特集《人工知能》

オープンソースソフトウェアを用いた人工知能による特許自動分類

会員　小川　延浩
国立情報学研究所　教授・理学博士　宇野　毅明

要　約

人工知能を実現するアプローチとして最近注目を浴びているのが機械学習である。メディアでもよく目にするディープラーニング（または深層学習）は機械学習の一手法である。この機械学習という技術は，最近ではオープンソースソフトウェア（以下 OSS とする）やクラウドサービスの普及によって高度な専門知識がなくても使える身近な存在になりつつある。機械学習を用いることで，例えば特許文献の自動分類（特許分類記号の自動付与）なども比較的簡単に行う事ができる。本稿では，自動分類を行うための技術を説明するとともに，その一手法について実現するステップを解説する。パソコンを片手に本稿を読み進める事で，読者が自分でも特許文献の自動分類が体験できるようにすることを目指す。

目次

1．はじめに

2016 年 8 月に次のような記事が WEB 上に掲載され，話題となった[(1)]。

「機械学習を知らない皮膚科医が，皮膚がんの診断 AI を試作できたワケ」

この記事では，皮膚を撮影した写真をアップロードすることで，皮膚がんである確率をコンピュータがアドバイスしてくれる WEB サービスを，AI 技術については素人であった皮膚科専門医が開発したということが紹介されている。この「機械学習」は，人工知能学会の WEB ページには，「観測センサーやその他の手段で収集されたデータの中から一貫性のある規則を見つけだそうとする研究」と説明されている[(2)]。つまり，事前に収集した皮膚がんの画像データを用いて一貫性ある皮膚がん画像の規則を機械学習によって見つけ出し，その規則をもとにコンピュータがアドバイスするシステムが作られたのである。

この開発を可能にさせたのは，Labellio[(3)] という画像認識モデルを作成する WEB サービスの存在であった。最近では，高度な技術であっても，短期間の内にオープンソースソフトウェア等で配布され，一般に利用可能となることも多い。その利用には，ある程度のプログラミング知識が必要となることも多いが，徐々に GUI 付のアプリケーションソフトウェアや，WEB サービスを通して利用できるケースも増えてきており，利用に際してのハードルが格段に下がりつつある。先の事例はこういった流れを端的に表したものと言えるだろう。

このような事例は知財業界でも当然に起こり得ると考えられる。機械学習やディープラーニングは画像処理分野で盛り上がりを見せているため，現在みられる事例の多くは画像処理が絡むものである。しかし，これらの技術は文字情報（テキストデータ）に対しても適用可能である。意匠や商標といった画像が絡むサービスはもちろん，明細書等の文字情報を対象とした

サービスについても，知財専門家がAI技術を用いて開発できる下地は既に整いつつある。

このような背景を踏まえ，本稿では，読者が実際に「機械学習」を体験するための説明を行いたい。AI技術を用いたサービスの一例として明細書の自動分類を取り上げ，読者がパソコンを片手に本稿を読み進める事で，機械学習を用いた簡単な明細書自動分類が体験できるようにすることを目指す。まず，次の2章では，自動分類を行うためにはどのような手法があるかについて説明する。そして，その次の3章では具体的な実現方法について説明する。最後に4章で，まとめを行う。

2．特許文献の自動分類手法

（1） 機械学習によるデータの分類

機械学習を用いて主に行われるのは，クラス分類とクラスタリングというタスクである。クラス分類とは，予めクラス（グループやカテゴリのことであり，例えばIPCのセクション・クラス・サブクラスなどがここで言うクラスに該当する）が決められており，一定のルールに従ってデータを各クラスに分類していく作業のことをいう。一方，クラスタリングとは，はじめにクラスが決められておらず，データ群の中から各データの関係性等を調べることでグループ化していく作業のことをいう。今回のターゲットは，特許文献の自動分類という，予め決められたクラスを各文献に対して付与する作業であるから，クラス分類にあたる。このクラス分類の中には二値分類と多値分類がある。二値分類とは2種類のクラスに分類することであり，例えばスパムメールフィルタのように，ある条件に該当するクラスと該当しないクラスに分類する作業である。多値分類とは，3種類以上のクラスに分類することであり，特許分類記号の付与はこの多値分類にあたる。

次に，クラス分類の手法について説明する。機械学習を用いてクラス分類を行う場合，まず教師データと呼ばれる予め用意したデータ群から分類ルールを作成する。そしてその作成した分類ルールを用いてターゲットとするデータ（テストデータと呼ぶ）の自動分類を行うことになる。分類ルールの作成手法には，決定木，サポートベクタマシン，ナイーブベイズなど複数の手法が存在する。ここでは，一例として「決定木」について説明する。その他の手法については，別途参考文献を示す[4]。

決定木は，条件分岐を繰り返すことで分類すべきクラスを決定する手法である。生成された分類ルールが，他の手法と比較して人間に理解しやすいという特徴がある。例えば，次のような教師データが与えられた場合に，どのような分類ルールが生成されるかを考える。

商品ID	商品種別	価格(円)	クラス
1	野菜	500	高額商品
2	野菜	300	低額商品
3	魚介	1000	高額商品
4	魚介	800	低額商品

表1　教師データの一例

この教師データは，商品ID，商品種別，価格から成り，それぞれの商品に対して「高額商品」か「低額商品」かのクラスが付与されている。野菜の場合は500円でも「高額商品」であるが，魚介の場合は800円でも「低額商品」であり，価格で一律にクラスが決まっているわけではない。そこで，決定木では，例えば次のような分類ルールが生成される。この分類ルールを用いることで，教師データに含まれず，クラスが未知のデータが来た際にも適切なクラスを付与することができるようになる（例えば，商品IDが5で，商品種別が野菜，価格が100円ならば，「低額商品」となる）。

