Exploration and Implementation of Classification Algorithms for Patent Classification

Darshana A. Naik, S. Seema, Geetika Singh and Abhinav Singh

Abstract Data mining techniques have seen tremendous increase in their usage in the past few years. Patent mining is one of the domains that utilize data mining techniques to a great extent. Patent mining consists of various tasks such as retrieval of patent, classification, patent valuation, patent visualization and detecting infringements. Among these, patent classification is an important task. It deals with the classification of patents into various categories. A common bottleneck in this task has been related to the automated classification of patents with better accuracy. The rapid increase in the number of patents being filed every year and the increasing complexity of the patent documents demand for advanced and revolutionized tools/machines to assist in performing patent classification in automated manner. Usually, the patents are examined thoroughly by patent analysts from various domains, who possess respective expertise and are well aware of the domain jargons. The main objective of such systems is to get rid of the time-consuming, laborious manual process and to provide patent analysts a better way for classifying patent documents. Also it helps in better management, maintenance and convenient searching of patent documents. Here, two prominent classification algorithms—Naïve Bayes and support vector machines (SVM)—are explored and implemented. Additionally, some preprocessing steps such as stop word removal, stemming, and lemmatizing are also done to obtain better accuracy. TF-IDF feature is also incorporated to obtain precise results.

Keywords Data mining · Patent mining · Patent classification · Classification algorithms · Naive Bayes · Bag of words · Support vector machine · TFIDF

D. A. Naik (✉) · S. Seema · G. Singh
Department of CSE, MSRIT, Banglaore, India
e-mail: darshananaik@msrit.edu

S. Seema
e-mail: seemas@msrit.edu

G. Singh
e-mail: geethika.singh20@gmail.com

A. Singh
CloudxLab Inc., Bangalore, India
e-mail: abhinav.foss@gmail.com

S.-L. Peng et al. (eds.), *Computing and Network Sustainability*, Lecture Notes in Networks and Systems 75, https://doi.org/10.1007/978-981-13-7150-9_12

1 Introduction

Patent documents play crucial role in the protection of invention of an individual or organization. Patent documents are hugely different from ordinary documents. Patent document has a specified format and consists of front page, description section, references and diagrams as shown in Fig. 1. However, patent documents are huge and contain myriads of technical jargons which require an extensive amount of efforts for analysis. Therefore, in order to assist patent analysts to perform certain processes such as patent data examination, processing, classifying and analyzing, a separate stream called as patent mining is created. Patent mining forms a part of data mining. Just the way data mining assists in exploring patterns in the data so does patent mining in patent content [1].

The drastic increase in the number of patents calls for the emergence of a need for designing a tool that automatically categorizes the patents. This would decrease the extensive amount of effort to be spent on performing manual patent classification. Hence a sophisticated tool developed in this direction will be of great advantage. A key bottleneck for such systems is the ability of discovering and utilizing the data stored in databases [2]. Apart from the above concern, a system must also have the ability to analyze the complexities involved in interlinking the patent information.

Patent mining comprises various tasks such as retrieval of patent, classification, patent valuation, patent visualization and detecting infringements. [1]. Patent classification is an important task. It helps in flexible management and easy maintenance of patent documents.

Usually, patent undergoes thorough examination by patent analysts from various domains, who possess respective expertise and are well aware of the domain jargons. WIPO (World Intellectual Property Organization) is one of the standard agencies that help in preserving and promoting the patents. WIPO designates patent analysts who examine the patent data and classify manually. Such manual classification is quite laborious and time-consuming.

Therefore, to classify patents efficiently, there emerges a requirement for a sophisticated system which will automatically classify the patents.

In past, many classification algorithms have been designed. However, the results are still far from obtaining the accuracy. There are many hurdles faced during the patent classification process such as (1) complex structure—patent is usually filled with over-usage of various jargons which makes classification unsuccessful because of difficulty in fetching useful features, (2) complex hierarchy of schema of patent classification and (3) increasing volume of patent documents [1]. To address these hurdles, researchers have worked on developing powerful classification algorithms. The research work was carried out with emphasis on (1) using various kinds of information provided in patent for performing classification and (2) analyzing how the different kinds patent documents behave when tested against different classification algorithms.

