Manuscript

Classification of Alice Patents

Dominik Jurek

October 28, 2023

The 2014 *Alice Corp. v. CLS Bank International* Supreme Court decision* (*Alice* decision) weakened the enforceability of existing software patents and limited the patentability of new software-related innovations. I introduce a novel approach that directly uses the patent claim language of patent applications that were rejected due to the *Alice* decision to train a binary natural language processing (NLP) classification algorithm and identify claim language that is not eligible after *Alice* anymore. The advantage of this method is that I can predict for a large set of patents whether they are invalid and identify treated firms by measuring the share of invalid patents in patent portfolios before the *Alice* shock. This manuscripts provides the context for the GitHub project that implements the methods.

---

* 573 U.S. 208 (2014)

1. NLP methodology

I use natural language processing (NLP) to analyze the language of patents and define the intellectual property portfolio held by firms and their exposure to the *Alice* Supreme Court decision. I first develop and train a model to identify claim language that is invalid under *Alice*. Second, I classify existing patents and create annual patent portfolios using patent assignments to firms. Finally, I validate my classification using data on lawsuits and find that my NLP-based classification predicts litigated patents after *Alice* better than the patent classes alone.

1.1. Background

Software-related patents are a good starting point when trying to learn more about the effect of patentability on firms and innovation. Software concepts are difficult to describe in patent claims leading to uncertainty about the right scope of patent-eligible subject matter. On the one hand, the incentive to innovate needs to be preserved while avoiding on the other hand 'patent thickets' that stifle competition and limit innovation by other inventors (Stroud and Kim 2017). As a consequence, Supreme Court decisions led to several regime shifts over the decades on what is and is not patent-eligible.

Starting with the *State Street* decision[1] in 1998, a relatively broad interpretation of patent eligibility was adopted. In the following years, overly broad software patents were linked to the rise of patent trolls and nonpracticing entities; a type of patent holder that generates income by claiming license fees and litigating against genuine innovators (see, e.g. Appel et al. 2019, Cohen et al. 2019, Lee et al. 2019, Lemley and Feldman 2016). This broad patentability regime came to an end with the *Alice* decision: on June 19, 2014, the U.S. Supreme Court affirmed the judgment of a lower court that the patent claims directed at a method for mitigating settlement risk were ineligible because the claims just implemented an abstract idea (Ren and Duprez 2019). This decision, as many commentators were quick to point out, increased uncertainty around the patent eligibility of software and business method claims and limited their scope, reversing the prior broad patentability regime for software innovations (see, e.g. DiNizo 2018, Tran 2015, 2016, Daily 2017, Stroud and Kim 2017, Chien 2016, Craig 2017). Recent empirical research has confirmed the impact of *Alice* on patenting and litigation: Chien and Wu (2018), Kesan and Wang (2020), and Toole and Pairolero (2020) all find that rejection rates and application abandonments among software and business method application have spiked after *Alice*, with some rejection rates for business methods growing from 25% to 81% in the month after *Alice*.

Overall, the *Alice* decision is one of the most important recent regime shifts for the patentability of certain types of innovations. For individual firms, this means an exogenous shock to the ability to file for patent protection of new inventions and enforce existing software patents.

[1] *State St. Bank & Trust Co. v. Signature Fin. Group, Inc.*, 149 F.3d 1368 (Fed. Cir. 1998).

To use this regime shift to identify the causal effect of patentability, I need to classify patents based on their exposure to the *Alice* decision. This is challenging since not one particular type of patent or patent class was invalidated, rather a certain type of language describing innovations was ruled to be patent-ineligible. Prior research uses United States Patent Classification (USPC) patent class 705 to identify the most treated patents (see, e.g., Wagner and Cockburn 2010, Contigiani 2020). I can confirm that this class is indeed the most affected USPC class by *Alice*, but USPC classes are not useful going forward since they were discontinued by the United States Patent and Trademark Office (USPTO) in 2013. Using USPC classes thus does not allow us to validate the empirical approach and identify treated patents after the court decision. The Cooperative Patent Classification (CPC) system, which replaced the USPC system, has several drawbacks for treatment identification as well: there is no direct concordance between USPC and CPC, and CPC groups are much broader than USPC classes, thus covering more technologies than the patent-ineligible inventions after *Alice*. Another approach to identify treated patents would be to directly observe litigated patents after *Alice* (Galasso and Schankerman 2015). While this certainly allows us to define which patents are not eligible under *Alice*, only about 1.5% of patents are ever litigated (Lemley and Shapiro 2005).

To allow for a consistent measure of treated patents across the entire sample period of 2010 to 2019, I develop and train an NLP model to identify claim language that is invalid under *Alice* and classify existing patents.

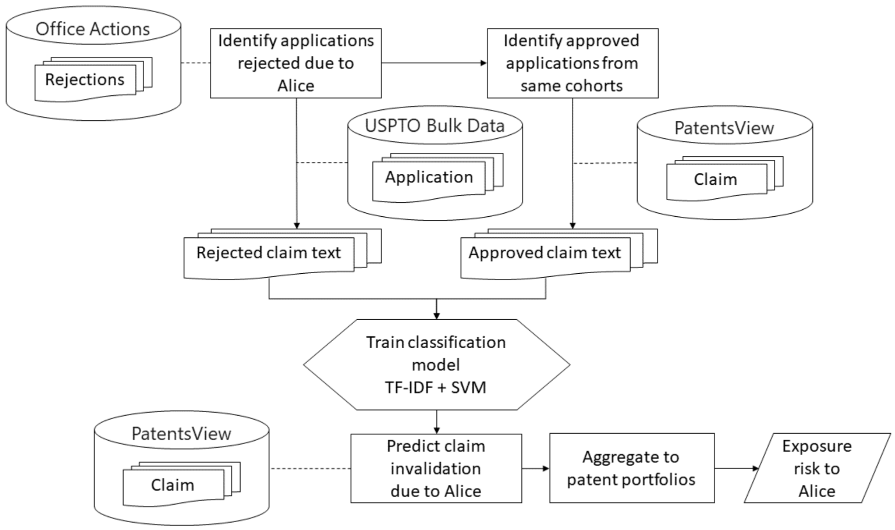
1.2. Natural language processing approach for identification

1.2.1. Overview I use rejected patent applications by the United States Patent and Trademark Office (USPTO) mentioning the *Alice* decision as reason to find examples of claim text formulations that are not eligible anymore after 2014. I select approved applications from the same filing years and patent classes as the rejected claims to train a binary classification model on the respective wording. I use this trained classification model on issued patents to predict their exposure to potential invalidation due to *Alice*. Finally, I can aggregate the patents that were assigned to firms to patent portfolios and define treatment as the share of invalid patents after *Alice*. Figure 1 illustrates the building and training of the model, the classification process of patent claims, and the definition of firm-wide *Alice* exposure in patent portfolios.

