# Binary Patent Classification Methods for Few Annotated Samples

Benjamin Meindl[a], Ingrid Ott[b,e], Ulrich Zierahn[c,d]

[a] *University of Lisbon*
[b] *Karlsruhe Institute of Technology (KIT)*
[c] *ZEW Mannheim*
[d] *CESifo Network*
[e] *IfW Kiel*

**Abstract**

In this paper, we develop binary patent classification algorithms for ambiguous concepts and small sample sizes. These are particularly useful for economic questions, which often require binary classification for implementing ambiguous and subjective concepts, where human classification is time-consuming, so that sample sizes are small. This covers examples such as whether workers are susceptible to automation or not, or whether a device is an automat or not. We compare the performance of naive Bayes, support vector machine, random forest and k-nearest neighbor classifiers with a the spaCy convolutional neural network (CNN) model, as well as spaCy CNN model pre-trained with patent data. The results show overall highest accuracy for the CNN models, with a significantly improved performance through pre-training. Our analysis suggests that the spaCy pre-trained CNN model provides a highly accurate NLP model, feasible for implementation without extensive computation capacity required. Pre-training was particularly beneficial for small sample sizes. Already 100 labeled patents lead to an accuracy of 77.2%. The low sample size required, may encourage researchers in various fields to use manually labeled patent data, for evaluating their specific question.

*Keywords:* patent classification, small sample size, convolutional neural network, language model pre-training, fast pre-training

## 1. Introduction

New technologies play a key role for economic development and wealth [1]. This covers a large and currently very active debate on the effects of automation technologies on the labor market [2, 3]. The economic debate often relies on binary classifications to analyze the effects of new technologies on the economy. For example, economists study whether technological change refers to automation or not (e.g. [4]), whether workers are susceptible or non-susceptible to automation (e.g. [5, 6]), how innovation vs. imitation affects the economy (e.g. [7]), or the role of process vs. product innovations for firms (e.g. [8]). Patent texts are well recognized indicators to describe the technological state of the art. As such, patents contain relevant information to measure the mentioned concepts, e.g., by classifying patents that refer to automats vs. non-automats [4]. This is often complex due to the ambiguity of the concepts and the similarity of patents that refer to distinct categories. Being able to assign patents to unique categories allows linking them to other economic data. Until now there only exist few and very broad concordances that allow assigning patents either to technologies [9] or to industries [10]. But these classifications are rather broad.

In this paper, we compare binary patent classifiers, which may be used for analyzing technological change. The main challenge not only lies in the complexity and ambiguity of the concepts, but also in the sample size. Sample sizes are often small, because human coders often require significant time for classifying such cases. These algorithms may be applied to other cases with complex and ambiguous binary classes and few training data.

The rest of this paper is organized as follows: Section 2 provides a description of the underlying patent data and Section 3 our machine learning algorithms. We present and discuss our results in Sections 4. Section 5 concludes.

*Published in "Proceedings of The 1**st Workshop on Patent Text Mining and Semantic Technologies (PatentSemTech 2019)**", Eds. L. Andersson, H. Aras, F. Piroi, A. Hanbury, PatentSemTech 2019, 12 September 2019, Karlsruhe, Germany. https://doi.org/10.34726/pst2019.4* 

## 2. Patent Data

We aim at developing a classifier which is able to handle cases with high ambiguity / large overlap. Additionally, it should provide sufficient precision even with low numbers of examples, as hand-classification is costly when human coders have to read large parts of a patent to classify it. In order to develop algorithms which are suited for such cases, we focus on data which contains a binary outcome variable with ambiguous classes. In particular, we rely on patent data, which is particularly suited to study technological change. Moreover, we focus on two selected cooperative patent classification (CPC) classes as our outcome variable to analyze a binary outcome. We focus on two CPC classes which are potentially hard to differentiate for an algorithm in order to train algorithms which are suited for ambiguous cases.