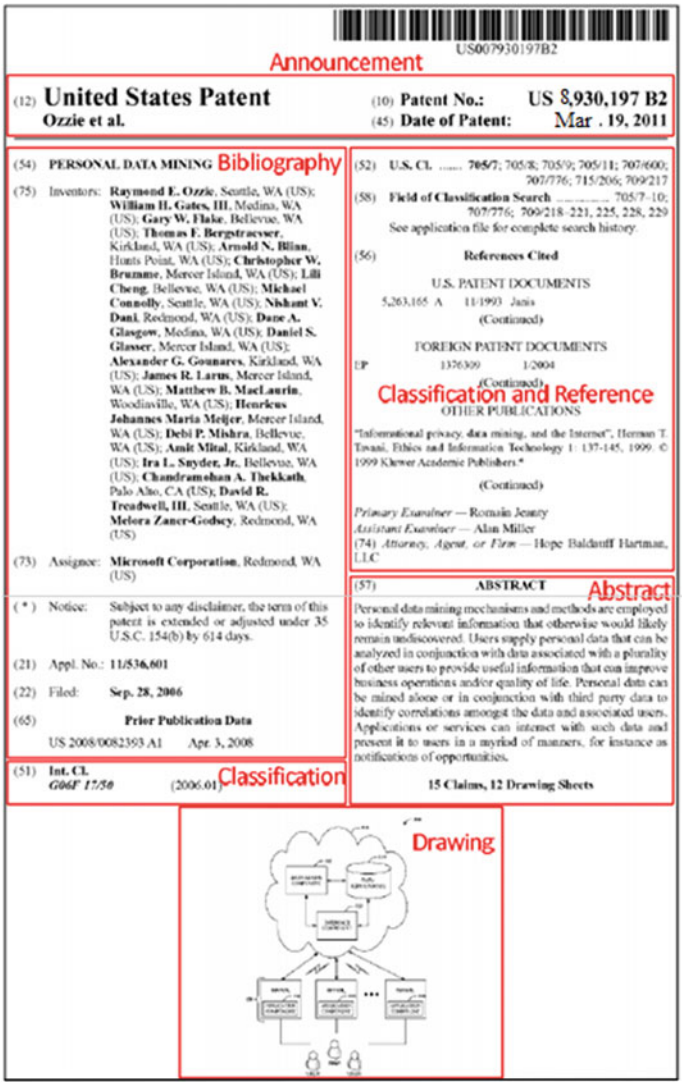


Fig. 1 Sample front page of patent

This paper makes an effort to explore, implement and test certain classification algorithms for patent classification. Several optimizations are also done gradually and results are recorded.

2 Bag-of-Words Model

Bag-of-words (BOW) model is used to represent textual data in machine learning domain. Similarly, BOW is used here to represent patent content. The unstructured text document is often represented in the form of BOW model. Some of the parameters in the dataset such as title, abstract and references can be represented in BOW model. Most of the classification algorithms use BOW model in their underlying implementation.

3 Dataset Generation

To perform the classification process in unsupervised manner, one requires dataset. If done manually, the dataset creation will require enormous effort. Hence, an attempt of implementing a crawler is done that will crawl the desired website or repository containing patent information.

The web crawler would crawl the pages or repository and will fetch the relevant information such as patent title, patent abstract and IPC code. One can choose to download the data in CSV format, XML or JSON, etc. Here, CSV format is chosen. The web crawler is written in python language. Table 1 shows the template of dataset collected.

4 Classification Process

The classification process generally consists of various phases such as retrieval of document, tokenization, stemming, stop word removal, vector representation and feature selection [4]. Each phase augments the process and leads to the direction of accuracy. Figure 2 illustrates all the phases involved in classification process as represented below.