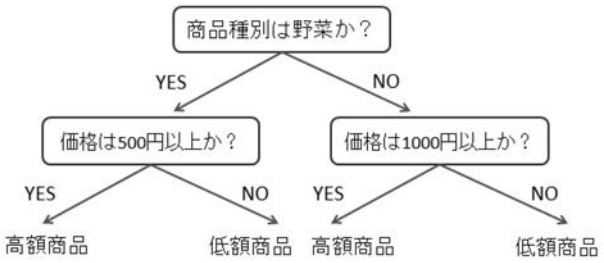


図1　決定木による分類の一例

（2） 文字（テキスト）データの分類

先ほどの教師データの例では，商品種別は名詞，価格データは数字であったため，判別が比較的容易であった。しかし，特許文献のように文章を含むデータに対してクラス分類する場合は，先ほどのようにはいかない。例えば，価格が次のような文字データとなった場合はどうだろうか。

商品 ID	商品種別	価格(円)	クラス
1	野菜	高い価格	高額商品
2	野菜	300 円	低額商品
3	魚介	1000 円札 3 枚	高額商品
4	魚介	800 円	低額商品

表２　教師データの一例

価格に記載された文字列が，「高い価格」または「1000 円札 3 枚」だった場合は「高額商品」というように分類ルールを決めることもできるが，この場合，価格が「1000 円札 2 枚」と記載された商品データは判別することができない。そこで，文字データの場合はまず，文字データを単語ベクトルに置き換えるという作業を行う。

商品 ID	商品種別	価格(円)	クラス
1	野菜	(高い，価格)	高額商品
2	野菜	(300 円)	低額商品
3	魚介	(1000 円札，3 枚)	高額商品
4	魚介	(800 円)	低額商品

表３　教師データの一例

この場合，例えば「高い」や「1000 円札」といったワードが出現すれば「高額商品」といったようなルールが生成されることになる。そうすると，先ほどのように，「1000 円札 2 枚」と価格に記載された商品は「高額商品」と分類されることになり，新たなデータに対して判別できる可能性を高めることができる。以上のように簡単な例を用いて文字データを分類する流れを見てみたが，これ以外にも文字データには数値データにはない特徴があり，数値データを分類する時より難易度が高い。例えば，日本語の場合は，英語のように単語がスペースで分かれているわけではないため，どこで切って単語ベクトルを作るかといった問題がある。これは「形態素解析」という自然言語処理技術を用いることで，文章から単語に分けることができるのであるが，それでも 100%正しい分け方ができるわけではない。従って，精度をどのように高めるかという点が課題となる。一方，英語の場合は，動詞は語尾が活用することや，名詞も単数形と複数形が存在するといった事情があるため，同じ意味であっても異なるデータとなるケースがある。このようなデータ上のゆらぎを吸収するために，「stemming」や「lemmatization」という処理が行われる。swim，swimming，swimmer などの文字データから幹となる部分を抽出し，すべて swim というデータとして扱うのである。

以上のように，文字データの分類は，もともとのデータを整形し，機械学習に適用するデータにする処理が難しく，そこに多くのノウハウ等が存在する。しかし，IBM Watson 等のように文字データが扱える機械学習のクラウドサービスも登場してきており，こういった難しい部分も将来的にはユーザーは考える必要がなくなりつつある。本稿では，機械学習を体験することがメイントピックであるため，この文字データを準備する部分は筆者側で別途準備した。本節で触れた技術等については自然言語処理の文献により詳細な説明がされている[5]。より詳細な技術内容に興味のある方はそちらを参照していただきたい。

（３）　自動分類の活用状況

本節では，特許文献の自動分類技術が，今現在，どのように活用されているかについて説明する。

文書・文献の自動分類は，様々な場面で現実に活用されており，例えばニュースサイトの各記事のタグ付けなどによく用いられている[6]。しかし，特許文献の場合，分類コードが階層化されていて，それぞれの分類コードが表す内容も細かいため，ニュース記事や電子メールといった短くコンパクトにまとまった内容を扱う場合とは異なるという指摘もされている[7]。実際に特許分類の付与業務を実施している財団法人工業所有権協力センター（IPCC：Industrial Property Cooperation Center）では，特許分類の自動推定に向けた取り組みとして調査研究は行っているが[8]，実用されているという報告はない[9]。これは，分類精度にまだ課題があるためであるが，調査研究は諸外国含めて精力的に行われており[10]，ディープラーニングや word2vec といった近年の機械学習技術・自然言語処理技術の発展によって，今後の実用化が期待される。一方，機械学習を用いた特許文献のクラスタリングについては，特許情報の分析を行うシステムで実用化が既にされており[11]，機械学習は特許情報の世界に着々と浸透しつつある。

３．自動分類処理の作成と実行

本章では，本稿のメイントピックである機械学習の体験を行う手順について説明する。まず，機械学習で何を行うかという目標について説明した後，作業の流

れを説明し，作業中の各ステップについての説明に移る。

（1） 目標設定

機械学習が利用できるOSSのWeka[12]を用いて特許文献の自動分類を体験する。1か月分の日本特許の公開公報を教師データとして学習を行い，翌月1か月分の公開公報をテストデータとして自動分類を試みる。

（2） 作業の流れ

今回行う作業は以下の4ステップからなる。まず，機械学習を行うために必要なハードウェアとソフトウェアを準備する（ステップ1)。その後，機械学習に用いるデータを準備する（ステップ2)。そして教師データを用いて学習を行い，分類ルールを生成する(ステップ3)。最後に生成した分類ルールを用いて，テストデータの自動分類を実行する（ステップ4)。

ステップ1	ハードウェア・ソフトウェア環境の準備
ステップ2	教師データとテストデータの準備
ステップ3	分類ルールの作成
ステップ4	自動分類の実行確認

表4　本稿で行う作業ステップ

以降では，これら4つのステップについてそれぞれ解説していく。紙面の都合上，多少説明を簡略化している部分もある。より詳しい説明を知りたい場合は，Wekaの説明が行われている書籍やサイトを巻末に示したので参照いただきたい[13]。

（3） ステップ1：ハードウェア・ソフトウェア環境の準備

①使用するハードウェア，ソフトウェア

本稿を執筆する上で筆者が用いた環境を以下に示す。下の環境は一例であり，必ずしも同じ環境である必要はないが，本稿ではWindows環境を前提に説明を行う。説明は省略させていただくが，WekaはMacやLinuxでも動作するので，本稿の説明内容に対して若干の修正を加えることで，他のハードウェアでも同じことが行える。