We motivate the choice of our patent sample by the recent interest in robot technologies and the widespread interest this technology field receives in current public and economic debate (e.g., [11, 12, 13]). The United States patent classification (USPC) class 901 - robot - has been mapped to the CPC with the most recent update being from 2012[1]. Most statistically relevant CPC classes related to the USPC class 901 are G 05D, A 61B, G 05B, B 25J, B 23K, B 06B, and G 01N.

Most similar from a technological perspective are CPC classes G 05B and G 05D.[2] We thus restrict our sample to the two sub-classes G 05D and G 05B and use these two classes as a natural delineation to train binary classifiers. G 05D refers to systems for controlling or regulating non-electric variables, e.g., for welding, pressure control, and so on. G 05B relates to control and regulating systems which are "clearly more generally applicable". The fact that G 05B refers to systems which are more generally applicable, whereas G 05D refers to those that control or regulate only non-electric variables, creates a certain ambiguity. Such an abiguity is often present in the economic examples noted above: Without a sufficient training it is often hard to assess for a human, whether a patent is sufficiently generally applicable to be classified as G 05B instead of G 05D. This challenge is similar to the economic samples described in the introduction, such as [4] who define an automat as a device that carries out a process *independently*. Their classification task (i.e., automats vs. non-automats) involves ambiguity, as devices typically require at least some kind of human involvement, so that the interpretation of *independence* remains a subjective assessment of the human coders.

Another objective of the algorithm is to achieve high accuracy with low sample data, as hand-classification is costly when human coders have to read large parts of a patent to classify a patent. [4], for example, build their analysis of patents describing "automats" on 560 hand classified patents. We will compare our algorithms for different sample sizes, to evaluate requirements on sample sizes for potential annotation tasks. We start with the smallest sample size of 100 patents only, which may be mainly relevant for early validation of the feasibility of an idea, and as an input for active learning, which is an early training of the model to select further patents for more efficient classification. Next, we include datasets with 250 and 500 patents. We expect 500 patents to be a potential minimum sample size for analysis, e.g., similar to [4]. Finally we build larger datasets of 1,500 and 5,000 patents, to evaluate the benefit of higher investment of resources for annotation.

We draw our sample data from the USPTO-2m patent abstract dataset [14], which is commonly used for patent classification benchmarking. For each dataset, we draw 50% each G 05D and G 05B examples, whereas patents with both labels are considered as G 05D. For evaluation, we use 250 randomly drawn patents of each category.

## 3. Patent Classification Algorithms

In our analysis, we compare different approaches for patent classification. [4] use a multinominal naive Bayes (MNB) algorithm to identify patents describing an "automat." Based on 560 manual annotations, they achieve a correct prediction of 80% of patents. One valuable feature of MNB is the ability to interpret results. [4], for example, extract tokens typical for "automats." Support vector machines (SVM) may outperform Naive Bayes [15, 16] or other approaches such as k-nearest neighbor [17] for text classification, and also allow for feature extraction. [18] performed best at the ALTA 2018 patent classification task, using a method based on SVMs.

Further approaches for patent classification are based on neural network (NN) models [19]. [20, 14] describe the potentially high precision of NNs for patent classification and [21] find that they may outperform SVM, particularly for shorter texts. Some recent advances in the field of natural language processing rely on pretraining and fine-tuning NN models (e.g., BERT [22], ULMFiT [23]). [24] outperformed previous approaches

[1] USPC has been deprecated in favor of CPC.

[2] compare https://www.uspto.gov/web/patents/classification/cpc/pdf/us901tocpc.pdf.

of patent classification using patent data to pre-train a BERT convolutional neural network (CNN) model.

Pre-training models such as BERT require extensive computational resources. Therefore, [25, 26] describe alternative models, achieving a significant reduction in computational resource requirements with nearly similar performance. A similar model, called Language Modelling with Approximate Outputs (LMAO) is implemented in the spaCy library[3].

