On the Patent Claim Eligibility Prediction Using Text Mining Techniques

Chia-Yu Lai
National Sun Yat-sen University
Taiwan
chiayu06@mis.nsysu.edu.tw

San-Yih Hwang
National Sun Yat-sen University
Taiwan
syhwang@mis.nsysu.edu.tw

Chih-Ping Wei
National Taiwan University
Taiwan
cpwei@ntu.edu.tw

Abstract

With the widespread of computer software in recent decades, software patent has become controversial for the patent system. Of the many patentability requirements, patentable subject matter serves as a gatekeeping function to prevent a patent from preempting future innovation. Software patents may easily fall into the gray area of abstract ideas, whose allowance may hinder future innovation. However, without a clear definition of abstract ideas, determining the patent claim subject matter eligibility is a challenging task for examiners and applicants. In this research, in order to solve the software patent eligibility issues, we propose an effective model to determine patent claim eligibility by text-mining and machine learning techniques. Drawing upon USPTO issued guidelines, we identify 66 patent cases to design domain knowledge features, including abstractness features and distinguishable word features, as well as other textual features, to develop the claim eligibility prediction model. The experiment results show our proposed model reaches the accuracy of more than 80%, and domain knowledge features play a crucial role in our prediction model.

1. Introduction

In recent years, the development of technology has gone beyond the tangible devices and evolved into computer-implemented algorithms incorporated in an unprecedented pace. With the rapidly increasing number of software patent applications, computer-implemented software patentability has become the most controversial issue, urging legislatures to step in and define patentability [1]–[3].

Should software-based innovation be patented? Recently many law cases have been made public to demonstrate the patentability of software. Each validated patent must be novel, nonobvious and fully described. In addition, an invention must first possess statutory subject matter eligibility ("SME") under Section 101 before the evaluation of novelty, obviousness, and specificity [4]. Under the law of Section 101 of the Patent Act, a software patent must be considered a "new and useful process, machine, manufacture or composition of matter or any new and useful improvement". However, Section 101 defines eligible subject matter very broadly, merely including three common law exception to SME: abstract ideas, laws of nature, and natural phenomena. These exceptions are designed for the goal of preventing a patent from preempting future research and innovation. The limits on patent eligibility established through common law are to prevent those exceptions to hinder entire patent law. Nevertheless, an application that is "patent eligible" may not necessarily be directed to be "patentable". Still, the most challenge theme is the ambiguity in patent eligibility, especially in the definition of abstract ideas exception [5, 6].

The court decision in *Bilski v. Kappos* was the start of a discussion about the abstract ideas limitation on the patentable subject matter [5]. *Bilski* was a business method that describes how buyers and sellers of commodities in the energy market can hedge against risks for price changes. The court utilized the "machine-or-transformation" test to evaluate SME. Another is *Alice Corp. v. CLS Bank* case, where *Alice* was a computer-implemented method for mitigating settlement risk by employing a computer system as a third-party intermediary [1]. In *Alice Corp. v. CLS Bank* case, the United States Supreme Court developed a general patentability test process for determining whether the patent claims are directed to one of the patent-ineligible concepts namely, abstract idea, law of nature and natural phenomenon. The court in *Alice Corp. v. CLS Bank* case concluded that the use of third party intermediary is a "building block of the modern economy," therefore, an abstract idea [5], [7].

URI: http://hdl.handle.net/10125/49962
ISBN: 978-0-9981331-1-9

After *Alice Corp. v. CLS Bank* case, courts have invalidated many patents for computer-implemented patent applications by citing *Alice Corp. v. CLS Bank* case with the two-step test that would determine one of three longstanding judicial exceptions: abstract ideas. Those patents were invalidated because they are abstract ideas that only transferred a process conducted by a human into a software-implemented computer that is not enough to confer SME. Without giving a specific definition on the term of "abstract ", USPTO offered several guidelines and previous cases in response to decisions from the U.S. Supreme Court on claims reciting judicial exceptions to help examiners determine whether patent claims are drawn to abstract ideas [1]. Although it has been described by USPTO Guidance on what might direct to be an "abstract idea", applying such a definition to other software patents tends to be more challenging in patent systems [5].

To grant a patent, examiners must establish a balance between encouraging and rewarding innovation and preventing the patent from too broad to preempt future research and innovation. Preemption is the critical foundation of the patentable subject matter concern. For instance, In *Alice Corp. v. CLS Bank* case, the Court has repeatedly accentuated that the abstract idea exception covers the basic tools of scientific and technological work that tend to be build blocks of human ingenuity. Thereby, USPTO issued several Interim Guidance after *Alice* to provide a framework to address the ambiguity of the "abstract idea" exceptions to SME [4].

