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)]

はじめに

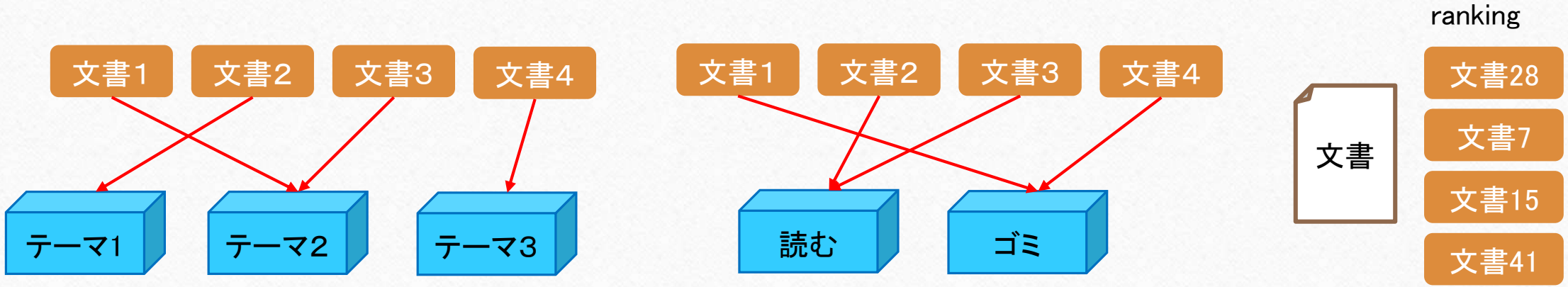
機械学習を用いて特許文書を扱う場面

・SDIで得られた特許文書をテーマ別に分類するとき

・SDIで得られた特許文書を精査するかゴミとして捨てるか判別するとき

・侵害予防調査における類似文書抽出

・新規性・進歩性判断



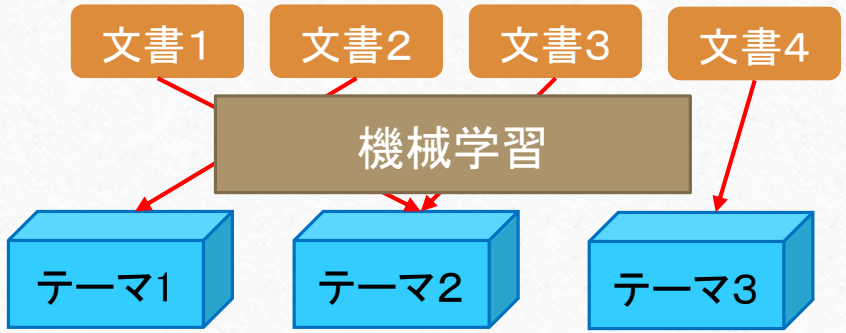
SDI:（Selective Dissemination of Information） あらかじめ検索式を登録しておき、定期的に新たな特許を抽出する機能）

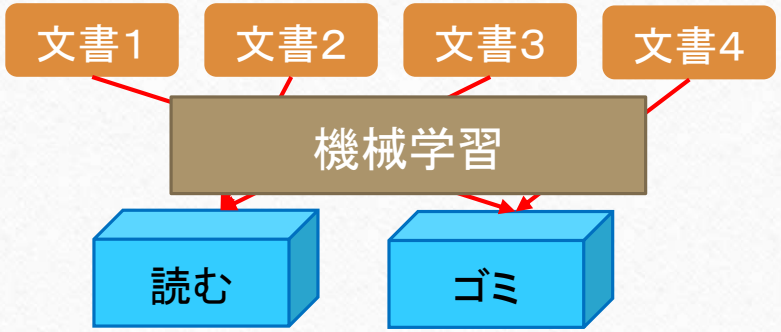
侵害予防調査：自らの事業活動が他人の特許に抵触することを回避するために行う調査

機械学習の問題点

機械学習の問題点

・機械学習のモデルや実装方法がブラックボックス



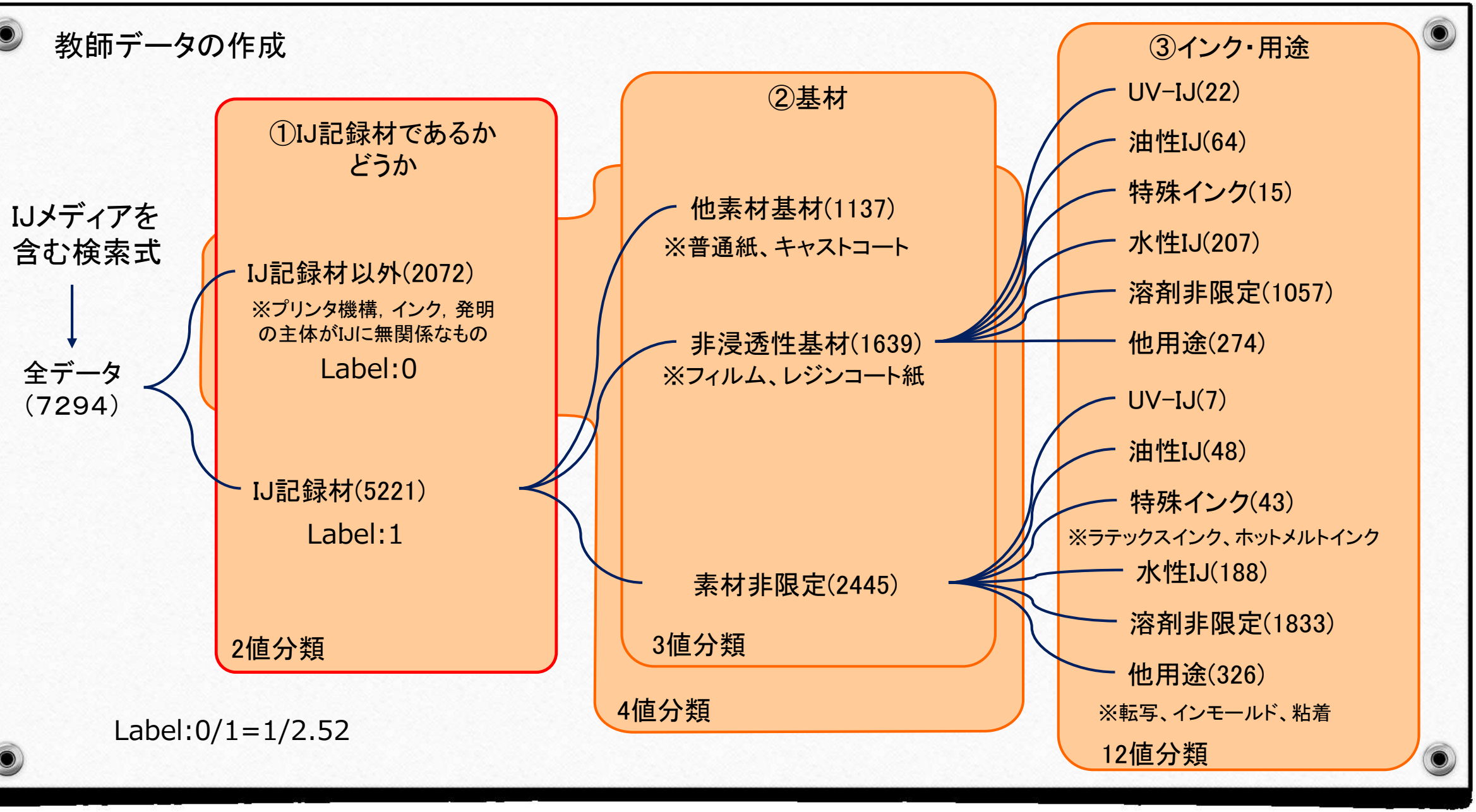


・教師データの作成が面倒　自動作成も難しい

さらに特許情報を扱う場合は

・企業から解析用データが提供されず、開発者が自前でデータを作る　ことが多い

教師データの作成


IJメディアを
含む検索式
全データ
（7294）
①IJ記録材であるか
どうか
IJ記録材以外(2072)
※プリンタ機構，インク，発明
の主体がIJに無関係なもの
Label:0
IJ記録材(5221)
Label:1
2値分類
②基材
他素材基材(1137)
※普通紙、キャストコート
非浸透性基材(1639)
※フィルム、レジンコート紙
素材非限定(2445)
3値分類
4値分類
③インク・用途
UV-IJ(22)
油性IJ(64)
特殊インク(15)
水性IJ(207)
溶剤非限定(1057)
他用途(274)
UV-IJ(7)
油性IJ(48)
特殊インク(43)
※ラテックスインク、ホットメルトインク
水性IJ(188)
溶剤非限定(1833)
他用途(326)
※転写、インモールド、粘着
12値分類
Label:0/1=1/2.52

機械学習用データ

	学習させるデータ(x)	クラスラベル(y)
学習	[[0.45330593 0.53856367 0.54323846 ... 0.6245525 0.54924446 0.34767133] [0.29302627 0.54410183 0.4921428 ... 0.5259048 0.4706858 0.65297157] [0.37936592 0.473032.. 0.4001513 0.4228363... ... [0.51291215 0.5794438 0.570155 ... 0.65197116 0.48016682 0.5000225] [0.5576463 0.54231286 0.43099108 ... 0.5447404 0.45836136 0.575232]]	[[0.] [1.] [1.] ... [0.] [0.]]
予測		

文書ベクトル

5835個

学習モデル $y = f(x)$

300次元

機械学習用データ

	学習させるデータ(x)	クラスラベル(y)
学習	[[0.45330593 0.53856367 0.54323846 ... 0.6245525 0.54924446 0.34767133] [0.29302627 0.54410183 0.4921428 ... 0.5259048 0.4706858 0.65297157] [0.37936592 0.473032 ... 0.4001513 0.4228363... ... [0.51291215 0.5794438 0.570155 ... 0.65197116 0.48016682 0.5000225] [0.5576463 0.54231286 0.43099108 ... 0.5447404 0.45836136 0.575232]]	[[0.] [[1. 0.] [[1. 0. 0.] [1.] [0. 1.] [0. 0. 1.] [1.] [0. 1.] [0. 1. 0.] ... ... [0.] [1. 0.] [1. 0. 0.] [0.]] [1. 0.]] [0. 1. 0.]]
予測	[[0.4253725 0.46143517 0.46506587 ... 0.5001111 0.5638405 0.47532108] [0.39791682 0.47899336 0.3949293 ... 0.5145807 0.47782594 0.52436244] ... [0.5402739 0.44076866 0.5362779 ... 0.4866611 0.4708381 0.5173907] [0.4652203 0.3781388 0.4520514 ... 0.5372063 0.403151 0.4744484]]	[[1.] [[0. 1.] [[0. 1. 0.] [0.] [0. 1.] [0. 0. 1.] ... ... [1.] [0. 1.] [1. 0. 1.] [0.]] [1. 0.]] [1. 0. 0.]]