1.2.2. Data and training sets As a first step, I need to identify text examples of claims that are affected by the *Alice* decision. For this, I use rejected patent application data provided by the USTPO and build on the work by Lu et al. (2017).[2]

[2] The data are sourced from the Office of the Chief Economist (OCE), URL: `www.uspto.gov/learning-and-resources/electronic-data-products/office-action-research-dataset-patents`.

Figure 1 Process diagram patent claim classification.



In this paper, the authors use natural language processing tools on "office actions" to systematically extract and classify the reasons for any rejections, objections, or requirements. An office action is the written notification to the applicant of the examiner's decision on patentability. Since these office actions include references that the applicant may find useful for responding to the examiner and deciding whether to continue pursing the application, Lu et al. (2017) can identify rejections due to the Supreme Court decision in *Alice Corp. v. CLS Bank International*. The authors restrict to published applications, thus I can use the Patent Examination Research Dataset (PatEx)[3] and published applications[4] to extract the respective independent claim texts and additional application details such as filing date, patent class, and application status.[5]. PatentsView also provides structured pre-grant published claim data for applications. I use applications in the four USPC patent classes

[3] Available from USPTO, URL: `www.uspto.gov/learning-and-resources/electronic-data-products/patent-examination-research-dataset-public-pair`.

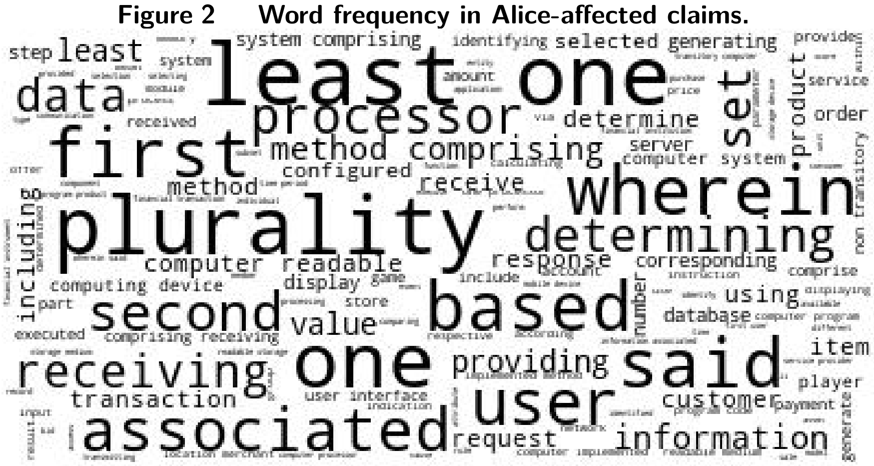
[4] Applications received since November 2000 are generally published within 18 months following the American Inventors Protection Act (AIPA) (Graham et al. 2015). XML files with weekly pre-grant publications of applications are available under `bulkdata.uspto.gov/`.

[5] I can find for more than 90% of all applications with *Alice* rejections the respective claim publications. If applicants only file in the U.S. and don't seek protection in other national jurisdictions, however, they can request non-publication of an application, which might introduce selection bias into my observable treated claims. Overall, only 3.4% of all *Alice*-treated applications do not have a recorded pre-grant publication in PatEx (making up around 36.9% of all missing observations). Furthermore, looking at the set of applications for which I am not able to find the published pre-grant publications more generally, I do not find evidence for any selection: the time between filing and recorded pre-grant publication is on average 317 days for the applications with texts and 284 days for the missing observations. The likelihoods of being ultimately granted are 57% and 50%, respectively. For the granted applications, the average time between filing and patent issuance is statistically not different between the two groups, with on average 1,519 days for the applications with texts and 1,474 days for the missing observations. Overall, the applications to which I cannot link claim texts are very similar to the applications with texts in terms of USPC classes, filing years, rejection rates, and processing time, suggesting that the missing claim publications are due to issues like typos and errors in the underlying XML files rather than applicants systematically deferring the publication of *Alice*-affected applications.

with the most rejections due to *Alice*, accounting for more than 91% of all *Alice*-based claim rejections in my sample.[6] Treated claims are those rejected claims that had no other issue identified in the office action and were either final rejections, were abandoned due to failure to respond to the office action, or were abandoned after the examiner's answer or board of appeals decision. In total, this results in 5,125 unique treated claim texts.

For the classification algorithm to be trained, I need to have a control group of claim texts that are valid under *Alice*. I select applications from the same USPC classes and the same filing years as the treated applications. I limit to applications that were granted without office action due to 35 U.S.C. §101, that is without issues referring to subject matter eligibility that could be related to the *Alice* decision.[7] I restrict to utility patents issued after the *Alice* decision in June 2014 and use the final claim text from PatentsView; I can be reasonably certain that these issued claim texts were deemed eligible under *Alice* by the patent examiners. To balance my training sets, I randomly draw independent claims as controls, proportional to the distribution of USPC classes and filing years as in my treated claims set.

Figure 2 Word frequency in Alice-affected claims.



