Multilabel classification based on NLP-derived features: a case study on autonomous vehicle technology patents

Djavan De Clercq[a,c], Benjamin Tan[a], Ilias Atigui[b], Eliott Atlani[b]

a. Department of Industrial Engineering and Operations Research, University of California, Berkeley
b. Department of Civil and Environmental Engineering, University of California, Berkeley
c. School of Environment, Tsinghua University

Abstract

The objective of this study is to train a high-accuracy multilabel classification algorithm capable of classifying patents based on a cooperative patent classification (CPC) labelset. This is a departure from binary classification and multi-class classification. These algorithms were applied to a case study involving metadata and full summary text from 3,000 autonomous vehicle patents. The methodology involved the following steps: first, latent dirichlet allocation (LDA) was used to extract features for model training from patent texts. Secondly, two methods were used for the classification: (1) the random forest algorithm adaptation method, and (2) the problem transformation method based on adaptive boosting. The results of this study were promising. The best algorithm performance scores based on example-based metrics such as subset accuracy, precision, recall, F1-score, and hamming loss were 0.489, 0.847, 0.554, 0.653, and 0.074 respectively. In addition, the multilabel classifiers showed strong performance on label-based metrics including BAC, AUC, MMCE, FNR, and FPR. The results of this study has positive implications for automated patent classification in patent offices around the world, especially for technology patents in emerging technology fields.

Highlights

- Latent dirichlet allocation was applied to autonomous vehicle patent text for feature extraction.
- Multilabel classification machine learning was applied to predicting 10 CPC target classes.
- Classifiers demonstrated strong performance across both example-based and label-based metrics.
- Interactive visualization of results was demonstrated with an open-source tool using R.

1. Introduction

Multi-label classification is a powerful tool for classifying data that has multiple labels. In traditional classification machine learning problems, each observation is generally associated with a single label. Such approaches involve binary, single-class problems (where output values are either 0 or 1) or multi-class problems (where the output value is one of *n* classes, i.e. either "red", "blue", or "green"). In a multi-label classification problem, however, observations are associated with a set of labels $L = \{\lambda_i\}$; inputting unseen instances into a multi-label classifier outputs a subset of labels taken from L (Rokach et al., 2014). Multi-label classification has been applied to many practical domains such as protein classification, media annotation, bioinformatics, music processing, image recognition, and text (Abe, 2015; Charte et al., 2015; Luo et al., 2017; Ramírez-Corona et al., 2016; Tahir et al., 2012). However, these methods hae not yet been applied to patent classification problems. This study applies a novel combination of natural language processing and multi-label classification to classifying patent data into different technology fields.

Technology patents contain a wealth of information which can assist scientists, engineers, and corporate/political decision makers throughout the inventive process (Madani and Weber, 2016). According to the World Intellectual Property Organization, 90% to 95% of inventions can be found in patent documents (Souili et al., 2015a), making patent texts an important resource for understanding the evolution of technologies over time. However, parsing through patent text consumes significant time and resources.

When patents are submitted to the United States Patent and Trademark Office (USPTO), they are classified into technological fields based on the Cooperative Patent Classification (CPC) system, recently developed cooperatively by the European and US Patent Offices. The main purpose of CPC classification is "to facilitate the retrieval of technical subject matter. In order to reliably retrieve technical subject matter using the CPC classification system it is important for all technical subject matter to be consistently classified." (United States Patent and Trademark Office (USPTO), n.d.) In addition, CPC classes allow for targeted searches without the use of keywords; since companies often use very unspecific or purposefully obfuscated language, using keywords for searching patents may require unreasonably large time investments (Eisinger et al., 2013).

Assigning CPC classes requires professional patent examiners to classify documents manually. This task is complicated, since patents contain a lot of information: patent text data includes text from the patent's title, abstract, background/summary, detailed description, and claims. Patent metadata includes data from fields such as the applicant, inventor, assignee, date of issue, patent examiner, and so on. Patent examiners are required to assign multiple CPC labels to a patent in order to reflect the various technology fields that a patent may pertain to. This process of patent classification suffers from low efficiency and high expense (Zhang, 2014).

Given the potential difficulties in analyzing text to assign labels to newly filed patents, the objective of this research is to apply a combination of natural language processing (NLP) and multi-label classification tools to automatically assign patents to multiple CPC classes. To that end, this study first applied latent dirichlet allocation – a topic modeling approach from the field of NLP – in order to extract topic features from patent texts. Next, multi-label classification algorithms were trained on these topic features (and also patent metadata-derived features) to predict the CPC classes of patents in an emerging technology field: autonomous vehicles. Put concisely, the research question is: which model can most accurately predict the CPC labelset of patents in an emerging technology field?

The remainder of this paper is structured as follows. First, we provide a literature review of natural language processing and machine learning tools that have been applied to patent data. Next, we present a detailed

overview of the methodology used in this study. Lastly, we present the results and a discussion of our analysis, followed by concluding remarks.

2. Literature review

Recent studies have applied either NLP, machine learning, or a mixture both, to a range of problems related to the extraction of information from patents. However, most of these studies have focused on technology forecasting, single-label patent classification, or research-oriented information extraction.

2.1. Machine Learning & Natural Language Processing applied to patent data for technology forecasting

Numerous studies have focused on applying machine learning to patent data for technological forecasting. For instance, Lee et al. (2018) employed feed-forward multilayer neural networks to assess the value of patents and build an indicator system to evaluate a technology's "emergingness" over time. Kyebambe et al. (2017) proposed an algorithm capable of clustering similar technologies based on patent feature vectors and predicting emerging technologies at least a year before they emerge. They applied classification algorithms based on naïve bayes, artificial neural networks, support vector machine and random forest in order to label a patent as "emerging" or "non-emerging". Suominen et al. (2017) applied latent dirichlet allocation, a topic modeling approach, to identifying the topical and temporal dynamics in telecommunications patents. Joung and Kim (2017) used a "technical keyword" approach based on term frequency-inverse document frequency to identify clusters of emerging technologies. Song et al. (2018) applied sentiment analysis to identify emerging technologies based on automobile industry patent data.

2.2. *Machine Learning & Natural Language Processing applied to patent classification*

In addition to studies on technology forecasting, other studies have sought to classify patents into variously defined buckets. However, to the authors' knowledge, none of these studies have applied multi-label classification techniques combined with NLP-based feature engineering. Zhang (2014) proposed an interactive patent classification algorithm based on multi-classifier fusion and active learning, and achieved F1 scores ranging from 0.737 to 0.842. This study, however, did not focus on CPC classes. Wu et al., (2016) developed an automatic patent quality analysis system capable of clustering previously published patents into different "quality groups". That study applied kernel principal component analysis and support vector machine to patent data on thin film solar cells. Venugopalan and Rai (2015) built a classifier based on document-term frequency and topic modelling in order to categorize 10,201 patents about solar photovoltaics by technology area. Their analysis achieved relatively high accuracy (87.7%) with simple unigram topic models.