文書ベクトル

5835個

学習モデル $y = f(x)$

1459個

2値分類表現(1次元/2次元)

one-hot ベクトル

SVM（サポートベクターマシン）

線形入力素子を利用して2クラスのパターン識別器を構成する手法

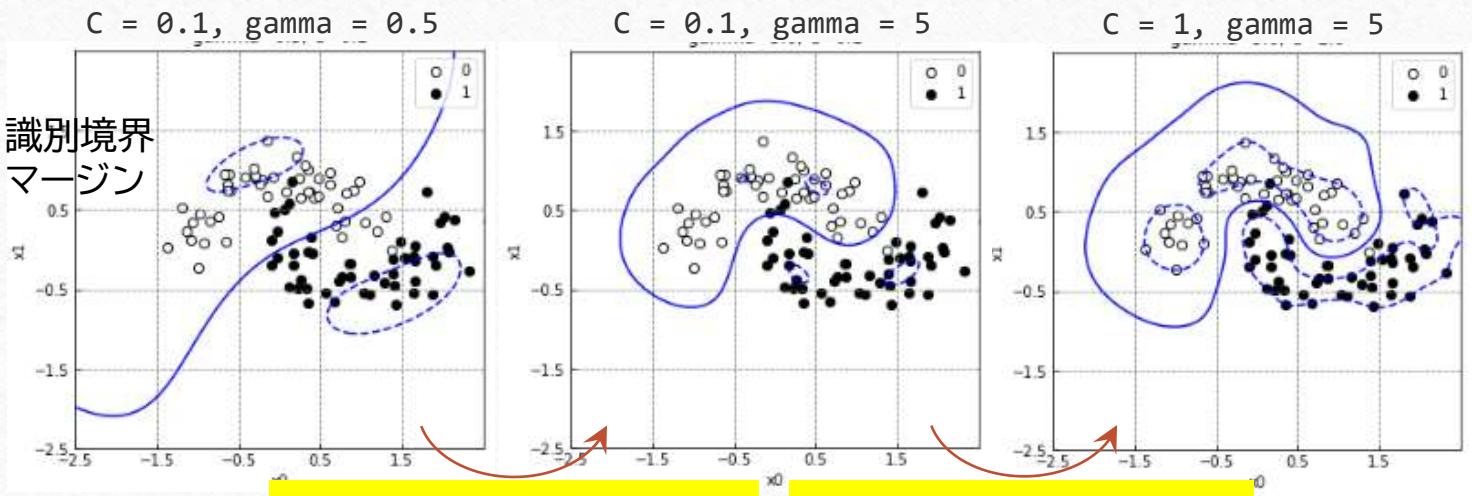
・多次元データを入力して2値分類できる

・・カーネル関数を用いてデータを2次元空間に写像

・ハイパーパラメータ c，γを与えて識別器を得る

・・学習は最も精度が高いc，γを見つける作業

コストパラメータ: C　　誤分類をどの程度許容するか
カーネルのパラメータ: γ　境界線の複雑さ



実線:識別境界
破線:マージン

γ増大:境界線が複雑に

C増大:マージン大きく

Scikit-learn がもつ
月形の２値データセットの場合

```
from sklearn.datasets import make_moons
```

・得られた識別器に未知の文書ベクトルを流すと分類結果が出力される → 照合

ニューラルネットワーク

学習

学習させるデータ(x)

```
[[0.45330593 0.53856367 0.54323846 ...
0.6245525  0.54924446 0.34767133]
 [0.29302627 0.54410183 0.4921428  ...
0.5259048  0.4706858  0.65297157]
 [0.37936592 0.47303295 0.5605437  ...
0.4001513  0.42283633 0.515934  ]
 ...
  [0.51291215 0.5794438  0.570155   ...
0.65197116 0.48016682 0.5000225 ]
 [0.5576463  0.54231286 0.43099108 ...
0.5447404  0.45836136 0.575232  ]]
```

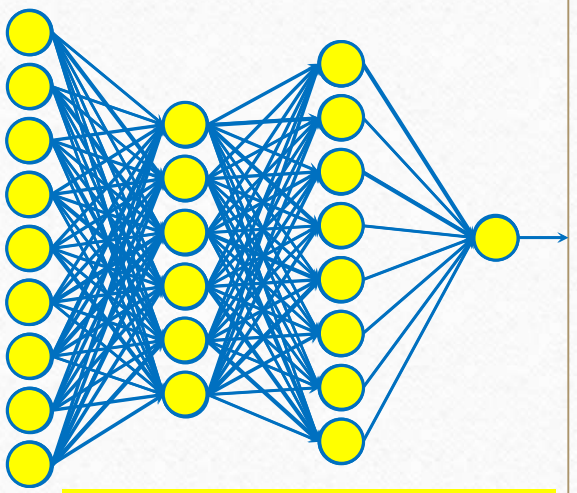
5835個

文書ベクトル

300次元

ニューラルネットワーク

学習モデル $y = f(x)$



300個

1個

5835文書 × 設定回数

クラスラベル(y)

```
[[0.]
 [1.]
 [1.]
 ...
 [0.]
 [0.]]
```

ニューラルネットワーク

学習

学習させるデータ(x)

```
[[0.45330593 0.53856367 0.54323846 ...
0.6245525  0.54924446 0.34767133]
 [0.29302627 0.54410183 0.4921428  ...
0.5259048  0.4706858  0.65297157]
 [0.37936592 0.47303295 0.5605437  ...
0.4001513  0.42283633 0.515934  ]
 ...
  [0.51291215 0.5794438  0.570155   ...
0.65197116 0.48016682 0.5000225 ]
 [0.5576463  0.54231286 0.43099108 ...
0.5447404  0.45836136 0.575232  ]]
```

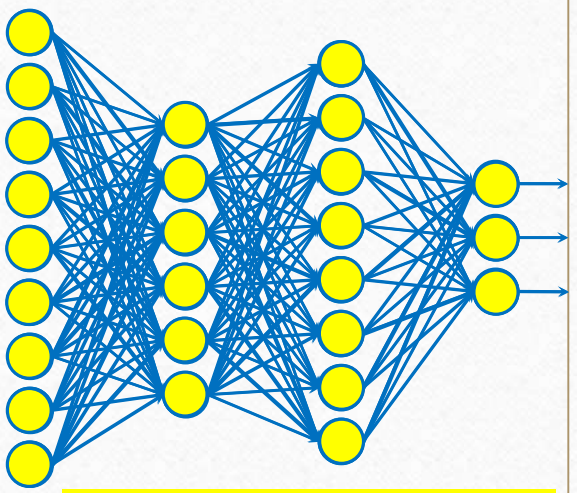
5835個

文書ベクトル

300次元

ニューラルネットワーク

学習モデル $y = f(x)$



300個

3個

5835文書 × 設定回数

クラスラベル(y)

```
[[1. 0. 0.]
 [0. 0. 1.]
 [0. 1. 0.]
 ...
 [1. 0. 0.]
 [0. 1. 0.]]
```

形態素解析（分かち書き）

文書ベクトル化最初の手順 ← 日本語、中国語は非分かち書き言語

MeCab

・辞書に従って分かち書き　係り受け解析もできる

・・辞書にない単語は未知語として切り出される

・・同梱されているIPA辞書

・・ユーザー辞書のNEologd辞書（Web上から得た新語）

・・J-GLOBAL MeSH辞書（J-GLOBAL科学技術用語）

Sentencepiece

・与えられた文字列のなかで頻度多い繰り返し単位を切り出す

・単語数を指定できる

・・解析に用いたIJに関する文書で作られたモデル

・・日本版Wikipediaで作られたモデル

・未知語がなくなる代わりに文法は無視される

形態素解析（分かち書き）

例文

有機溶剤、顔料、及び塩化ビニル－酢酸ビニル共重合体樹脂を含有する非水性インクジェットインキであって、前記塩化ビニル－酢酸ビニル共重合体樹脂の重量平均絶対分子量（｜Mw｜）が20，000～40，000であり、

MeCab（同梱されているIPA辞書）

有機溶剤 、 顔料 、 及び 塩化ビニル － 酢酸ビニル 共 重合体 樹脂 を 含有 する 非 水性 インクジェット インキ で あっ て 、 前記 塩化ビニル － 酢酸ビニル 共 重合体 樹脂 の 重量 平均 絶対 分子量 （ ｜ Ｍｗ ｜ ） が ２ ０ ， ０ ０ ０ ～ ４ ０ ， ０ ０ ０ で あり 、