Table 1 Template of dataset

Name:	PatentDataset.csv
Columns	Title—contains the title of the patent Abstract—contains the brief summary of the patent Class—represents the IPC class code of the patent Class_code—represents the equivalent of class column in numerical format

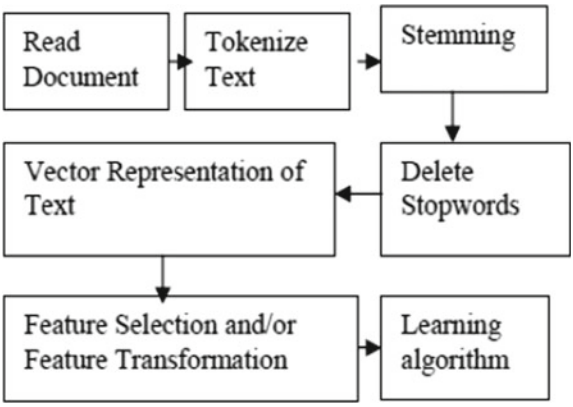


Fig. 2 Classification process

4.1 Tokenization

In this phase, a stream of text is divided into an indivisible element called as token. The tokens produced are fed into the subsequent phases of classification. A token can be defined as a continuous sequence of alphabetic characters or numerical characters. Tokens are delimited by whitespace characters such as single space, tab space, line break or special punctuation marks. Many algorithms contain inbuilt support to perform tokenization. Tokenization might seem an easy task in English language but is difficult for the languages that lack word boundaries like Chinese language.

4.2 Stemming

Stemming takes care of diminishing the deviation of inflectional forms and derivational related words to the original form. Words with similar base do not contribute toward classification process and must be disregarded.

4.3 Stop Word Removal

Stop words do not contribute anything toward classification process and hence must be filtered out in prior. Instead for few systems, such words cause difficulties and degrade the performance. There is no pre-defined list of stop words. A collection of words can be considered as stop words. There are certain common words that are used in some search engines.

Python provides various APIs to facilitate stop word removal. One can simply specify whether to ignore the stop words or not by mentioning the language to be

considered for stop words and setting the appropriate flag. The list of stop words is pre-decided but can be customized as per one's desire. Here, the accuracy obtained was 18% before removing the stop words but was increased to 59% after removal of stop words using naive Bayes algorithm.

4.4 Vector Representation

Often the documents are represented in an algebraic model. The text documents are often represented as vectors of identifiers. Such identifiers can be index terms that can be used in various processes like retrieval, indexing, ranking on relevancy, etc. [5].

An array of words forms a document. Vocabulary or feature set is defined as group of words of a training group [6]. Python again provides support for such representation.

4.5 Feature Selection and Transformation

Feature selection is a process of diminishing the dimensionality of input dataset by removing the irrelevant data or by fetching useful information from the given data.

This helps in increasing the performance.

5 Classification Algorithms

There are several text classification algorithms available. Some of the algorithms are Naive Bayes, SVM, Winnow, TRIZ-based algorithm and KNN (K-nearest Neighbor). This paper attempts to consider Naive Bayes and SVM algorithms. It also performs the comparison of their results. An observation is also done by considering stop words, stemming, lemmatizing, tfidf, etc. The overall classification architecture can be depicted as shown in Fig. 3.

5.1 Naive Bayes Algorithm

Naïve Bayes algorithm has been a proven traditional classification algorithm and has been used since many years. Naïve Bayes classifiers are a group of classification algorithms that uses Bayes theorem. It is generally a collection of algorithms with a common principle. Naïve Bayes is consistently used in various text classification applications due to its simplicity and effectiveness [7].

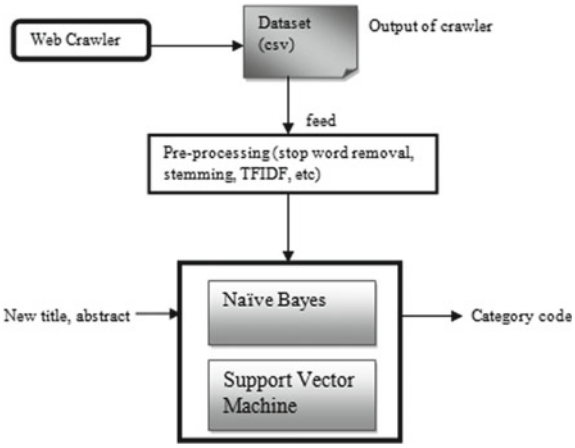


Fig. 3 Overall architecture

Here, an attempt is made to perform classification using Naïve Bayes for a dataset of 530 records in Python language. Python provides support for Naïve Bayes by offering certain packages and libraries. Following is the pseudo code:

```
Import packages
Read the dataset
Split the dataset – Testing and Training Dataset (Testing
is 20% of original Dataset)
Apply SnowBallStemmer with ignore_stopwords = true
Build the vectorizer for the training dataset
#Build the pipeline by passing vectorized dataset, Tfidf
#transformer, navie bayes classifier
Build the Pipeline (vectorizer, transformer, classfier)
Fit the input dataset and output features
Test the testing dataset against this final model
```