Figures 2 and 3 show word clouds of the most frequent words in the treated and control claim corpora, with more frequent words being larger. The clouds look quite similar, which is to be expected

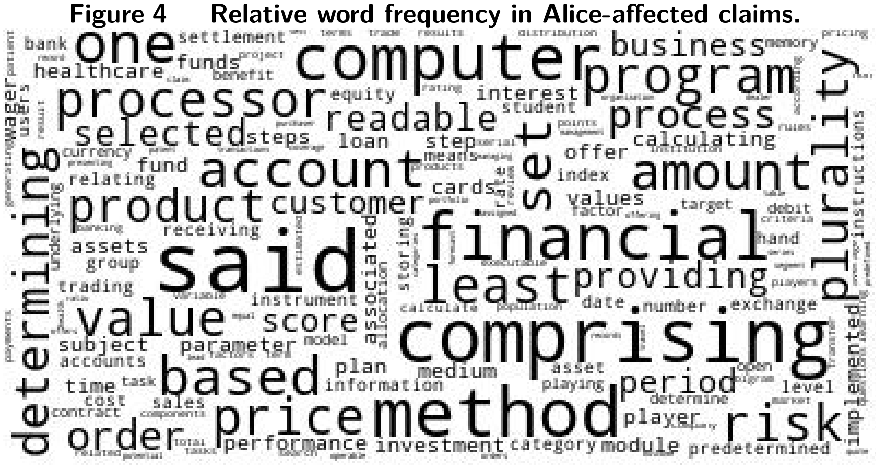
[6] Those classes are: 705 (Data processing: financial, business practice, management, or cost/price determination), 463 (Amusement devices: games), 702 (Data processing: measuring, calibrating, or testing), 434 (Education and demonstration); with the 705 class accounting for 82.4% of all rejected claims, followed by class 463 with 9.9%, class 702 with 4.6%, and class 434 with 3.1%, each class containing at least 100 rejected claims.

[7] Title 35 of the United States Code (U.S.C.), Section 101 governs which inventions are eligible for patents and reads: "Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title." The *Alice* decision significantly limited the scope of eligible subject matter under the law for certain software and business method inventions that were deemed patent ineligible as a mere computer-based implementation of abstract ideas.

Figure 3 Word frequency in control claims.

for claims from the same patent classes and the same cohorts. The terms are weighted by raw term frequencies without any pre-selection of terms, thus some of the most prominent terms in both groups are terms with little individual meaning such as ‘plurality’, ‘least’, ‘wherein’, reflecting the legal nature of the claim formulations.

Figure 4 Relative word frequency in Alice-affected claims.



Figures 4 and 5 show word clouds by weighting terms based on the relative frequency differences, i.e., how more frequent a word is in one group compared to the other. Words like ‘amount’, ‘financial’, ‘risk’, and ‘account’ are more prominent for *Alice*-affected claims, since the invalidated patents in the case were about an electronic escrow service for facilitating financial transactions. While the eligible claims contain more diverse words including specifications of physical implementations like ‘device’, ‘server’, ‘display’, *Alice*-affected claims prominently feature abstract descriptions about business processes such as ‘computer’, ‘determining’, ‘method’, and ‘program’. This is in line with the Supreme Court ruling stating that abstract claims are not eligible just by introducing a computer into the

Figure 5 Relative word frequency in control claims.

claim.[8] The wording of the treated and control claims thus have distinct characteristics which are not merely explained by patent class or filing cohort but by their exposure to the *Alice* decision.

1.2.3. Model building and evaluation I construct a binary classification model and train it on the rejected claims with the eligible claims as control set to classify patent claims based on their exposure to the *Alice* decision.[9]

Since most classification algorithms require numerical features as input, I vectorize claim texts into arrays representing the words of the claims. I split the text of the claims into separate words (tokens), and remove all numbers, special characters, HTML tags, punctuation, and words shorter than three characters. I also remove stop words, i.e., common words that occur with high frequency but carry little substantive information like 'the', 'and', 'like'. Finally, I convert the resulting strings to lowercase and apply the Porter stemming algorithm. Reducing words to their stems allows to identify families with similar meanings, e.g., the words 'playing', 'played', and 'play' all have the same stem 'plai'.[10]

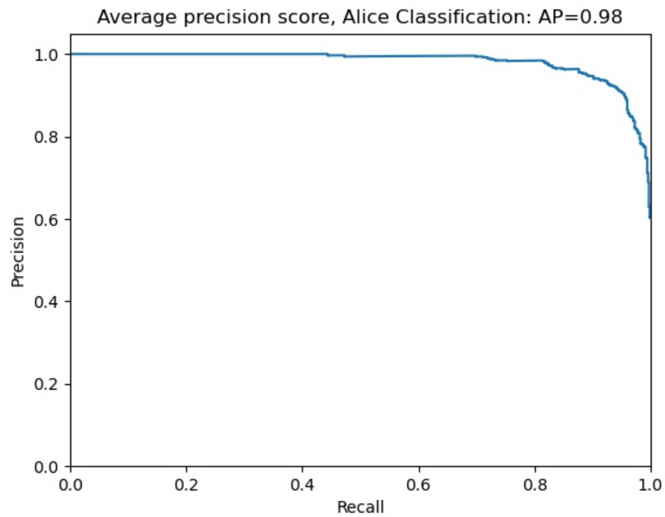
To turn the resulting arrays of tokens into a numeric vector representation, I count the occurrences of each token found in the corpus for each document. I use here uni- and bigrams, i.e., sequences of up to two tokens as features. A single claim is represented as a vector with each element representing the number of occurrences of a token, i.e., the term frequency. The collection of claims thus can be represented by a matrix with one row per claim and one column per token occurring in the corpus.

[8] See the Syllabus to *Alice Corporation v. CLS Bank International*, Supreme Court, No. 13–298.: " "Simply appending conventional steps, specified at a high level of generality," to a method already "well known in the art" is not "enough" to supply the " 'inventive concept' " needed to make this transformation [from an abstract idea into a patent-eligible invention]. [...]. The introduction of a computer into the claims does not alter the analysis. Neither stating an abstract idea "while adding the words 'apply it,' " ... is enough for patent eligibility."

[9] All calculations and executions were performed in Python 3.7.8 on a GNU/Linux operating system.

[10] All these pre-processing steps are performed using the 'gensim' library in Python 3 with the function 'preprocess_string'.

Figure 6 Precision-recall curve.



Since tokens that are very common among claims have little value in classifying them, I weigh the term frequency with the inverse document frequency, i.e., the frequency of the term in the overall corpus. The more common a term is across all the claims, the lower the weight of the respective column in the weighted term frequency matrix.[11] Overall, the texts of treated and control claims are represented as numeric vectors in a Term Frequency–Inverse Document Frequency (TF-IDF) matrix, giving more weight to the unique language of claims.