For our analysis, we want to a compare binary classification performance of a pre-trained CNN with alternative approaches. Naive Bayes has been used as a baseline for similar efforts [27]. We use a Bernoulli naive Bayes (BernoulliNB) classifier as a baseline for our work, which accounts particularly for the binary decision. Further, we evaluate an SVM based model, which has been successfully used for various patent classification tasks. Also, we implement a random forest classifier (RandomForest) and a k-nearest neighbor classifier (k-NN) for comparison.

BernoulliNB, SVM, RandomForest, and K-NN classifiers are implemented using Scikit-learn. Therefore, we lemmatize words (using NLTK[4]), remove stopwords, and extract the most relevant words per document through term frequency-inverse document frequency scores (TF-IDF), using unigrams as well as bigrams. [28] finds that TF-IDF analysis using bigrams (instead of unigrams only) may lead to higher accuracy, as it accounts for complex multi-word expressions. We use the Scikit-learn model selection, GridSearchCV, for optimization of model parameters.

We implement a CNN based classifier using spaCy, which is a library aiming at providing a combination of high accuracy and speed. This is especially relevant for patent classification, as it enables research on large patent data sets with reasonable resources. Further, it allows resource efficient LMAO pre-training for patent specific context.

Our analysis includes two spaCy based approaches. First, we use the default large English language model. Second, we use the same model pre-trained with patent data (we refer to it as $\text{spaCy}_{\text{pre}}$). To assure high contextual relevance of pre-training, we use the 25,212 patents in the class G 05 from the USPTO-2m dataset. The algorithm ran 200 passes over the dataset until the loss function did not further decrease. In addition, we run the same models with the software prodigy[5]. Prodigy builds on spaCy and allows for straightforward implementation of natural language processing analysis and annotation. It provides a simple API requiring only basic knowledge in programming. We want to evaluate whether using the tools compromises performance compared to a manual implementation of spaCy.

[3] https://spacy.io/

[4] https://www.nltk.org

[5] https://prodi.gy

## 4. Results and Discussion

A comparison of the different algorithms shows that the pre-trained CNN model outperforms remaining models (see table 1) for each sample size. The regular spaCy model performs second best for all sample sizes. From the remaining models, the BernoulliNB classifier performed best for all sample sizes but the largest one. The performance of the SVC model fluctuated strongly for different sample sizes, and did even decrease, e.g., comparing the 1,500 dataset with the 250 dataset. RandomForest and k-NN were within lowest performing classifiers for all sample sizes, however, they reach a reasonable accuracy for the largest dataset. We thus find that the pre-trained CNN model performs best as a binary patent classifier for hard-to-classify concepts.

The results further show a significant increase in performance through pre-training with patent data. The benefits are strongest for small sample sizes, where 100 annotations led to accuracy scores of 77.2%, compared to a score of 72.5% for the CNN without pre-training. This score suggests, that pre-trained neural network may be well suitable for active learning, which aims at increasing the efficiency of annotations through active learning [29]. The performance advantage of pre-training, however, decreases with sample size and almost disappears for the largest data set. Accordingly, we find that pre-training is particularly useful for small data sets, but provides negligible performance advantages with large data sets of around 5,000 or more annotated samples. Future research may evaluate, whether more expensive pre-training methods provide even stronger models.

Our best-performing CNN achieves an accuracy of 0.832 and 0.866 with sample sizes of 500 and 1500 patents. These accuracy scores may be appropriate for a number of further analyses and may encourage future researchers to use labeled patent data for their analyses.

Moreover, the spaCy LMAO pre-training does not require extensive computation capacity. Therefore, the described methods may be suitable for a broad range of researchers, providing high accuracy and enabling efficient implementation.

