Cover Page: Patent Quest: An Unsupervised Learning Algorithm that Returns the Most Common Words Found in Recently Filed Patents and Abstracts (Group P4)

Ryan Mott, Benjamin Rogers, Ralph Keyser

rmmott, bsrogers, rkeyser

Team Member Contributions: It is the opinion of each team member that each of us contributed equally to various aspects of the project, which are laid out in detail below. Since the very beginning, the team members worked collaboratively through a Discord channel, and kept in constant communication with one another as they pushed forward to overcome difficult challenges presented in this project. There is a great deal of mutual respect shared amongst the team members and we have an excellent working relationship.

1. Data Cleaning
    a. Ryan Mott took responsibility for locating the relevant datasets hosted online and identifying the most important fields to search for and select for use in patent analytics; as well as taking responsibility for the creation of patent-specific stop words to be used for visualizations.
    b. Benjamin Rogers conceived of, designed and implemented the API that was used to pull our dataset from PatentsView.
    c. Ralph Keyser was responsible for conceiving of means to flatten the dataset and overcome data-specific conflicts and issues for means of processing.
2. Methods Development
    a. Ryan Mott took lead in design of the search query, similar patent algorithm, and Naive Bayesian classification algorithm using one hot encoding.
    b. Benjamin Rogers was responsible for the tokenization and lemmatization processing, as well as the TF-IDF classification algorithm.
    c. Ralph Keyser took primary responsibility for the creation of the WordCloud algorithm and discovering tools for tokenization and lemmatization.
3. Exploration
    a. Ryan Mott explored various ways to iterate machine learning through the large datasets; circulated analytics to the team; and reviewed literature concerning machine learning in a patent context, which is included in the presentation.
    b. Benjamin Rogers explored PatentQuest API's and available datasets for patents and patent abstracts.
    c. Ralph Keyser explored various methods for visualizing the data as well as word cloud manipulation and masking.
4. Results Analysis
    a. Ryan Mott prepared algorithms to test convergent accuracy of the classification tools, reviewed the results of patent similarity predictions, and curated industry-specific WordClouds for use in presentations.
    b. Benjamin Rogers took responsibility for TF-IDF analysis as well as prediction accuracy vs training data prevalence

   c. Ralph Keyser advised appropriate methods to use for accuracy measurements, considering efficiency and computational reasonability

5. Conclusion Drawing
   a. Ryan Mott worked to form reasoned conclusions based on proposed hypotheses in view of the results, discussed these conclusions with the team, and drafted sections of the report concerning them.
   b. Benjamin Rogers worked on conclusions related to data size and handling.
   c. Ralph Keyser interpreted the outcomes with the group, understanding the results and how the data should be presented meaningfully.
6. Presentation Preparation
   a. Ryan Mott prepared slides concerning experimentation, as well as compiled the other member’s audio into a video presentation.
   b. Benjamin Rogers prepared slides for data processing and retrieval
   c. Ralph Keyser prepared the initial draft of the presentation, including slides on background, approach, process, and WordCloud visualizations
7. Final Report Creation
   a. Ryan Mott drafted sections on background, methodology, and experimentation, as well as the Naive Bayes classifier.
   b. Benjamin Rogers was responsible for the TF-IDF-related aspects section and made contributions to concluding.
   c. Ralph Keyser began the drafting of the final report and was responsible for learning and implementing the document into LaTex.

Patent Quest: An Unsupervised Learning Algorithm that Returns the Most Common Words Found in Recently Filed Patents and Abstracts (Group P4)

Ben Rogers
Department of Computer Science
North Carolina State University
Raleigh, NC 27695
bsrogers@ncsu.edu

Ryan Mott, J.D.
Department of Nuclear Engineering
North Carolina State University
Bronxville, NY 10708
rmmott@ncsu.edu

Ralph Keyser
Department of Computer Science
North Carolina State University
Raleigh, NC 27695
rkeyser@ncsu.edu

