Speeding up the Manifesto Project: Active learning strategies for efficient automated political annotations

Felix Biessmann[1], Philipp Schmidt[2]

January 29, 2018

[1]felix.biessmann@gmail.com
[2]schmidtiphil@gmail.com

Disclaimers

- (For us) This is just a hobby / open source project
- It has nothing to do with our job

Disclaimers

- (For us) This is just a hobby / open source project
- It has nothing to do with our job

Motivation

- Everyday loads of political content is published
- Too much text to handle by humans

→ Automated political bias prediction required for
  - Political scientists
  - Journalists
  - Average media consumer

??

Motivation

- Everyday loads of political content is published
- Too much text to handle by humans

→ Automated political bias prediction required for

  - Political scientists
  - Journalists
  - Average media consumer

??

Motivation

- Everyday loads of political content is published
- Too much text to handle by humans

→ Automated political bias prediction required for

  - Political scientists
  - Journalists
  - Average media consumer

??

Motivation

- Everyday loads of political content is published
- Too much text to handle by humans

→ Automated political bias prediction required for

  - Political scientists
  - Journalists
  - Average media consumer

??

Motivation

- Everyday loads of political content is published
- Too much text to handle by humans

$\rightarrow$ Automated political bias prediction required for
  - Political scientists
  - Journalists
  - Average media consumer

??

Motivation

- Everyday loads of political content is published
- Too much text to handle by humans

→ Automated political bias prediction required for
  - Political scientists
  - Journalists
  - Average media consumer

??

Motivation

- Machine Learning models need fresh training data
- But annotation budget is often limited:
  - Temporal constraints (before elections) ?
  - Online news media (too much content)

→ How to select which texts to annotate?

Motivation

- Machine Learning models need fresh training data
- But annotation budget is often limited:
  - Temporal constraints (before elections) **?**
  - Online news media (too much content)

$\rightarrow$ How to select which texts to annotate?

Motivation

- Machine Learning models need fresh training data
- But annotation budget is often limited:
  - Temporal constraints (before elections) **?**
  - Online news media (too much content)

→ How to select which texts to annotate?

Motivation

- Machine Learning models need fresh training data
- But annotation budget is often limited:
  - Temporal constraints (before elections) **?**
  - Online news media (too much content)

→ How to select which texts to annotate?

Motivation

- Machine Learning models need fresh training data
- But annotation budget is often limited:
    - Temporal constraints (before elections) **?**
    - Online news media (too much content)

$\rightarrow$ How to select which texts to annotate?

Active Learning

- Train best model with limited budget
- Annotate difficult ones[3] first
- Why?
  - Intuition:
    Model learns most from difficult examples
  - Math:
    Gradient of loss function larger for difficult examples

[3] For which model is most uncertain.

Active Learning

- Train best model with limited budget
- Annotate difficult ones[3] first
- Why?
    - Intuition:
      Model learns most from difficult examples
    - Math:
      Gradient of loss function larger for difficult examples

[3] For which model is most uncertain.

Active Learning

- Train best model with limited budget
- Annotate difficult ones[3] first
- Why?
  - Intuition:
    Model learns most from difficult examples
  - Math:
    Gradient of loss function larger for difficult examples

[3] For which model is most uncertain.

Active Learning

- Train best model with limited budget
- Annotate difficult ones[3] first
- Why?
  - Intuition:
    Model learns most from difficult examples
  - Math:
    Gradient of loss function larger for difficult examples

[3] For which model is most uncertain.

Active Learning

- Train best model with limited budget
- Annotate difficult ones[3] first
- Why?
  - Intuition:
    Model learns most from difficult examples
  - Math:
    Gradient of loss function larger for difficult examples

[3] For which model is most uncertain.

