機械学習を用いた特許文書分類における入力ベクトルの影響

○西尾潤 [1)2)], 安藤俊幸 [1)3)]
アジア特許情報研究会[1)], 株式会社ユポ・コーポレーション [2)], 花王株式会社 [3)]
〒101-0062　東京都千代田区神田駿河台4－3
Tel: 03-5281-6675　FAX: 03-5281-0819
E-mail: Nishio.Jun@mk.yupo.co.jp

Influence of input vector on patent document classification using machine learning

NISHIO Jun[1)2)], ANDO Toshiyuki[1)3)]
Society of Asia Patent Information[1)], YUPO Corporation[2)], Kao Corporation[3)]
4-3, Kanda-surugadai, Chiyoda-ku, Tokyo 101-0062 Japan
Phone: +81-3-5281-6675　Fax: +81-3-5281-0819
E-mail: Nishio.Jun@mk.yupo.co.jp

【発表概要】

特定技術分野における「特許請求の範囲」を入力文とし、人為的に分類ラベルを付与したデータセットを自作し、教師あり機械学習で文書分類を行うとき、機械学習モデルに入力する文書ベクトルの違いが精度に及ぼす影響について報告する。

機械学習モデルは、Tensorflow をバックエンドとする Keras で 1 次元 CNN を使用するニューラルネットワークと、非線形 SVM とを実装した。

形態素解析は MeCab と sentencepiece とを比較検討した。

また、入力ベクトルは辞書 ID 列を Keras のエンベッド層に入力する方法、形態素頻度情報、TF-IDF、Word2Vec による分散表現のそれぞれを Keras の全結合層に入力する方法及び SVM に入力する方法を比較検討した。

また、入力文字列の長さがが文書によってまちまちである点について着目し、文字列の後方をカットしたときの影響についても考察する。

本検討はアジア特許情報研究会における 2018 年のワーキングである。

【キーワード】

自然言語処理, 特許文書, 形態素解析, MeCab, sentencepiece, doc2vec, Tfidf, Support Vector Macine, IPA 辞書, NEologd 辞書, J-GLOBAL MeSH 辞書, 畳み込みニューラルネットワーク, Embedd 層, one-hot ベクトル, 規格化

1. はじめに

機械学習を用いて特許文書を扱うことは、SDI（予め設定した検索式に該当する特許情報を定期的にチェックし、必要なデータを収集・管理すること）によって得られた文書をテーマ別に分類すること（多値分類）、SDI 文書を査読するか否か判定すること（二値分類）、侵害予防調査における類似文書抽出、新規性・進歩性判断等に有用であるとして研究が進められ、一昨年からはいくつかのツールが市販されている。

一方で、機械学習のモデルや実装方法はブラックボックスであり、データがどのように処理されているか窺い知ることはできなかった。

近年、機械学習の最新技術を無料で実装できるようになり、また実装例や学習モデルが数多く公開されているため、機械学習の環境が整ってきたといえるが、特許情報を解析すること自体が企業活動と直結することもあって、解析用データセットが公開されておらず、これを自作しなければならない状況である。

2. 実験手法

2-1. 分析対象とする文書の取得

分析の対象とする技術分野をインクジェットメディアとし、インクジェットメディアに関する検索式を立て、7294 件のデータセットを得た。もっとも古い特許文献は 1983 年 2 月 16 日に公開されたもので、OCR（光学的文字認識）によってテキスト化された文献があり、OCR による誤読取はそのままとしてこれを入力文とした。

テキストは特許請求の範囲の全文であり、正規表現を使用して以下のパターンを除去するデータクレンジングを行った。

・「【」と「】」との間に1以上の文字が含まれる

・(57) など WIPO（世界知的所有権機関）が定める INID コード[1)]

2-2. 正解ラベルの付与

上記 7294 件のデータセットに対して人為的にラベルを付与した。

・2 値分類用ラベル（7294 件）

インクジェットメディアでない・・・0

インクジェットメディアである・・・1

さらに 1 を付与した文書を以下の通り細分類して 3 値分類用ラベルを付与した。

・3 値分類用ラベル（5230 件）

非浸透性（樹脂）基材・・・1

1 以外で素材が限定される基材・・・2

素材が限定されない基材・・・3

2-3. 形態素解析

MeCab[2)]と Sentencepiece[3)]とを用いた。

2-4. 文書ベクトル化

Python Gensim に実装されている Doc2vec[4)]を用い、形態素解析済みの文書を学習させて得たモデルから、分散表現ベクトルを得た。

加えて、Python scikit learn に実装されている TF-IDF（TfidfVectorizer[6)]）、Python collection ライブラリを用いて得た単語 ID を Keras の Embedd 層に入力する方法を比較検討した。

2-5. 分類器

Python scikit learn に実装されている SVM[7)] (Support Vector Macine)と Keras Sequential モデル [8)]で構成した 1 次元 CNN[9)] (畳み込みニューラルネットワーク)を 2 層持つニューラルネットワークとを用いた。

SVM では線形カーネル、多項式カーネル、RBF カーネル、シグモイドカーネルのそれぞれにおいてガンマ値およびｃ値をスライドさせて最適化する方法と

した。

教師データ(trainning):検証データ(validation):予測データ(predict)の比率を5:2/3:1/3とした5分割(5-hold)法による交差検証を行い、5個の精度の平均値を評価値とした。

3. 結果と考察

3-1. 形態素解析

MeCabに同梱されたIPA辞書[9]とユーザー辞書としてNEologd辞書[10](Web上から得た新語)、J-GLOBAL MeSH辞書[11](J-GLOBAL科学技術用語)、およびSentencepieceとを用いて形態素解析を行った結果を表1に示す。

表1. 形態素解析方法による違い

Mecab (IPA 辞書)
['界面活性剤', '、', '有機溶剤', '、', '及び', '樹脂', 'を', '含有', 'する', 'インク', 'で', 'あっ', 'て', '、', '前記', '界面活性剤', 'として', 'シリコーン', '界面活性剤', '及び', 'フッ素', '界面活性剤',・・・]
Mecab (NEologd 辞書)
['界面活性剤', '、', '有機溶剤', '、', '及び', '樹脂', 'を', '含有', 'する', 'インク', 'で', 'あっ', 'て', '、', '前記', '界面活性剤', 'として', 'シリコーン', '界面活性剤', '及び', 'フッ素', '界面活性剤',・・・]
Mecab (J-GLOBAL MeSH 辞書)
['界面', '活性', '剤', '、', '有機', '溶剤', '、', '及び', '樹脂', 'を', '含有', 'する', 'インク', 'で', 'あっ', 'て', '、', '前記', '界面', '活性', '剤', 'として', 'シリコーン', '界面', '活性', '剤', '及び', 'フッ素', '界面', '活性', '剤',・・・]
Sentencepiece
['_', '界面活性剤', '、', '有機溶剤', '、', '及び樹脂を含有するインク', 'であって', '、', '_前記界面活性剤', 'として', 'シリコーン界面活性剤', '及び', 'フッ素界面活性剤', 'を含み', '、', '_前記', '有機溶剤として', '', 'フッ素界面活性剤',・・・]

IPA辞書に対してNEologd辞書による結果は全く変化がなかったが、Web上に出現する新語に科学技術用語が極めて少ないことによると考えられる。本検討では以後NEologd辞書による形態素解析結果を使用しない。

IPA辞書に対してJ-GLOBAL MeSH辞書による結果は科学技術用語の粒度が小さい傾向がある。

Sentencepieceによる結果は、科学技術用語に対しては粒度が大きい傾向がある。たとえば、シリコーン界面活性剤とフッ素界面活性剤とは異なる語として切り出された。また、文頭に「_」(アンダーバー)が付与されている特徴がある。