CPU	Intel Core ™ 2 Quad CPU 2.83GHz
メモリ	4GB
ハードディスク	500MB
OS	Windows7 Service Pack1(64bit)
ソフトウェア(＊)	Java 1.8.0_101 Weka 3.7.11

表5　ハードウェア・ソフトウェア環境の一例
(＊)以降のステップでインストールを説明する

②Javaのインストール

まず，Wekaの動作に必要なJavaというソフトウェアをインストールする。Oracle社の公式サイト（https://java.com/ja/download/）からJavaのインストーラをダウンロードする。Windows版の場合，ファイル名はjxpiinstall.exeであり，このファイルをダウンロード後に実行し，インストールを行う。

図2　Oracle社のJavaダウンロードサイト
https://java.com/ja/download/

図３　Java のインストール完了後に表示されるダイアログ

③ Weka のインストール

原稿執筆時点での Weka 最新バージョンは 3.8 であるが，本稿では 3.7.13 を利用する[14]。64 ビット版ウインドウズのバイナリは Sorceforge のサイト（https://sourceforge.net/projects/weka/files/weka-3-7-windows-x64/3.7.13/）からダウンロードできる。このファイルを実行することで Weka がインストールされる。インストール時のポイントとして，インストール先フォルダを例えば”c:\Weka”のように変更しておくことをお勧めする。標準のインストール先フォルダのまま進めると，Windows の UAC（ユーザアカウント制御）機能が働き，以降の設定変更時につまずいてしまう可能性があるからである。

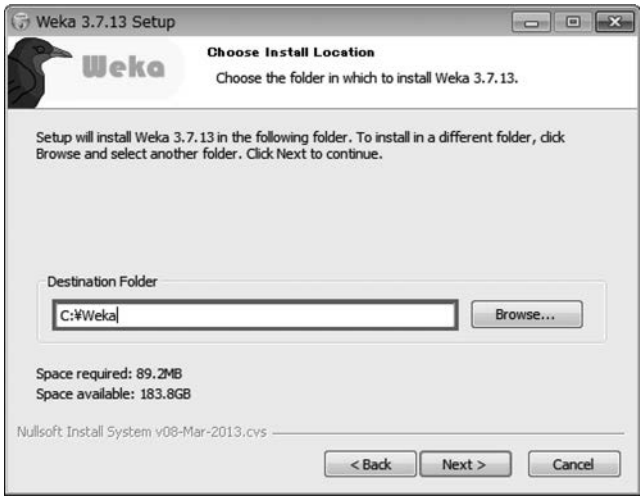


図４　Weka インストール時のインストール先フォルダの選択画面

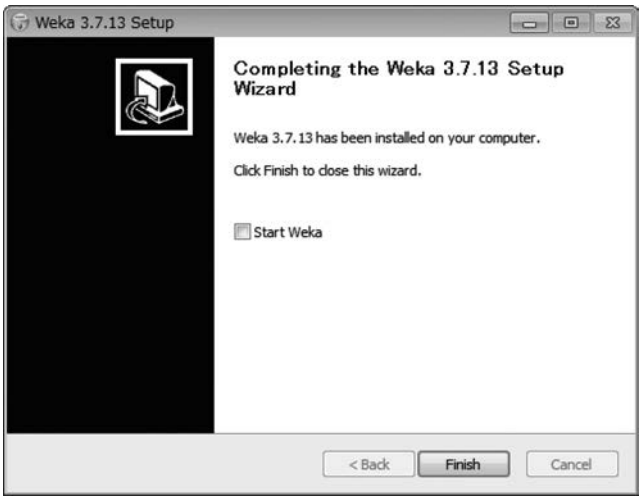


図５　Weka のインストール完了を知らせるダイアログ

インストールが完了すると，今度は Weka の設定変更を行う。Weka を実行している場合はいったん終了する。今回利用する教師データ，テストデータは文字コードが UTF-8 という形式のデータである。Weka が扱うデータの文字コードを UTF-8 に指定する必要がある。これまでの手順に従っている場合，C:\Weka にインストールされているので，そのフォルダの中にある RunWeka.ini というファイルをダブルクリックする。すると，メモ帳等のエディタが起動するので，下の図のように fileEncoding=xxx という部分の xxx を”utf-8”に変更し，保存する。これで Weka の設定変更が完了する。

図６　Weka の設定変更

（4） ステップ2：教師データとテストデータの準備

今回は，特許庁の公報発行サイトからダウンロードした2016年1月と2月に発行された公開公報を利用し，1月分を教師データ，2月分をテストデータとして準備する。教師データとテストデータは同じ形式で，各行が1件の公報に対応する（但し，先頭行は見出し）。そして，列には，公報番号，公報中から抽出したキーワード（今回は抄録部分からTF-IDFによって重みづけを行い抽出した上位キーワード），IPC特許分類（サブクラス記号）の3要素を配置する（具体例は次の図を参照）。テストデータには本来は3列目のIPC特許分類は不要であるが、自動分類結果との答え合わせに便利なので追加している。この列を削除すると後述するKnowledge Flowを若干変更する必要があるため、今回はこの形式で進めていただきたい。

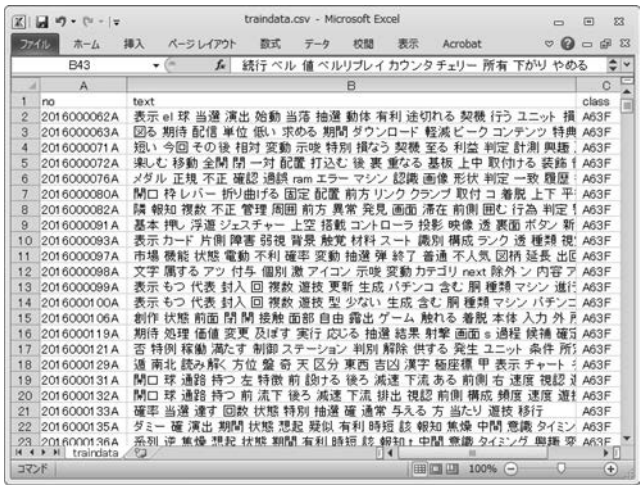


図7　2016年1月発行の公報群から作成した教師データ

図8　2016年2月発行の公報群から作成したテストデータ

この作成ステップはやや煩雑であるため，今回は以下のサイトにデータを準備した。文末の補足の章にてデータ作成について簡単に述べるが，まずはこのデータをダウンロードして次のステップに進んでいただきたい。