Abstract

The United States Patent and Trademark Office ("USPTO") maintains the records of millions of issued patents in an online database and categorizes each patent by a number of attributes, including inventor; location; applicant; and technological class. We have devised a method of finding, in response to a given query, the most common keywords used in the abstracts of each patent, and generating a WordCloud-style visualization that can be used to quickly surmise the general subject matter of a group of patents. Using these most common keywords, we have developed two analytical tools: first, we developed a means to search a dataset for patents that concern similar subject matter, based on the similarity of the most common keywords; and second, we developed two machine learning models, Naive Bayesian and TF-IDF, which, using only the abstract's keywords, predicts which technological class ("CPC subclass") a given patent may fall under. Over 1000 trials, these predictive models have been shown effective, with the Naive Bayesian model having a 41% chance of correctly classifying a patent among nearly 700 possible CPC subclasses in the training set.

1 Background

From the Singer sewing machine (see U.S. Patent No. 8,294) to the Edison lightbulb (see U.S. Patent No. 223,898) to the Wright Brothers' aircraft (see U.S. Patent No. 821,393) , much of American history can be told through the stories of invention and innovation that propelled it forward into new eras. A window into these stories exists through the U.S. patent system, which publishes these inventions, and the best way of making and using them, in documents called "patents."

The United States Patent and Trademark Office ("USPTO") is in charge of issuing these patents and making them accessible to the public, which it has done perpetually since its inception in 1790[6]. The USPTO also assigns to each patent one or more technological classes, called "CPC Subclasses," in order to categorize inventions into their scientific field. As of 2023, there are now nearly 13 million

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

patents that have been published through the U.S. patent system, which remain freely accessible by all[7].

The patents themselves are written by the inventor (and usually, a patent attorney), and include, among other things, a title, an abstract, a brief description of the invention and a set of patent claims[3]. Patents are notoriously complex, and the exact nature of what they describe and claim is often subject to intense disagreement, leading to multi-year legal battles that can cost millions of dollars in legal fees[2].

But often, the exact nature of what a patent describes may be unimportant compared to the general subject matter that the patent represents. For example, a business may want to surveil the general progression of its competitor's research and development efforts, in order to ensure that it is pursuing similar research goals and market developments. Or perhaps, the business may want to see if the competitor is venturing into its own patents research efforts - and possibly infringing on patents of its own. As another example, a data scientist may want to collect information on general R&D trends in a given industry or region, and thus would care more about aggregated information about many patents than the exact meaning of a single patent.

Currently, there are many available means to analyze and visualize patents based on their known parameters. For example, resources such as PatentsView.com allow the public to look up patents based on a given inventor, and obtain information on what company that the inventor works for, or the technological field that is typically identified in the inventor's patents. Further, researchers have increasingly used machine learning and deep learning methods for patent analysis. Some have sought to use machine learning to identify emerging technologies[1], while others have sought to classify patents[4], or even draft new patents as an "AI patent lawyer"[10]. In general, the trend seems to be to use machine learning models to accomplish more and more complex patent-related tasks.

However, there continues to exist a need for simpler models, which can quickly and accurately compile information about the general subject matter of publicly available patents and patent applications. In this context, a machine learning analysis limited to the abstracts of patents – which themselves contain highly condensed, valuable information about the subject matter of the patents – is largely absent from the literature.

2 Method

2.1 Create a Data Mining Methodology Using the Words in Patent Abstracts

This project seeks to use machine learning methods for the purpose of gleaning information about the subject matter of patents by analyzing the text found in patents' abstracts.

Using only the words in the patent abstracts, we will enable allow a user to perform a basic search – by competitor, inventor, geographic region or technological class – within a specified date range, and then produce a word cloud visualization, showing the most common keywords used in the patents that are included in the search.

We will demonstrate the usefulness of this method by carrying out two further experiments: first, we will return a list of similar patents, judged by the presence of the same common keywords in their abstracts; and second, we will make a prediction as to the technological class of the patent, and report its accuracy. These metrics will demonstrate this method's utility as an efficient, simple method to quickly convey essential patent information, without committing large resources to analyzing the entire text of the patents as other machine learning methods have done.