However, many questions and issues have been raised. For instance, the Court directly refused to define what constitutes an abstract idea. In addition, what are the requirements of the inventive concept to become significantly more than an abstract idea? The Court suggests the examiners to analogize the patent to those from previous cases. While SME has been a threshold inquiry, the subject matter of a patent is determined by the language of the claims [8].

In this study, we propose a framework to examine the patent eligibility based on evaluating patent claims. While prior studies in patent analysis research have proposed many methods based on patent textual documents [9]–[12], patent claim data has been ignored. To the best of our knowledge, applying text mining techniques to patent claim eligibility has not been studied.

This paper is organized as follows. Section 2 provides the relevant legal background for subject matter eligibility and patent examination process. Section 3 reviews the literature on patent analysis. We then illustrate our approach for each proposed model in Section 4. The preliminary experimental results with the empirical data in claim eligibility model is presented in Section 5. In the last section, we conclude our research and point out our future research directions.

2. Legal Background

In this section, we briefly introduce subject matter eligibility on software patent and the concept of "abstract" under U.S. patent Law.

Software is relatively a new subject within the framework of copyrights and patents. The first copyrighted software was granted in 1964 by the US Copyright Office. Besides, software patents have been a much shorter history than software copyright to be recognized as patentable to a limited extent since the U.S. Supreme Court's *Diamond vs. Diehr* decision in 1981 [13].

After *Alice Corp. v. CLS Bank* case, the Supreme Court has applied two-step patent eligibility analysis followed by *Mayo* for determining whether a patent claim is directed to patent-eligible subject matter. Therefore, USPTO started to update the Interim Guidance on Patent Subject Matter Eligibility. An analysis flowchart is provided by the Guidance under Section 101 to help examiners to clarify how to identify abstracts ideas by comparing claims to other examples. In the first step, the courts want to determine whether the claims were directed to the abstract idea. If so, then in the second step of the analysis, the courts should examine whether the claim contained any inventive concept which can add significantly more than an abstract to transform the claim into patentable subject matter.

The update of the Interim Guidance provides four categories to find a claim is directed to an abstract idea: (1) fundamental economic practices; (2) certain methods to organize human activity; (3) an idea "of itself" and (4) mathematical relationships or formulas. In addition, examiners are required to identify the abstract idea by reciting previous claims and have to explain the reasons that it corresponds to an identified abstract idea when rejecting a claim based on the abstract idea exception [2], [4].

3. Related Work

Patent information is regarded as a valuable database for discovering technology trends and establishing innovation strategies. In addition, patent documents are easy to acquire and fully open to the public. For instance, USPTO recently provides several bulk database download websites for public access and further research

(https://www.uspto.gov/learning-and-resources/bulk-data-products).

Recent research in patent documents has resulted in the development of various tools and techniques. The automated tools are established to explore the patent data through visualization, citation analysis, patent map analysis, through many techniques, including text-mining [11, 12].

Several previous studies are devoted to the clustering of patent documents for their quality evaluation. These text-mining techniques mainly extract textual features from the documents such as TF-IDF (term-frequency – inverse document frequency) and *n*-gram keyword extraction [14]. For instance, Tseng et al. [12] represented a series of text mining techniques employing the analytical process on keyword extraction and clustering analysis to create visualized patent map for further patent analysis. Some studies compared several keyword selection criteria by employing keyword frequencies in documents, variances of keyword frequencies across patent documents, and TF-IDF values [15, 16], while others explore the different parts of patents' textual documents and extract keywords, such as titles, abstracts, claims, and descriptions [17].

In addition, some studies have been conducted on evaluating the quality of patent and trying to improve it by the administrative process [18]–[21]. For instance, Rai [20] introduced a predictive modeling approach by text-mining and several machine learning techniques based on the various features extracted from patents to predict the patent legal validity and patent quality. Furthermore, Hido et al. [22] proposed a model computing the patentability score based on a set of feature variables including text contents of patent documents. Following this line of research, we adopt the TF-IDF and machine learning techniques on patent documents, especially on patent claims to build a binary classification model.

Patent claims refer to the scope of the protection sought in a patent application. Therefore, patent claims in many respects should be the most important part of the patent application because it is the claims that define the invention scope for which the Patent Office has granted protection. A patent contains at least one independent claim describing the essential features of an invention, potentially followed by several dependent claim elements covering additional details. Thereby, these claims can be vertically linked to each other based on the structure [24]. Lee et al. [9] proposed to apply semantic patent claim analysis to evaluate patent infringement risks for a more general model. Moreover, Hasan et al. [25] proposed a Claim Originality Analysis (COA) to build a patent ranking software, that the value of the patents by evaluating the important phrases that appear in the patent claims. In [26], they also examined several indicators including patent claim originality to predict patent quality. In our work, we try to apply extensive text-mining techniques on patent claims and claim constructions.