データダウンロード用サイト
https://github.com/jpa17267/patent-autoclassification

データのダウンロードは上のURLにブラウザソフトでアクセスし，次の図のように① Clone or download ボタン，② Download ZIP ボタン，を押すことで完了する。ZIPファイルがダウンロードされるので，そのファイルを解凍することで”traindata.csv”（教師データ），”testdata.csv”（テストデータ）が入手できる。

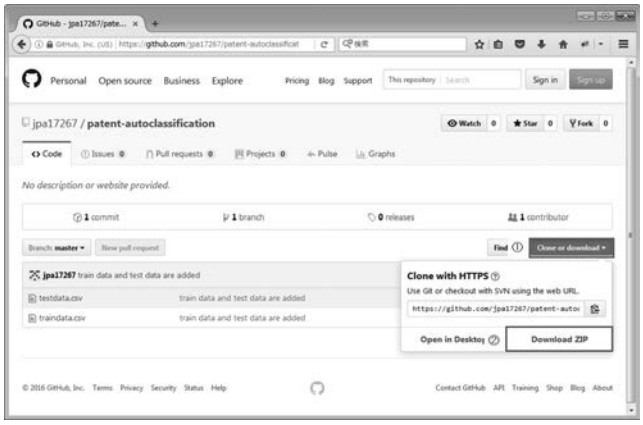


図9　データのダウンロードサイト
https://github.com/jpa17267/patent-autoclassification

今回の教師データ，テストデータでは，クラス数を出現頻度の上位4つ（A63F，G06F，H04N，H01M）に絞っている。これら以外のクラスに属するデータについては排除した。あまりサンプル数が少ないと十分な学習が行えないためである。これら以外の特許分類に対しても学習を行いたい場合には，別の月の公開公報を利用する等をして学習データを増やすこと等を検討する必要がある。

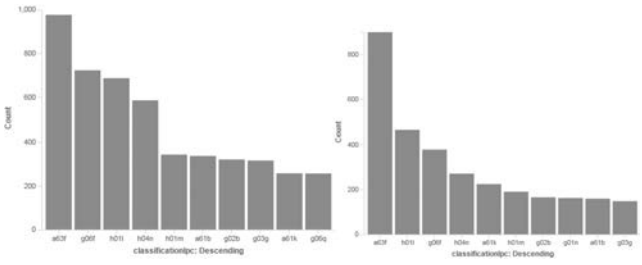


図10　2016年1月（左）と2月（右）の出現頻度上位10位の特許分類

（5） ステップ3：分類ルールの作成

① Weka の実行

まず，スタートメニューやデスクトップのショートカットをクリックし，Weka を実行する。すると，以下のような画面が表示されるので，Applications のメニューから KnowledgeFlow というボタンをクリックする。すると，Weka KnowledgeFlow Environment というウインドウが出現し，このウインドウで分類ルールの作成，および作成済の分類ルールを用いた自動分類の実行を行うことができる。

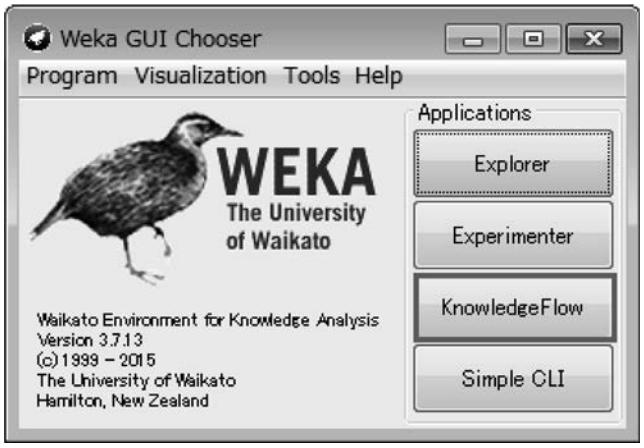


図 11　Weka 起動後の画面

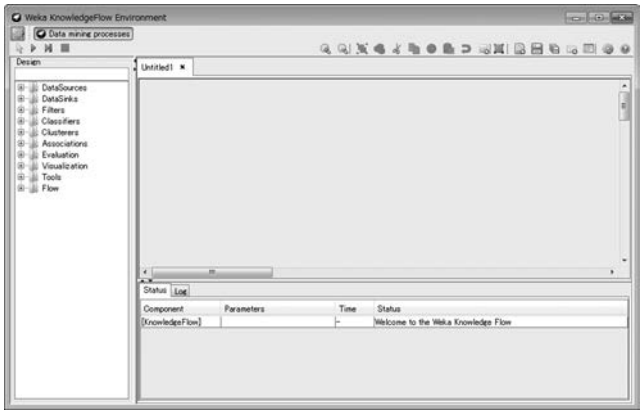


図 12　Weka KnowledgeFlow Environment ウインドウ

② Knowledge Flow による分類ルールの作成

ここでは，教師データである”traindata.csv”を用いて機械学習を行い，4つのクラス（A63F，G06F，H04N，H01M）への分類ルールを作成する。

まずは入力ファイルを設定する。左の Design と書かれた領域から DataSources/CSVLoader（ツリービューで DataSources を選択して展開し，その後 CSVLoader を選択することを本稿では以降このように記載する）を選択し，中央のメインエリアをクリックすると，次の図のように CSVLoader のアイコンが表示される。