MeCab（J-GLOBAL MeSH辞書）

技術用語の粒度が小さい

有機 溶剤 、 顔料 、 及び 塩化 ビニル － 酢酸 ビニル 共 重合 体 樹脂 を 含有 する 非 水性 インクジェット インキ で あっ て 、 前記 塩化 ビニル － 酢酸 ビニル 共 重合 体 樹脂 の 重量 平均 絶対 分子 量 （ ｜ Ｍｗ ｜ ） が ２ ０ ， ０ ０ ０ ～ ４ ０ ， ０ ０ ０ で あり 、

形態素解析(分かち書き)

例文

有機溶剤、顔料、及び塩化ビニル－酢酸ビニル共重合体樹脂を含有する非水性インクジェットインキであって、前記塩化ビニル－酢酸ビニル共重合体樹脂の重量平均絶対分子量(｜Mw｜)が20，000～40，000であり、

MeCab(同梱されているIPA辞書)

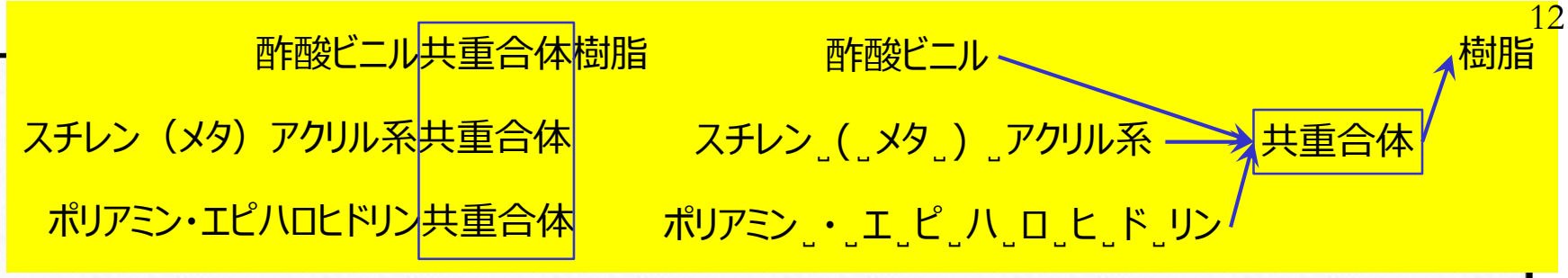
有機溶剤 、 顔料 、 及び 塩化ビニル － 酢酸ビニル 共 重合体 樹脂 を 含有 する 非 水性 インクジェット インキ で あっ て 、 前記 塩化ビニル － 酢酸ビニル 共 重合体 樹脂 の 重量 平均 絶対 分子量 （ ｜ Ｍｗ ｜ ） が ２ ０ ， ０ ０ ０ ～ ４ ０ ， ０ ０ ０ で あり 、

Sentencepiece(解析に用いたIJに関する文書で作られたモデル)

先頭・空白に_、粒度が大きい

_有機溶剤 、 顔料 、 及び塩化ビニル - 酢酸ビニル共重合体樹脂 を含有する 非水性インクジェットインキ であって 、 _前記塩化ビニル - 酢酸ビニル共重合体樹脂 の 重量平均絶対分子量 (｜ Mw ｜) が ２ 0,000～40,000 であり 、

形態素解析（分かち書き）



例文

有機溶剤、顔料、及び塩化ビニル－酢酸ビニル共重合体樹脂を含有する非水性インクジェットインキであって、前記塩化ビニル－酢酸ビニル共重合体樹脂の重量平均絶対分子量（｜Mw｜）が20，000～40，000であり、

Sentencepiece（解析に用いたIJに関する文書で作られたモデル）　粒度が大←語彙数多くとれる

▁有機溶剤_、_顔料_、_及び塩化ビニル_-_酢酸ビニル共重合体樹脂_を含有する_非水性インクジェットインキ_であって_、_▁前記塩化ビニル_-_酢酸ビニル共重合体樹脂_の_重量平均絶対分子量_(|_Mw_|)_が_2_0,000～40,000_であり_、

Sentencepiece（単語数を32000から2000に減らした例）　粒度が小さくなる

▁_有機溶剤_、_顔料_、_及び_塩化ビニル_-_酢酸ビニル_共重合体_樹脂_を含有する_非_水性_インクジェット_インキ_であって_、_▁前記_塩化ビニル_-_酢酸ビニル_共重合体_樹脂_の_重量_平均_絶_対_分子量_(_|_M_w_|_)_が_20_,000_～40_,000_であり_、

形態素解析（分かち書き）

例文

有機溶剤、顔料、及び塩化ビニル－酢酸ビニル共重合体樹脂を含有する非水性インクジェットインキであって、前記塩化ビニル－酢酸ビニル共重合体樹脂の重量平均絶対分子量（｜Mw｜）が20，000～40，000であり、

Sentencepiece（解析に用いたIJに関する文書、単語数8000の例）

▁ 有機溶剤 、 顔料 、 及び 塩化ビニル - 酢酸ビニル共重合体 樹脂を含有する 非水性インクジェットインキ であって 、 ▁前記 塩化ビニル - 酢酸ビニル共重合体 樹脂 の 重量 平均 絶 対 分子量 （ ｜ Mw ｜ ） が 2 0,000 ～40 ,000 であり 、

単語数が同じでも学習させる文書によって粒度が全く異なる

Sentencepiece（wikipedia日本語版、単語数8000の例）

▁ 有機 溶 剤 、 顔料 、 及び 塩化 ビ ニル - 酢酸 ビ ニル 共 重 合体 樹脂 を 含有 する 非 水 性 インク ジェット イン キ であって 、 ▁前 記 塩化 ビ ニル - 酢酸 ビ ニル 共 重 合体 樹脂 の 重量 平均 絶対 分子 量 （ ｜ M w ｜ ） が 20,000 ～ 40 , 000 であり 、

形態素解析（分かち書き）

粒度の確認

形態素解析器	辞書/モデル文書/語彙数設定		粒度	最大文書長	単語数	粒度
MeCab	IPA	–	中	15755	18806	塩化ビニル␣–␣酢酸ビニル␣共␣重合体␣樹脂
	MeSH	–	小	16322	17387	塩化␣ビニル␣–␣酢酸␣ビニル␣共␣重合␣体␣樹脂
Sentencepiece	特許文書	2000	中	20496	2681	塩化ビニル␣-␣酢酸ビニル␣共重合体␣樹脂
		8000	大	17443	8665	塩化ビニル␣-␣酢酸ビニル共重合体␣樹脂
		32000	大	16380	32609	塩化ビニル␣-␣酢酸ビニル共重合体樹脂
	wiki	8000	小	16677	9836	塩化␣ビ␣ニル␣-␣酢酸␣ビ␣ニル␣共␣重␣合体␣樹脂

文書長、単語数は計算量に影響する

MeCabの場合、Mesh辞書を使い単語粒度を小さくする

sentencepieceの場合、単語数8000に設定すると
計算効率が高くなりそう

文書ベクトル（文字列を数値に変換）

TF-IDF（文書に含まれる単語の重要度を示す指標）

1文書内での頻度情報（tf(t,d)）と全文書に対する単語のレア度（idf(t)）を掛け合わせたもの

$$\text{tf-idf}(t,d) = \text{tf}(t,d) \times \text{idf}(t).$$

$$\text{idf}(t) = \log \frac{1+n}{1+\text{df}(t)} + 1.$$

分母に1を足しているのは0割り算を防ぐため

tf(t,d) 単語[*t*]の出現数
idf(t) 単語[*t*]のレア度

n 文書セット内の文書の総数
df(*t*) 単語[*t*]を含む文書の数

・scikit-learnのTfidfVectorizerで同時計算

頻度情報

最小値 0.
最大値 1820.
（最大頻度）

```
[[ 3.  0.   0.  ...  0.  0.  0.]
 [ 2.  18.  44. ...  0.  0.  0.]
 [ 0.  0.   14. ...  0.  0.  0.]
 ...
 [ 0.  0.   0.  ...  0.  0.  1.]
 [ 2.  0.   0.  ...  0.  0.  0.]
 [ 3.  0.   0.  ...  0.  0.  0.]]
```

単語数（18806次元）

TF-IDF

最小値 0.
最大値 48.54

```
[[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. 0.0231.]
 [0.0683 0.     0.     ...  0. 0. 0. ]
 [0.0913 0.     0.     ...  0. 0. 0. ]]
```

単語数（18806次元）

文書数
（7294次元）

文書ベクトルの数字同士に力関係がある・・・SVMやニューラルネットワークに直接入力可

文書ベクトル（文字列を数値に変換）

単語ID（Tokenizer）

形態素解析器で分割された単語に一意のIDを振る

・未知語に0を割り当てる

```
from keras.preprocessing.text import Tokenizer
```

単語ID

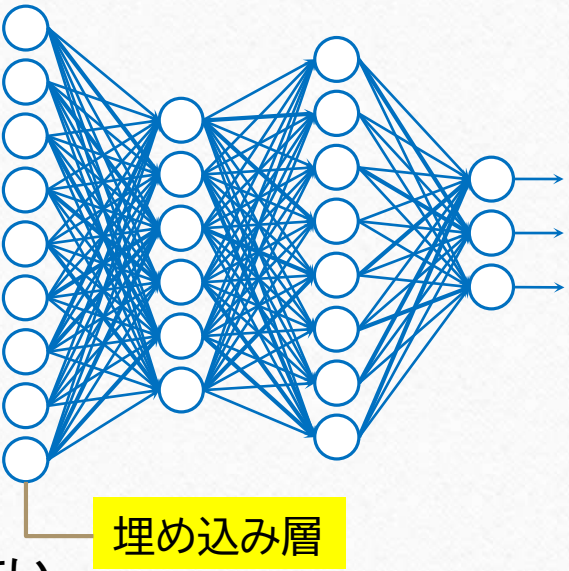
```
[[2645  3    172938 ...  0   0    0]
 [4     5843 23     ...  0   0    0]
 [7     21   1      ...  0   0    0]
 ...
 [26821 54   83343  ...  352 5447 63]
 [83343 115  13     ...  0   0    0]
 [7     21   75     ...  0   0    0]]
```