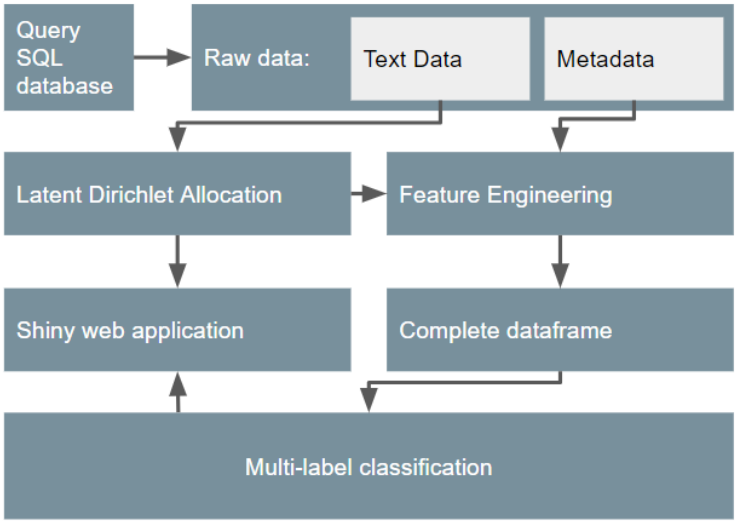
2.3. Studies focused on information extraction from patent data for innovation researchers

Patent analysts must regularly examine a large amount of patent data, and analytical aids that facilitate the patent analysis workflow are in high demand. Souili et al. (2015a) proposed an NLP-based method that provides design engineers with an efficient tool that re-structures a patent corpus in a problem graph to better highlight state of the art of given field of knowledge. Souili et al. (2015b) used NLP to provide automatic extraction of inventive design method (IDM) data from patent documents. Codina-Filbà et al. (2017) proposed a patent summarization technique that takes the idiosyncrasies of the patent genre (such as the unbalanced distribution of the content across the different sections of a patent, excessive length of the sentences in the claims, abstract vocabulary, etc.) into account to obtain a comprehensive summary of the invention. Cao et al. (2016) used knowledge mining tools to extract key methods used in sustainable design from patent data, with the objective of providing technical support for transforming traditional products into sustainable products.

3. Methodology

This study was based on the methodology shown in figure 1. Detailed descriptions of each step are provided in the subsections below.

Figure 1: Overview of methodology



3.1. Data Query

Autonomous vehicle patent data was retrieved via SQL regular expression queries with more than 100 synonyms of the phrase "autonomous vehicle". Data was retrieved from SQL databases hosted on Google Cloud, and queried using Google BigQuery. The first SQL database was the one put together by Balsmeier et al. (2016), which contains full titles, abstracts, and other metadata for 6.5 million patents. The second is the PatentsView SQL database hosted on Google BigQuery which contains the full texts of 5.8 million patents. The patents retrieved by the search query were further verified for relevance manually. The full data can be found in the supplementary information.

3.2. Latent Dirichlet Allocation

In order to extract additional features from the patent texts, this study applied Latent Dirichlet Allocation (LDA), a topic modeling algorithm. LDA is a generative probabilistic model for collections of discrete data such as text corpora, first described in Blei et al. (2003); in the original model, LDA is used to model a collection of unlabeled documents as a mixture of topics, where each topic is a distribution over fixed terms. LDA has emerged as one of the most popular unsupervised learning models for document and word clustering (Momtazi, 2018).

In the LDA model, we take a $M * V$ co-occurrence table to indicate our patent corpus, where M is the number of documents and V denotes the size of the vocabulary. This table contains the frequency of occurrences $n(\boldsymbol{w}_i, \boldsymbol{d}_j)$ for word $\boldsymbol{w}_i$ in document $\boldsymbol{d}_j$. LDA assumes that this corpus contains K latent hidden topics $(z_1, z_2, \dots, z_k)$, and that documents in the corpus are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words (Blei et al., 2003; Li et al., 2018).

Given the parameter α, the probability density can be expressed as (Li et al., 2018):

$$p(\theta|\alpha) = \frac{\Gamma(\sum_{i=1}^{K} \alpha_i)}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \theta_1^{\alpha_1 - 1} \dots \theta_K^{\alpha_K - 1}, \quad (1)$$

where $\Gamma(\cdot)$ is the gamma function and θ is the topic mixture. The joint distribution of θ, topics z, and words w for the given parameters α and β is (Blei et al., 2003):

$$p(\theta, z, w|\alpha, \beta) = p(\theta|\alpha) \prod_{n=1}^{N} p(z_n|\theta) p(w_n|z_n, \beta) \quad (2)$$

where N is the number of topics. A document's marginal distribution is obtained by:

$$p(w|\alpha, \beta) = \int p(\theta|\alpha) \left(\prod_{n=1}^{N} \sum_{z_n} p(z_n|\theta) p(w_n|z_n, \beta) \right) d\theta \quad (3)$$

Lastly, to estimate the topic distribution z for a given document, the posterior distribution is computed as:

$$p(\theta, z|w, \alpha, \beta) = \frac{p(\theta, z, w|\alpha, \beta)}{p(w|\alpha, \beta),} \quad (4)$$

Where $p(\theta, z, w|\alpha, \beta)$ is obtained by equation (2) and $p(w|\alpha, \beta)$ is obtained by equation (3).

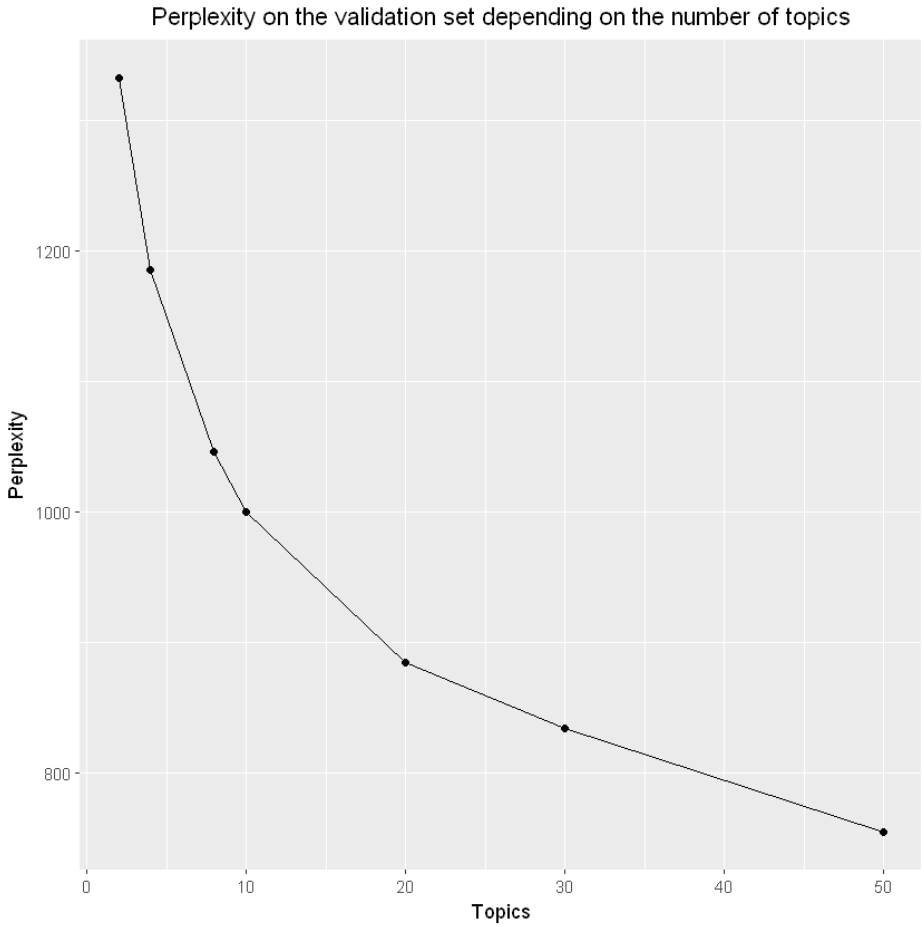
In the context of this study, the objective of LDA is to find the mixture of words that is associated with each of the *K* topics, and to determine the mixture of topics that describes each document. Patents would then be labelled with *K* topic probabilities in order to inform the machine learning models used to predict CPC classes. In other words, the additional set of *n* features (i.e. topics in this case) generated via the LDA process would correspond to vectors of probabilities, indicating the likelihood that a given autonomous vehicle patent is related to topic *n*.

