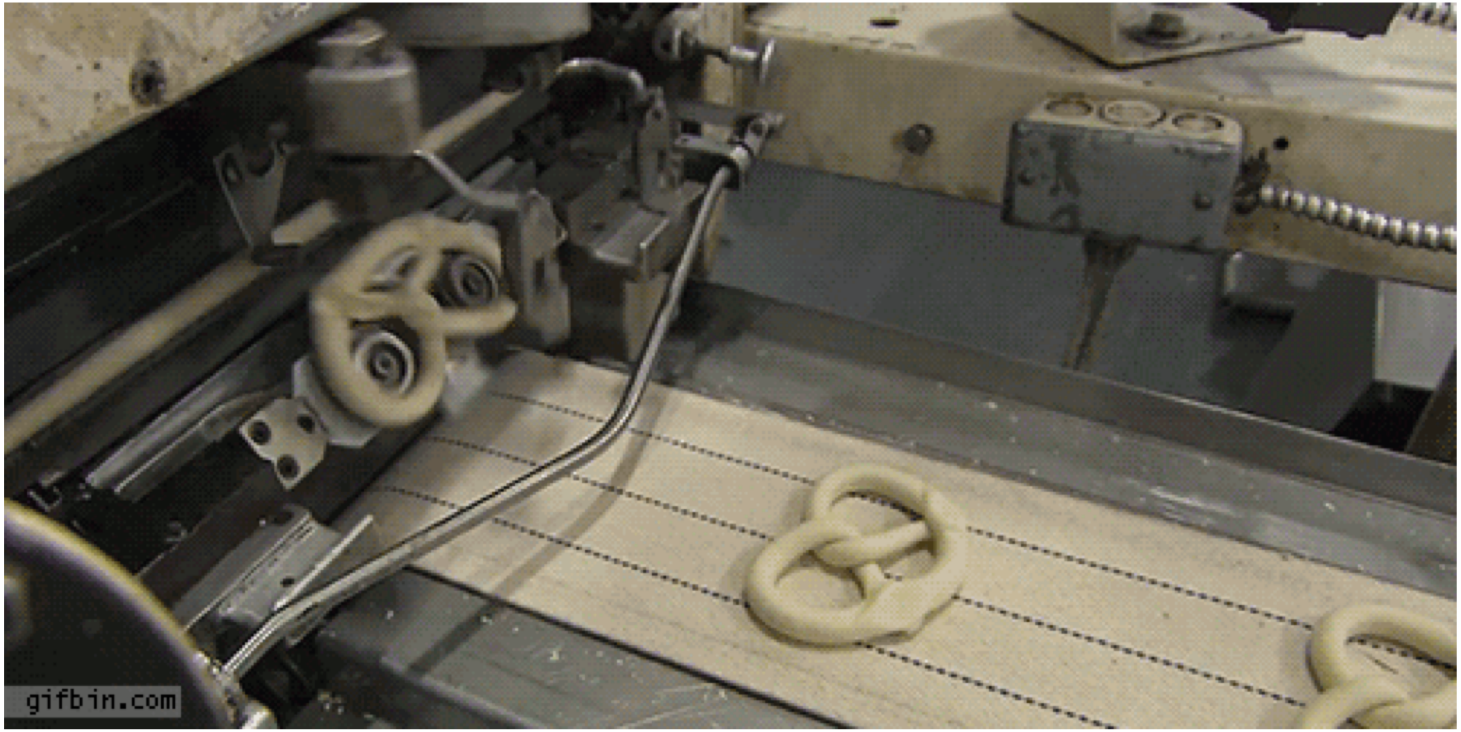
Binary Patent Classification Methods for Few Annotated Samples

September 12, 2019

Benjamin Meindl
Ingrid Ott
Ulrich Zierahn

PatentSemTech, 2019

Source: https://www.technologyreview.com/; https://giphy.com/



Source: https://giphy.com

Patents can serve as an indicator for technological progress

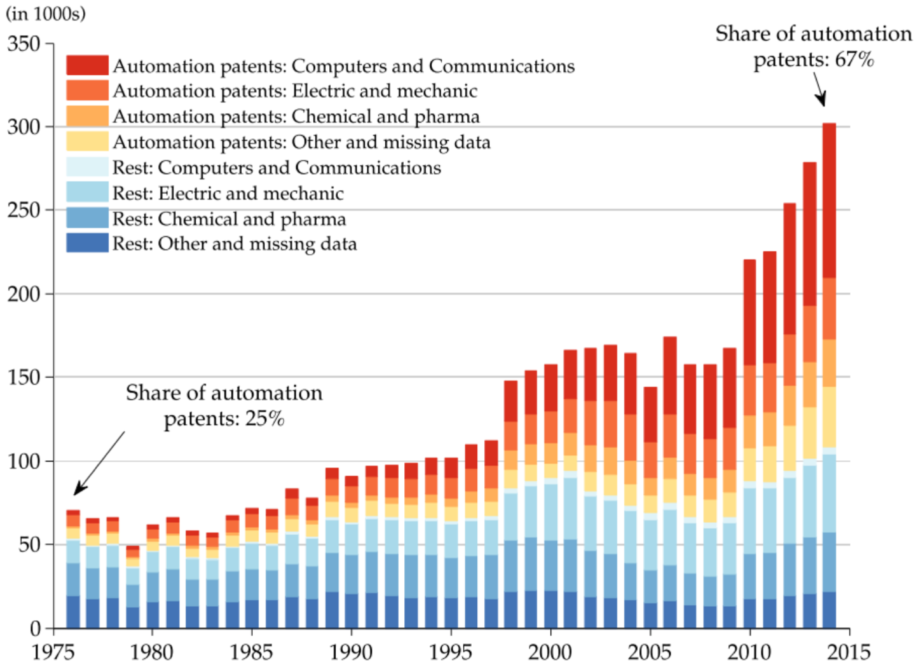
- Impact of technological resources on corporate diversification (Silverman 2002)
- Impact of automation technology on labor market (Mann & Püttmann 2017)
- Relation of wages and automation innovations (Dechezleprêtre et al. 2019)

Patent classifications are the basis for the analysis of their impact

- Patent to industry mapping (Van Looy et al. 2015)
- Patents used vs. produced (Silvermann 2002)
- Patents describing “Automats” (Mann & Püttmann 2017)

Manual classification is resource intensive!

Figure 2: Patents, 1976-2014



Share of automation patents: 67%

Share of automation patents: 25%

Note: See text for classification of automation patents and assignment of patents to categories.

Source: USPTO, Google, Hall, Jaffe, and Trajtenberg (2001) and own calculations.

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

Patent classification algorithms

- Patent category classification
  - Support vector machine and multinomial Naïve Bayes performed best at benchmark (Fall et al. 2003)
  - Support vector machine (Benites et al. 2018)
  - Word embedding and neural network (Li et al. 2018)
  - Word embedding and BERT[1] neural network (Lee & Hsiang 2019))

- Other classification
  - Bernoulli Naïve Bayes (Mann & Püttmann)

[1] Bidirectional Encoder Representations from Transformers

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

F. Benites, S. Malmasi, M. Zampieri, *Classifying Patent Applications with Ensemble Methods*, Proceedings of Australasian Language Technology Association Workshop (2018) 89–92

J. Lee, J. Hsiang, PatentBERT: Patent classification with fine-tuning a pre-trained BERT model (2019)

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

Objective

- Binary patent classification
- Small sample size
- Simple implementation (complexity, resources)

We compare the accuracy of binary classification algorithms

- Bernoulli naive Bayes (BernoulliNB)
- Support vector machine (SVC)
- Random forest
- k-nearest neighbor
- SpaCy CNN
- SpaCy CNN pre-trained/fine tuned

Data

- USPRO-2m dataset (title, abstract, sub-class
- Robotic related patents (G05B, G05D)
- 100, 200, 500, 1500, 5000 patents for training
- 250 patents for evaluation

G05B

CONTROL OR REGULATING SYSTEMS IN GENERAL; FUNCTIONAL ELEMENTS OF SUCH SYSTEMS; MONITORING OR TESTING ARRANGEMENTS FOR SUCH SYSTEMS OR ELEMENTS

G05D

SYSTEMS FOR CONTROLLING OR REGULATING NON-ELECTRIC VARIABLES

Implementation

- USPTO-2m
- Scikit-learn
  - Lemmatization (WordNet Lemmatizer)
  - Stopword removal
  - TF/IDF (ngram 2,3)
  - Classification (grid search)
- SpaCy
  - Pre-training
  - Classification

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**

The highly efficient method is simple to implement

- Low resources
  - Pre-training 2 days
  - Training 10 min
- Simple code
  - SpaCy is well documented and simple to use
  - Prodigy (license required) requires only few lines of code with similar results

Code do train the algorithm

Pretrain model

```
>> python -m spacy pretrain 'data/pretrain/pretrain_G06'
'data/pretrain/pretrained_model_vec'
```

Create dataset

```
>> python3 -m prodigy db-in patent_G05D
"data/categories/G05D_G05B_500.jsonl”
>> python3 -m prodigy db-in patent_G05D_eval
"data/categories/G05D_G05B_eval.jsonl"
```

Train algorithm

```
>> python3 -m prodigy textcat.batch-train patent_G05D en_core_web_lg --
output 'models/en_categories_B05G_250' --dropout 0.6 --batch-size 16 -n 20
--eval-id patent_G05D_eval --init-tok2vec
'data/pretrain/pretrained_model_vec/model200.bin'
```

Thank you

benjamin.meindl@tecnico.ulisboa.pt