Data

- All annotated German texts from `https://manifestoproject.wzb.eu/`
- Only texts with more than 1000 observed labels

Preprocessing

- Basic text cleaning (regexps, stopwords)
- Unigram Bag-of-Words features
- Hashing Vectorizer

Classification Model: Multinomial Logistic Regression

Manifestocode prediction is modelled as

$$p(y = k|\mathbf{x}) = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}} \text{ with } z_k = \mathbf{w}_k^\top \mathbf{x}. \tag{1}$$

With

- Labels $y \in \{1, 2, \ldots, K\}$ (manifesto code)
- $\mathbf{w}_1, \ldots, \mathbf{w}_K \in \mathbb{R}^d$ weight vectors of kth manifesto code
- L_2 norm regularization of weights

Active Learning Strategies

- Random Baseline: Uniform random sampling
- Uncertainty Sampling: Only top-prediction counts

$$\mathbf{x}_i = \underset{i,k}{\operatorname{argmax}}\,(1 - p(y = k|\mathbf{x}_i, \mathbf{W})) \tag{2}$$

- Entropy Sampling: All predictions count

$$\mathbf{x}_i = \underset{i}{\operatorname{argmax}} \sum_k p(y = k|\mathbf{x}_i, \mathbf{W}) \log(p(y = k|\mathbf{x}_i, \mathbf{W})) \tag{3}$$

- Margin Sampling: Top 2 predictions count

$$\mathbf{x}_i = \underset{i}{\operatorname{argmin}}\,(p(y = k_1|\mathbf{x}_i, \mathbf{W}) - p(y = k_2|\mathbf{x}_i, \mathbf{W})) \tag{4}$$

Active Learning Experiments

- Train model on 1%, 10%, 20%, . . . , 100% of training data
- Vary sampling strategies to select from unlabelled texts
- Evaluate agains 'perfect' model trained on all data

Results: 'Perfect' Reference Model

manifesto code	precision	recall	f1-score	support
107	0.60	0.48	0.53	774
201	0.51	0.55	0.53	1194
202	0.63	0.57	0.60	983
305	0.46	0.59	0.52	783
403	0.52	0.48	0.50	1281
411	0.39	0.60	0.47	1535
501	0.61	0.55	0.58	1380
502	0.65	0.41	0.50	587
503	0.46	0.52	0.49	2083
506	0.63	0.48	0.54	1026
605	0.56	0.44	0.49	576
701	0.59	0.39	0.47	1123
avg / total	0.50	0.48	0.48	17559

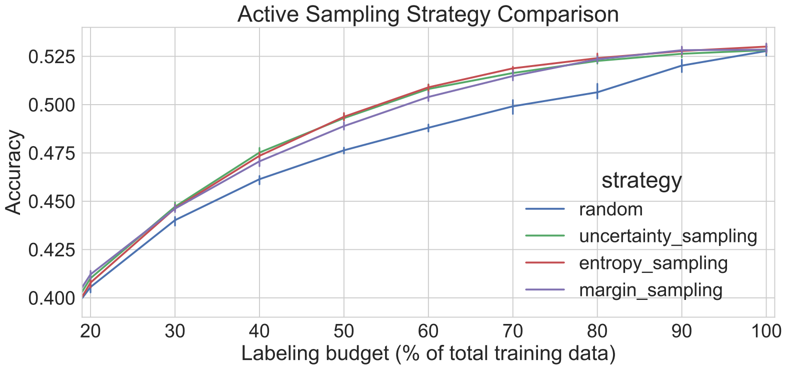
Table: Precision, recall, F1 score and number of instances per class.

Results: Out-of-domain Predictions

Table: **Tested on manifesto quasi-sentences**

	prec.	recall	f1-score	N
cducsu	0.26	0.58	0.36	2030
fdp	0.38	0.28	0.33	2319
gruene	0.47	0.20	0.28	3747
linke	0.30	0.47	0.37	1701
spd	0.26	0.16	0.20	2278
total	0.35	0.31	**0.30**	12075

Active Learning Results



Median accuracy and the 5th/95th percentile across 100 repetitions

Conclusion

- Political text analysis requires automation
- Automation requires annotations for training models
- Limited budged for annotations of political texts
- Active Learning
  - Helps to select which texts to annotate
  - Perfect model with 80% of data
  - Almost perfect (over 95%) with 50% of data

→ Active learning can speed up political annotations.

- Demo: `http://rightornot.info`

References