Instead of setting *K* to an arbitrary value, this study referred to the perplexity measure to determine the appropriate number of topics. Perplexity is a common measure of the probability distribution's predictive ability; appropriate distributions have relatively low perplexity (Wang and Xu, 2018). Perplexity can be calculated with the following equation (Pavlinek and Podgorelec, 2017):

$$per(D_{test}) = \exp\left(\frac{\sum_{d=1}^{M} \log p(w_d)}{\sum_{d=1}^{M} N_d} \right) \quad (5)$$

Where D_{test} denotes the held out data, M denotes the number of documents in a collection, w_d denotes the words, and N_d describes the number of words in a given document d.

Figure 2: Perplexity on the validation set depending on the number of topics



As mentioned earlier, lower values of perplexity indicate lower misrepresentation of the words of the test documents by the trained topics. Figure 2 demonstrates the perplexity of LDA models with varying levels of *K*; perplexity continues to decrease up until 50 topics. Perplexity is not the only criteria in deciding on the final value for *K*; choosing an appropriate *K* also depends on domain knowledge and human evaluation of whether the words associated with each topic make structural sense. Based on these considerations, this study selected a *K* value of 30, implying 30 topics; this value had a reasonable value for perplexity, while retaining topic interpretability.

The LDA analysis in this study computes the probability (gamma value) of a given patent being associated with one of the 30 topics. Regarding feature engineering, this study appended the gamma probabilities for each patent across the 30 topics to the final dataframe. These 30 additional topic probability features, together with the patent metadata-derived features, were used as predictors in training the multilabel classification models.

3.3. Dataset

The data was prepared for applying machine learning by merging the features derived from selected patent metadata with the features based on the vectors of topic probabilities. In addition, the final dataframe included the 10 most common (in the retrieved autonomous vehicle data) four-digit (layer 1) CPC codes as target variables. Table 1 provides an overview of the variables included in this study.

Table 1: Predictor and target variables used in the machine learning models

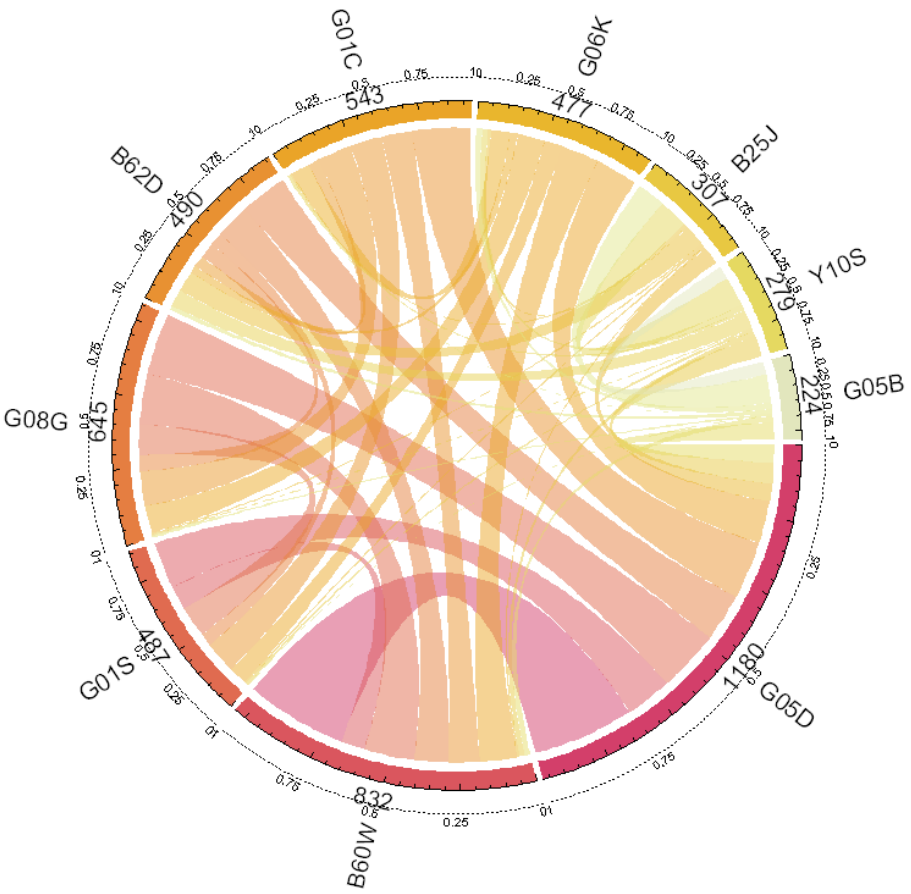
Variable	P/T*	Type	Explanation
TitleLen	P	Integer	Number of characters in the patent title
AbLen	P	Integer	Number of characters in the patent abstract
FullTextLen	P	Integer	Number of characters in the patent full summary text
Nber_Layer1CPCs	P	Integer	The number of CPC classes associated with patent
IssueDate	P	Integer	Year that the patent was granted
GovernmentInterests?	P	Binary	Whether the government had interests in this invention
OneInventorInUS?	P	Binary	Whether or not inventor in US
OneAssigneeInUS?	P	Binary	Whether or not assignee company in US
As_Google?	P	Binary	Whether or not assignee company is Google
As_GM?	P	Binary	Whether or not assignee company is GM
As_Ford?	P	Binary	Whether or not assignee company is Ford
As_Toyota?	P	Binary	Whether or not assignee company is Toyota
BIG4_Google_GM_Ford_Toyota?	P	Binary	Whether or not assignee company is a top 4 company
No_Citations?	P	Binary	Number of citations in the patent
NberTimes.Cited?	P	Binary	Number of times the patent has been cited
InvState_CA?	P	Binary	Whether or not inventor in California
InvState_MI?	P	Binary	Whether or not inventor in Michigan
InvState_MA?	P	Binary	Whether or not inventor in Massachussets
InvState_PA?	P	Binary	Whether or not inventor in Pennsylvania
InvStateCAorMIorMAorPA?	P	Binary	Whether or not inventor in top 4 states
MoreThanOneExaminer?	P	Binary	Did more than one patent examiner evaluate patent?
LernerDavidLittenbergKrumholzMentlik?	P	Binary	Whether patent associated to this law firm
McDonnellBoehnenHulbertBerghoff?	P	Binary	Whether patent associated to this law firm
FishRichardson?	P	Binary	Whether patent associated to this law firm
DarrowChristopherDarrowMustafa?	P	Binary	Whether patent associated to this law firm
NotInTop4LawFirms?	P	Binary	Whether or not patent was in top 4 law firms
Topic 1, 2, … , 30 probability **	P	Numeric	Probability of patent being associated to topic n
CPC 1, 2, … , 10 ***	T	Binary	Layer 1 CPC class associated with a patent

*P: Predictor; T: Target; **30 topic features in total; *** 10 CPC target variables in total.

3.4. Multi-label classification

The objective of multi-label classification was to predict the CPC labelset associated with autonomous vehicle patents. Prior to conducting the multilabel classification, the data was split into a training set (for model training) and a test set (for model evaluation). However, the occurrence of CPC labels was imbalanced (some CPC classes occurred far more frequently than others). Figure 3 shows a SCUMBLE plot which demonstrates this imbalance. In order to balance the label distribution among the training and test set partitions, the stratified sampling method was used. Additional information on the data can be found in the supplementary information. Figure 3 shows that G05D was the most frequent CPC label, while G05B was the least frequent label. The arcs extending from each label to another in the chart demonstrate the level of concurrence between labels.

Figure 3: SCUMBLE plot of the 10 CPC target variables

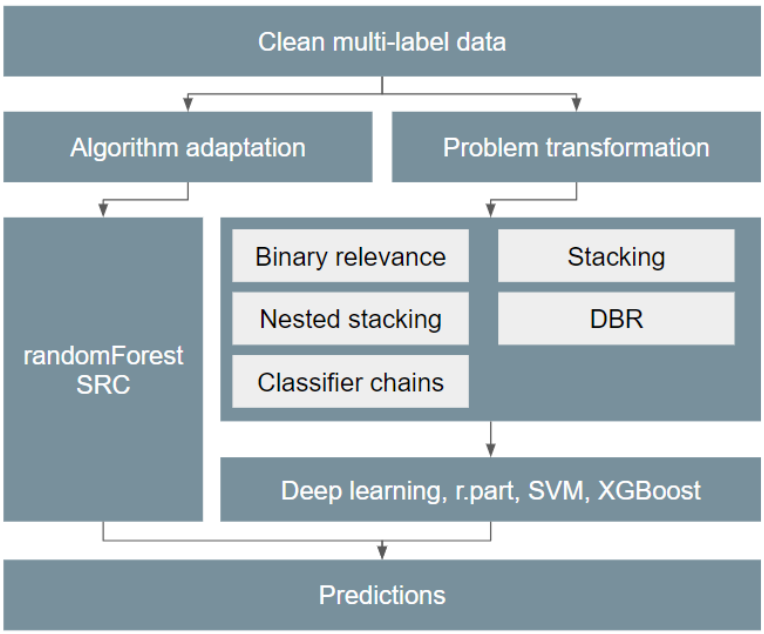


After stratified partitioning of the data, two multilabel classification approaches were used on the data: data/problem transformation and method adaptation. Data transformation techniques convert multilabel datasets into binary or multiclass datasets, which can be processed with any traditional classification algorithms (including recursive partitioning, support vector machine, or neural networks). Method adaptation involves adapting traditional classification algorithms to produce multiple output labels instead of just one.

This study applied both approaches, as demonstrated in figure 4. For the algorithm adaptation approach, the multilabel adaptation of random forest was applied (using the randomForestSRC package in R). For the

problem transformation/ensemble classification approaches, we used the following problem transformation methods (with adaptive boosting as the base learner): binary relevance, classifier chains, nested stacking, dependent binary relevance (DBR), and stacking. The mlr, mldr, and mldr.datasets packages in R were used for the problem transformation method. For a comprehensive treatment of some of the multilabel classification methods used in this study, see: Gibaja and Ventura, 2015; Herrera et al., 2016; Probst et al., 2017; and Tsoumakas and Katakis, 2007. In addition, online R documentation provides information on implementation of these methods.

Figure 4: Overview of the problem transformation and algorithm adaptation methods used



3.5. Multilabel performance assessment metrics

Under the traditional scenario, with only a single class as output, the prediction can only be correct or incorrect. Multilabel predictions, however, can be either fully correct, partially correct/incorrect (at varying degrees), or completely incorrect (Herrera et al., 2016). This additional complexity calls for more nuanced evaluation metrics.

Performance metrics can be categorized into two types. The first is example-based metrics, which evaluate multilabel classification performance based on the average differences of the actual and predicted sets of labels over all examples of a given evaluation dataset. The second is label-based metrics, which decompose the evaluation process into separate evaluations for each label (Giraldo-Forero et al., 2015).

Although more than twenty distinct performance metrics have been defined in the literature, this study evaluates the model based on commonly accepted multilabel metrics. For example-based metrics, we use hamming loss, precision, recall, F1-score, and subset accuracy (these measures are defined below). For label-based metrics, we use balanced accuracy (BAC), area under the curve (AUC), mean misclassification error (MMCE), false negative rate (FNR), and false positive rate (FPR). More details on these metrics can be found in (Herrera et al., 2016).

Example-based metrics

Hamming loss can be computed by

$$Hamming\ loss = \frac{1}{n}\frac{1}{k}\sum_{i=1}^{n}|Y_i \Delta Z_i| \tag{6}$$

Where *n* is the number of data points and *k* the number of labels. Hamming loss computes the symmetric difference (Δ operator) between the predicted labelset Y_i and true values Z_i, counting the number of mismatches (Charte et al., 2018). metric accounts for both omission errors (a correct label is not predicted) and prediction errors (an incorrect label is predicted) normalized over the total number of classes and total number of examples (Gibaja and Ventura, 2015). Hamming loss should be minimized; the smaller the value, the better the model performance.

Precision (also called positive predictive value) is computed as

$$Precision = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Z_i|}{|Z_i|} \tag{7}$$

Precision is defined as the proportion of positive examples that are truly positive. When a model predicts a positive class, this measure captures how often the model is correct.

Recall (also called true positive rate), which is often used in conjunction with precision, is given by

$$Recall = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Z_i|}{|Y_i|} \tag{8}$$

Recall denotes the ratio of correct classifications made to the number of classifications that should have been made, or the number of true positives over the total number of positives.

The F-score is an additional measure of model performance that combines precision and recall into a single number. The F-score is a combination of precision and recall based on the weighted harmonic mean of the two (Visentini et al., 2016). It is given by

$$F1\ score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{9}$$

Lastly, subset accuracy (also known as classification accuracy or labelset accuracy) is used as the most strict evaluation metric. It is computed by

$$Subset\ Accuracy = \frac{1}{n}\sum_{i=1}^{n}[\![Y_i = Z_i]\!] \tag{10}$$

Under this this measure, the full real/predicted labelsets are compared for full equality. Full accuracy would thus imply that every label in the predicted labelset matches that in the true labelset.

Label-based metrics

Label-based metrics are briefly summarized here, since they are commonly used in binary and multiclass classification literature. FPR refers to the percentage of misclassified observations in the positive class. FNR refers to the percentage of misclassified observations in the negative class. AUC refers to the integral over the graph resulting from computing FPR and TPR (recall) for different classification probability

thresholds. BAC refers to the mean of the true positive rate and the true negative rate, and accounts for imbalanced class ratios (García et al., 2009). Lastly, MMCE refers to the mean of responses that are not equal to the actual truth.

3.6. Shiny web application

In addition to the baseline analysis, this study built an interactive tool that allows researchers to interactively explore our machine learning and NLP results. The web-based tool was developed with R Shiny, an open source package for developing graphical user interfaces. The tool allows users to adjust certain multilabel classification parameters and evaluate subsequent performance. NLP-based visualization (based on term frequency-inverse document frequency and LDA) of patent data is also a feature of the tool.

3.7. Reproducibility of results

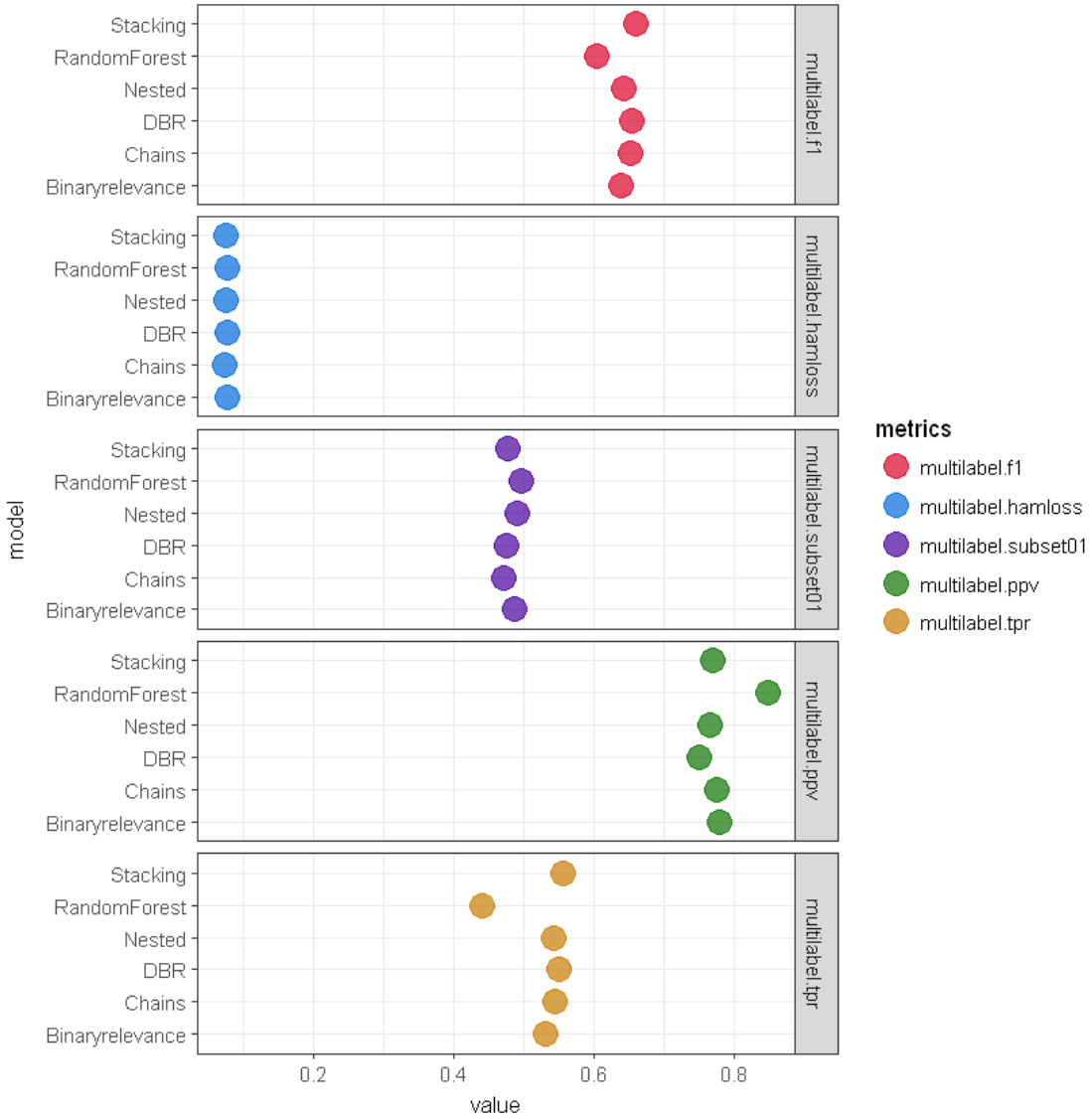
The machine learning models in this study were implemented using various packages in R, including "mlr", "mldr", "mldr.datasets", and many more. The full R code behind the NLP, multilabel classification, and R Shiny tool development is provided on github so that researchers may replicate and/or extend upon the results of this research.

4. Results and discussion

4.1.Model performance on example-based metrics

Figure 5 illustrates the performance of the machine learning models used for multilabel CPC classification; the algorithm adaptation models (random forest) and the problem transformation-based model (adaptive boosting) are compared across the metrics introduced in the methodology section.

Figure 5: Model evaluation based on five multi-label classification performance metrics



4.1.1. Subset accuracy and hamming loss (figure 5: multilabel.subset01 and multilabel.hamloss)

Regarding subset accuracy, the performance for random forest (algorithm adaptation method) and adaptive boosting (via stacking, nested stacking, dependent binary relevance/DBR, classifier chains, and binary relevance) respectively were 0.496, 0.476, 0.489, 0.474, 0.471, and 0.486 respectively. The random forest algorithm adaptation method performed best under this metric. Given that subset accuracy is the most stringent evaluation metric, it is logical that values are approximately 0.50. This implies that our algorithm

manages to correctly predict the presence/absence of *all* 10 CPC labels in the target labelset about 50% of the time.

For hamming loss, the performance values respectively were 0.077, 0.074, 0.075, 0.077, 0.074, and 0.076. Given that hamming loss is a metric to be minimized, the classifier chains and stacking problem transformation methods (with adaptive boosting) performed best. This metric shows the fraction of the wrong labels to the total number of labels, indicating that the best-performing model classified only 7.4% of labels incorrectly on average. This implies that autonomous vehicle patents were being misclassified at a very low frequency.

4.1.2. Precision, Recall, and F1 score (Figure 5: multilabel.ppv, multilabel.tpr, and multilabel.f1)

For precision, the performance values for random forest (algorithm adaptation method) and adaptive boosting (via stacking, nested stacking, DBR, classifier chains, and binary relevance) respectively were 0.847, 0.769, 0.765, 0.750, 0.774, and 0.777; the random forest algorithm adaptation method performed best. As for recall, the values were respectively 0.439, 0.554, 0.543, 0.550, 0.544, and 0.530; DBR with adaptive boosting performed best. The F1-score performance values were 0.603, 0.658, 0.642, 0.653, 0.650, 0.638 respectively; stacking with adaptive boosting performed best.

In this context, precision refers to the fraction of patents that were correctly assigned their CPC class (true positives) divided by the total amount of patents that were either correctly classified (true positives) or incorrectly classified (false positives). In other words, precision reveals to what extent the autonomous vehicle patents are being assigned incorrect labels. High values for precision indicates that the algorithm is doing a good job at preventing patents from being assigned incorrect labels, even though it is not predicting all labels accurately with an exact match.