文書数（7294次元）

文書最大長（15755次元）

最小値 0
最大値 18805
（単語数18806）

文書ベクトルの数字同士に力関係がない



・ニューラルネットワークの入力層に埋め込み（Embed）層を用いて、特定IDの単語が入力されたことを知らせる

・頻度順に単語IDを割り振ると低頻度のもの、高頻度のものをカットしやすい

文書ベクトル（文字列を数値に変換）

入力文書の長さはまちまち

・単語が存在しないところに0（未知語）を割り当てる ・・・ zero padding

最小値 0.
最大値 1820.
（最大頻度）

頻度情報

```
[[ 3.  0.   0. ...  0.  0.  0.]
 [ 2.  18.  44. ...  0.  0.  0.]
 [ 0.  0.   14. ...  0.  0.  0.]
 ...
 [ 0.  0.   0.  ...  0.  0.  1.]
 [ 2.  0.   0.  ...  0.  0.  0.]
 [ 3.  0.   0.  ...  0.  0.  0.]]
```

単語数（18806次元）

単語ID

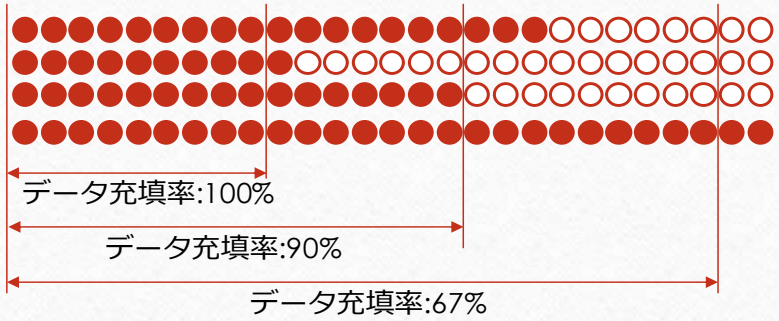
```
[[2645  3    172938 ...  0   0    0]
 [4     5843 23     ...  0   0    0]
 [7     21   1      ...  0   0    0]
 ...
 [26821 54   83343  ...  352 5447 63]
 [83343 115  13     ...  0   0    0]
 [7     21   75     ...  0   0    0]]
```

文書最大長（15755次元）

文書数
（7294次元）

文書ベクトル（文字列を数値に変換）

入力文書の長さはまちまち



・最大文書長を有限の値で切り落とすこともできる

最小値 0
最大値 18805
（単語数）

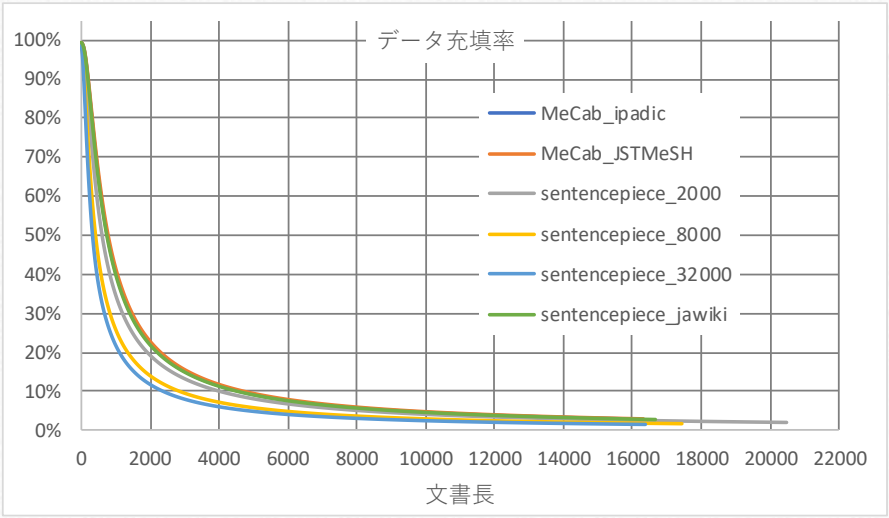
```
[[ 0.  32.  22. ...  0.  0.  0.]
 [ 0.  18.  30. ...  0.  0.  0.]
 [ 0.  25.  44. ...  0.  0.  0.]
 ...
 [ 0.  2.   3.  ...  0.  0.  1.]
 [ 0.  7.   8.  ...  0.  0.  0.]
 [ 0.  6.   7.  ...  0.  0.  0.]]
```

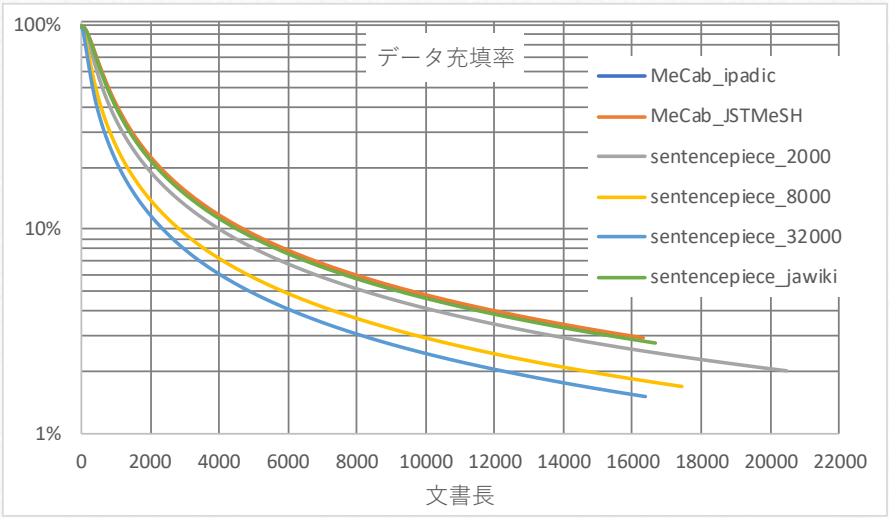
後方切り（2000次元）

文書数
（7294次元）

・後方切りによってデータ充填率が上がる（計算量に影響）

・後方切り甚だしいと特徴語が切り捨てられる？（精度？）





文書ベクトル（文字列を数値に変換）

Doc2Vec

word2vec（Googleの研究者らによって提唱されたモデル）

・CBOW ・・・周辺の語から中央の語を予測（語順は順不同）

特許	?	範囲	酸化	?	剤
	請求			防止	

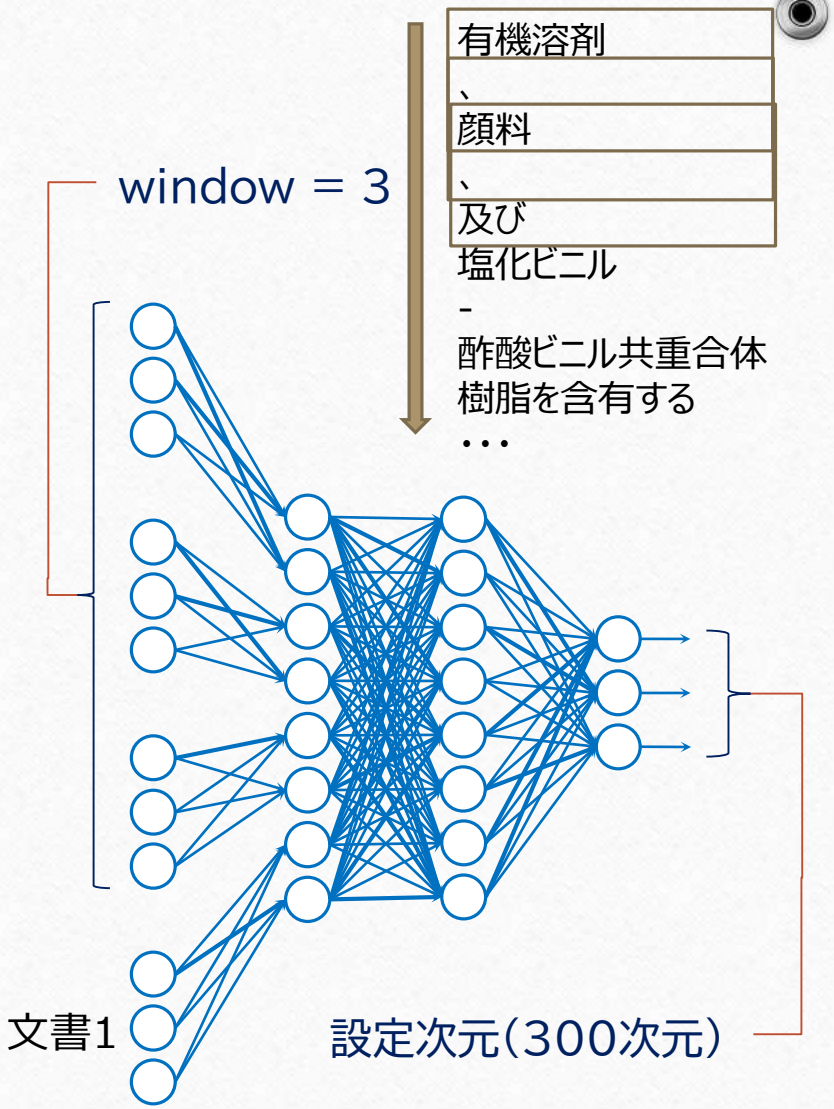
・Skip-gram ・・・中心語が与えられ周囲の語を予測

?	請求	?	?	防止	?
特許		範囲	酸化		剤

・Doc2Vecでは文書IDを組み合わせて文書ベクトルを得る

文書内からランダムに選択された単語を予測する→ DBOW

中心の単語を予測する→ PV-DM



文書ベクトル（文字列を数値に変換）

TF-IDF

最小値 0.
最大値 48.54

```
[[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. ]]
```