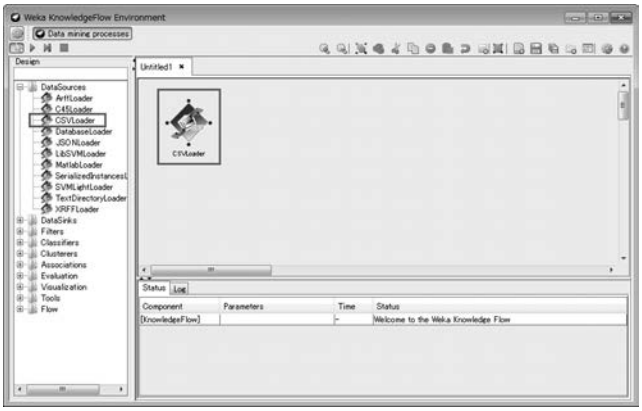


図 13　CSVLoader を配置したところ

その後，CSVLoader のアイコンをダブルクリックすると，以下のダイアログが表示されるので，Filename のところに教師データ（“traindata.csv”）を指定し，stringAttribute の値を 2 に変更する。これは，”traindata.csv”の2列目が今回の学習対象となる抄録データであることを示している。

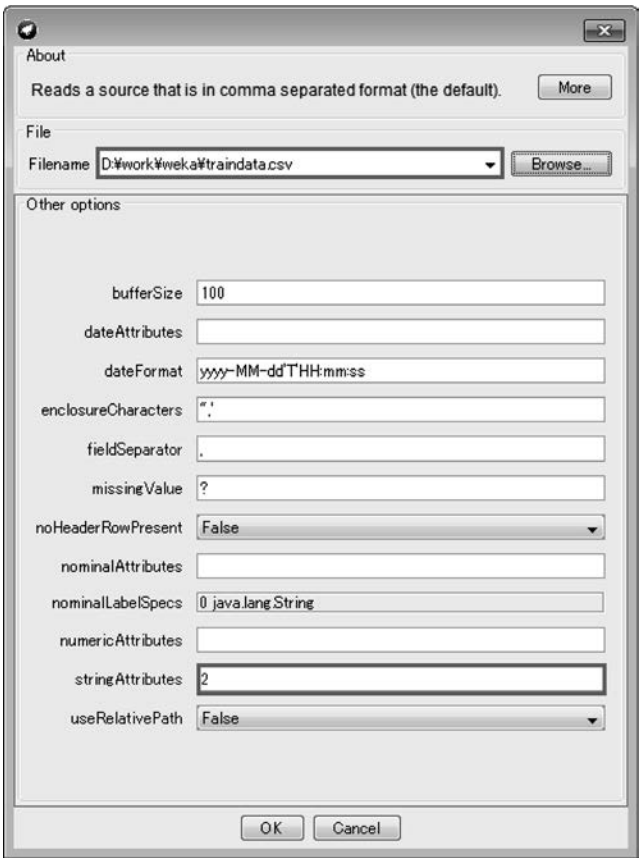


図 14　CSVLoader の設定

次に，教師データ中のクラスデータ（正解データ）を設定する。そのためには，Filters/unsupervised/attribute/ClassAssigner を選択し，CSVLoader と ClassAssigner を接続する。接続するためには，

CSVLoader を右クリックして開いたコンテクストメニューから dataSet を選択した後，ClassAssigner をクリックすればよい。ClassAssigner を接続すると，”traindata.csv”の最後の列がクラスデータと指定したことになる。

図 15　CSVLoader と ClassAssigner を接続したところ

次に，抄録データをスペースで単語に分割するため，Filters/unsupervised/attribute/StringToWordvectorを ClassAssigner の後ろに接続する。そして，StringToWordVector をダブルクリックし，設定画面で attributeIndices を 2 に設定する。これによって，単語分割する対象が 2 列目であることを指定する。

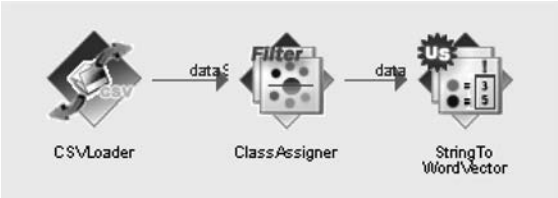


図 16　StringToWordVector を接続したところ

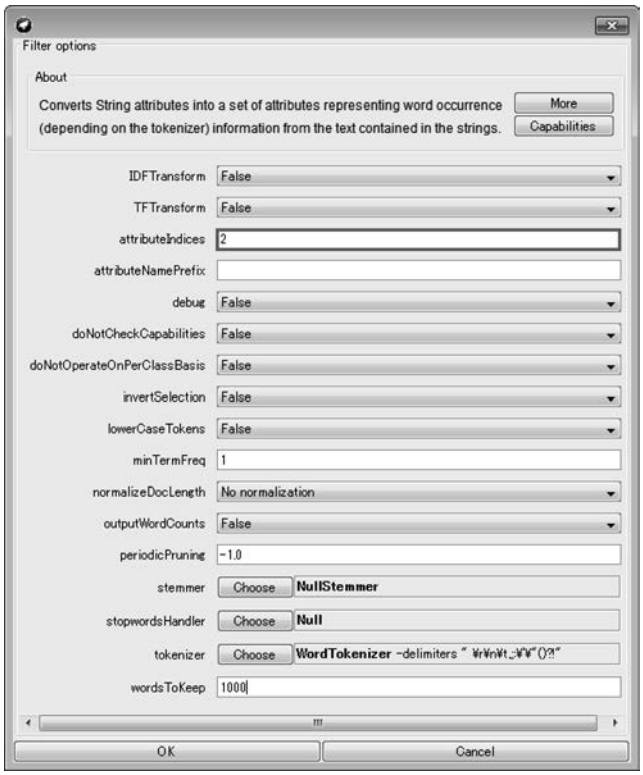


図 17　StringToWordVector の設定

次に，Evaluation/TrainingSetMaker を接続し，さらに Classifiers/trees/J48 を接続する。J48 は決定木を用いた分類ルールの生成器である。J48 を接続する際，これまでのように dataSet で接続するのではなく，trainingSet を選択して接続する。最後にDataSinks/SerializedModelSaverを J48 から batchClassifier を選択して接続し，SerializedModelSaver をダブルクリックして出力ファイルとファイル名の prefix（接頭文字列）を設定する。以下の例では，Prefix として“mymodel_”を指定しているため，分類ルールのデータは mymodel_J48～.model というファイルが作成される（～の部分はその他の設定によって変わる）。