I train a support vector machine and construct a hyperplane to separate eligible from ineligible claims in the TF-IDF matrix. I use a quadratic kernel to solve for non-linear relations.[12] The full model thus takes as input the full text of a claim and gives a predicted probability between 0 and 1 of whether this claim was affected by the *Alice* decision. I use 85% of the training data for model building and the rest to evaluate the quality of the fit. Table 1 reports the performance of my classification model for this test dataset, figures 8, 6, and 7 give a graphical interpretation with the confusion matrix, precision-recall curve, and receiver operating characteristic (ROC) curve.

The confusion matrix plots the relative share of correctly predicted labels in the diagonal from the upper left to the bottom right corner. Related measures are precision, defined as the number of true positives over the number of true positives plus the number of false positives, and recall, defined as the number of true positives over the sum of true positives and false negatives. I want to achieve a

[11] The specific inverse document frequency weight for feature t is $idf(t) = \ln \frac{1+n}{1+df(t)} + 1$ with n being the total number of claims in the training set and $df(t)$ the number of claims in the corpus containing the term t. The weighted vectors are normalized to unit length.

[12] Using a linear or RBF kernel does not change the results. I use the ‘scikit-learn’ module in Python to implement the TF-IDF matrix and the support vector classification (SVC).

Figure 7 ROC curve.

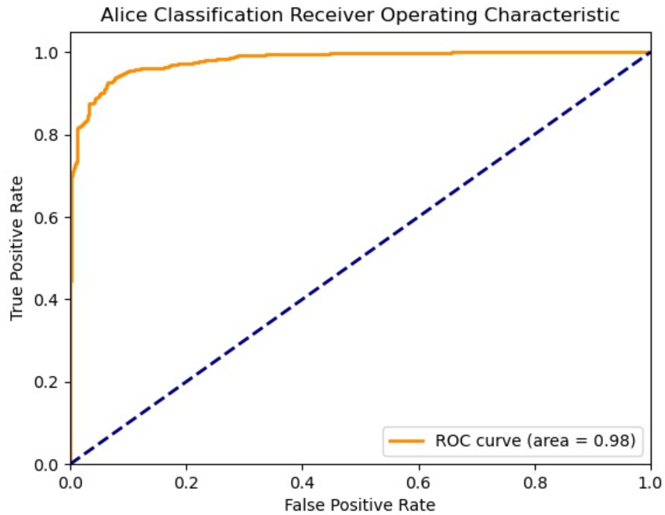
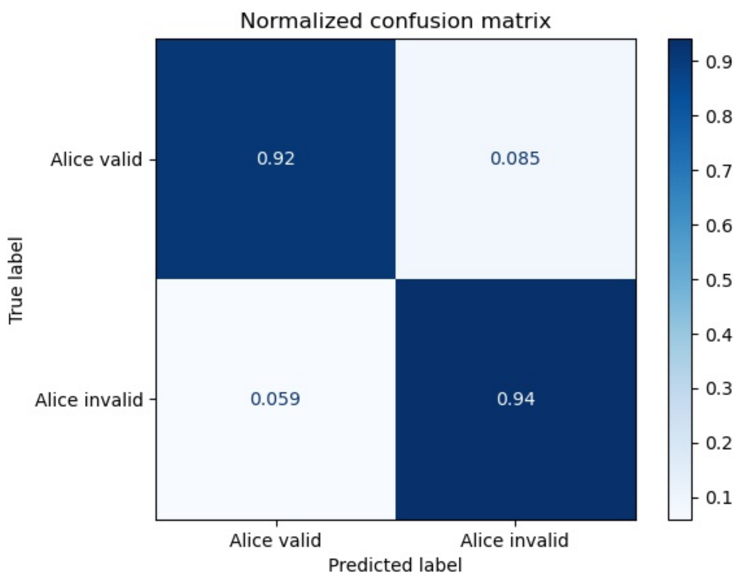


Figure 8 Confusion matrix.



high precision, i.e., my model correctly classifies claims as affected, and a high recall, i.e., my model finds most instances of *Alice*-affected claims.

There is a trade-off between these two measures since higher thresholds for being classified as affected return fewer results, but most of the predicted labels are correct (i.e., high precision). On the other hand, if the threshold is very low, the model returns many results, i.e., has a high recall, but most of its predicted labels are incorrect. The precision-recall curve plots this trade-off for different threshold levels with a larger area under the curve indicating a better classification model. The ROC curve similarly plots the fraction of true positives out of the invalid claims against the fraction of false positives out of the valid claims at various threshold settings. The 45-degree line in the ROC

Table 1 Classification report for prediction model.

	Precision	Recall	F1 Score	Support
Valid Claims	0.940	0.915	0.927	696
Invalid Claims	0.917	0.941	0.929	696
Precision Score	0.928			
Recall Score	0.928			
Accuracy Score	0.928			
F1 Score	0.928			
MCC	0.857			

Table 2 Comparison of different classification models.

Model Name	Precision Score	Recall Score	F1 Score	Accuracy Score	MCC
AdaBoost	0.724	0.739	0.731	0.728	0.457
Decision Tree	0.746	0.777	0.761	0.756	0.513
Gradient Boosting	0.767	0.792	0.779	0.776	0.552
K-nearest Neighbors	0.769	0.773	0.771	0.770	0.540
Logistic Regression	0.813	0.829	0.821	0.819	0.638
Naïve Bayes	0.802	0.819	0.810	0.808	0.617
Random Forest	0.876	0.905	0.890	0.889	0.778
Stochastic Gradient Descent	0.852	0.869	0.861	0.859	0.719
Support Vector Machine	0.893	0.935	0.914	0.912	0.824

plot would represent random guessing while a good model would be in the upper left corner with a high true positive rate and a low false positive rate.

All relevant performance measures are shown in the classification report in table 1. In the lower panel of the classification report, I also report the overall precision and recall scores as well as the accuracy score, which is the average of correctly classified labels. A common measure combining precision and recall is the F1 score, defined as $F1 = 2\frac{precision*recall}{precision+recall}$. Furthermore, the Matthews Correlation Coefficient (MCC) is frequently used to measure the quality of binary classification models.