2.2 Detailed Description of the Statistical Methods

The first step in the methodology is to prepare the WordCloud visualization and associated text information. This includes gathering the data from publicly-available datasets and using lemmatization, tokenization, and stop words to create a robust dataset upon which the experimentation can take place. As part of this process, a dictionary of "patent-specific" stop words is created by randomly querying the dataset and identifying patents that appear in abstracts regardless of search query, and combined with publicly available stopword sets such as that available from NLTK. The dataset comprises all words identified in responsive patent abstracts that meet the following criteria, along with their frequency. Thus,

$$tokenized\_abstract\_words, frequency = \sum_{i=1}^{i=100} [(lemmatized\_words - stop\_words) * frequency]_i \tag{1}$$

, where, i is a word's ranking in terms of frequency in the dataset.

The second step in the methodology is to identify patents that have abstracts that are similar to an identified patent or group of patents. Using the WordCloud text information found in the first step, the algorithm searches for all patents within the same date range and technological class or classes of a patent or patents. Each patent is then provided a similarity score, and ranked. The similarity score is calculated as follows:

$$Similarity\,Score = \sum_{i=1}^{i=100} \begin{cases} frequency_i, & \text{if } tokenized\_abstract\_words_i \, in \, patent \\ 0, & \text{else} \end{cases} \tag{2}$$

Thus, if a given word is found within a patent's abstract, then the patent's similarity score is increased by the frequency that the given word was found in the data that generated the WordCloud. Using this approach, if a term is more often used in the WordCloud, it will be weighed more heavily in the similarity score of a given patent, which is a more robust approach than simply weighing each word equally.

The third step in the methodology is to identify a patent's technological class using a Naive Bayes algorithm. For this, we use the methodology of step 1 to create WordClouds for each of the nearly 700 technological classes assigned by the patent office to newly issued patents from 2013 - 2022[1]. The text and frequency data underlying these WordClouds, along with the frequency that each technological class appeared in the dataset, is then used to create a training dataset. The training dataset is then one-hot encoded, wherein each unique word obtained for the 700 classes is provided a separate cell. Using this one-hot encoding matrix, which consists of 672 classes[2] and 7200 unique words, as well as frequency data for each class, a comprehensive Naive Bayes algorithm could be implemented more simply, and quickly, than using the tokenized data and frequencies of the initial training set.

The Naive Bayes algorithm itself proceeds as follows. Ultimately, we are seeking the probability that, given the presence of words in a patent's abstract, the patent will fall within a given class, or:

$$P(class \,|\, presence \, of \, words) \tag{3}$$

This is found by computing the probability of being in that particular class:

$$P(class) \;=\; \frac{frequency(class)}{frequency(all \, classes)} \tag{4}$$

Multiplied by the probability of having a word in the body if it is of a given class:

$$P(class \,|\, presence \, of \, word) \;=\; P(class) \;*\; P(presence \, of \, word \,|\, class) \tag{5}$$

When expanded to multiple words, i.e.,

$$P(class \,|\, presence \, of \, words) \tag{6}$$

The probability is found by multiplying the frequency of the class by the individual probabilities of the multiple words:

$$P(class \,|\, presence \, of \, words) \;=\; P(class) * \tag{7}$$

[1] Data from 2023 was reserved as a testing set.

[2] Some of the technological classes did not appear in issued patents in 2013 - 2022. Therefore, although there are 683 possible classes, not all appeared in the training set, and thus, were not included in the model

$$\prod_{i}^{n} P(presence\, of\, word_i \mid class)$$

The ultimate conclusion as to what class it falls in is:

$$argmax(P(class \mid presence\, of\, words)) \tag{8}$$

In the case that a word appears in the abstract of the patent-to-be-classified, and in the corpus of 7200 words, but not in a given class, we nevertheless set P(word|given class) to be slightly more than zero – specifically, one over the largest class obtained in the dataset – to avoid the so-called "zero frequency problem" in Naive Bayes algorithms:

$$P(word \mid class,\; word\, not\, present) \;=\; \frac{1}{max(frequency(class))} \tag{9}$$