3-2. 文書ベクトル化

ここでは形態素を「単語」と称する。

頻度情報は各文書における単語の出現頻度を行列にしたもので、文書数(行)×単語数(列)のベクトルが得られた。

表2. 頻度情報で得られた文書ベクトル例

```
[[ 0.   32.   22.   ...  0.   0.   0.]
 [ 0.   18.   30.   ...  0.   0.   0.]
 [ 0.   25.   44.   ...  0.   0.   0.]
 ...
 [ 0.   2.    3.    ...  0.   0.   1.]
 [ 0.   7.    8.    ...  0.   0.   0.]
 [ 0.   6.    7.    ...  0.   0.   0.]]
```

TFIDFは文書に含まれる単語の重要度を示す指標で、1文書内での頻度情報(tf)と全文書に対する単語のレア度(idf)を掛け合わせたものである。スパース(疎)な文書ベクトルが得られた。

表3. TFIDFで得られた文書ベクトル例

```
[[0.0650  0.      0.      ...  0.  0.  0. ]
 [0.0577  0.0286  0.1386  ...  0.  0.  0. ]
 [0.      0.      0.0311  ...  0.  0.  0. ]
 ...
 [0.      0.      0.      ...  0.  0.  0. ]
 [0.0683  0.      0.      ...  0.  0.  0. ]
 [0.0913  0.      0.      ...  0.  0.  0. ]]
```

単語ID列は、あらかじめ頻度が多い順に並べた単語辞書(ID と単語)を作り、形態素解析済みの文書を単語IDに置換して作ったもので、文書数(行)×最大単語数(列)のベクトルが得られた。最大単語数に満たない文書は文の後方を0で埋めてある(0パッディング)。

表 4.　単語 ID 列で得られた文書ベクトル例

```
[[2645    0    172938 ...  0    0    0]
 [4       5843 23     ...  0    0    0]
 [7       0    1      ...  0    0    0]
 ...
 [26821   0    83343  ...  0    0    0]
 [83343   0    13     ...  0    0    0]
 [13      0    75     ...  0    0    0]]
```

単語 ID 列の特徴は、文書ベクトル中の数値同士が直接関係しない点である。従って SVM には適用できず、ニューラルネットワークには Embedd 層に入力する。

また、前述の通り文書の長さはまちまちであることから、0 パッディングによって各文書の長さ(行列の列次元)を揃えている。文書の先頭からどこまでを区切るかによって有効データの割合(データ充填率)を調整できる。区切りを短くすることでデータ充填率が上がることと計算コストが低下することが期待されるが、仮に文の後方に重要な単語があったときには分類精度が低下する懸念がある。

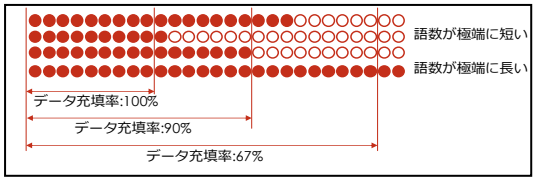


図 1.　データ充填率の概念

分散表現は単語を高次元の実数ベクトルで表現する技術であるが、これを応用して文書ベクトルを得ることができる [4]。

表 5.　分散表現で得られた文書ベクトル例

```
 [[-0.0818314   -0.03471915   0.20251338 ...
0.01916058 -0.19111413   -0.05480235]
 [-0.02651453   0.04319183 -0.10081583 ...
0.07938665   0.164984   -0.03279184]
 [-0.01340349   0.00436418   0.10938133 ...
0.02093608 -0.01491436    0.05139681]
 ...
 [ 0.1016487   -0.08065614   0.0379178   ...
0.08526418   0.0536143    0.04634216]
 [ 0.00051857   0.05571419 -0.1199406   ...
0.11025959   0.07439005    0.05453833]
 [-0.07418571   0.22156447 -0.0777843   ...
0.11831082   0.02078068   -0.00275185]]
```