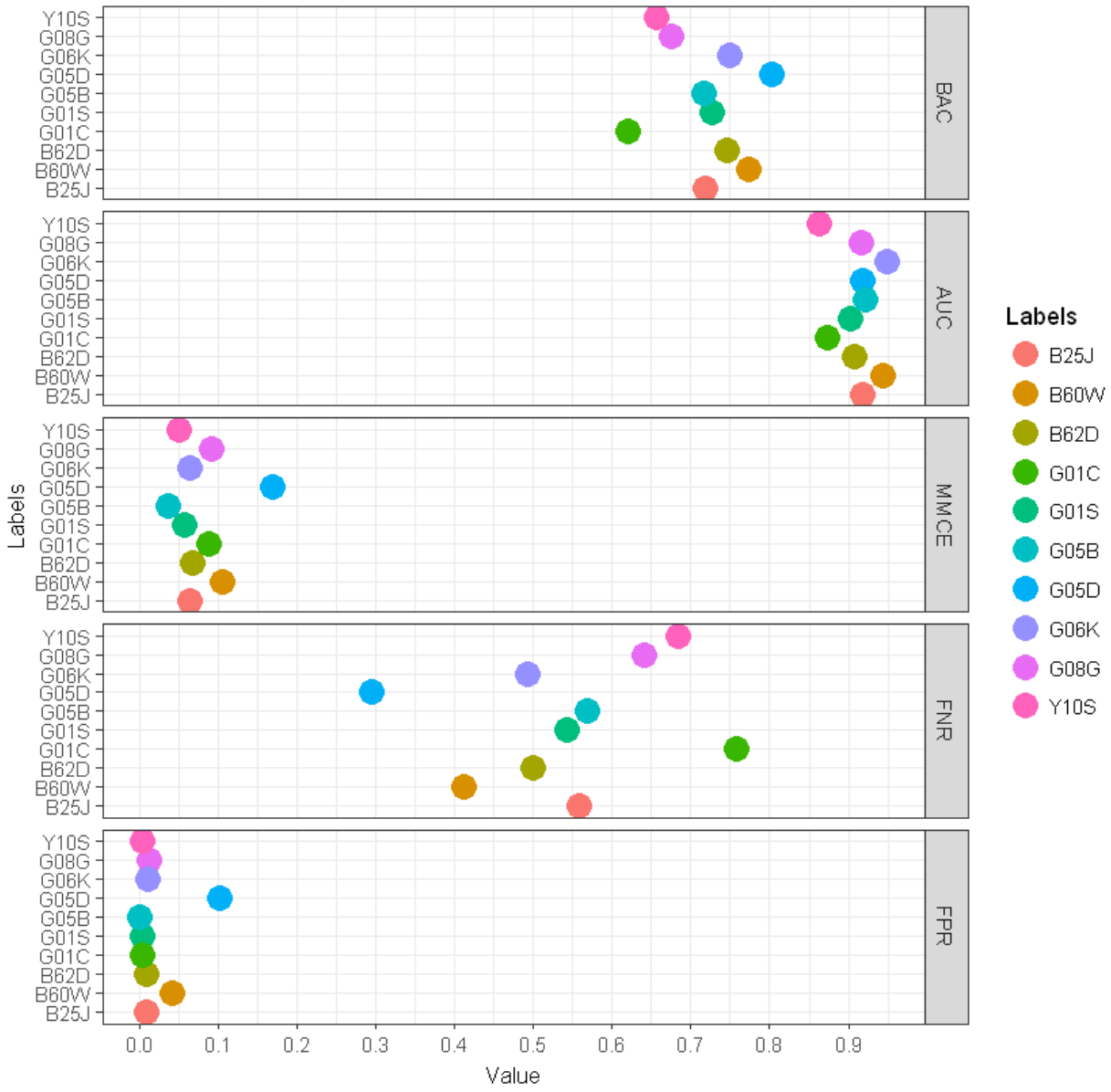
Recall, in this context, refers to the fraction of patents that were correctly assigned to their CPC class (true positives) divided by the total amount of patents that were either correctly assigned (true positives) or missed completely (false negatives). Recall indicates to what extent a patents' true CPC class was not identified at all (as opposed to being classified incorrectly, which is captured by precision).

In the context of CPC classification, it is more important that machine learning models perform well on precision than on recall. The logic is as follows: a patent researcher cares less that a patent related to autonomous vehicle sensing technology is actually classified as sensing technology, compared to the scenario where the patent is incorrectly classified into an unrelated technology field. In other words, the confusion arising from patent misclassification is more pernicious than if a patent is not classified at all.

4.2. Label-based performance metrics

In addition to the aggregate metrics provided above, label-based evaluation metrics introduced in section 3 (BAC, AUC, MMCE, FNR, FPR) were computed in order to provide a more nuanced perspective on model performance. Figure 6 provides the label-based performance metrics of the random forest algorithm adaptation method. For the sake of brevity (and since the performance values are similar), additional label-based performance metrics for the problem transformation method with adaptive boosting are provided in the supplementary information.

Figure 6: label-based performance for the random forest algorithm adaptation method



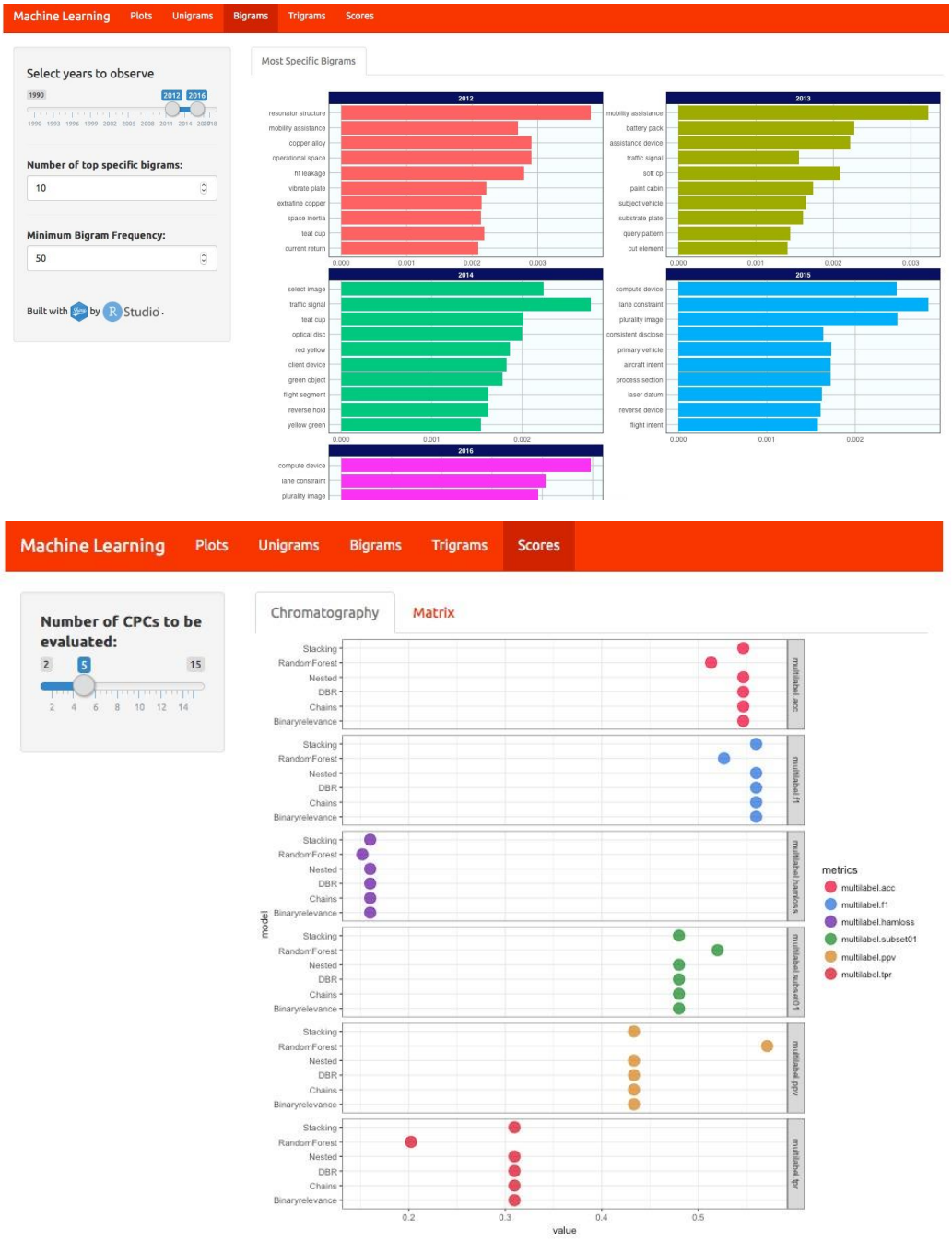
The random forest model shows promising results according to the label-based metrics. Firstly, the false positive rate across all ten labels is very low, ranging from 0 to 0.101. This indicates that very few labels are being misclassified; the model is effective in translating input features to finding relevant labels. The false negative rate is somewhat high for certain classes, with a range of 0.295 to 0.758. These values indicate

that a significant number of labels are not being classified at all (rather than being misclassified). This was already alluded to based on the precision and recall performance. The random forest model had high AUC values, ranging from 0.862 to 0.95 across the ten labels; this is significantly higher than the random baseline predictor AUC value of 0.5, indicating that the model is valuable. MMCE ranges from 0.035 to 0.168, indicating that few labels are misclassified on average. Lastly, BAC (which accounts for imbalanced class ratios) ranges from 0.619 to 0.802 across the ten class labels.

4.3. Interactive visualization

In addition to the static results of the analysis presented above, this study developed an interactive tool which allows users to explore the patent data based on NLP methods and adjust parameters in the machine learning models applied. The tool allows for basic understanding of patent topics based on text analysis and also allows users to evaluate the machine learning performance metrics based on the size of the CPC labelset to be classified (classification is scalable from 2 to 15 labels); smaller labelsets generally have better performance in terms of accuracy, recall, etc.

Figure 7: illustration of the NLP



5. Conclusion

This study extracted topic features from patent text data in an emerging technology field based on latent dirichlet allocation. Together with patent metadata, these topic features were used as inputs for multi-label classification algorithms. Multi-label classification was conducted based on the algorithm adaptation method (using the multi-label adaptation of random forest) and problem transformation method based on adaptive boosting (binary relevance, classifier chains, dependent binary relevance, stacking, and nested stacking).

The classification algorithms showed promising results when judged on example-based metrics. The highest values for subset accuracy, precision, recall, and F1-score computed as 0.489, 0.847, 0.554, and 0.653 respectively. High precision indicated that the best model was successful in preventing patents from being assigned incorrect labels. Hamming loss, a metric to be minimized, was low. The best performing model classified only 7.4% of labels incorrectly on average. In addition, the adapted random forest classifier showed encouraging performance on label-based metrics including BAC, AUC, MMCE, FNR, and FPR.

In conclusion, this study provides some foundations for patent metadata and NLP-based multilabel classification, using autonomous vehicle patent data for a case study.

Acknowledgements

We would like to thank Dr. Lee Fleming and Guan-Cheng Li for their supervision over this research project. We would also like to thank Berkeley's Fung Institute for Engineering Leadership for their support.

References

Abe, S., 2015. Fuzzy support vector machines for multilabel classification. Pattern Recognit. 48, 2110–2117. https://doi.org/https://doi.org/10.1016/j.patcog.2015.01.009

Blei, D.M., Ng, A.Y., Jordan, M.I., 2003. Latent Dirichlet Allocation. J. Mach. Learn. Res. 3, 993–1022.

Cao, G., Luo, P., Wang, L., Yang, X., 2016. Key Technologies for Sustainable Design Based on Patent Knowledge Mining. Procedia CIRP 39, 97–102. https://doi.org/https://doi.org/10.1016/j.procir.2016.01.172

Charte, F., Rivera, A.J., Charte, D., del Jesus, M.J., Herrera, F., 2018. Tips, guidelines and tools for managing multi-label datasets: The mldr.datasets R package and the Cometa data repository. Neurocomputing 289, 68–85. https://doi.org/https://doi.org/10.1016/j.neucom.2018.02.011

Charte, F., Rivera, A.J., del Jesus, M.J., Herrera, F., 2015. Addressing imbalance in multilabel classification: Measures and random resampling algorithms. Neurocomputing 163, 3–16. https://doi.org/https://doi.org/10.1016/j.neucom.2014.08.091

Codina-Filbà, J., Bouayad-Agha, N., Burga, A., Casamayor, G., Mille, S., Müller, A., Saggion, H., Wanner, L., 2017. Using genre-specific features for patent summaries. Inf. Process. Manag. 53, 151–174. https://doi.org/https://doi.org/10.1016/j.ipm.2016.07.002

Eisinger, D., Tsatsaronis, G., Bundschus, M., Wieneke, U., Schroeder, M., 2013. Automated Patent Categorization and Guided Patent Search using IPC as Inspired by MeSH and PubMed. J. Biomed. Semantics 4, S3–S3. https://doi.org/10.1186/2041-1480-4-S1-S3

García, V., Mollineda, R.A., Sánchez, J.S., 2009. Index of Balanced Accuracy: A Performance Measure for Skewed Class Distributions BT - Pattern Recognition and Image Analysis, in: Araujo, H., Mendonça, A.M., Pinho, A.J., Torres, M.I. (Eds.), . Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 441–448.

Gibaja, E., Ventura, S., 2015. A Tutorial on Multilabel Learning. ACM Comput. Surv. 47, 52:1--52:38. https://doi.org/10.1145/2716262

Giraldo-Forero, A.F., Jaramillo-Garzón, J.A., Castellanos-Domínguez, C.G., 2015. Evaluation of Example-Based Measures for Multi-label Classification Performance BT - Bioinformatics and Biomedical Engineering, in: Ortuño, F., Rojas, I. (Eds.), . Springer International Publishing, Cham, pp. 557–564.

