まず、全体的にword2vecが出すsimilarityの数値がBoWに比べて高かった。そのため、BoWを用いた実行結果の”草食男子”の３番目はsimilarity = 0で全く関連性のない文書が出力されたことに対して、word2vecは３番目においてsimilarity = 0.342で内容も”草食男子”というwordにふさわしいものであると感じた。全体を通して、word2vecはBoWと比べてsimilarityの値が高く、内容も主観的にwordに乗っ取られたものと感じ取れたため、用意した文書(DataFrame)にはword2vecが適していると考えられる。

・(4)５分割検定

Xにはvector\_w2v、Yにはdf[“class”]を設定し行った。

Df[“class”]は後付けのクラスであり、textを読んで恋愛や結婚関係である記事(text)を”love”、それ以外を”others”としてラベリングした。

モデルはLinearSVC、SVC を採用。それぞれハイパラメータをC=0.5, 1.0, 1.5に設定し、評価を行った。



結果LinearSVCがSVCよりもハイパラメータが全ての場合において、評価が上回った。その中でもC=1.0のときのLinearSVCが一番評価が高い結果となった。

LinearSVC(C=1.00)のときの混合行列plot結果

グラフ

自動的に生成された説明

考察

上のplot画像において、”love”であると予想されたtext(12件)中、実際に”love”とラベリングされていたものは10件であった。また、”others”と予想されたtext(8件)中、実際に”others”とラベリングされていたものは8件であった。ただ混合行列をprint文で出力した場合は２次元配列となり、今回用いたSeabornのヒートマップを用いた方がより美しく可視化されていたことがわかった。今回はclass分けが2つのみであったが、それ以上のclass分けがされたdataframeに対して使用するとグラデーション等の機能がより有効なものになると考える。

参考文献

・Seabornのヒートマップで混同行列を美しくつくる, https://evaluelog.com/post-122/ , 2022/07/02

・Scikit-learnでよく使う分類モデルまとめ, <https://betashort-lab.com/データサイエンス/機械学習/scikit-learnでよく使う分類モデルまとめ/>, 2022/07/02