表 5 に示すように、パラメータで設定した次元を持つ密なベクトルが得られた。

3-3. SVM と文書ベクトル入力による分類

TFIDF、Doc2Vec(PV-DBOW)、Doc2Vec(PV-DM)のそれぞれ 150 次元および 300 次元ベクトルで比較した。

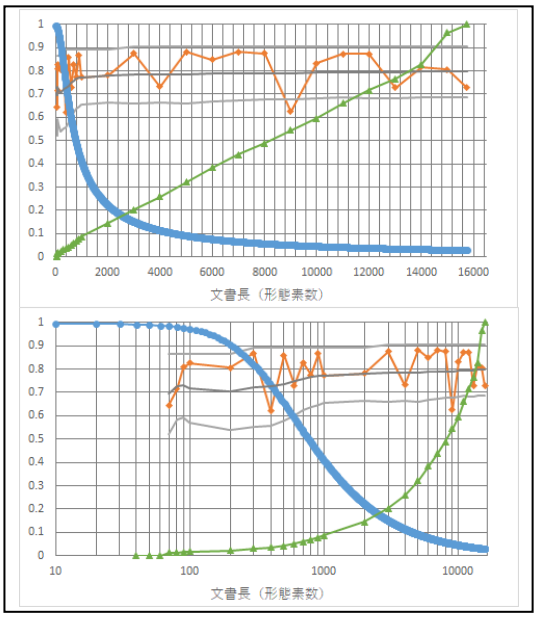
表 6 に所要時間と分類精度例を示す。同サイズのベクトルでもベクトル化手法によって所要時間が大きく変動する。

表 6.　SVM 実行例

ﾍﾞｸﾄﾙ化手法	ｻｲｽﾞ	ｶｰﾈﾙ	精度	所要時間
TFIDF	150dim	rbf	0.893	3396
TFIDF	300dim	rbf	0.893	3568
D2V(DBOW)	150dim	rbf	0.893	3666
D2V(DBOW)	300dim	rbf	0.891	6590
D2V(DM)	150dim	rbf	0.870	1785
D2V(DM)	300dim	rbf	0.870	3296

3-4. 1 次元 CNN と辞書 ID 入力による分類

MeCab(IPA 辞書)形態素解析による入力文書行列の列次元を変更したときのデータ充填率，精度，計算に要する時間を図 2 と図 3(横軸対数表示)に示す。



（上）図 2.　1 次元 CNN 実行例
（下）図 3.　同　対数表示

データ充填率（青，●）は文書長 100 で 97.2%であるが、ここから急激に減少し、文書長 1000 で 40.8%、文書長 2000 で 22.3%となった。文書長が Keras Embedd 層の入力ベクトルサイズになるため、所要時間（緑，▲）は文書長にほぼ比例した。無印は学習時の検証データ精度[val_acc]で、文書長が長いとやや上がるが急激な変化はなかった。予測精度（橙，◆）の変動の方が大きい。

3-5. 1 次元 CNN と文書ベクトル入力による分類

カーネルサイズ 3 の 1 次元畳み込み層を持ち、出力 1 次元（活性化関数：sigmoid）または 2 次元（活性化関数：softmax）の 2 種類のニューラルネットワークを構成し、ラベルもそれに合わせて 1 次元ベクトルと 2 次元 one-hot ベクトルとを使用した。

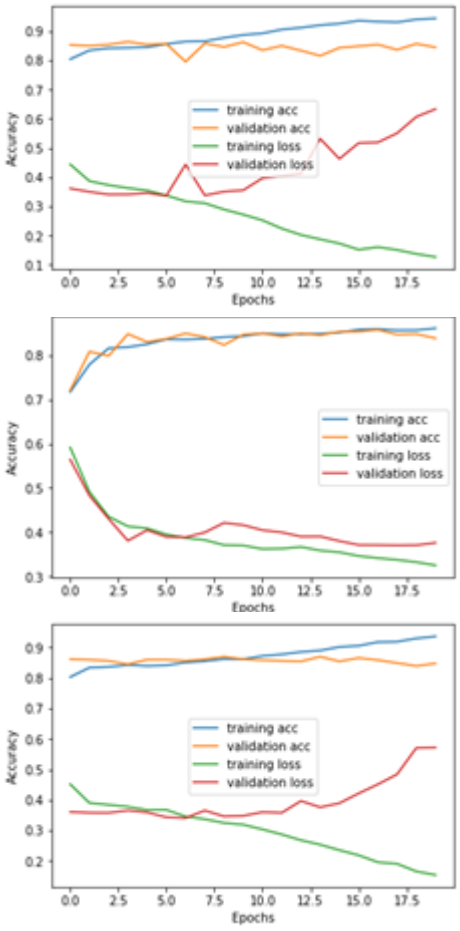
図 4～6 に入力ベクトルの規格化の効果を示す。特に validation loss を示す赤線に着目すると、1～0 の規格化の時に過学習が抑制されていることがわかる。以降、1～0 規格化を行って検討した。