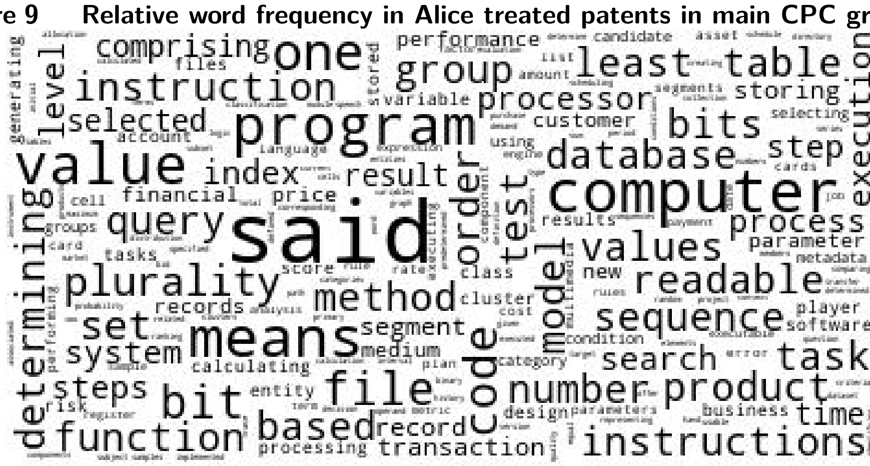
Overall, all measures show a very good classification model with high precision, high recall, and more than 85% MCC correlation, meaning a very high correlation between the true and predicted labels.

Finally, I use the different performance metrics to compare popular classification models on the test data in table 2 to validate my model choice. All models are used with their default parameters and with the same TF-IDF matrix of claim texts. Among all classifiers, the support vector machine proves to be the best.

1.3. Patent claim classification

Using the trained classification model, I predict the probability of being affected by *Alice* for independent patent claims for issued patents in the treated USPC classes. I do this for all patents issued since 1990 using the claims data from PatentsView.[13] This renders a total 393,100 classified claims in 115,630 unique patents. Since the USPTO moved to CPC classification in 2013, I repeat the classification process for the five CPC groups most consistent with the treated USPC classes.[14] Classifying patent claims based on CPC groups gives 3,236,420 classified independent claims in 983,229 patents. This is notably more than using the USPC classification due to the USPC classification ending in 2015 in PatentsView data and the affected CPC groups being much broader than the USPC classes.[15] This broader definition of treated patent classes under CPC can also be seen in figures 9, 10, 11, and 12 where I generate word clouds from the claim texts that are predicted to be treated and untreated under *Alice* for the CPC and USPC classes: the CPC group clouds are in general more diverse, though the overall image is similar with words like 'computer', 'means', 'value', 'device', 'signal', and 'display' showing up prominently in both classes and groups.[16]

Figure 9 Relative word frequency in Alice treated patents in main CPC groups.



To understand better which words are most important for the classification decision, I use a Local Interpretable Model-agnostic Explanations (LIME) algorithm on 1000 randomly selected patent claims. The basic idea behind the LIME algorithm is to perturb the underlying text by leaving out words and measuring how much and in which direction this changes the classification. Doing this gives for each tested text a list of words with the highest positive or negative impact on the classification. In table 3, I show the 20 most frequent words associated with a classification as eligible and ineligible.

[13] URL: `patentsview.org/download/claims`.

[14] There is no direct mapping from the USPC classes into CPC groups; however, the groups A63F (Video games), G07F (coin-freed or like apparatus), G06F (Digital data processing), H04L (Transmission of digital information), and G06Q (Data processing systems) represent around 50% of the affected patents in the treated main USPC classes 705, 463, 434, and 702.

[15] The four treated USPC classes account for a total of around 1.1% of all USPC class assignments, while the five corresponding CPC groups make up around 9.2% of all CPC assignments.

[16] I restrict to a sample of 1 million claims from the patents in the classified CPC groups to limit memory usage for the word cloud rendering.

Figure 10 **Relative word frequency in Alice treated patents in main USPC classes.**