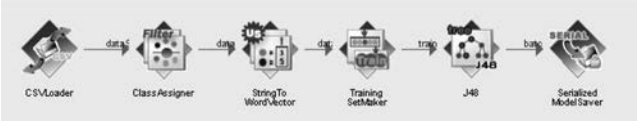


図 18　分類ルールを生成する KnoledgeFlow

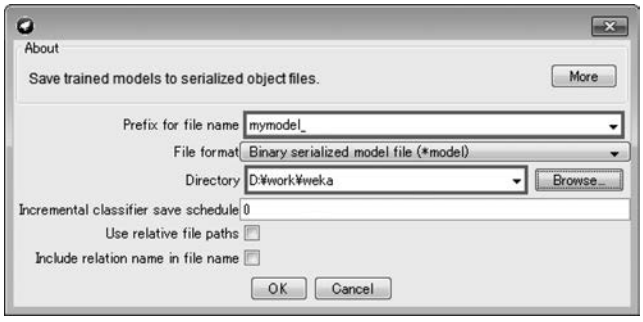


図 19　SerializedModelSaver の設定

KnowledgeFlow が完成すると，メニューバーにある実行ボタンを押すことで作成作業が開始する。作成完了すると，右下の Status がすべて Finished になる。この処理は若干時間を要し，筆者の動作環境では完了まで約 50 秒かかっている。

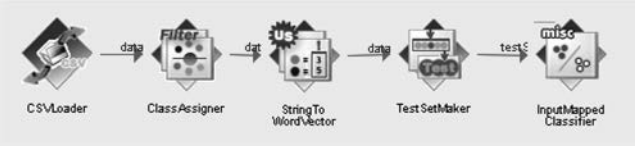
図 20　分類ルールの生成完了時の画面

（6）　ステップ 4：自動分類の実行確認

①分類処理の作成

ここでは，先に作成した分類ルールを用いて testdata.csv の自動分類を実行する。

まず，分類ルールを作成したときと同じくCSVLoaderでtestdata.csvを読み込む設定をし，ClassAssignerとStringToWordVectorを接続する。それぞれの設定も同じように行う。ここから，分類ルールの作成時とは異なるが，Evaluation/TestSetMakerを接続し，TestSetMakerを右クリックして出てくるtestSetにてClassifiers/misc/InputMappedClassifierと接続する。



InputMappedClassifierは，先ほど作成した分類ルールによって分類を行う分類器なので，分類に使う手法としてJ48，分類ルールファイルとして先ほど作成したファイル（“mymodel_J48_1_1.model”）を設定にて指定する。

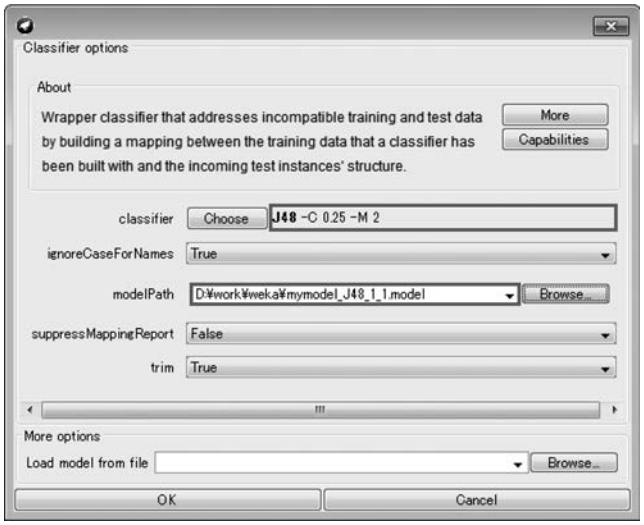


図21　InputMappedClassifierの設定画面

ここまで出来たら，あとは作成された分類結果データを見やすくするための整形作業である。Evaluation/PredictionAppenderをInputMappedClassifierの右クリックでコンテクストメニューに表示されるbatchClassifierを選択して接続し，PredictionAppenderを以下の設定にする。

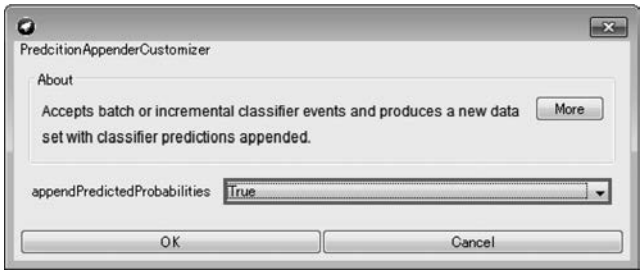


図22　PredictionAppenderの設定

この設定によって，分類結果でクラスごとに属する可能性が確率表示されるようになる。次に，分類結果を見やすくするために，余分なデータを削除する。Filters/unsupervised/attribute/RemoveByNameをPredictionAppenderの右クリックでコンテクストメニューに出てくるdataSetにて接続する。RemoveByNameの設定では，InputMappedClassifierの結果以外を削除するように，以下の設定を行う。

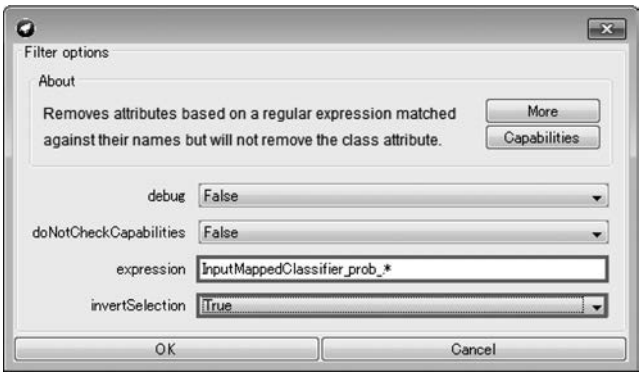


図23　RemoveByNameの設定

この結果をDataSink/CSVSaverに接続することで，分類結果をCSVファイルに出力させることができる。出力ファイルの指定はCSVSaverの設定で行える。