In addition to Naive Bayes, we also experimented with term-frequency, inverse document frequency (TF-IDF). This metric can be used in place of frequency when predicting a patents class score with the following equations:

$$TF \;=\; \frac{number\, of\, times\, the\, term\, appears\, in\, the\, document}{total\, number\, of\, terms\, in\, the\, document} \tag{10}$$

$$IDF \;=\; log\left(\frac{number\, of\, documents\, in\, the\, corpus}{number\, of\, documents\, in\, the\, corpus\, containing\, the\, term}\right) \tag{11}$$

$$TF\, IDF \;=\; TF * IDF \tag{12}$$

With the above equations, corpus includes all patents within the training data set and document includes all patents for a given class within the training data. The 100 words with the highest TF-IDF score for each class are saved and joined with the highest TF-IDF scores for all other CPC scores and assigned a score of 0. Patent class is then predicted by summing the score of each classes TF-IDF score for each word in the given patents abstract and taking the class that has the highest cumulative score.

$$\begin{aligned} &Predicted\, Class_{patent} \;= \\ &max(\forall CPC\, Classes(similarity\, score \;=\; \sum_{word} TFIDF_{word}))\, where\, word \;\in\; patent\, abstract \end{aligned} \tag{13}$$

3 Plan and Experiment

Dataset: All Patent Abstracts from Patents Issued from 2013-2023

Although full text patents are available directly as bulk downloads from the United States Patent and Trademark Office, the sheer volume of the data – which includes every word written in every patent issued since 2002 – made it cumbersome to use. Each day, around 1,000 U.S. patents are issued, almost a patent a minute, with each patent typically having dozens of pages of dual-column text. Since our project is focused primarily on a small subset of this text (the abstract and the title), collecting the data in bulk form would have been unwieldy and enormously inefficient.

Fortunately, data from patents (since 1976) and patent applications (since 2001) are also available as part of a mySQL relational database hosted at PatentsView.org[8]. Using the available API query tools[9], one can pull a selection of fields that enables the ability to search, within a given data range, all patent abstracts based on the following fields: patent ID, title, inventor, company (assignee), technological class (CPC subclass), city, state, and country. Using this approach, one can retrieve a dataset containing the previous ten years of patent titles and abstracts - from 2013 to 2023 - and store that data in a manageable file size. The data can be separated by year into separate .csv files, which are capable of being loaded into pandas dataframes using the pandas read_csv() function.

Using our methodologies, we present the following hypotheses to confirm through experimentation on the dataset referenced above:

3.1 Experiment 1

Hypothesis 1. The text from patent abstracts can be queried according to specific needs, and can be processed to create WordCloud-style visualizations that provide useful information concerning the subject matter of the query.

To test this hypothesis, we can develop a search query to search the data and retrieve relevant patent abstracts, process those abstracts, and create WordClouds associated with well known companies to see if they convey useful information.

For this experiment, we developed the following functions:

- **queryPatents.py** searches the dataset for patents based on date range, inventor, assignee, location, or technological class
- **makeStopWords.py** makes a corpus of stop words to ignore when processing the data, which we generated by including standard stopwords provided by NSTK, as well as patent-specific stopwords we learned from conducting random samplings of the dataset and identifying common words that appear regardless of search query
- **makeSortedTokens.py** makes a lemmatized, tokenized list of the most common words of the abstract or abstracts found in the search query
- **makeWordCloud.py** makes a Word Cloud visualization based on the output from makeSortedTokens.py

3.2 Experiment 2

Hypothesis 2. The queried text from patent abstracts can be used as a basis of comparison to the abstracts of other patents, which can be ranked using a similarity metric and presented to the user.

To test this hypothesis, we can develop use the results of the functions from Experiment 1 to identify a set of tokenized words found in retrieved patent abstracts, and then, use those words to search for other patents.

For this experiment, we developed the following function:

- **findSimilarPatents.py** which uses the methodology described above to retrieve and rank patents by a similarity score based on the frequency of words present in both the patents and the retrieved patent abstracts

3.3 Experiment 3

Hypothesis 3. The queried text from patent abstracts can be used to identify patents as being part of a particular class, with an accuracy comparable to the results of other reduced-dimensionality classification regimes, such as that published by Kamateri, et al[5].

To test this hypothesis, we can use two separate algorithms, Naive Bayes and TF-IDF, to attempt to classify individual patents based on technological subclass; and then devise large-scale accuracy-checking algorithms to perform a defensible measure of accuracy.

For this experiment, we developed the following functions:

- **findNBSubclass.py**, which uses the Naive Bayesian methodology described above to predict a patent's class
- **tf-idf.py**, which uses the TF-IDF methodology described above to predict a patent's class

4 Results

4.1 Experiment 1. Word Cloud Visualizations

Below are pictured the four word clouds for the test companies, Cisco, Pfizer, Tesla and Hasbro, from the year 2019.

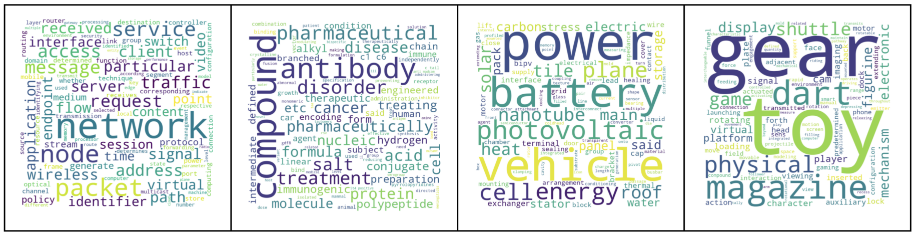
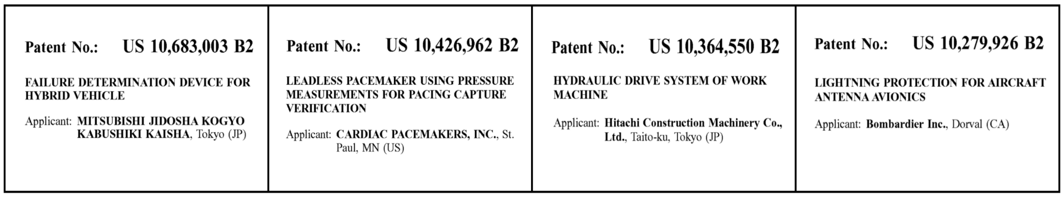


Figure 1: WordCloud representations of patents: Cisco, Pfizer, Tesla, Hasbro (L to R)

Each WordCloud contains a series of words that align with the types of innovation that one would expect from these companies, with larger words, such as “toy” in the Hasbro cloud, characterizing a defining part of the company’s business. Moreover, words that are common in most patents, but do not convey useful information - such as “invention,” “wherein,” “comprising,” “present” and “disclose” - are missing, indicating that the corpus of stop words successfully filtered the words out. It is apparent that the text from patent abstracts can indeed be queried according to specific needs, and can be processed to create intuitive and useful WordCloud-style visualizations.

4.2 Experiment 2. Retrieving Similar Patents

Below are pictured the most similar patents found for the test companies, Toyota, Medtronic, Caterpillar, and Boeing, in the years 2019-20.



Patent No.: US 10,683,003 B2

FAILURE DETERMINATION DEVICE FOR HYBRID VEHICLE

Applicant: **MITSUBISHI JIDOSHA KOGYO KABUSHIKI KAISHA**, Tokyo (JP)

Patent No.: US 10,426,962 B2

LEADLESS PACEMAKER USING PRESSURE MEASUREMENTS FOR PACING CAPTURE VERIFICATION

Applicant: **CARDIAC PACEMAKERS, INC.**, St. Paul, MN (US)

Patent No.: US 10,364,550 B2

HYDRAULIC DRIVE SYSTEM OF WORK MACHINE

Applicant: **Hitachi Construction Machinery Co., Ltd.**, Taito-ku, Tokyo (JP)

Patent No.: US 10,279,926 B2

LIGHTNING PROTECTION FOR AIRCRAFT ANTENNA AVIONICS

Applicant: **Bombardier Inc.**, Dorval (CA)

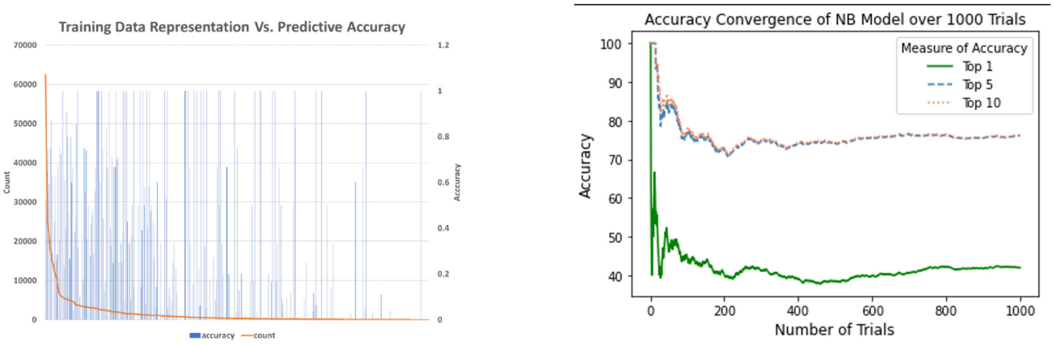
Figure 2: Patents most similar to company’s patent portfolio: Toyota, Medtronic, Caterpillar, and Boeing (L to R)

The retrieved patents each appear to originate from competitors of the test companies, and seem to concern core aspects of the test companies’ business model. For example, Toyota, as the largest distributor of hybrid vehicles in the world, holds a large portfolio of intellectual property relating to hybrid technology. The retrieved patent, a Mitsubishi patent relating to a diagnostic device for a hybrid vehicle. Similar results can be seen for electronic medical device supplier Medtronic, which returned a cardiac pacemaker patent; and for Caterpillar and Boeing, which retrieved patents made by Hitachi Construction, and Bombardier Inc., respectively. Thus, from a functionality testing perspective, it is possible to obtain facially similar patents looking at only the abstract, using the simple frequency-counting methodology articulated above.

4.3 Experiment 3. Predicting Technological Class

Pictured below are the results of the accuracy test over the course of 1000 trials for the Naive Bayesian algorithm. For each trial, a random patent was selected from a uniform distribution of all patent IDs issued in 2023, which comprised the testing set.

As shown, after 1000 runs, the Naive Bayes model converged to a single prediction accuracy of 0.41. This means the model’s top prediction was one of the patents actual assigned classes 41% of the



Accuracy per CPC, and Most Frequent CPCs

Accuracy Convergence, 1000 Predictions

Figure 3: Analytics from TF_IDF (L) and Naive Bayes (R)

time. Additionally, it had five class prediction accuracy of 0.76, and ten class prediction accuracy of 0.77, meaning that at least one of the top five predictions was in the patents actual assigned classes 76% and 77% of the time, respectively. A similar accuracy test was performed on the TF-IDF model. TF-IDF as a metric for predicting patent class had single prediction accuracy of 0.294, and a five and ten class prediction accuracy of 0.70. Additionally, it had five class prediction accuracy of 0.70 meaning at least one of the top five predictions was in the patents actual assigned classes 70% of the time. Models were validated using K-Fold validation, and were stable across all folds ranging from 0.28-0.31 for single prediction, and 0.68 to 0.73 for five class prediction. K-Folds were made by training on one year's data, and testing on all other years' data.