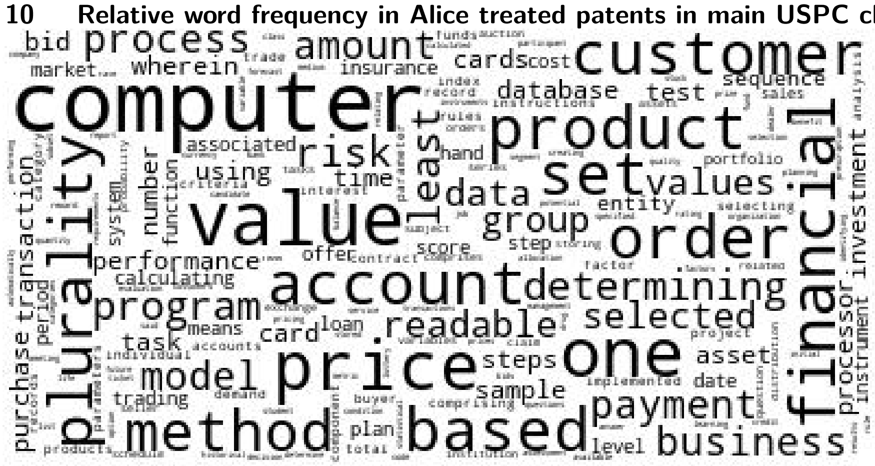


Figure 11 **Relative word frequency in untreated patents in main CPC groups.**

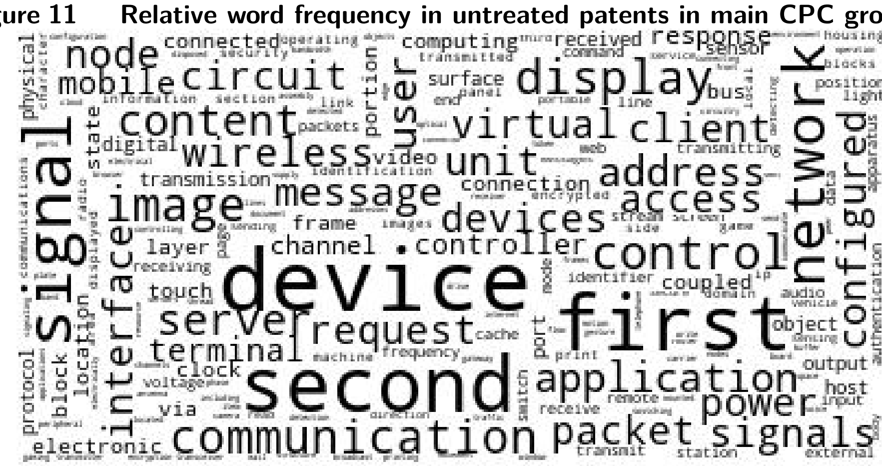
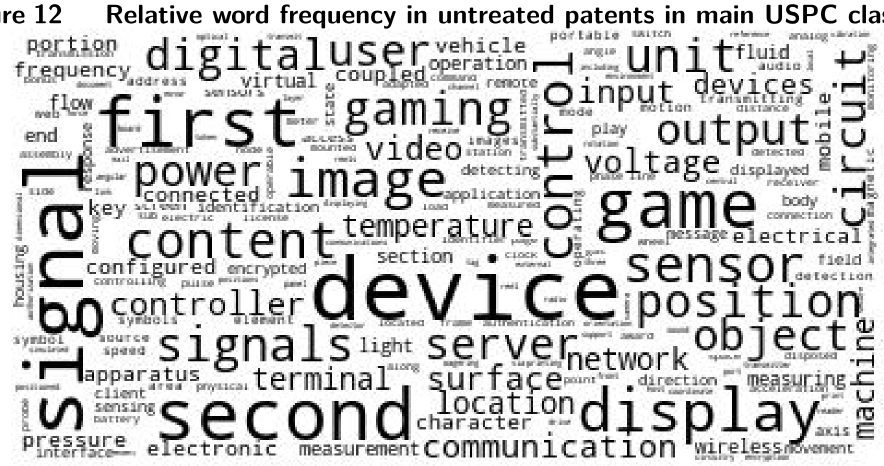


Figure 12 **Relative word frequency in untreated patents in main USPC classes.**



The general picture from the word clouds of figures 4 and 5 is confirmed: more precise wording with references to the actual implementation of claims improves the eligibility. More abstract and general formulations of methods and processes render claims ineligible. This is consistent with the findings of Dugan (2018) and the literature stating that more specific formulations in claims are eligible under

Alice compared to abstract ideas and processes (DiNizo 2018, Tran and Benevento 2019, Craig 2017). Notably, the differences between important words for patents in affected CPC groups and USPC classes are small, confirming that the classification model correctly predicts *Alice*-affected patents in the larger CPC groups.

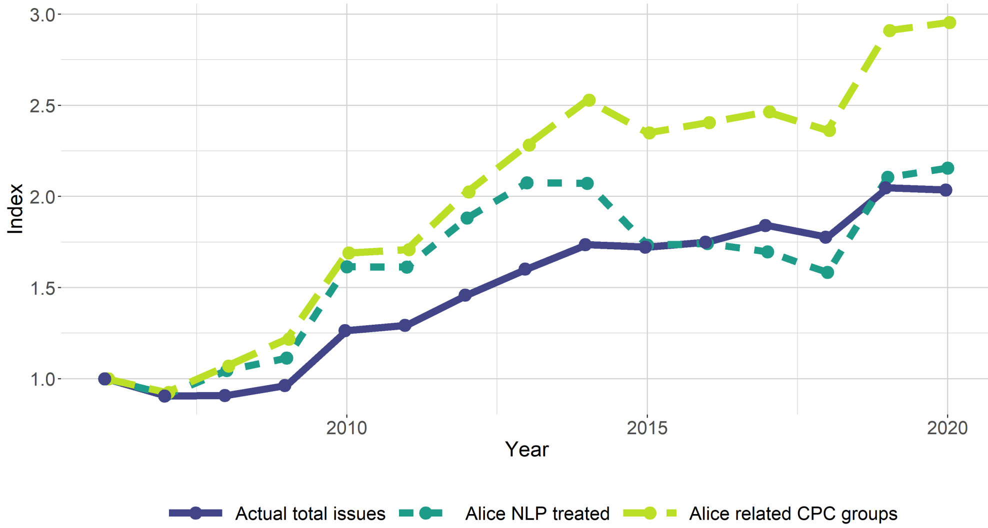
Table 3 LIME important words for classification in 1000 claims sample.

USPC classes		CPC groups	
Ineligible words	Eligible words	Ineligible words	Eligible words
said	second	said	second
means	device	method	device
method	data	means	data
value	signal	value	signal
steps	plurality	program	control
program	display	memory	plurality
player	game	comprising	network
function	control	steps	user
processor	signals	file	information
step	user	group	display
processing	sensor	module	request
condition	information	processing	signals
module	position	processor	application
transaction	server	code	content
comprising	image	level	server
reference	time	sequence	image
level	network	bit	circuit
number	electronic	bits	node
selected	application	input	address
test	circuit	logic	virtual

The words with the highest impact on the classification are ‘said’ for ineligible texts and ‘second’ for eligible texts. These words also show up in the respective word clouds in figures 4, 5, 10, 9, 12, and 11. ‘Said’ and ‘second’ are examples of sentence structures in eligible and ineligible claims that cannot be captured by, e.g., keyword search alone: while ‘said’ frequently refers back to a broader subject at the beginning of a claim, ‘second’ is common in sentences about how several components of an invention are interacting, thus giving a more technical description of the operation. My trained NLP model can uncover and leverage formulations in claim texts like these to more precisely predict whether a claim is eligible or ineligible under *Alice*.

Since most patents granted after 2013 have a CPC classification rather than a USPC classification, I focus on the CPC-based patent classification in the following. I define a patent as being treated if its first claim is predicted to be invalid under *Alice*. The first claim is the broadest claim of a patent and thus sets the scope for the technology protected by the patent (see, e.g., Kuhn and Thompson

Figure 13 Normalized patenting index 2006 to 2020.