Herrera, F., Charte, F., Rivera, A.J., Jesus, M.J., 2016. Multilabel Classification Problem Analysis, Metrics and Techniques, 1st ed. Springer, Switzerland. https://doi.org/https://doi.org/10.1007/978-3-319-41111-8

Joung, J., Kim, K., 2017. Monitoring emerging technologies for technology planning using technical keyword based analysis from patent data. Technol. Forecast. Soc. Change 114, 281–292. https://doi.org/https://doi.org/10.1016/j.techfore.2016.08.020

Kyebambe, M.N., Cheng, G., Huang, Y., He, C., Zhang, Z., 2017. Forecasting emerging technologies: A supervised learning approach through patent analysis. Technol. Forecast. Soc. Change 125, 236–244. https://doi.org/https://doi.org/10.1016/j.techfore.2017.08.002

Lee, C., Kwon, O., Kim, M., Kwon, D., 2018. Early identification of emerging technologies: A machine learning approach using multiple patent indicators. Technol. Forecast. Soc. Change 127, 291–303. https://doi.org/https://doi.org/10.1016/j.techfore.2017.10.002

Li, H., Yang, X., Jian, L., Liu, K., Yuan, Y., Wu, W., 2018. A sparse representation-based image resolution improvement method by processing multiple dictionary pairs with latent Dirichlet allocation model

for street view images. Sustain. Cities Soc. 38, 55–69. https://doi.org/https://doi.org/10.1016/j.scs.2017.12.020

Luo, F., Guo, W., Yu, Y., Chen, G., 2017. A multi-label classification algorithm based on kernel extreme learning machine. Neurocomputing 260, 313–320. https://doi.org/https://doi.org/10.1016/j.neucom.2017.04.052

Madani, F., Weber, C., 2016. The evolution of patent mining: Applying bibliometrics analysis and keyword network analysis. World Pat. Inf. 46, 32–48. https://doi.org/https://doi.org/10.1016/j.wpi.2016.05.008

Momtazi, S., 2018. Unsupervised Latent Dirichlet Allocation for supervised question classification. Inf. Process. Manag. 54, 380–393. https://doi.org/https://doi.org/10.1016/j.ipm.2018.01.001

Pavlinek, M., Podgorelec, V., 2017. Text classification method based on self-training and LDA topic models. Expert Syst. Appl. 80, 83–93. https://doi.org/https://doi.org/10.1016/j.eswa.2017.03.020

Probst, P., Au, Q., Casalicchio, G., Stachl, C., Bischi, B., 2017. Multilabel Classification with R Package mlr. R J. 9, 352–369.

Ramírez-Corona, M., Sucar, L.E., Morales, E.F., 2016. Hierarchical multilabel classification based on path evaluation. Int. J. Approx. Reason. 68, 179–193. https://doi.org/https://doi.org/10.1016/j.ijar.2015.07.008

Rokach, L., Schclar, A., Itach, E., 2014. Ensemble methods for multi-label classification. Expert Syst. Appl. 41, 7507–7523. https://doi.org/https://doi.org/10.1016/j.eswa.2014.06.015

Song, K., Kim, K., Lee, S., 2018. Identifying promising technologies using patents: A retrospective feature analysis and a prospective needs analysis on outlier patents. Technol. Forecast. Soc. Change 128, 118–132. https://doi.org/https://doi.org/10.1016/j.techfore.2017.11.008

Souili, A., Cavallucci, D., Rousselot, F., 2015a. A lexico-syntactic Pattern Matching Method to Extract Idm- Triz Knowledge from On-line Patent Databases. Procedia Eng. 131, 418–425. https://doi.org/https://doi.org/10.1016/j.proeng.2015.12.437

Souili, A., Cavallucci, D., Rousselot, F., 2015b. Natural Language Processing (NLP) – A Solution for Knowledge Extraction from Patent Unstructured Data. Procedia Eng. 131, 635–643. https://doi.org/https://doi.org/10.1016/j.proeng.2015.12.457

Suominen, A., Toivanen, H., Seppänen, M., 2017. Firms' knowledge profiles: Mapping patent data with unsupervised learning. Technol. Forecast. Soc. Change 115, 131–142. https://doi.org/https://doi.org/10.1016/j.techfore.2016.09.028

Tahir, M.A., Kittler, J., Bouridane, A., 2012. Multilabel classification using heterogeneous ensemble of multi-label classifiers. Pattern Recognit. Lett. 33, 513–523. https://doi.org/https://doi.org/10.1016/j.patrec.2011.10.019

Tsoumakas, G., Katakis, I., 2007. Multi-Label Classification: An Overview. Int. J. Data Warehous. Min. 3, 1–13. https://doi.org/10.4018/jdwm.2007070101

United States Patent and Trademark Office (USPTO), n.d. 905 Cooperative Patent Classification [R-07.2015] [WWW Document]. URL https://www.uspto.gov/web/offices/pac/mpep/s905.html (accessed 4.4.18).

Venugopalan, S., Rai, V., 2015. Topic based classification and pattern identification in patents. Technol. Forecast. Soc. Change 94, 236–250. https://doi.org/https://doi.org/10.1016/j.techfore.2014.10.006

Visentini, I., Snidaro, L., Foresti, G.L., 2016. Diversity-aware classifier ensemble selection via f-score. Inf.

Fusion 28, 24–43. https://doi.org/https://doi.org/10.1016/j.inffus.2015.07.003

Wang, Y., Xu, W., 2018. Leveraging deep learning with LDA-based text analytics to detect automobile insurance fraud. Decis. Support Syst. 105, 87–95. https://doi.org/https://doi.org/10.1016/j.dss.2017.11.001

Wu, J.-L., Chang, P.-C., Tsao, C.-C., Fan, C.-Y., 2016. A patent quality analysis and classification system using self-organizing maps with support vector machine. Appl. Soft Comput. 41, 305–316. https://doi.org/https://doi.org/10.1016/j.asoc.2016.01.020

Zhang, X., 2014. Interactive patent classification based on multi-classifier fusion and active learning. Neurocomputing 127, 200–205. https://doi.org/https://doi.org/10.1016/j.neucom.2013.08.013

Supplementary Information

Code

All datasets and R code involved in this analysis are provided online to ensure full reproducibility. These resources can be found at https://github.com/djavandeclercq/MultiLabelPatentClassification.

A note on the CPC classifiers predicted in this study

The 10 CPC target variable in this study included:

- G05D: Systems for controlling or regulating non-electric variables.
- B60W: conjoint control of vehicle sub-units of different type or different function; control systems specially adapted for hybrid vehicles; road vehicle drive control systems for purposes not related to the control of a particular sub-unit.
- G01S: Radio direction-finding; radio navigation; determining distance or velocity by use of radio waves; locating or presence-detecting by use of the reflection or reradiation of radio waves; analogous arrangements using other waves.
- G08G: Traffic control systems.
- B62D: Motor vehicles; trailers.
- G01C: Measuring distances, levels or bearings; surveying; navigation; gyroscopic instruments; photogrammetry or videogrammetry.
- G06K: Recognition of data; presentation of data; record carriers; handling record carriers.
- B25J: Manipulators; chambers provided with manipulation devices.
- Y10S: Technical subjects covered by former uspc cross-reference art collections.
- G05B: Control or regulating systems in general; functional elements of such systems; monitoring or testing arrangements for such systems or elements.