5.2 SVM

Support vector machine is categorized as supervised learning model. This algorithm uses kernel trick to transform the data and then considering these transformations as base, it distinguishes among the possible outputs. Technically, it produces a hyper plane which classifies the new dataset using the knowledge of the original dataset given [8]. Python again provides great support for implementing SVM. The pseudo code is as follows:

```
Import packages
Read the dataset
Split the dataset – Testing and Training Dataset (Testing
is 20% of original Dataset)
Apply SnowBallStemmer with ignore_stopwords = true
```

```
Build the vectorizer for the training dataset
#Build the pipeline by passing vectorized dataset, Tfidf
#transformer, clf svm classifier
Build the Pipeline (vectorizer, transformer, classfier)
Fit the input dataset and output features
Test the testing dataset against this final model
```

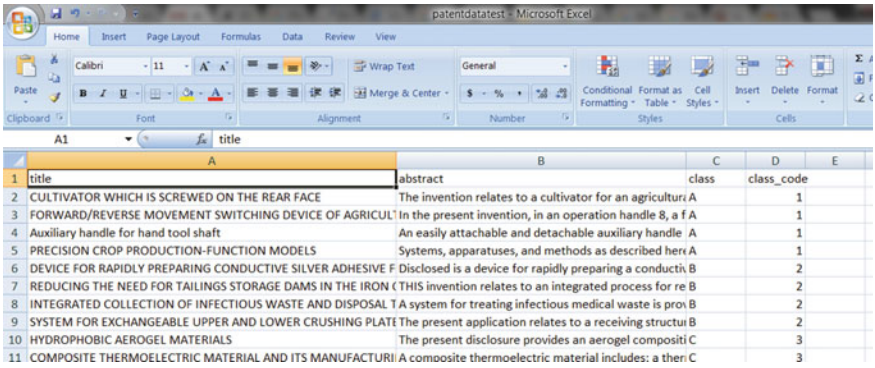
6 TFIDF

Term weighting is an useful concept in determining the accuracy of classification process. In a particular context, every word carries different level of importance. Such importance is denoted by term weight which is tagged with every word. This is called as term frequency (TF). There is another related phrase inverse document frequency (IDF) which is a weight that relies on the distribution of every word in the repository.

Python provides Tfidf transformer, which is responsible for calculating the tf-idf weights for our term frequency matrix.

7 Results

The classification of patents is achieved by using two algorithms—Naïve Bayes, SVM and the results of both are compared. SVM proves to be better. But the dataset upon which these algorithms are applied is obtained using the web crawler. The web crawler is written to crawl WIPO Web site and the resultant dataset is as shown in Fig. 4.



	A	B	C	D	E
1	title	abstract	class	class_code	
2	CULTIVATOR WHICH IS SCREWED ON THE REAR FACE	The invention relates to a cultivator for an agricultur	A	1	
3	FORWARD/REVERSE MOVEMENT SWITCHING DEVICE OF AGRICUL	In the present invention, in an operation handle 8, a f	A	1	
4	Auxiliary handle for hand tool shaft	An easily attachable and detachable auxiliary handle	A	1	
5	PRECISION CROP PRODUCTION-FUNCTION MODELS	Systems, apparatuses, and methods as described here	A	1	
6	DEVICE FOR RAPIDLY PREPARING CONDUCTIVE SILVER ADHESIVE F	Disclosed is a device for rapidly preparing a conducti	B	2	
7	REDUCING THE NEED FOR TAILINGS STORAGE DAMS IN THE IRON C	THIS invention relates to an integrated process for re	B	2	
8	INTEGRATED COLLECTION OF INFECTIOUS WASTE AND DISPOSAL T	A system for treating infectious medical waste is pro	B	2	
9	SYSTEM FOR EXCHANGEABLE UPPER AND LOWER CRUSHING PLATE	The present application relates to a receiving structu	B	2	
10	HYDROPHOBIC AEROGEL MATERIALS	The present disclosure provides an aerogel compositi	C	3	
11	COMPOSITE THERMOELECTRIC MATERIAL AND ITS MANUFACTURI	A composite thermoelectric material includes: a ther	C	3	

