Speeding up the manifesto project: Active learning strategies for efficient automated political annotations

November 3, 2017

Abstract

The Manifestoproject Corpus is an exceptional data source for political education as it combines political texts with valuable annotations by human experts. However the amount of data human annotators can label is very limited compared to the ever increasing amount of political texts published in manifestos, news media and social networks. The discrepancy between labeling budget and data that needs to be labelled highlights the necessity of automated annotations by means of machine learning (ML). Taking into account label consistency across human experts, which is often not perfect and thus requires to collect at least three labels per data point, this discrepancy is even worse.

When only a small fraction of the available data can be labelled in order to train an ML model, the priority of which samples should be labelled first is important: Some samples are easy to classify - those should not be labelled with high priority, as the classifier will not learn much from them. For these samples it is relatively safe to have them labelled automatically by an ML model without wasting the time of human annotators. Other data points are difficult for the ML model; when labels for those are obtained first, the model will reach its optimal classification performance faster.

In this study we leverage this insight, known in the ML community as active learning. We present offline experiments on the manifesto corpus showing that active learning strategies significantly speed up training of ML models for manifesto code annotations. This shows the potential of ML methods as assistive technology for political science. To demonstrate the benefits of this approach we present an implementation of a web based active learning annotation system that can be readily used for speeding up the manifesto annotations as well as annotations of other political texts.

1 Introduction

2 Results

Basically: it works, see also Figure 1.

Acknowledgements

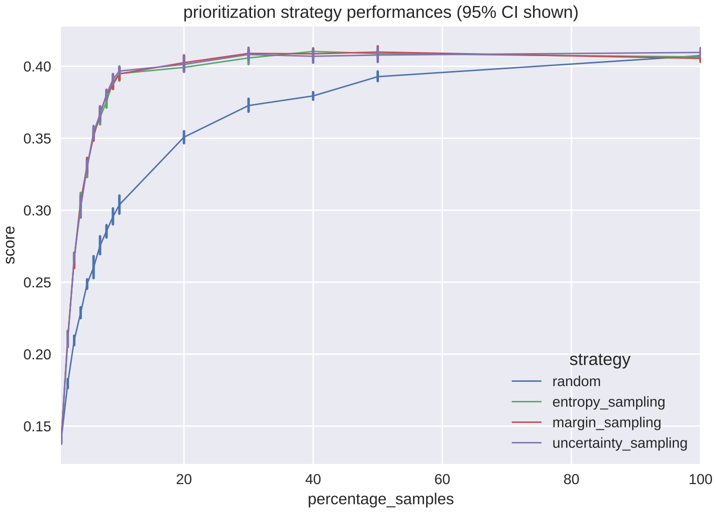
prioritization strategy performances (95% CI shown)
score
0.40
0.35
0.30
0.25
0.20
0.15
percentage_samples
20
40
60
80
100
strategy
random
entropy_sampling
margin_sampling
uncertainty_sampling

Figure 1: