5P-05

word2vec を用いたブレインストーミングシステムの提案

小山　琢也　　荻野　正

明星大学　情報学部情報学科

1. はじめに

新しいアイデアを生み出すための一つの手法として、ブレインストーミングと呼ばれる発想法が活用されている。また、ブレインストーミングで出されたアイデアを整理するための手法として、MindMap 法や KJ 法がある。今日では、研究や学習、ビジネスなど多岐にわたる分野で活用されていることもあり、これらの発想法を支援するアプリケーションが多く存在している。

しかし、ブレインストーミングの原則とされている「自由奔放」で「質より量」を重視したアイデアを出すことは、これらの発想法に不慣れなユーザーにとっては難度が高い。したがって、ある程度主題に関連した語句が提示されていれば、参加している人に対して発想支援ができるのではないかと考えた。

本研究では、word2vec と呼ばれるニューラルネットワークを用いて、ブレインストーミングにおけるユーザーのアイデア発散、整理を支援するシステムを提案する。

2. word2vec

word2vec とは、Tomas Mikolov ら[1]によって提案されたニューラルネットワークの一つであり、単語の分散表現を作る、つまり単語を密な実数ベクトルに対応付けることを目的に作られた。word2vec の特徴として、一つはベクトルの足し引きができるということである。もう一つは、異なる言語でも同じ意味の単語は同じベクトルになっているということが挙げられる[2]。

つまり、単語間の類似度算出や、単語の意味の数値化ができることで、類似度に沿った関係性の可視化が可能であり、また、２つ以上の単語から、新たに類似した単語を導くことができる。本研究では、word2vec がいわゆる連想機能を持ち合わせているということに着目して、提案するブレインストーミング支援システムに活用する。

Proposal of Brainstorming System Using word2vec
Takuya Koyama,Tadashi Ogino
Meisei University

3. 提案システム

提案するシステムについて説明する。まず、ユーザーが主題を入力することで、主題から連想される語句を表示することができる（図１）。ユーザーは、主題から連想される語句をもとに、アイデアを発散させていく事ができる。

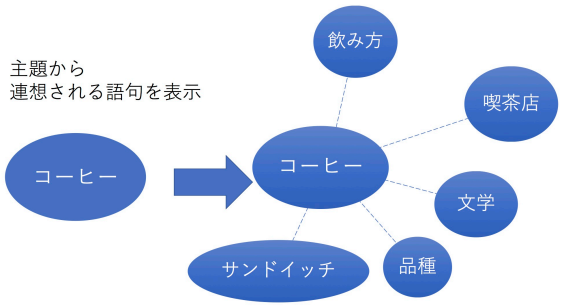

図１.主題から連想される語句を表示

また、発散途中のアイデアからも連想される語句を表示することができる（図２）。

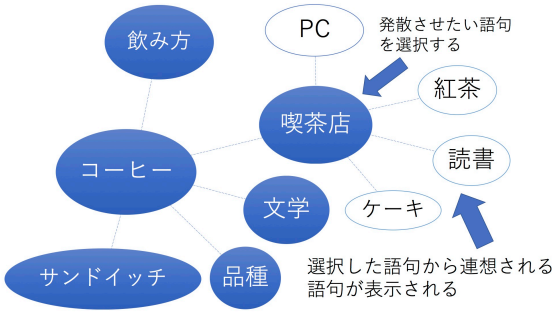

図２. 発散途中のアイデアから連想される語句を表示

そして、出されたアイデア間の関係性を自動的に可視化する事ができる (図３)。

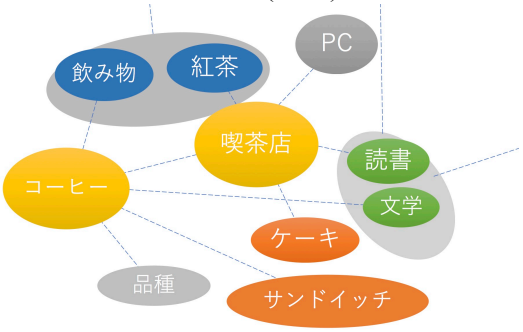

図３. アイデア間の関係性の可視化

４. 実験

本実験では、提案するシステムのうち、主題から連想される語句の表示と、関係性の可視化を行うことを目標とする。具体的には、python 上で word2vec を用いて、ある単語の類似度を算出し、その結果を processing に読み込み、可視化させる。

はじめに、word2vec での学習に用いるコーパスの生成を行う。今回は、コーパスとして Wikipedia 日本語版のデータを使う。MeCab を用いて、文章を単語に分割する「わかち書き」を行う。今回生成されたコーパスは、約 120 億単語、約 7GB のコーパスとなった。

次に、生成したコーパスを用いて学習を行う。今回 word2vec で学習する際には、ベクトルの次元を 200 次元とし、最大 5 単語までの文脈を見るというオプション設定を行った。この学習結果のデータを用いて、単語類似度を算出する。実際に“日本語”という単語を引数にしてプログラムを実行した結果を図４に示す。

```
python wikidataout.py 日本語
1    中国語   0.7827805280685425
2    英語   0.7775366902351379
3    原語   0.7461384534835815
4    韓国語   0.7423421144485474
5    朝鮮語   0.7285013198852539
6    普通話   0.7228590250015259
7    外来語   0.7198395729064941
8    台湾語   0.7055975198745728
9    外国語   0.7045632600784302
10    フランス語   0.701770544052124
```

図４.“日本語”に対する類似した単語１０件の表示

さらに、Processing でデータを扱うために、結果を python 内で JSON ファイルに出力する。

例として、日本語の類似度を算出した結果の JSON ファイルの内容は次のようになる。

```
{   "1": {
"name": "中国語",
"s": 0.7827805280685425
},
"2": {
"name": "英語",
"s": 0.7775366902351379
},
・・・
```

図５. 出力された JSON ファイルの内容

最後に、出力した JSON ファイルを processing に読み込み、単語類似度を可視化する。プログラムを実行した結果を図６に示す。

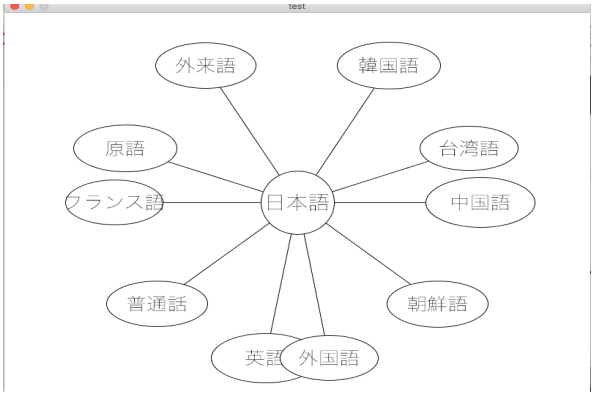

図６. 単語類似度の可視化

５. 考察

今回の実験では、ある単語の類似度を算出し、その単語の可視化を行うことを目標としたが、単純な可視化には成功したと言える。しかしながら、中心の単語と各単語間の類似度がわかりにくいため、改善する必要がある。

また、出てきたアイデアについて深掘りしたいとき、ユーザーがいつでも連想語句をリクエストできる仕組みを作ることも一つの課題である。そして、連想語句を提示する際に、どうやってばらつきを持たせるかと言うのが問題となってくる。これを解決するためには、連想語句を多めに取得し、そのうちの何件かをランダムに選ぶという方法が挙げられる。

６. おわりに

本研究では word2vec で得られる単語の分散表現と、これを用いた単語間の類似度計算に着目し、議題における連想語句の提示や、アイデアの結合による新しいアイデアの発想を支援するシステムの提案を行った。実験により、word2vec による単語類似度の算出と、processing を用いた類似度の可視化をすることができた。また、

１） processing と python 間のデータのやり取りの円滑化
２） 連想語句のランダム生成法の検討
３） ユーザーからの連想語句リクエスト方法の検討

以上の事項については、今後の課題とする。

参考文献

[1] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean, “Efficient Estimation of Word Representations in Vector Space”,Cornell University Library arXiv.org, arXiv;1301.3781v3[cs.CL],2013

[2] 西尾泰和, “word2vec による自然言語処理” O’Reilly Japan, 2014