Despite the above-mentioned novel approaches, their analyses are based on very basic textual patent documents such as patent descriptions, abstract, patent titles, etc. In this study, building on the prior works we take into account patent claims and apply more comprehensive text-mining techniques, such as RST and text quality analysis, to derive more features in predicting claim eligibility. Additionally, our prediction model is based on the state-of-the-art gradient boosting model that achieves a higher accuracy than the traditional classifiers, such as Logistics Regression used in others papers.

To the best of our knowledge, our work is the first to predict patent subject matter eligibility, while employing various text-mining techniques to build up the claim eligibility predictive model.

4. Methodology

4.1. The research framework for patent claim eligibility predictive model

Figure 1 shows the research framework of our claim eligibility model, which consists of two main modules, namely model training and model prediction, as shown in Figure 1. The suggested approach employs various text-mining methods to extract features about patent claims and analyze them. The patent claims are used to construct a claim eligibility model, which can be used to predict the SME of a given claim.

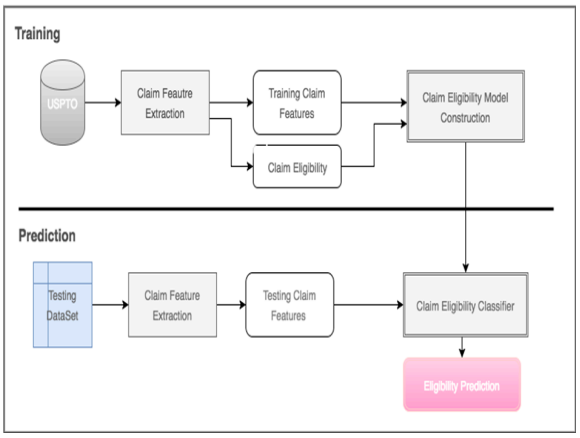

Figure 1. The research framework for Claim Eligibility model

4.2. Data Collection and Data Processing

In this paper, we scrape our data from the USPTO website, since it is most the representative patent databases. The database is well organized, providing historical data back to 1975 in electronic textual files. We collect both granted patent and application in full-text XML files and store into MySQL database.

After data collection, we employ various methods on each phrase to create sets of features of the predictive model.

4.3. Feature Extraction

Patent claims can be distinguished into dependent and independent claims based on their structures. Each patent must have at least one independent claim, followed by several dependent claims. An independent claim is a claim that defines an invention with all the necessary elements to be stand-alone. Drawing upon Interim Guidance, the released documents only examine independent claims to determine whether the claim is directed to patent subject matter eligibility. Thereby, in this research, we only examine independent claims and extract their textual features.

4.3.1. Baseline features. TF-IDF (Term Frequency – Inverse Document Frequency) is absolutely the most widely-used text feature extraction technique [27]. However, the computational cost increases linearly with the number of words used. Thus, in this paper, we select tf-idf features before building the model by using ExtraTreeClassifier, a python sklearn ensemble package, which utilizes randomized decision trees to select the best features [28].

4.3.2. Domain Knowledge Features. Since *Alice Corp. v. CLS Bank* case, USPTO released several interim guidance documents and worksheets for examiners to determine the patent claims eligibility. In this paper, we utilize the 66 cases based on USPTO official guidance document (July 2015 Update: Subject Matter Eligibility). This document provides further information regarding how examiners identify abstract ideas for each case.

The courts avoid giving a definition on abstract ideas, other than by cases. Therefore, examiners are trained to refer to these precedents to identify abstract ideas by way of comparison to those abstract concepts already identified. Accordingly, the 2015 updated guidance document provides further information about identifying abstract ideas drawing upon Supreme Court and Federal Circuit eligibility decisions with judicial descriptors. In this document, each case contains claims that are classified into four abstract idea categories, namely economic, human, idea, and math. Following each eligibility decision on abstract idea categories, we classify each claim based on 66 USPTO cases into abstract idea categories. Thereby, we design four abstractness features by comparing each claim with claims in the four classified categories. The value of an abstractness feature is the maximum cosine-similarity of each claim and the claims in the pertaining category. The algorithm for computing abstractness for each claim is shown in Figure 2.

```
Compute Claim Abstractness (s: An Independent claim):
{
  abstractness_similarity=[];
  For each category C in (economic, human, math, idea) do:
      Let Claims(C) be a set of claims associated with category C;
      Similarity(C) = Max_{c∈C}(Cosine(s, c))
          abstractness_similarity = [
                similarity ["economic"],
                similarity["human"],
                similarity ["idea"],
                similarity ["math"]
          ];
  return abstractness_similarity ;
}
```