Considering there are 683 possible classes, this model performed significantly better than random guess which would have an expected accuracy of 0.0015. An interesting trend common to both models is the unexpected similarity between the five class and ten class prediction accuracy (0.76/0.77 for Naive Bayes, and 0.70/0.70 for TF-IDF). The similarity in accuracy suggests that there were very few instances where the models could identify the class correctly in the top ten, but not in the top five. In addition, it seems as though the models were better at predicting some classes than others. For example, in the TF-IDF classification model, one interesting thing to note is that at most 400 unique classes, or about 60% of classes were predicted using single class prediction. The left image of Figure 3 is a plot showing, for the TF-IDF model, the accuracy of the model with respect to some of the most frequent CPC classes. The blue vertical lines represent the model's accuracy for each class, while the orange line represents the number of times that each class was represented in the training set.

As shown by the higher concentration of blue lines to the left, the representation of the class in the dataset was relevant to whether a class would be predicted at all. However, it was not dispositive. Indeed, the model correctly predicted CPC subclasses with as low as 6 representative samples and had 0% accuracy for CPC subclasses with as high as 14,313 samples. In general, data did not appear to be indicative of predictive accuracy beyond there being 500 representative samples. Overall, it appears that the queried text from patent abstracts can be used to identify patents as being part of a particular class, with an accuracy that is somewhere around 30-40%, depending on the algorithm. Compared to existing models of patent classification, which show accuracies between 52% and 67%[5], the presented models are less accurate. It is important to note, however, that unlike the prediction methods described in the literature, the presented models are based on the patent abstracts alone, and are presented primarily to provide a verifiable metric of usefulness of the patent abstract in determining the content of the patent, rather than creating a robust method of patent classification. Still, it is clear that with some additional refinements – such as the ensemble methods discussed in Kamateri, et al. – there may be ways to improve the classification methodologies and produce more accurate predictions.

4.4 Conclusions

Overall, this project has demonstrated the feasibility and effectiveness of employing data mining techniques to extract valuable insights from millions of patent abstracts. By devising a methodology to systematically create WordClouds of patents or patent portfolio, we enabled a means to quickly compare large volumes of patent data. The usefulness of this data was confirmed by its ability to predict, using Naive Bayes and TF-IDF methods, a patent's given class with 30-40% accuracy. Most of the challenges involved in this process revolved around the quantity of data. Even limiting data to the patent abstracts from the past ten years, the sheer size of the patent dataset (3 million patents) tested the limit of physical memory for our computers, which required us to adopt some memory management and batch processing approaches that were novel to us.

While we are satisfied with the predictive accuracy with our classification models, we think there is room for improvement. One of the issues we encountered was that the vast majority of patents have multiple classes. Because of this, words are counted in multiple CPC subclasses causing some cross pollination of words in classes that they may not really apply to. Carefully pruning the training dataset to include only those patents which fall under a single class may result in higher testing accuracy, a subject which may be explored in future work.

5 Repository

Our project files can be found at: https://github.ncsu.edu/rmmott/-engr-ALDA-Fall2023-P4

References

[1] Patrick Thomas Anthony Breitzman. The emerging clusters model: A tool for identifying emerging technologies across multiple patent systems, Feb 2015.

[2] American Intellectual Property Law Association. AIPLA 2019 report of the economic survey, 2019.

[3] Machelen (BE); Peter Jan Leonard Mario Quaedflieg Waalre (NL) Bart Rudolf Romanie Kesteleyn Berlare (BE); Dominique Louis Nestor Surlereaux. Method for the preparation of hexahydro-furo-[2,3-b]furan-3-ol, U.S. Patent 7 126 015, Oct. 2006.

[4] Caitlin Cassidy. Parameter tuning naïve bayes for automatic patent classification, Jun 2020.

[5] Eleni Kamateri, Vasileios Stamatis, Konstantinos Diamantaras, and Michail Salampasis. Automated single-label patent classification using ensemble classifiers, 2022.

[6] The United States Patent and Trademark Office. Milestones in u.s. patenting, 2023.

[7] The United States Patent and Trademark Office. Update to the patent examination research dataset (patex) now available, Oct 2023.

[8] USPTO PatentsView. API endpoints.

[9] USPTO PatentsView. API query language.

[10] Ralf Krestel; Renukswamy Chikkamath; Christoph Hewel; Julian Risch. A survey on deep learning for patent analysis, Jun 2021.