# テキストのベクトル化のための手法

富山県立大学工学部電子情報工学科　　指導教員：奥原浩之
1715013　江崎菜々

## 1 はじめに

### テキストのベクトル化

word2vec 以外にテキストをベクトル化する方法を探してみることにした。
今回は Book of Word(BoW)、Kares、Skip-thought の 3 つについて紹介する。

### その 1　BoW

手法の中でも単純で、文章から単語が何回出てきたかカウントするのみ。
-例題文-
The sun is shining
The weather is sweet
The sun is shining, the weather is sweet, and one and is two

BoW は大文字小文字の区別はしないので「the」と「The」は同じ単語扱いになる。
そうすると、単語数は 9 個、文章は 3 個になる。
文字カウントには scikit-learn という機械学習のライブラリの CountVectorizer を使う。
まずアルファベット順に単語ごとに番号を割り振る。

出力結果
'is': 1, 'one': 2, 'sweet': 5, 'two': 7, 'shining': 3, 'sun': 4, 'the': 6, 'weather': 8, 'and': 0
]
つづいて各行に 9 個の単語が何個あるか調べる
出力結果
010110100　1 行目
and が 0 個、is が 1 個、one が 0 個、shining が 1 個、sun が 1 個、sweet が 0 個、the が 1 個、two が 0 個、weather が 0 個
010100101　2 行目
232111211　3 行目
終了

### その 2　Kares

厳密にいうと Kares の Tokenizer クラスでやる。3 つの方法で得られる情報が違う
-例題文-
I am a student. He is a student, too.
She is not a student.

まず BoW 同様文章の数、単語ごとの出現回数、単語に割り当てられたばんごうを出力する。

与えられた文章の数　:　2
与えられた文章内の単語ごとの出現回数　: 'i', 1, 'am', 1, 'a', 3, 'student', 3, 'he', 1, 'is', 2, 'too', 1, 'she', 1 'not', 1
単語ごとに割り振られた番号　:　'a': 1, 'student': 2, 'is': 3, 'i': 4, 'am': 5, 'he': 6, 'too': 7, 'she': 8, 'not': 9

○バイナリ表現
回数問わず文章中に出現したら 1 になる先頭は 0 番目で、0 番目に割り振られた番号はないからいつだって 0
0. 1. 1. 1. 1. 1. 1. 1. 0. 0.　1 行目
0. 1. 1. 1. 0. 0. 0. 0. 1. 1.　2 行目

○カウント表現
各単語を番号順に並べ、各行で出現した回数を表示する。これも先頭は 0
0. 2. 2. 1. 1. 1. 1. 1. 0. 0.　1 行目
0. 1. 1. 1. 0. 0. 0. 0. 1. 1.　2 行目

○ TF-IDF 表現
単語の重要度を表す。出現頻度が高い　＋　いくつもの文章で出現しない単語ほど数値は高くなる。この文章で一番高いのは「a」次は「student」
1 行目
0. 0.86490296 0.86490296 0.51082562 0.69314718 0.69314718 0.69314718 0.69314718 0. 0.
2 行目
0. 0.51082562 0.51082562 0.51082562 0. 0. 0. 0. 0.69314718 0.69314718

### その 3　word2vec

言葉の引き算により関係性を求めることが可能
具体的には単語の意味を加えたり、抜いたりできる
単語同士の意味の近さをベクトル化により数値化できる。
⇒進化するニュートラルネットワークの中で自然言語を演算処理できる。
例：国王ー男＋女＝女王
例：ゴリラ≒マウンテンゴリラ←勝手な予想

例題：
The sun is shining
この文章のみで考えると、各単語のベクトル表記は
the=1000, sun=0100, is=0010, shining=0001 、と文章に出てきた順に番号がつく

### その 4　Skip-thought

word2vec と特徴は似ているが、Skip-thought は長文に対して、類似文との比較ができるということである。

Woed2vec との違い。
Word2vec ⇒入力に対し、周辺に位置する単語を予測して単語の共起関係を学ぶ。
Skip-thoug ⇒入力単語の系列をエンコード、前後の文の単語を出力として順番に予測。エンコード結果をベクトル化する。
使用例：類似した文の検索、長文もいける。

### まとめ

持ち運びが楽なスマホを使ったデータ収集
集めたデータをクラスタ分析して類似性を見つける

## 参考文献

[1] 環境・生体データからの勾配・制約を考慮した粒子群最適化による行動パターン解析