Figure 2: Abstract Similarity Features

Besides, we employ information gain to identify the top 30 words in patent claims that are best in distinguishing eligible and ineligible claims, focusing on the verb and noun words. The occurrence of each word in the independent claim serves as the corresponding feature value. We adopt information gain as a measure for distinguishing eligible and ineligible claims. Let c_i denote the set of categories and information gain of each term *t* formula is defined by following expression [29]:

$$IG(t) = -\sum_i \Pr(c_i) \log \Pr(c_i) + \Pr(t) + \Pr(t) \sum_i \Pr(c_i \mid t) \log \Pr(c_i \mid t) + \Pr(\bar{t}) \sum_i \Pr(c_i \mid \bar{t}) \log \Pr(c_i \mid \bar{t})$$

It is widely used as a term goodness criterion in textual documents. Higher information gain indicates that a term is a better indicator to distinguish between eligible claims from ineligible claims. After computing all the similarities and information gain score, we extract 34 features to be our domain knowledge feature set (see Table1).

Table 1. Domain Knowledge Features

Feature Name	Data Type	Description
Abstract_economic	float	The similarity value with economic practice category of abstractness
Abstract_human	float	The similarity value with human activity category of abstractness
Abstract_idea	float	The similarity value with only idea itself category of abstractness
Abstract_math	float	The similarity value with mathematical formula category of abstractness
30 distinguishable Words	int	Frequency of top 30 words in information gain

4.3.3. Common Text Features Lastly, we examine the quality of patent claims by their readability index [30]. Several previous studies verify the quality of text based on the readability of the text, the reputation of the writers, and various content features based on the content terms [31]. To enhance our text quality analysis, we also consider readability. Readability of text measures how accessible the texts are. Existing research in many fields has proved that readability is a simple but very effective indicator about the writer's capabilities [32]. We consider the four popular readability indicators, namely ARI, FKGrade, CLIndex, GFog. For example, the Automated Readability Index (ARI) is used for estimating how easily the text can be read.

Moreover, we also examine the textual structure of the claims. Rhetorical Structure Theory (RST), which is proposed in [33], provides a precise framework that presents the tree-like discourse structure between text spans in a passage. RST defines 23 relations to describe the connection between two text spans, a meaningful piece in a passage, which can be a sentence or a clause. At least one of the two text spans are marked as "nucleus", which indicates that the text span holds the main idea, while the others are "satellite" that are considered less important. Take the relation "Elaboration" as an example, a two-clause structure is defined, and both of the elements participating in this relation must be nuclei to show the explainable situation between the two clauses [34].

We utilize an RST tree to represent the result of a patent claim parsed by RST parsers. Many researchers that incorporate RST in the domains of text mining are already published [35, 36]. An RST tree is composed of a set of clause nodes and relation nodes, which may consist of two child nodes or more. Thus, the nucleus-satellite indicator can be recorded and the type of RST relation of the node will also be specified. In the clause nodes, the text of the sub-sentence and the n-s indicator are recorded. RST defines 24 kinds of relations between text spans, which can be paragraphs, sentences, or clauses After parsing each independent claim by RST parser, we aggregate 7 relations as our features to examine the textual structure of the claims.

We select a state-of-art RST research, which RST parser can achieve the high accuracy on discourse labeling. Based on Feng et al. [37], this research provides detailed RST-style discourse parser written in python on their website. We can implement this software after setting up all requirements. However, we still need to modify it to fit our environment. After installing the RST programs, we start to construct an RST tree through RST parsers by inputting each patent independent claim text. Through the discourse parsing process, each claim is parsed to a complete structure of RST relations for constructing a tree. Afterwards, combined with four readability index features, we extract 16 features into our common text feature sets (see table 2).

Table 2. Common Text Features

Feature Name	Data Type	Description
readability_ari	float	Average automated readability index of the patent claims
readability_FKGrade	float	Average automated readability index of the patent claims
readability_CLIndex	float	Average automated readability index of the patent claims
readability_GFog	float	Average automated readability index of the patent claims
RST_Elaboration_N	int	Count numbers appear in RST Elaboration relation in Nucleus
RST_Elaboration_S	int	Count numbers appear in RST Elaboration relation in Satellite
RST_Attribution_N	int	Count numbers appear in RST Attribution relation in Nucleus
RST_Attribution_S	int	Count numbers appear in RST Attribution

		relation in Satellite
RST_same_unit_N	int	Count numbers appear in RST same_unit relation in Nucleus
RST_Joint_N	int	Count numbers appear in RST Joint relation in Nucleus
RST_Manner_Means_N	int	Count numbers appear in RST Manner_Means relation in Nucleus
RST_Manner_Means_S	int	Count numbers appear in RST Manner_Means relation in Satellite
RST_Enablement_N	int	Count numbers appear in RST Enablement relation in Nucleus
RST_Enablement_S	int	Count numbers appear in RST Enablement relation in Satellite
RST_Background_N	int	Count numbers appear in RST Background relation in Nucleus
RST_Background_S	int	Count numbers appear in RST Background relation in Satellite