Doc2Vec

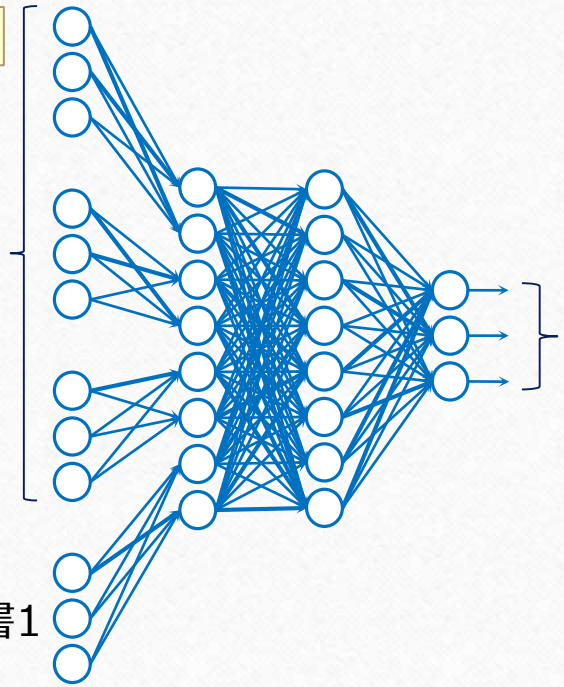
word2vec（Googleの研究者らによって提唱されたモデル）

- 空白で区切った文章を入力
- 設定次元数の文書ベクトルが得られる

```
from gensim.models import Doc2Vec
```

有機溶剤 、顔料 、及び 塩化ビニル - 酢酸ビニル共重合体 樹脂を含有する 非水性インクジェットインキ であって 、 ・・・

空白で区切った文書



ユニークな文書番号　文書1

Doc2Vec

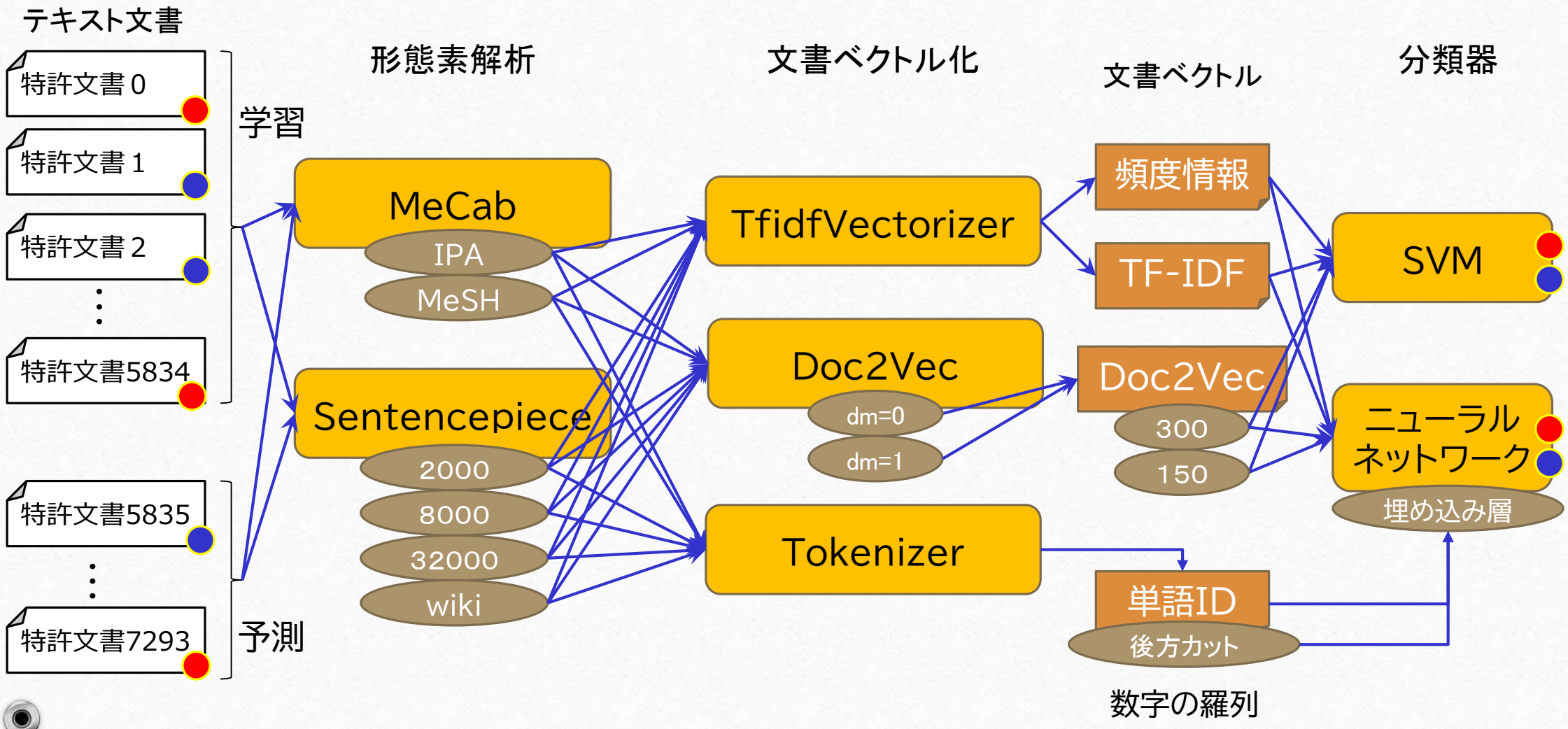
最小値 -2.5768.
最大値 2.7639.

```
[[-0.081  -0.0347  0.2025 ...
0.0191 -0.1911  -0.0548]
 [-0.0265  0.0431 -0.1008 ...
0.0793  0.1649  -0.0327]
...
 [ 0.0005  0.0557 -0.119  ...
0.1102  0.0743   0.0545]
 [-0.0741  0.2215 -0.077  ...
0.1183  0.0207  -0.0027]]
```

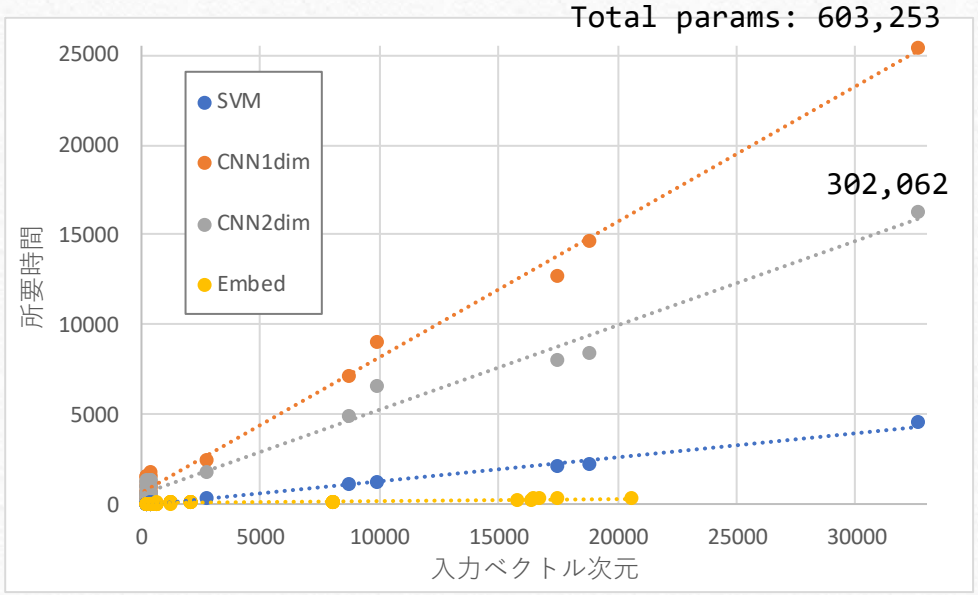
文書数
（7294次元）

設定次元（300次元）

文書ベクトル化と分類処理の流れ



所要時間

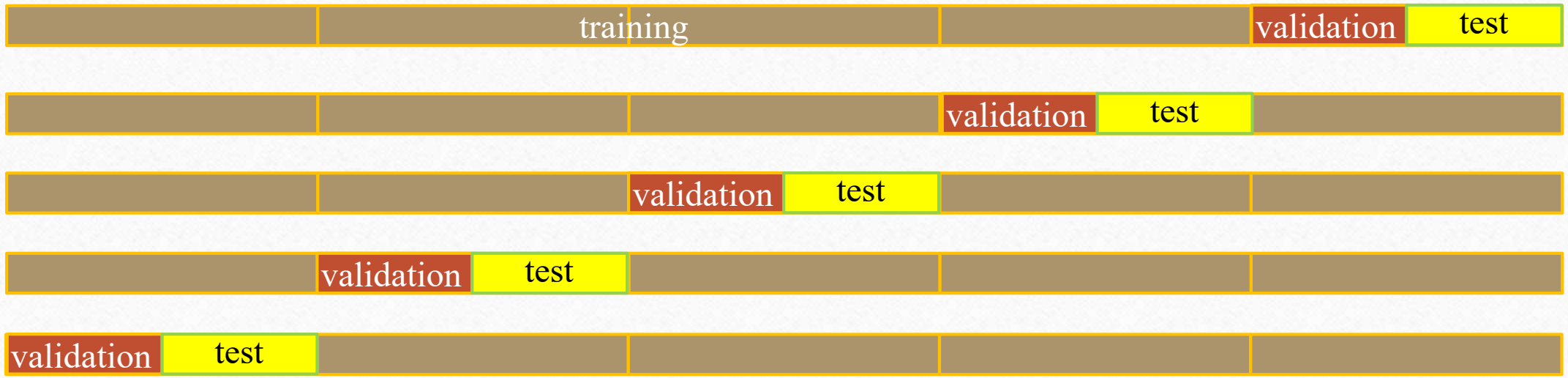


同じ層構成なら、入力次元にほぼ比例

層(タイプ)	出力次元	パラメータ数	出力次元	パラメータ数
reshape	150, 1	0	8000, 1	0
dropout1d	150, 1	0	8000, 1	0
Conv1D	150, 64	256	8000, 64	256
MaxPooling1D	75, 64	0	4000, 64	0
dropout1d	75, 64	0	4000, 64	0
batch_normalization	75, 64	256	4000, 64	256
Conv1D	75, 4	772	4000, 4	772
batch_normalization	75, 4	16	4000, 4	16
Flatten	300	0	16000	0
Dense	16	4816	16	256016
Dense	8	136	8	136
Dense	1	9	1	9
	Total params:	6,261		257,461