In addition to the results shown in the table, we ran the spaCy models through the Prodigy software. The

| Model | Sample size | | | | |
|---|---|---|---|---|---|
| | 100 | 250 | 500 | 1,500 | 5,000 |
| BernoulliNB | 0.706 | 0.776 | 0.798 | 0.808 | 0.842 |
| SVC | 0.612 | 0.536 | 0.794 | 0.774 | 0.858 |
| RandomForest | 0.590 | 0.668 | 0.752 | 0.770 | 0.836 |
| K-NN | 0.598 | 0.704 | 0.716 | 0.772 | 0.838 |
| spaCy | 0.726 | 0.786 | 0.806 | 0.858 | 0.872 |
| **spaCy$_{\text{pre}}$** | **0.772** | **0.800** | **0.832** | **0.866** | **0.874** |

Table 1: Comparison of patent classification performance. The models implemented are Bernoulli naive Bayes (BernoulliNB), support vector machine (SVC), random forest, k-nearest neighbour, spaCy large English model, and a spaCy model pre-trained with patent data. The models have been tested with different sample sizes, of 100, 250, 500, 1,500, and 5,000 patents in categories G 05D, and G 05B. Scores relate to recognition of G 05D.

results were similar to both spaCy models and are thus not listed in Table 1. This implies that relying on a simple API that requires only basic knowledge in programming comes at little performance costs, rendering the methods proposed in this paper potentially accessible to researchers from disciplines with typically less training in programming, such as e.g. economists.

## 5. Conclusions

Patent classification, in general, is an active research field. Besides pre-classification of patent applications, which is highly relevant for patent offices [17], also other fields may benefit from advances in this area. Particularly economists may benefit from improved methods of patent analyses. [30], for example, describe the lack of high-quality data and empirically informed models as a key challenge for a better understanding of automation technologies. Patent data may be a rich source of data to address this challenge.

Our work contributes to patent as well as NLP research by evaluating a powerful pre-trained CNN based approach for binary patent classification. The proposed method offers a fast, high accuracy tool enabling a broad range of researchers conducting patent classification or other text classification tasks. We find that pre-training significantly raises performance particularly in small samples of annotated data, while the performance surplus declines for larger samples.

We further find that the methods provide a high accuracy, do not require high computational resources, and that relying on Prodigy as a simple API does not result in noticeable performance losses. This implies that the methods proposed here are both useful and potentially accessible to researchers from other disciplines.

## Competing Interests

We declare that we have no significant competing financial, professional, or personal interests that might have influenced the performance or presentation of the work described in this manuscript.

## References


[1] D. Acemoglu, Introduction to Modern Economic Growth, Princeton University Press, 2009.

[2] J. Mokyr, C. Vickers, N. L. Ziebarth, The history of technological anxiety and the future of economic growth: Is this time different?, Journal of Economic Perspectives 29 (3) (2015) 31–50.

[3] D. Autor, Why are there still so many jobs? the history and future of workplace automation, Journal of Economic Perspectives 29 (3) (2015) 3–30.

[4] K. Mann, L. Püttmann, Benign effects of automation: New evidence from patent texts (2018).

[5] C. B. Frey, M. A. Osborne, The future of employment: How susceptible are jobs to computerization?, Technological Forecasting and Social Change 114 (2017) 254–280.

[6] M. Arntz, T. Gregory, U. Zierahn, Revisiting the risk of automation, Economics Letters 159 (2017) 157–160.

[7] P. S. Segerstrom, Innovation, imitation, and economic growth, Journal of Political Economy 99 (4) (1991) 807–827.

[8] A. Bartel, C. Ichniowski, K. Shaw, How does information technology affect productivity? Plant-level comparisons of product innovation, process improvement, and worker skills, Quarterly Journal of Economics 122 (4) (2007) 1721–1758.

[9] U. Schmoch, Concept of technology classification for country comparisons, Final report to the World Intellectual Property Organisation (WIPO), Fraunhofer Institute for Systems and Innovation Research (ISI) (2008).

[10] eurostat, Patent statistics: Concordance ipc v8 - nace rev.2, Tech. rep., Eurostat (2014).
URL https://circabc.europa.eu/sd/a/d1475596-1568-408a-9191-426629047e31/2014-10-16-Final%20IPC_NACE2_2014.pdf

[11] D. Acemoglu, P. Restrepo, Robots and jobs: Evidence from US labor markets, NBER Working Paper 23285 (2017).