4.4. Model building

To predict the eligibility of claims, we need to build the predictive model. In our work, a boosting machine learning method is applied to construct the prediction model. Boosting is a general approach that is used to improve the performance of any machine-learning models, including regression and classification algorithms. The basic idea of boosting is to gradually reduce the error in every iteration, and of the various morphs of boosting machines, the definition of error and ways to reduce it differ. For example, Adaboost fits an ensemble model in a forward stage-wise manner, which means in each iteration, the machine introduces a weak learner on the data and tries to label the misclassified data in the previous stage with correct class [38]. In Adaboost, the shortcomings to be minimized are identified by high-weight data points, which are the aforementioned misclassified data. The gradient boosting machine shares the same concept as Adaboost and correct the mistake throughout the iterations and further combines the gradient descent and boosting in order to minimize the error function by moving in the opposite direction of the gradient [39]. In our work, we use the eXtreme Gradient Boosting (XGBoost), which extends the gradient boosting and strengthen its ability in sampling and multithreaded process, as our boosting algorithm to rank the importance of our proposed feature sets [40].

5. Evaluation

5.1. Data Collection

We first collect 66 USPTO sample cases based on subject matter eligibility court decisions: judicial exceptions on abstract ideas, appearing in the 2014 Interim Eligibility Guidance. Then, we search for 51 software patent applications with appeal decisions to validate the patentability. Finally, with the growing number of court and the Patent Trial and Appeal Board (PTAB) decisions that declined to overturn patent claims, we identify 161 cases after Alice Corp. v. CLS Bank court decision. Thus, we have 278 patent cases that serve as our training dataset, shown in Table 3.

Table 3. Dataset Statistics

	# of Patent	# of granted	# of claims	# of eligible claims
USPTO cases	66	21	2,712	1,188
Patent Appeal	51	36	1,009	979
After Alice	161	151	5,305	4,733
Total	278	208	9,026	6,900

5.2. Experiment Design

We apply NLTK package[41] on the dataset to determine feature values of patent claims. First, we run the tokenize module of NLTK to segment the patent claim text into words. Next, the POS module is conducted to label the POS tag of each token. For example, after the POS module, each word is labeled by its POS, e.g., ('method', 'NN'), ('managing', 'VBG'), and ('consumption', 'NN'), where NN is a noun and VBG is a verb. We only lemmatize the tokens which are NN and VBG to return to the base form of the word for extracting 30 distinguishable words by information gain based on all 278 cases. We also adopt the stop-word removing module to remove the stop-words, punctuation, and numbers. After the processing, we determine the values for both baseline features (TF-IDF) and text-mining features, which are used to construct the prediction model.

5.3. Evaluation Result in Claim-eligible Model

We develop a claim prediction model using 1,079 independent claims, including 822 eligible claims and 257 ineligible claims. The balancing issue occurs when the is a large difference between the numbers

of samples in the different classes. While the imbalanced ratio is greater, the algorithm will favor the class with the larger number of samples, the majority class. Thus, during our experiment, we utilize the imbalance learning tools from Python imbalance-learn package [42]. This method can generate noisy samples by interpolating new sample between marginal outliers and inliers in which we balance the number of eligible and ineligible claims.

To build our baseline features, we must compare the TF-IDF selected words to be our baseline features. After optimal modeling and overfitting avoidance, we decide 60 words and 25 words to be our baseline feature sets. In this study, we compare the performance of the baseline classifier to our proposed model. First, we build the classifiers using baseline features which only using TF-IDF selected words, separated 60 words and 25 words. We trained multiple classifiers, Logistic Regression (LR), Random Forests (RF), Adaboost (ADA), Gradient Boosting (GRD), with 10-fold cross validation to compare their performance and the result is shown in Table 4 - 8.

To automatically determine our relative importance of all features listed in Table 1 & Table 2, We apply Xgboost package in Python to rank the importance of all features. Figure 3 shows the corresponding percentage of relative importance over the feature sets. We can identify the claims in abstractness of human activities contribute the most influence on the eligibility. Table 4 - 8 provide performance results for various classifiers with different feature set combinations. As can be seen, pure TF-IDF achieves the worst performance (Table 4). By adding domain knowledge feature set the performance increases across different classifiers (Table 5 and 6). However, incorporating common text features incurs little increase in performance (Table 7). This is because the common text features may have the similar effect as TF-IDF. For comparison purpose, when we incorporate all features, the resultant prediction model achieves the best performance (Table 8).