Note. Count of issued utility patents by year, normalized to 2006 levels. The three patent indices are the actual count of all issued utility patents, the count of patents that are classified by the NLP algorithm as treated under *Alice*, and the count of issued patents in the five CPC groups closely related to *Alice* (A63F (Video games), G07F (Coin-freed or like apparatus), G06F (Digital data processing), H04L (Transmission of digital information), and G06Q (Data processing systems)). Patent data are sourced from PatentsView.

2019). Alternative specifications of treatment are as follows: a patent is treated if the majority of its independent claims are predicted to be *Alice*-treated, the average (continuous) predicted treatment probability for all independent claims is above 50%, or at least one independent claim is predicted to be *Alice*-treated. The different treatment definitions on the patent level are highly correlated with correlation coefficients between 71.1% and 88.0%.[17] The only effective difference is in how many patents are predicted to be treated: of the 1,062,883 classified patents, 18.5% are predicted to be treated using the first claim. Using the average treatment probability of claims leads to the fewest patents being predicted to be affected by *Alice* (14.4% of classified patents) and defining treatment if at least one claim is predicted to be ineligible to the most treated patents (24.9% of classified patents). For any of the definitions, the number of treated patents is large with more than 150,000 in each case. In practice, the different definitions of treated patents have little impact on the economic analysis, thus I use for the main analysis the treatment status of the first claim to define treatment on the patent level.

1.3.1. Graphic validation To validate my approach, figure 13 shows the indices of treated patent issuances defined by my NLP method (*Alice* NLP treated), patents in the CPC groups closely

[17] Patent classification with the main definition using the treatment status of the first claim has a correlation coefficient of 82.6% with the patent classification defining treatment if at least one claim is treated and 88.0% with the definition of treatment if most claims are treated, thus the first claim is representative for the treatment status of the rest of the patent.

related to *Alice* (*Alice*-related CPC groups), and all utility patent issuances in a given year (Actual total issues). Following 2014, there was a clear downward shock for both, the *Alice*-related CPC groups as well as my *Alice* NLP-treated patents, while actual total issues did not decrease. My NLP classification method identifies treated patents better than the affected CPC groups alone, which are broad and cover more technologies than are treated by *Alice*: the *Alice*-related CPC groups saw a decline in patent issuance from 2014 to 2015 of 7.1%, while *Alice* NLP-treated patent issuances declined by more than 16.3%.

1.4. Patent litigation - validation analysis

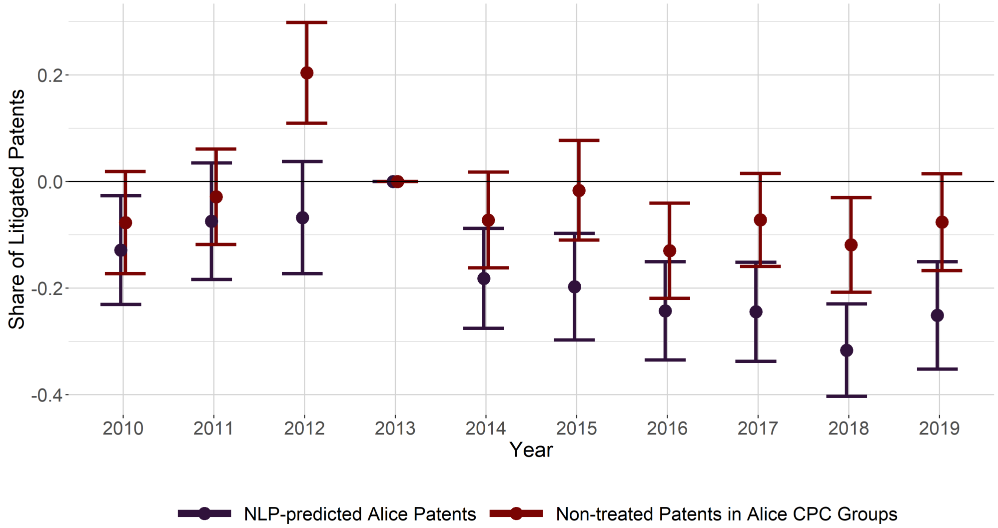
I use the Stanford NPE Litigation Dataset, based on Miller et al. (2018), to assess if my classification mechanism can predict real-world patent challenges. The Stanford NPE Litigation Dataset is the "first ever publicly available database to track comprehensively how practicing entities, non-practicing entities, and patent assertion entities (PAEs) claim patent ownership rights in litigation. [...] [R]esearchers are tracking every lawsuit filed in U.S. district courts from 2000 to the present and identifying each patent plaintiff as either a practicing entity or as one of eleven types of NPEs."[18] Since confounding legal changes around 2014/2015 have limited the ability for NPEs to file lawsuits (Appel et al. 2019), I focus on lawsuits with product companies as plaintiff. Product companies manufacture products, sell products, or deliver services generally, or are IP enforcement subsidiaries of practicing entities. This category account for around 52.4% of all patent litigation cases in the database. Lawsuits by product companies are also less likely to be 'opportunistic' (Cohen et al. 2019) and the patents defended are related to actual products and used technologies. Thus, if I find that patents predicted to be treated by the NLP method are less likely to be litigated by product companies after 2014, I can confirm that my algorithm identifies patents that are less enforceable after *Alice*.

To test this prediction, I distinguish in figure 14 between the share of treated patents among the litigated patents in cases (blue), and the share of patents in the affected CPC groups that are classified as not treated (red). Thus, I compare the fractions of treated and control patents within a patent infringement case filed by a product company. I regress each of these outcomes on year dummies for the filing year of the respective lawsuit and normalize the coefficients for 2013 to zero, the year immediately before *Alice*.[19] If my NLP method can identify ineligible patents better than the CPC group alone, I should see a distinct difference in the shares of litigated patents after 2014.

[18] URL: `law.stanford.edu/projects/stanford-npe-litigation-database/`

[19] For comparability, I scale both outcomes to zero mean and unit variance. All coefficients include error bars for the 95% confidence intervals.

Figure 14 Share of treated and non-treated patents in litigation by product companies.