図24　最終的に出来上がったKnowledgeFlow

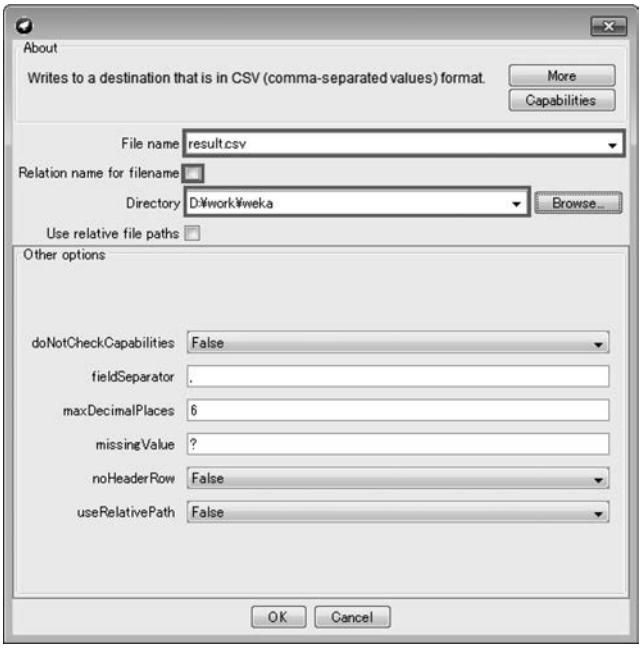


図25　CSVSaverの設定

②分類結果の確認

KnowledgeFlow を実行すると，”result.csv”というファイルが作成されるので，その中身を見ることで分類結果を知ることができる。以下の図は，result.csv の先頭列に，testdata.csv の先頭列を追加したデータである。B 列が実際に付与されていた正解クラス，C 列から F 列は自動分類がそれぞれ A63F，G06F，H01L，H04N と分類される可能性を 1（100%）から 0（0%）で示している。黄色で示した行のように誤って分類しているケースもあるが，単純な処理にも関わらず正解率は約 85% と比較的高い。さらに改善するためには，教師データの変更，決定木以外の別手法への変更，キーワードの抽出等の自然言語処理の改良，等のアプローチをとっていくことになる。興味のある読者は専門書を手に取って検討を進めていただきたい。

	A	B	C	D	E	F
1	no	' class'	'InputMappedClassifier_prob_ A63F'	'InputMappedClassifier_prob_ G06F'	'InputMappedClassifier_prob_ H01L'	'InputMappedClassifier_prob_ H04N'
2	201601597OA	' A63F'	1	0	0	0
3	201601597 1 A	' A63F'	1	0	0	0
4	201601598 2 A	' A63F'	1	0	0	0
5	201601598 7A	' A63F'	1	0	0	0
6	201601598 8A	' A63F'	1	0	0	0
7	201601598 9A	' A63F'	1	0	0	0
8	201601599OA	' A63F'	0.060606	0.909091	0	0.030303
9	201601599 1 A	' A63F'	1	0	0	0
10	201601599 3A	' A63F'	1	0	0	0
11	201601599 4A	' A63F'	1	0	0	0
12	201601599 9A	' A63F'	1	0	0	0
13	201601600 4A	' A63F'	0	0.95	0	0.05
14	201601600 5A	' A63F'	1	0	0	0
15	201601600 6A	' A63F'	1	0	0	0
16	201601600 7A	' A63F'	0	0.95	0	0.05
17	201601600 9A	' A63F'	1	0	0	0
18	201601601 1 A	' A63F'	1	0	0	0
19	201601601 3A	' A63F'	1	0	0	0
20	201601601 4A	' A63F'	1	0	0	0
21	201601601 5A	' A63F'	0.074866	0.791444	0.064171	0.069519
22	201601601 6A	' A63F'	1	0	0	0
23	201601601 9A	' A63F'	1	0	0	0
24	201601602 0A	' A63F'	1	0	0	0
25	201601602 1 A	' A63F'	1	0	0	0
26	201601602 3A	' A63F'	1	0	0	0
27	201601602 4A	' A63F'	1	0	0	0
28	201601602 5A	' A63F'	1	0	0	0
29	201601602 6A	' A63F'	1	0	0	0
30	201601602 7A	' A63F'	1	0	0	0
31	201601602 8A	' A63F'	1	0	0	0
32	201601602 9A	' A63F'	1	0	0	0
33	201601603 4A	' A63F'	1	0	0	0
34	201601603 5A	' A63F'	1	0	0	0
35	201601604 5A	' A63F'	1	0	0	0

図 26　分類結果の例

4．むすび

MIT スローン経営大学院のエリック・フォン・ヒッペル教授は，自著「民主化するイノベーションの時代」にて，次のように述べている。

「ユーザーが高品質の製品やサービスを自ら開発する能力は，劇的かつ急速に向上している。コンピュータのソフトとハードが進歩し続けているため，イノベーションに必要なツールの能力が次第に向上する一方，価格は確実に下がり，特別なスキルや訓練の必要性も日を追って少なくなってきた。」

エリック・フォン・ヒッペル，民主化するイノベーションの時代，ファーストプレス，p158

今後もこの傾向は更に進むと考えられる。人工知能技術を使うのは，政府や企業といった大きな組織だけではなく，弁理士や特許事務所という個人や少人数でも十分に可能であり，数年後には Excel を使うのと同じような感覚で人工知能技術を使われているかもしれない。人工知能によって仕事が奪われるといった議論もされているが，我々知財業界に従事する者が積極的に人工知能を使いにいくことで，これまで以上に付加価値の高い業務を行えるようになる可能性がある。本稿がわずかでもそのきっかけ作りに貢献できれば幸いである。