Our first observation is that for the four methods, much better performance was achieved when ensemble algorithms are applied, compared to using logistic regression and the conventional classification method. Secondly, when the domain knowledge features are incorporated, our accuracy achieved better performance compared with Common text features. It reveals that our domain knowledge approaches can help improve the performance. The recall values of our approaches using domain knowledge and abstractness features are 0.78 and 0.77 respectively, which are about 9% higher than baseline 0.69. In terms of precision, our approaches can obtain 0.91 and 0.90 by using knowledge and common features respectively, which are about 10% higher than baseline 0.81. It indicates that our proposed approaches are better to model TF-IDF for predicting patent claim eligibility.

Table 4. Performance of Claim Eligibility Model with only TF-IDF Features

Algorithm	Claim-eligible model by TF-IDF Features							
	TF-IDF Selected Words 60				TF-IDF Selected Words 25			
	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1
LR	86%	79%	56%	66%	83%	76%	46%	57%
RF	87%	73%	77%	75%	84%	73%	58%	64%
ADA	87%	72%	76%	74%	84%	72%	57%	64%
GRD	90%	81%	69%	75%	86%	80%	55%	65%

Table 5. Performance of Claim Eligibility Model with Abstractness Features

Algorithm	Claim-eligible model by Abstractness Features							
	TF-IDF Selected Words 60				TF-IDF Selected Words 25			
	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1
LR	85%	69%	66%	67%	81%	59%	64%	61%
RF	91%	86%	79%	82%	89%	76%	74%	76%
ADA	87%	87%	72%	79%	83%	66%	75%	70%
GRD	90%	86%	77%	81%	87%	74%	73%	73%

Table 6. Performance of Claim Eligibility Model with Domain Knowledge Features (Abstractness and top 30 Distinguishable Words Features)

Algorithm	Claim-eligible model by Domain Knowledge Features							
	TF-IDF Selected Words 60				TF-IDF Selected Words 25			
	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1
LR	84%	71%	75%	73%	81%	59%	69%	63%
RF	92%	87%	81%	84%	91%	80%	79%	80%
ADA	90%	79%	83%	81%	88%	73%	82%	77%
GRD	91%	87%	78%	83%	90%	77%	79%	78%

Table 7. Performance of Claim Eligibility Model with Common Text Features (RST and Readability)

Algorithm	Claim-eligible model by Common Text Features							
	TF-IDF Selected Words 60				TF-IDF Selected Words 25			
	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1
LR	80%	53%	63%	58%	82%	78%	47%	59%
RF	92%	91%	70%	79%	88%	82%	58%	68%
ADA	88%	76%	73%	74%	84%	73%	82%	77%
GRD	91%	90%	67%	76%	88%	77%	79%	78%

Table 8. Performance of Claim Eligibility Model with All Features

Algorithm	Claim-eligible model by All Features							
	TF-IDF Selected Words 60				TF-IDF Selected Words 25			
	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1
LR	84%	63%	74%	68%	80%	56%	75%	64%
RF	93%	94%	77%	84%	90%	88%	68%	77%
ADA	92%	84%	82%	83%	89%	73%	83%	78%
GRD	92%	91%	75%	82%	90%	77%	79%	78%

In addition, to automatically determine our relative importance of each feature listed in Table 1 & Table 2, we apply boosting models to prove the significance of our proposed features. Xgboost package [39] in Python can rank the importance of all features. Figure 3 shows the corresponding percentage of relative importance over the feature sets. We can identify the claims in abstractness and readability features contribute the most on the prediction of patent claim eligibility.

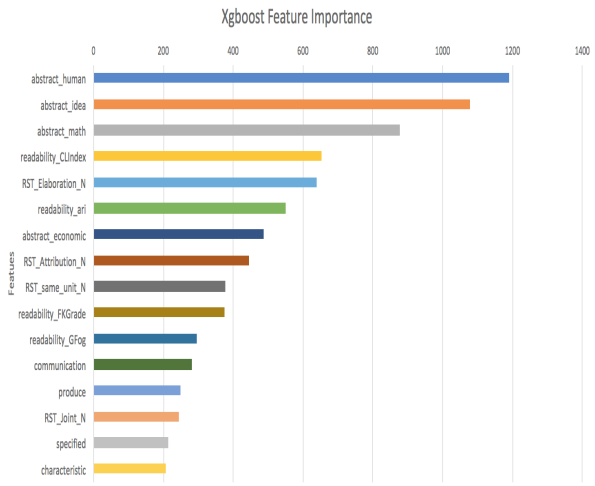

Figure 3. XgBoost Relative Importance of Features

6. Conclusion and Future Work

In this paper, we adopt advanced machine learning and text mining techniques to address a key problem in software patent subject matter eligibility (SME), namely providing abstractness measures of the patent claim. A major challenge is the limited availability of reliable labels for patent claim eligibility. We address this limitation by reviewing the USPTO released interim Guidance and court decisions, to confirm each claim eligibility from each patent. In addition, we extract a number of text-mining techniques to capture features via TF-IDF techniques. The output from abstractness, when combined with other text-based features, achieves accuracy score close to 0.90 for predicting patent claim eligibility. Our work is the first attempt to apply rigorous machine learning methods with text-based features to the problem of predicting patent claim eligibility.