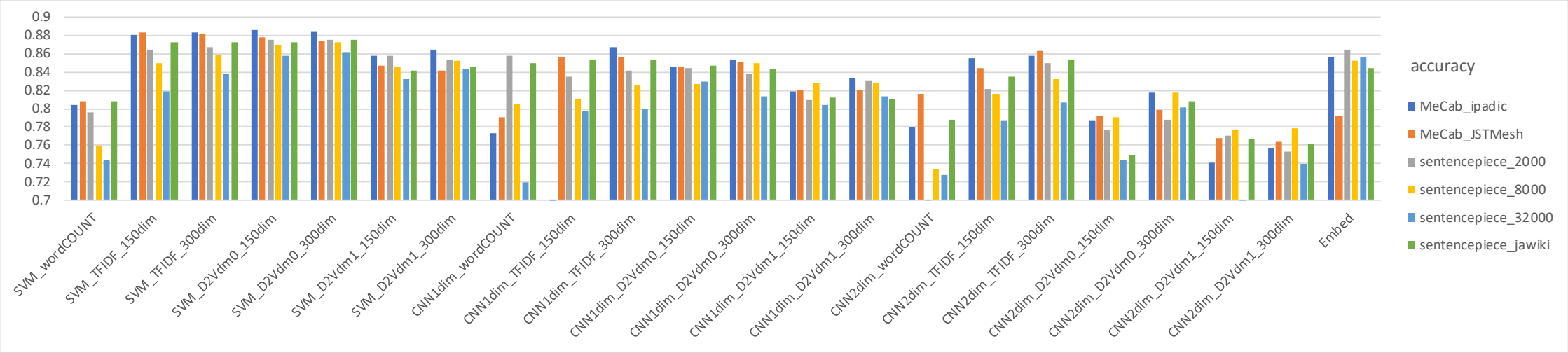
評価方法

変形5-hold 法



学習時：training と validation を使用
予測時：test を使用
　検証用データとテストデータとを分離
精度算出には5回の平均値を使用

評価方法



ラベル ＼ 予測値	0 IJでないもの	1 IJであるもの	
0	a（正答）	b	再現率
1	c	d（正答）	
	適合度		

Label:0/1=1/2.52

正確度(accuracy)

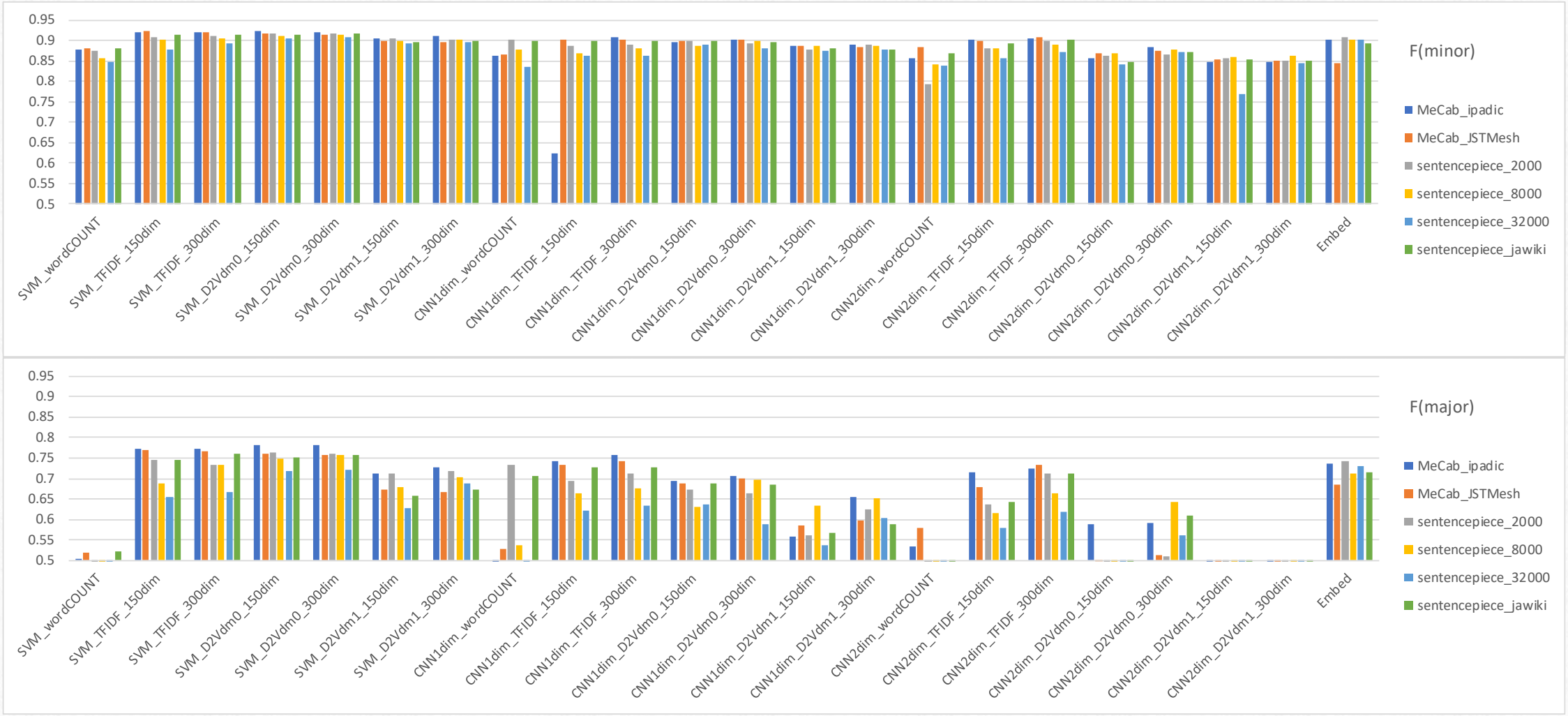
$$\frac{a+d}{a+b+c+d}$$

F値(F value) 再現率と適合度の調和平均

$$\frac{2a}{2a+b+c} \quad \text{(minor)}$$

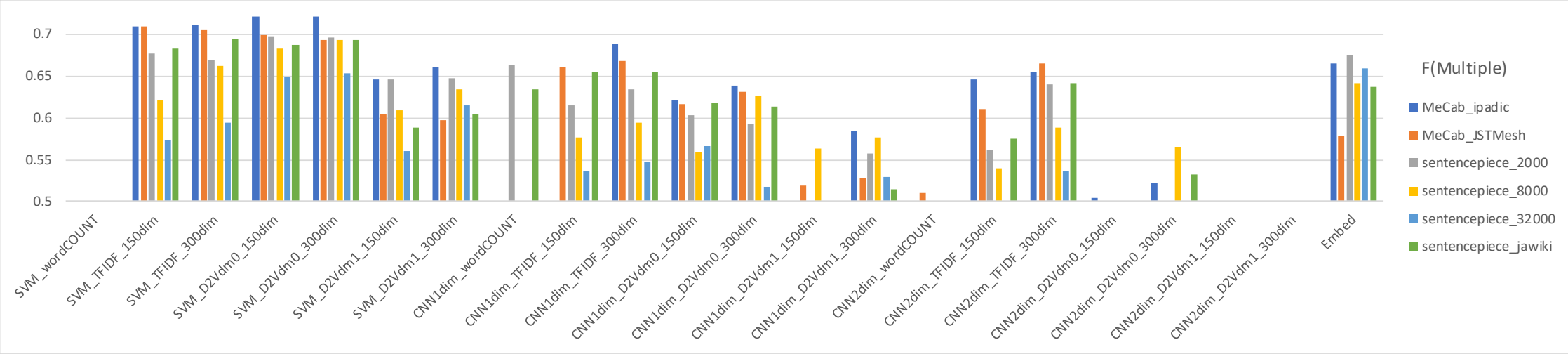
$$\text{(major)} \quad \frac{2d}{2d+b+c}$$