Fig. 4 Resultant dataset

Table 2 Performance of different classifiers

Classifiers	Accuracy (%)
Naive Bayes	59
SVM	74

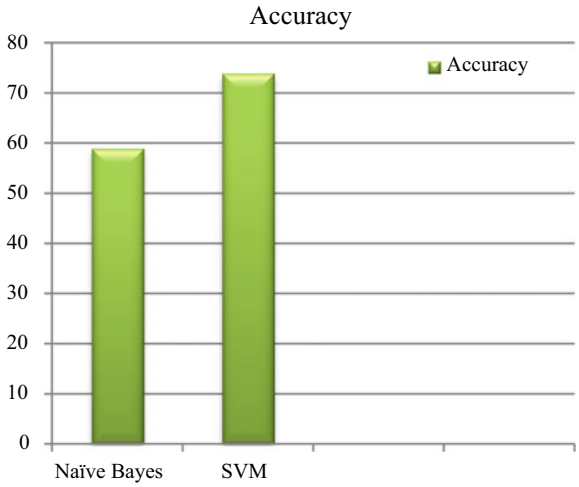


Fig. 5 Performance of dataset on different classification algorithms

Naïve Bayes algorithm gave an accuracy of 59% with the dataset having 530 records. And support vector machine presented an accuracy of 74% as shown in Table 2.

Figure 5 shows the performance of classifiers considering accuracy as parameter.

8 Conclusion

Patent classification task is a crucial task as it helps in management and further maintenance of patent documents. But because of drastic increase in number of patent documents every year, there emerges a need for a sophisticated system that automatically categorizes the patents. Some of the classification algorithms such as Naïve Bayes and SVM that are adopted in text classification and natural language processing were explored, implemented and compared for patent classification task. These classification algorithms are further tweaked or advanced by adding some pre-processing steps such as stop words removal, stemming, adding TF-IDF to provide better accuracy. Eventually, SVM proves to be better.

9 Future Enhancements

The classification of patents is done at higher category level that is section level. The hierarchy contains section at highest level followed by class level, subclass level and group level. One can perform patent classification further down the hierarchy. That is one can classify patents at class level, subclass level and group level also. As per IPC taxonomy, patent classification is hierarchical. There are eight sections at the higher level. Each section contains some classes, every class has subclasses and each subclass has some main groups that further have subgroups.

To obtain better accuracy one can additionally include complex attributes (as criteria) such as image depicted in front page of patent. The images can be transformed to appropriate models and the image classification process can be carried out. Generally, color, texture, shape, dimensions, etc., are fetched.

References can also be used in patent classification as one of the criteria to obtain better accurate results. The irrelevant character such as whitespace, tab space, punctuation marks and other special characters can be removed before performing classification.

Technically, one can combine several algorithms to obtain better accuracy. For example, one can create an ensemble of SVM and Naïve Bayes algorithms.

To perform efficient patent classification, neural networks can also be applied on larger dataset.

References

1. Zhang L, Li L, Li T (2014) Patent mining: a survey. ACM SIGKDD explorations newsletter archive, 16(2), Dec 2014
2. Abbas A (2013) A literature review on the state-of-the-art in patent analysis . Elsevier 37, Dec 2013
3. Sumathy KL, Chidambram M (2013) Text mining: concepts, applications, tools and issues—an overview. Int J Comput Appl 80(4), Oct 2013
4. Dasari DB, Rao VGK (2012) Text categorization and machine learning methods: current state of the art. Global J Comput Sci Technol Softw Data Eng 12(11) 1.0 Year 2012
5. Ikonomakis M, Kotsiantis S, Tampakas V (2005) Text classification using machine learning techniques 4(8) Aug 2005
6. https://towardsdatascience.com/document-feature-extraction-and-classification-53f0e813d2d3
7. https://en.wikipedia.org/wiki/Naive_Bayes_classifier
8. https://en.wikipedia.org/wiki/Support_vector_machine