This is an ongoing research and hence, has substantial room for improvement. Our study comes with some limitations. Most notably, our sample size based on USPTO cases and litigated cases are relatively small, resulting in potential overfitting of the results. We are able to remedy at least parts of this data limitations through utilizing unbalance learning and ensemble learning, which can be used when the number of variables is much larger than the number of observations. Second, a large share of patents are software patents and so potentially are not representative of all patent eligibility. Lastly, RST tools need to set up in a very complicated process thereby being difficult to replicate.

In summary, our results provide a promising step towards inferring the impacts of text-mining features on claim eligibility. In addition, our broad text-based feature sets could be applied to many other fields, not only on software. In the future, besides predicting claim eligibility, we hope to extend our model to predict patentability and patent value with more meta-data involved.

7. Reference

[1] P. Alice and C. L. S. Bank, "Exploring the Abstract : Patent Eligibility," vol. 2347, no. 2014, pp. 2015–2016, 2016.

[2] C. P. Moreno, "They Know It When They See It: Patentable Subject Matter after Alice," *Intellect. Prop. Technol. Law J.*, vol. 27, no. 1, p. 6, 2015.

[3] D. L. Burk, "Beyond abstraction: applying the brakes to runaway patent ineligibility," *J. Law Biosci.*, vol. 3, no. 3, pp. 697–703, 2016.

[4] J. P. Kesan and C. M. Hayes, "Patent Eligible Subject Matter After Alice," vol. 309, no. 112, pp. 1–23, 2016.

[5] S. Karakashian, "Software Patent War: The Effects of Patent Trolls on Startup Companies, Innovation, and Entrepreneurship, A," *Hast. Bus. Law J.*, vol. 11, pp. 119–156, 2015.

[6] A. L. Durham, "Two Models of Unpatentable Subject Matter," *31 St. Cl. High Tech. L.J. 251*, vol. 31, no. 2, 2015.

[7] A. Landers, “Patentable Subject Matter as Policy Driver,” no. December, 2015.

[8] B. D. R. Steinberg, T. E. Anderson, and M. H. Smith, “USPTO Issues Updated Guidance on Patent Subject Matter Eligibility,” vol. 27, no. 2, pp. 20–23, 2015.

[9] C. Lee, B. Song, and Y. Park, “How to assess patent infringement risks: a semantic patent claim analysis using dependency relationships.,” *Technol. Anal. Strateg. Manag.*, vol. 25, no. 1, pp. 23–38, 2013.

[10] H. Niemann, M. G. Moehrle, and J. Frischkorn, “Use of a new patent text-mining and visualization method for identifying patenting patterns over time: Concept, method and test application,” *Technol. Forecast. Soc. Change*, vol. 115, pp. 210–220, 2015.

[11] L. L. Zhang *et al.*, “A literature review on the state-of-the-art in patent analysis,” *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, vol. 37, no. 0, pp. 1–19, 2014.

[12] Y. H. Tseng, C. J. Lin, and Y. I. Lin, “Text mining techniques for patent analysis,” *Inf. Process. Manag.*, vol. 43, no. 5, pp. 1216–1247, 2007.

[13] A. Mossoff, “A Brief History of Software Patents (and Why They’re Valid),” 2014.

[14] R. Feldman and J. Sanger, “The text mining handbook: advanced approaches in analyzing unstructured data,” *Imagine*, vol. 34, p. 410, 2007.

[15] S. Lee, B. Yoon, and Y. Park, “An approach to discovering new technology opportunities: Keyword-based patent map approach,” *Technovation*, vol. 29, no. 6–7, pp. 481–497, 2009.

[16] H. Noh, Y. Jo, and S. Lee, “Keyword selection and processing strategy for applying text mining to patent analysis,” *Expert Syst. Appl.*, vol. 42, no. 9, pp. 4348–4360, 2015.

[17] Z. Xie and K. Miyazaki, “Evaluating the effectiveness of keyword search strategy for patent identification,” *World Pat. Inf.*, vol. 35, no. 1, pp. 20–30, 2013.

[18] S. R. Boalick, *Patent Quality and the Dedication Rule*, vol. 11, no. 2. 2004.