評価方法



F値(major)を用いると、データ数に偏りがある場合の精度が目に見えやすくなる

評価方法

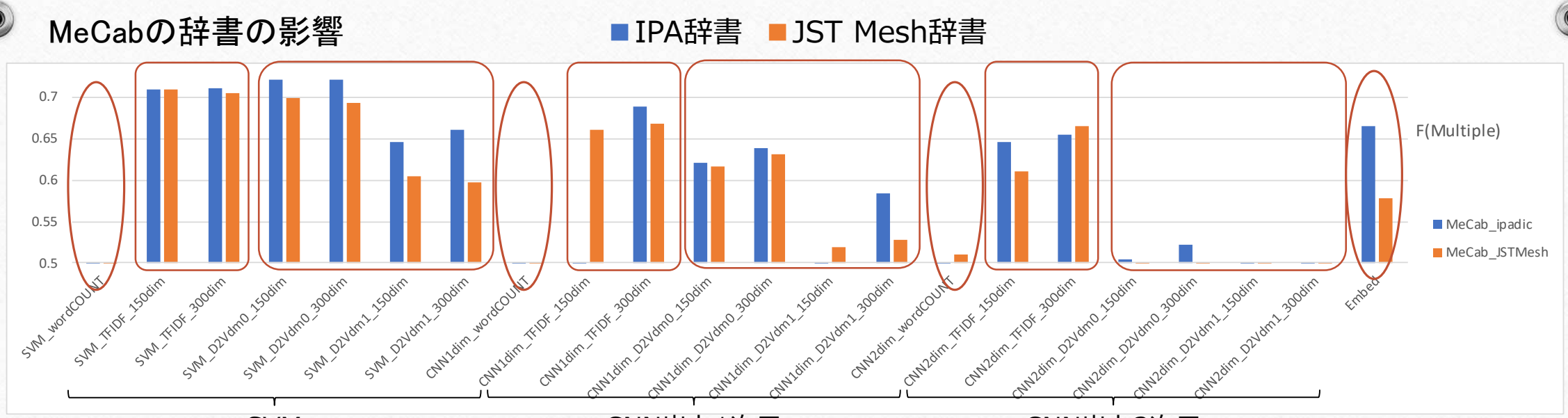


F(Multiple)

$$\frac{2a}{(2a+b+c)} \times \frac{2\mathrm{d}}{(b+c+d)}$$

F値(minor)× F値(major)で評価

MeCabの辞書の影響



頻度情報

```
[[ 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.]]
```

MeCabでは精度出ない

TF-IDF

```
[[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.]]
```

高精度

Doc2Vec

```
[[-0.081  -0.0347 0.2025  ...
  0.0191  -0.1911 -0.0548]
 [-0.0265 0.0431  -0.1008  ...
  0.0793  0.1649  -0.0327]
...
 [0.0005  0.0557  -0.119 ...
  0.1102  0.0743  0.0545]
 [-0.0741 0.2215  -0.077 ...
  0.1183  0.0207  -0.0027]]
```

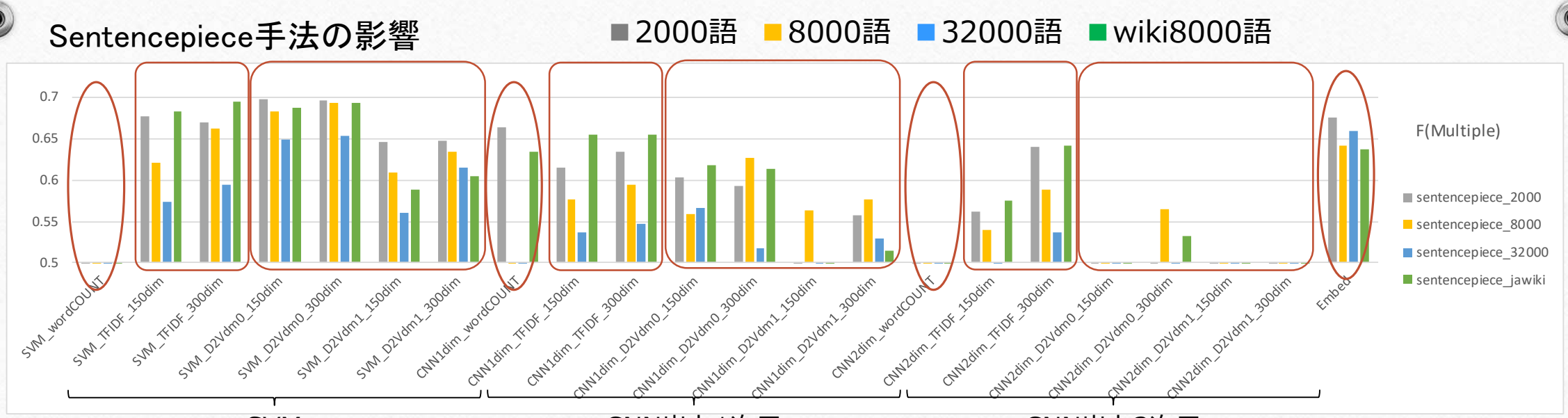
DBOWでSVMと相性良い

単語ID

```
[[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]]
```

IPA辞書で高精度

Sentencepiece手法の影響



頻度情報

```
[[ 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.]]
```

CNNで特異的に良い

TF-IDF

```
[[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.]]
```

語彙数少または
wikiとの相性が良い

Doc2Vec

```
[[-0.081  -0.0347 0.2025  ...
  0.0191  -0.1911 -0.0548]
 [-0.0265 0.0431  -0.1008 ...
  0.0793  0.1649  -0.0327]
 ...
 [0.0005  0.0557  -0.119 ...
  0.1102  0.0743  0.0545]
 [-0.0741 0.2215  -0.077 ...
  0.1183  0.0207  -0.0027]]
```

DBOWでSVMと相性良い

単語ID

```
[[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]]
```

高精度

高精度ランキング

ランキング20位まで/132通り

- Doc2VecとMeCabとの組み合わせ
- TF-IDFとMeCabとの組み合わせ
- Doc2VecとSentencepieceとの組み合わせ
- TF-IDFとSentencepieceとの組み合わせ
- 単語ID（Embed）とSentencepiece（2000）との組み合わせ

・MeCabはIPA辞書がJST Mesh辞書より精度高い

・Sentencepieceは
Doc2Vecに用いるとき、
特許文書2000次元 ＞ 特許文書8000次元 ＞ wikipedia8000次元
TF-IDFに用いるとき、
wikipedia8000次元 ＞ 特許文書2000次元
（順位逆転）

1	D2Vdm0_150dim_MeCab_ipadic
2	D2Vdm0_300dim_MeCab_ipadic
3	TFIDF_300dim_MeCab_ipadic
4	TFIDF_150dim_MeCab_JSTMeSH
5	TFIDF_150dim_MeCab_ipadic
6	TFIDF_300dim_MeCab_JSTMeSH
7	D2Vdm0_150dim_MeCab_JSTMeSH
8	D2Vdm0_150dim_sentencepiece_2000
9	D2Vdm0_300dim_sentencepiece_2000
10	TFIDF_300dim_sentencepiece_jawiki
11	D2Vdm0_300dim_sentencepiece_8000
12	D2Vdm0_300dim_MeCab_JSTMeSH
13	D2Vdm0_300dim_sentencepiece_jawiki
14	TFIDF_300dim_MeCab_ipadic
15	D2Vdm0_150dim_sentencepiece_jawiki
16	D2Vdm0_150dim_sentencepiece_8000
17	TFIDF_150dim_sentencepiece_jawiki
18	TFIDF_150dim_sentencepiece_2000
19	Embed_sentencepiece_2000
20	TFIDF_300dim_sentencepiece_2000

文書ベクトル長と精度

入力文書の長さはまちまち

・最大文書長を有限の値で切り落とすこともできる

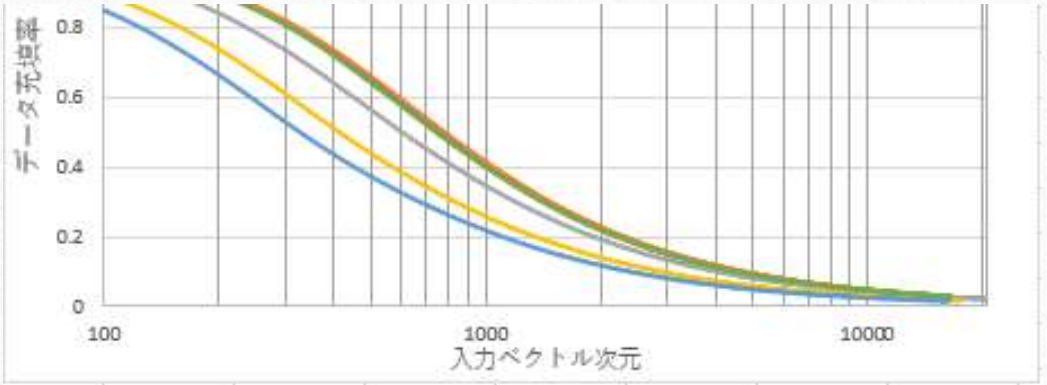
最小値 0
最大値 18805
(単語数)

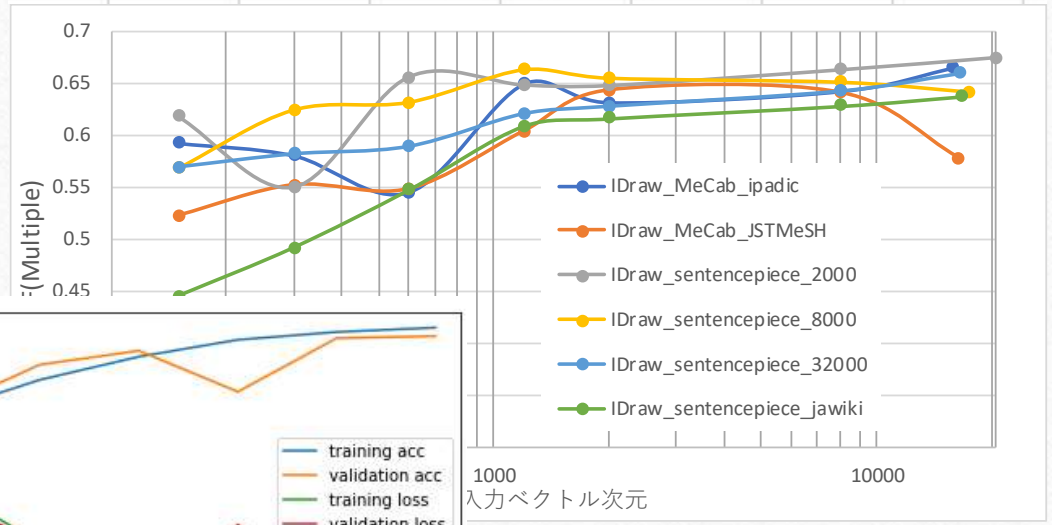
```
[[ 0.  32.  22. ...  0.  0.  0.]
 [ 0.  18.  30. ...  0.  0.  0.]
 [ 0.  25.  44. ...  0.  0.  0.]
 ...
 [ 0.  2.   3.  ...  0.  0.  1.]
 [ 0.  7.   8.  ...  0.  0.  0.]
 [ 0.  6.   7.  ...  0.  0.  0.]]
```