表 7 に実行例を示す。

表 7.　1 次元 CNN 文書ベクトル実行例

ﾍﾞｸﾄﾙ化手法	ｻｲｽﾞ	出力層	精度	所要時間
TFIDF	150dim	1dim	0.850	125
		2dim	0.851	116
TFIDF	300dim	1dim	0.860	211
		2dim	0.860	215
D2V(DBOW)	150dim	1dim	0.848	131
		2dim	0.856	122
D2V(DBOW)	300dim	1dim	0.856	212
		2dim	0.860	221
D2V(DM)	150dim	1dim	0.826	133
		2dim	0.833	133
D2V(DM)	300dim	1dim	0.835	217
		2dim	0.836	231

ベクトル化手法としては TFIDF と PV-DBOW による Doc2Vec の精度がやや高い。ベクトルサイズは 300 次元のほうが精度がやや高い。また、Doc2Vec のときに限り出力 2 次元の方が出力 1 次元



（上）図 4.　規格化なしでの実行例
（中）図 5.　1～0 規格化での実行例
（下）図 6.　1～-1 規格化での実行例

よりやや精度が高かった。

4. おわりに

執筆時点では SVM による分類が最も精度が高かったが、ニューラルネットワークでの分類精度は使用したモデルの構成、教師データ／検証データ／予測データの数に大きく影響されるため、精度は絶対的なものではない。

特許由来文章には科学技術的な意味を持たない語が高頻度で含まれるが、これらを簡便な手法で除去したときの精度について検証中である。

本検討はアジア特許情報研究会における 2018 年のワーキングである。

安藤さんをはじめ、同研究会のメンバーに深謝の意を表する。

5. 参考文献

[1] INIDコード一覧表, https://www.inpit.go.jp/content/100029977.pdf(参照 2019-04-13)

[2] MeCab, http://taku910.github.io/mecab/(参照 2019-04-13)

工藤拓 ほか. Conditional Random Fields を用いた日本語形態素解析. 情報処理学会研究報告自然言語処理(NL). 2004, vol 47, p. 89-96

[3] sentencepiece, https://github.com/google/sentencepiece(参照 2019-04-13)

工藤拓, Google 合同会社. サブワード正則化: 複数のサブワード分割候補を用いたニューラル機械翻訳. 言語処理学会 第 24 回年次大会 発表論文集. 2018, p. 37-40

[4] genism, doc2vec, https://radimrehurek.com/gensim/models/doc2vec.html (参照 2019-04-13)

Le, Q., Mikolov, T., Distributed Representations of Sentences and Documents, CoRR, abs/1405, 4053, 2014, p. 1-9

[5] scikit learn, TfidfVectorizer, https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html(参照 2019-04-13)

[6] scikit learn, SVC https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html (参照 2019-04-13)

[7] Keras Sequential モデル API https://keras.io/ja/models/sequential/ (参照 2019-04-13)

[8] Keras Conv1D https://keras.io/ja/layers/convolutional/ (参照 2019-04-13)

[9] ipadic version 2.7.0 ユーザーズマニュアル http://chasen.naist.jp/snapshot/ipadic/ipadic/doc/ipadic-ja.pdf (参照 2019-04-13)

[10] mecab-ipadic-NEologd : Neologism dictionary for MeCab https://github.com/neologd/mecab-ipadic-neologd (参照 2019-04-13)

[11] J-GLOBAL MeSH 辞書 https://dbarchive.biosciencedbc.jp/jp/mecab/data-2.html (参照 2019-04-13)