[19] C. J. Guerrini, “Defining patent quality,” *Fordham Law Rev.*, vol. 82, no. 6, pp. 3091–3143, 2014.

[20] A. K. Rai, “Improving (Software) Patent Quality Through the Administrative Process,” *Houst. Law Rev.*, vol. 51, no. 2, pp. 503–543, 2013.

[21] T. Watanabe, “Predictive modeling of patent quality by using text mining,” *Int. Assoc. Manag. Technol.*, no. 1, pp. 1–15, 2010.

[22] S. Hido *et al.*, “Modeling patent quality: A system for large-scale patentability analysis using text mining,” *Inf. Media Technol.*, vol. 7, no. 3, pp. 1180–1191, 2012.

[23] S. Graham, A. Marco, and R. Miller, “The USPTO Patent Examination Research Dataset : A Window on the Process of Patent Examination,” p. 117, 2015.

[24] M. Lupu, K. Mayer, J. Tait, and A. J. Trippe, *Current challenges in patent information retrieval*, vol. 29. Springer, 2011.

[25] M. Al Hasan, W. S. Spangler, T. D. Griffin, and a Alba, “{COA}: Finding Novel Patents through Text Analysis,” *Proceeedings 15th ACM SIGKDD Conf. Knowl. Discov. Data Min.*, pp. 1175–1184, 2009.

[26] Y. Liu, P. Hseuh, R. Lawrence, S. Meliksetian, C. Perlich, and A. Veen, “Latent graphical models for quantifying and predicting patent quality,” *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discov. data Min.*, pp. 1145–1153, 2011.

[27] S. Robertson, “Understanding Inverse Document Frequency : On theoretical arguments for IDF,” *J. Doc.*, vol. 60, no. 5, pp. 503–520, 2004.

[28] P. Geurts, D. Ernst, and L. Wehenkel, “Extremely randomized trees,” no. June 2005, 2006.

[29] J. O. Yang, Y., & Pedersen, “A comparative study on feature selection in text categorization,” *Icml*, vol. 97, pp. 412–420, 1997.

[30] Y. Liu, X. Huang, A. An, and X. Yu, “Modeling and predicting the helpfulness of online reviews,” *Proc. - IEEE Int. Conf. Data Mining, ICDM*, pp. 443–452, 2008.

[31] A. Ghose and P. G. Ipeirotis, “Estimating the Helpfulness and Economic Impact of Product Reviews,” *IEEE Trans. Knowl. Data Eng.*, pp. 1–15, 2010.

[32] Q. Gao and M. Lin, “Economic Value of Texts: Evidence from Online Debt Crowdfunding,” pp. 1–46, 2016.

[33] W. C. Mann and S. A. Thompson, “Rhetorical structure theory: Toward a functional theory of text organization,” *Text*, vol. 8, no. 3, pp. 243–281, 1988.

[34] A. Kumar and M. Stonebraker, “Semantics based transaction management techniques for replicated data,” *Proc. 1988 ACM SIGMOD Int. Conf. Manag. data - SIGMOD ’88*, vol. 17, no. 3, pp. 117–125, 1988.

[35] T. Hirao, M. Nishino, Y. Yoshida, J. Suzuki, N. Yasuda, and M. Nagata, “Summarizing a Document by Trimming the Discourse Tree,” *IEEE/ACM Trans. Speech Lang. Process.*, vol. 23, no. 11, pp. 2081–2092, 2015.

[36] L. Chengcheng, I. Engineering, and Li Chengcheng, “Automatic Text Summarization based on Rhetorical Structure Theory,” *2010 Int. Conf. Comput. Appl. Syst. Model. (ICCASM 2010)*, vol. 13, no. Iccasm, pp. V13-595-V13-598, 2010.

[37] V. W. Feng, Z. Lin, G. Hirst, and S. P. Holdings, “The Impact of Deep Hierarchical Discourse Structures in the Evaluation of Text Coherence.,” in *COLING*, 2014, pp. 940–949.

[38] Y. Freund and R. Schapire, “A desicion-theoretic generalization of on-line learning and an application to boosting,” *Comput. Learn. theory*, vol. 55, no. 1, pp. 119–139, 1995.

[39] J. H. Friedman, “Greedy function approximation: A gradient boosting machine,” *Ann. Stat.*, vol. 29, no. 5, pp. 1189–1232, 2001.

[40] T. Chen and C. Guestrin, “XGBoost : Reliable Large-scale Tree Boosting System,” *arXiv*, pp. 1–6, 2016.

[41] S. Bird, “NLTK Documentation,” 2017.

[42] G. Lemaˆ, F. Nogueira, W. S. West, O. Mv, and C. K. Aridas, “Imbalanced-learn : A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning,” vol. 18, pp. 1–5, 2017.