[12] W. Dauth, S. Findeisen, J. Südekum, N. Wössner, German robots - the impact of industrial robots on workers, IAB Working Paper 30/2017.

[13] G. Graetz, G. Michaels, Robots at work, Review of Economics and Statistics 100 (5) (2018) 753–768.

[14] S. Li, J. Hu, Y. Cui, J. Hu, DeepPatent: patent classification with convolutional neural networks and word embedding, Scientometrics 117 (2) (2018) 721–744. doi:10.1007/s11192-018-2905-5.

[15] T. Joachims, Text Categorization with Support Vector Machines : Learning with Many Relevant Features, European conference on machine learning (1998) 137–142.

[16] C. J. Fall, A. Törcsvári, K. Benzineb, G. Karetka, Automated Categorization in the International Patent Classification, Acm Sigir Forum 37 (1) (2003) 10–25.

[17] M. Krier, F. Zacc, Automatic categorisation applications at the European patent office, World Patent Information 24 (2002) 187–196.

[18] F. Benites, S. Malmasi, M. Zampieri, Classifying Patent Applications with Ensemble Methods, Proceedings ofAustralasian Language Technology Association Workshop (2018) 89–92.

[19] A. Abbas, L. Zhang, S. U. Khan, A literature review on the state-of-the-art in patent analysis, World Patent Information 37 (2014) 3–13. doi:10.1016/j.wpi.2013.12.006.

[20] M. F. Grawe, C. A. Martins, A. G. Bonfante, Automated Patent Classification Using Word Embedding, 16th IEEE International Conference on Machine Learning and Applications (2017) 408–411doi:10.1109/ICMLA.2017.0-127.

[21] W. Zaghloul, S. M. Lee, S. Trimi, Text classification: neural networks vs support vector machines, Industrial Management & Data Systems 109 (5) (2009) 708–717. doi:10.1108/02635570910957669.

[22] J. Devlin, M. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (2019) 4171–4186.
URL https://www.aclweb.org/anthology/N19-1423

[23] J. Howard, S. Ruder, Universal Language Model Fine-tuning for Text Classification, Proceedings ofthe 56th Annual Meeting ofthe Association for Computational Linguistics (Volume 1: Long Papers) (2018) 328–339.

[24] J. Lee, J. Hsiang, PatentBERT: Patent classification with fine-tuning a pre-trained BERT model (2019). arXiv:1906.02124.
URL http://arxiv.org/abs/1906.02124v2

[25] L. H. Li, P. H. Chen, C.-J. Hsieh, K.-W. Chang, Efficient contextual representation learning without softmax layer (2019). arXiv:1902.11269.

[26] S. Kumar, Y. Tsvetkov, Von mises-fisher loss for training sequence to sequence models with continuous outputs (2018). arXiv:1812.04616v3.

[27] D. Mollá, D. Seneviratne, Overview of the 2018 ALTA shared task: Classifying patent applications, in: Proceedings of the Australasian Language Technology Association Workshop 2018, Dunedin, New Zealand, 2018, pp. 84–88.
URL https://www.aclweb.org/anthology/U18-1011

[28] E. D'hondt, S. Verberne, C. Koster, L. Boves, Text Representations for Patent Classification, Computational Linguistics 39 (3) (2013) 755–775. doi:10.1162/COLI.

[29] S. Tong, D. Koller, Support Vector Machine Active Learning with Applications to Text Classification, Journal of Machine Learning Research (2001) 45–66.

[30] M. R. Frank, D. Autor, J. E. Bessen, E. Brynjolfsson, M. Cebrian, D. J. Deming, M. Feldman, M. Groh, J. Lobo, E. Moroa, D. Wang, H. Youn, I. Rahwan, Toward understanding the impact of artificial intelligence on labor, PNAS 116 (14) (2019) 6531–6539. doi:10.1073/pnas.1900949116.