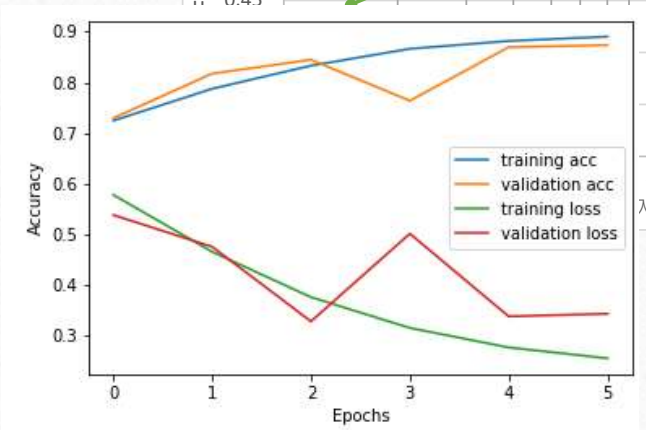
後方切り(次元設定)

先頭から1200次元まででカットしても精度はさほど変わらない(データ充填率20～40%)

それより短くすると精度の低下がみられる







まとめ

生文を2つのツール6パターンで形態素解析した

単語ID、頻度情報、TF-IDF、Doc2Vecによる分散表現を作り、SVMおよびニューラルネットワーク（CNN）に入力した

F値(minor)× F値(major)を用いるとモデルの良し悪しが明瞭になった
・予測結果の正誤は正確度(accuracy)でよいが、偶然も考えられるため

MeCabは、同梱されているIPA辞書を用いると精度が高いので今後も主流であり続ける
・分類タスクに対してJ-GLOBAL MeSH辞書は有用でなかった

Sentencepieceは、語彙数を少なくしたり、分析対象外の文章(Wikipedia)で学習したモデルを用いたりしたほうが精度が良い （粒度大きいとあまりよくない）
・頻度情報はSentencepieceとの組み合わせで有用になる

単語IDを入力する場合、先頭から1200語程度を使えば十分な精度が出る

同一モデルにおいて、文書ベクトル長と処理時間との間にはほぼ比例関係がある

文書ベクトルの作成方法 と 学習モデル との異なる組み合わせを用いて多数決モデルを作ることによって精度向上が期待できる

謝辞

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。

共同発表者の安藤さんをはじめ、同研究会のメンバーに深謝の意を表する。

アジア特許情報研究会　https://sapi.kaisei1992.com/

CNN 出力1層モデル

層(タイプ)	出力次元	パラメータ数	出力次元	パラメータ数
reshape	150, 1	0	8000, 1	0
dropout1d	150, 1	0	8000, 1	0
Conv1D	150, 64	256	8000, 64	256
MaxPooling1D	75, 64	0	4000, 64	0
dropout1d	75, 64	0	4000, 64	0
batch_normalization	75, 64	256	4000, 64	256
Conv1D	75, 4	772	4000, 4	772
batch_normalization	75, 4	16	4000, 4	16
Flatten	300	0	16000	0
Dense	16	4816	16	256016
Dense	8	136	8	136
Dense	1	9	1	9
	Total params: 6,261		Total params: 257,461	

CNN 出力2層モデル

層(タイプ)	出力次元	パラメータ数	出力次元	パラメータ数
reshape	150, 1	0	8000, 1	0
Conv1D	150, 64	256	8000, 64	256
MaxPooling1D	75, 64	0	4000, 64	0
dropout1d	75, 64	0	4000, 64	0
Conv1D	75, 4	772	4000, 4	772
MaxPooling1D	37, 4	0	2000, 4	0
Flatten	148	0	8000	0
Dense	16	2384	16	128016
Dense	8	136	8	136
Dense	2	18	2	18
	Total params: 3,566		Total params: 129,198	

Embedモデル

層(タイプ)	出力次元	パラメータ数	出力次元	パラメータ数
Embedding	150, 1	8665	8000, 1	8665
dropout1d	150, 1	0	8000, 1	0
MaxPooling1D	50, 1	0	2661, 1	0
Flatten	50	0	2666	0
Dense	256	13056	256	682752
Reshape	256, 1	0	256, 1	0
Conv1D	256, 32	128	256, 32	128
MaxPooling1D	85, 32	0	85, 32	0
Flatten	2720	0	2720	0
Dense	256	696576	256	696576
Dense	16	4112	16	4112
Dense	1	17	1	17
	Total params: 722,554		Total params: 1,392,250	

文書ベクトル（文字列を数値に変換）

高頻度・低頻度単語

・出現頻度1から2の単語（種類）が全体の60%

・・これをカットすると語彙数が減るため計算時間短縮

出現頻度	単語数	割合	頻度	割合
1	32914	42.5%	32555	3.1%
2	14477	18.7%	28663	2.8%
3～10	21112	27.3%	104682	10.0%
11～20	4060	5.2%	58344	5.6%
21～30	1492	1.9%	31292	3.0%
31～100	2334	3.0%	128915	12.4%
101～6000	1009	1.3%	458867	44.0%
6001～	15	0.0%	198588	19.1%

・出現頻度上位15位までの単語はインクジェットの特徴と無関係

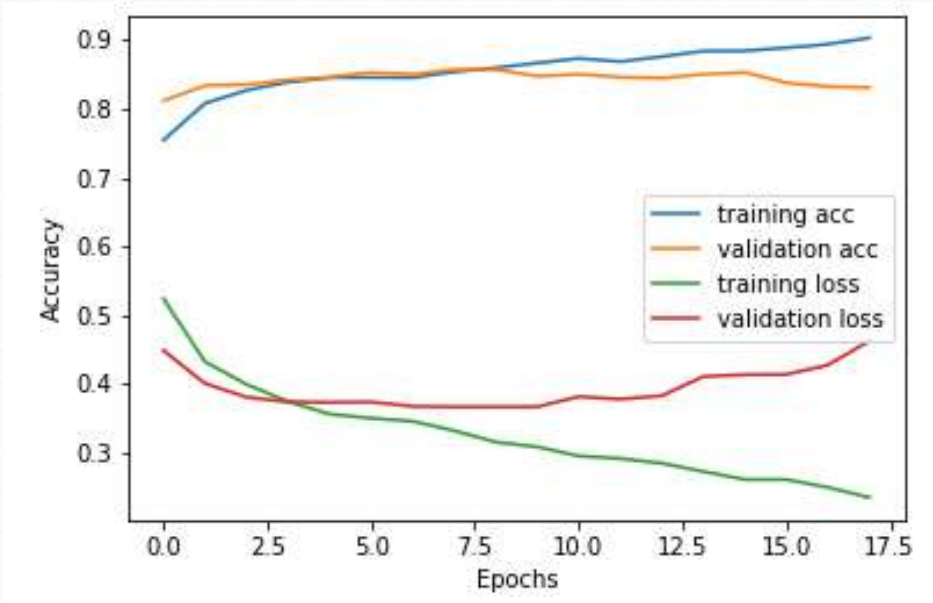
・・データの19%を占める

・・これをカットするとデータ数が減るため計算時間短縮
　かつ精度向上が期待できる

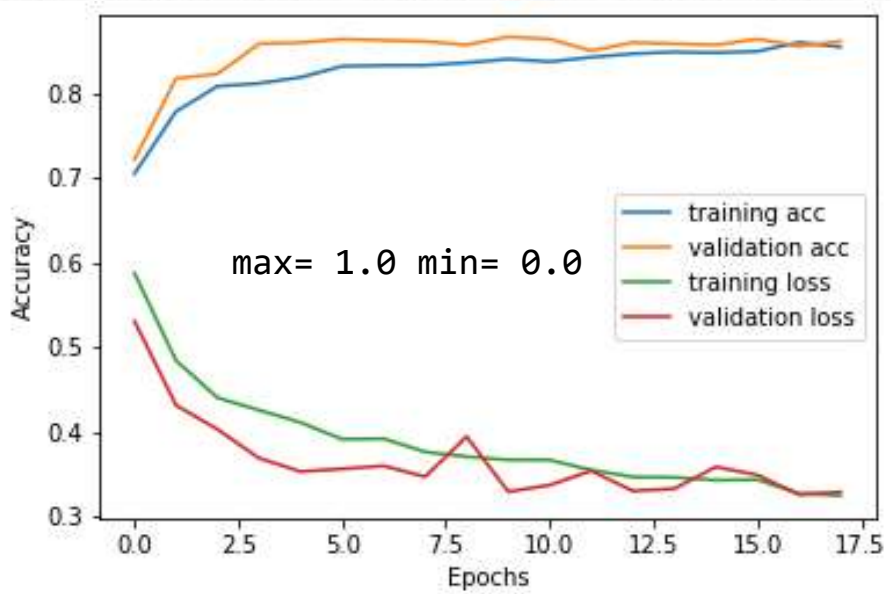
1	こと	31546
2	する	30074
3	特徴	28091
4	ある	25263
5	有する	11740
6	範囲	9917
7	含む	9376
8	なる	9156
9	2	8642
10	1	8316
11	含有する	7885
12	製造方法	7456
13	R	7232
14	あり	6421
15	3	6371
16	インク受容層	5411

規格化

max= 2.605609 min= -2.5078473



max= 1.0 min= 0.0



max= 1.0 min= -1.0