補足：教師データ・テストデータの作成

教師データ・テストデータの作成は次の 3 ステップからなる。

ステップ 1：公報発行サイトからデータをダウンロードする
ステップ 2：XML ファイルから必要なデータを抽出する
ステップ 3：テキストデータからキーワードベクトルを作成する

各ステップにつき簡単に説明する。

ステップ 1：公報発行サイトからデータをダウンロードする

公報発行サイト（https://www.publication.jpo.go.jp/index.action）にアクセスし，一括ダウンロードボタンを押して，該当する年月の公開公報を選択する。平成 28 年 1 月の公開公報をダウンロードする場合は，以下の 6 つのファイルをダウンロードすることになる。


特許庁 インターネット利用による 公報発行サイト

当サイトでは、「公開公報」、「特許・実用新案公報」、「登録実用新案公報」、「意匠公報」、「商標公報」、「公開・国際商標公報」、「審決公報」及び「特許庁公報」を発行します。

過去に発行された公報を一括ダウンロードできます。

ダウンロードしたい公報種別及び発行年月を選択してください。

公開公報 平成 28 年 1 月 表示

平成28年1月発行の公開公報です。

公報種別	発行日	号数	掲載範囲	ZIPファイル	TARファイル
公開公報	平成28(2016).1.7	28(2016)-001[682]	2016-000001 ～2016-001996	3,376(MB) ダウンロード	4,239(MB) ダウンロード
公開公報	平成28(2016).1.12	28(2016)-002[683]	2016-001997 ～2016-005434	3,162(MB) ダウンロード	4,258(MB) ダウンロード
公開公報	平成28(2016).1.14	28(2016)-003[684]	2016-005435 ～2016-007136	3,317(MB) ダウンロード	4,223(MB) ダウンロード
公開公報	平成28(2016).1.18	28(2016)-004[685]	2016-007137 ～2016-010315	3,337(MB) ダウンロード	4,283(MB) ダウンロード
公開公報	平成28(2016).1.21	28(2016)-005[686]	2016-010316 ～2016-013056	3,290(MB) ダウンロード	4,244(MB) ダウンロード
公開公報	平成28(2016).1.28	28(2016)-006[687]	2016-013057 ～2016-015887	3,298(MB) ダウンロード	4,251(MB) ダウンロード

図 27　公報発行サイトのダウンロード画面（平成 28 年 1 月分）
https://www.publication.jpo.go.jp/ik_pub/index.action?lang=ja_JP

ステップ２：XML ファイルから必要なデータを抽出する

XML ファイルから公報番号，抄録，クラスのデータを抽出し，CSV ファイルを生成する。XML ファイル中，公報番号は <publication-reference>の下の<doc-number>に，抄録は <abstract> に，クラスは <classification-ipc> にデータが格納されている。

ステップ３：テキストデータからキーワードベクトルを作成する

ステップ 2 で変換した abstract の文字列データをキーワードベクトル（品詞ごとにスペースで区切られたデータ）に変換する。これは mecab を用いて行うことができる。自動分類の精度を高めるためには，単にスペースで区切ったデータにするだけではなく，TF-IDF 等を用いてキーワードを選別する。

以上

（注）

(1) 日経 BP ITpro 2016/8/3 記事より（http://itpro.nikkeibp.co.jp/atcl/watcher/14/334361/072900634/?rt=nocnt）

(2) 人工知能学会 WEB ページより（http://www.ai-gakkai.or.jp/whatsai/AIresearch.html）

(3) Labellio は，ディープラーニング技術を用いて簡単に画像認識モデルを作成できる WEB サービスである。（https://www.labell.io/ja/）

(4) 機械学習アルゴリズムについては，例えば Microsoft が提供する機械学習も利用可能なクラウドサービス Microsoft Azure のサポートサイトで詳しく解説されている。参考 URL（https://azure.microsoft.com/ja-jp/documentation/articles/machine-learning-algorithm-choice/）

(5) 自然言語処理技術については，例えば「Steven Bird 他，入門　自然言語処理，O’REILLY」に説明がある。

(6) 一例として「はてなブックマーク 2」が挙げられる。この WEB サイトでは，Complement Naïve Bayes という機械学習の手法を用いて記事の自動分類を行っている。参考記事（http://ascii.jp/elem/000/000/186/186430/index.html）

(7) 奥村学，特許情報処理：言語処理的アプローチ，コロナ社，P74

(8) 小林英司，特許分類の自動推定に向けた取り組み－機械学習による自動分類推定の課題と今後の展開－，Japio Year Book 2015，p275

(9) 実際に運用されている事例の報告としては，特許分類の一元付与業務システムつけつけ君 II が挙げられる（古屋野浩志，特許分類等の付与精度向上への取り組み，Japio Year Book 2007，p119）

(10) 例えば，WIPO は自動分類の研究のために教師データの提供を行っている。参考サイト（http://www.wipo.int/classifications/ipc/en/ITsupport/Categorization/dataset/wipo-alpha-readme.html）

(11) 一例として，INNOGRAPHY が挙げられる。検索結果から特許文書のグルーピングを自動で行い可視化する機能を持っている。参考サイト（https://www.innography.com/why-innography/visualizations）

(12) Weka（Waikato Environment for Knowledge Analysis）は，Waikato 大学によって開発された機械学習を利用するためのオープンソースソフトウェアである。詳しい説明については，Weka のホームページを参照のこと。（http://www.cs.waikato.ac.nz/ml/index.html）

(13) 本稿の手順をより詳細に説明した補助資料として，「知りたいわかりたい人の体験する機械学習（高橋佑幸著・リックテレコム発行）」が挙げられる。また，Weka のより詳細な使い方を説明した文献として，「フリーソフトではじめる機械学習入門（荒木雅弘著・森北出版）」が挙げられる。

(14) 補助資料として挙げている文献「知りたいわかりたい人の体験する機械学習（高橋佑幸著・リックテレコム発行）」でも Weka 3.7.11 を利用していることから本稿でも利用するバージョンを 3.7 系に合わせることにした。紙面の都合上，説明を簡略化している部分もあり，わかりにくい箇所があれば補助資料を参照いただきたい。

（原稿受領 2016. 10. 24）