While the share of non-treated patents in product company lawsuits remains relatively unchanged, we see a clear downward shift after 2014 for the share of treated patents (there is one outlier for non-treated patent shares in 2012, which is not part of a trend though). The magnitude of the shock is large; the share of treated patents before *Alice* was 4.82%, which falls to 2.49% after 2014, a drop of more than 48%. The share of non-treated patents was 16.9% and falls by 2.91 percentage points, a much smaller decrease of just 17.2%. Thus, my method can predict the enforceability of patents after *Alice*, even for patents filed within the same CPC groups. The prediction method is not perfect and there is a small treatment effect for patents that are predicted to be non-treated. However, the difference in magnitudes of the treatment effects is large, confirming that my NLP algorithm identifies the exposure to *Alice*.[20]

[20] A more detailed investigation of whether my model can predict the invalidation of patents in infringement cases is beyond the scope of this paper. To do this, case records would need to be searched in Public Access to Court Electronic Records (PACER) data and analyzed for rulings that invalidate patents and reference to the *Alice* decision. As I have shown above, greater exposure to *Alice* decreases the likelihood of being included in an infringement case since the court would likely strike down the patent as ineligible. Thus, patents that are invalidated in a court case might be less representative of affected patents in general.

References

Appel, I., J. Farre-Mensa, and E. Simintzi (2019, September). Patent trolls and startup employment. *Journal of Financial Economics 133*(3), 708–725.

Chien, C. V. (2016). Software Patents As A Currency, Not Tax, On Innovation. *Berkeley Technology Law Journal 31*(3), 1669–1723. Publisher: [University of California, Berkeley, University of California, Berkeley, School of Law].

Chien, C. V. and J. Y. Wu (2018, October). Decoding Patentable Subject Matter. SSRN Scholarly Paper ID 3267742, Social Science Research Network, Rochester, NY.

Cohen, L., U. G. Gurun, and S. D. Kominers (2019, September). Patent Trolls: Evidence from Targeted Firms. *Management Science*, mnsc.2018.3147.

Contigiani, A. (2020, April). Experimentation and Appropriability in Early-Stage Ventures: Evidence from the Software Industry. SSRN Scholarly Paper ID 3282261, Social Science Research Network, Rochester, NY.

Craig, J. A. (2017). Deconstructing Wonderland: Making Sense of Software Patents in a Post-Alice World Patent Law. *Berkeley Technology Law Journal 32*(Annual Review Issue), 359–378.

Daily, J. E. (2017). Alice's Aftermath: Changes in Patentee Behavior since Alice v. CLS Bank Bridging the Gap between the Federal Courts and the United States Patent & Trademark Office: The Journal of Science and Technology Law Symposium. *Boston University Journal of Science and Technology Law 23*(2), 284–303.

DiNizo, A. M. J. (2018). From Alice to Bob: The Patent Eligibility of Blockchain in a post CLS Bank World. *Case Western Reserve Journal of Law, Technology and the Internet 9*, 1–28.

Dugan, B. (2018). Mechanizing Alice: Automating the Subject Matter Eligibility Test of Alice v. CLS Bank. *University of Illinois Journal of Law, Technology & Policy 2018*(1), 33–80.

Galasso, A. and M. Schankerman (2015, February). Patents and Cumulative Innovation: Causal Evidence from the Courts *. *The Quarterly Journal of Economics 130*(1), 317–369.

Graham, S. J. H., A. C. Marco, and R. Miller (2015, November). The USPTO Patent Examination Research Dataset: A Window on the Process of Patent Examination. SSRN Scholarly Paper ID 2702637, Social Science Research Network, Rochester, NY.

Kesan, J. P. and R. Wang (2020, March). Eligible Subject Matter at the Patent Office: An Empirical Study of the Influence of Alice on Patent Examiners and Patent Applicants. SSRN Scholarly Paper ID 3556216, Social Science Research Network, Rochester, NY.

Kuhn, J. M. and N. C. Thompson (2019, January). How to Measure and Draw Causal Inferences with Patent Scope. *International Journal of the Economics of Business 26*(1), 5–38.

Lee, J.-S., N. Kim, and Z.-T. Bae (2019, December). The effects of patent litigation involving NPEs on firms' patent strategies. *Technological Forecasting and Social Change 149*, 119758.

Lemley, M. A. and R. Feldman (2016, May). Patent Licensing, Technology Transfer, and Innovation. *American Economic Review 106*(5), 188–192.

Lemley, M. A. and C. Shapiro (2005, June). Probabilistic Patents. *Journal of Economic Perspectives 19*(2), 75–98.

Lu, Q., A. Myers, and S. Beliveau (2017, November). USPTO Patent Prosecution Research Data: Unlocking Office Action Traits. SSRN Scholarly Paper ID 3024621, Social Science Research Network, Rochester, NY.

Miller, S. P., A. Aravind, B. Bengfort, C. De La Cerda, M. Dragoni, K. Gibson, A. Itai, C. Johnson, D. Kannappan, E. Kehoe, H. Kim, K. Mladinich, R. Pinho, J. Polansky, and B. Weissenberg (2018). Who's Suing Us? Decoding Patent Plaintiffs since 2000 with the Stanford NPE Litigation Dataset. *Stanford Technology Law Review 21*(2), 235–275.

Ren, X. and R. Duprez (2019, May). Financial impact of legal practice on software patent subject matter eligibility: Effectiveness of a workable standard. *The Journal of High Technology Management Research 30*(1), 50–58.

Stroud, J. and D. M. Kim (2017). Debugging Software Patents after Alice. *South Carolina Law Review 69*(1), 177–220.

Toole, A. A. and N. A. Pairolero (2020, April). Adjusting to Alice. Working Paper 3, United States Patent and Trademark Office.

Tran, J. L. (2015). Software Patents: A One-Year Review of Alice v. CLS Bank. *Journal of the Patent and Trademark Office Society 97*(3), 532–550.

Tran, J. L. (2016). Two Years after Alice v. CLS Bank. *Journal of the Patent and Trademark Office Society 98*(3), 354–373.

Tran, J. L. and J. S. Benevento (2019). Alice at Five. SSRN Scholarly Paper ID 3454922, Social Science Research Network, Rochester, NY.

Wagner, S. and I. Cockburn (2010, March). Patents and the survival of Internet-related IPOs. *Research Policy 39*(2), 214–228.