Naive Bayesian vs. Keyword-Based Spam Filtering

by I. Androutsopoulos, J. Koutsias, K. V.Chandrinos and C. D. Spyropoulos

Adrian Scoică
(as2270@cam.ac.uk)

February 12, 2013

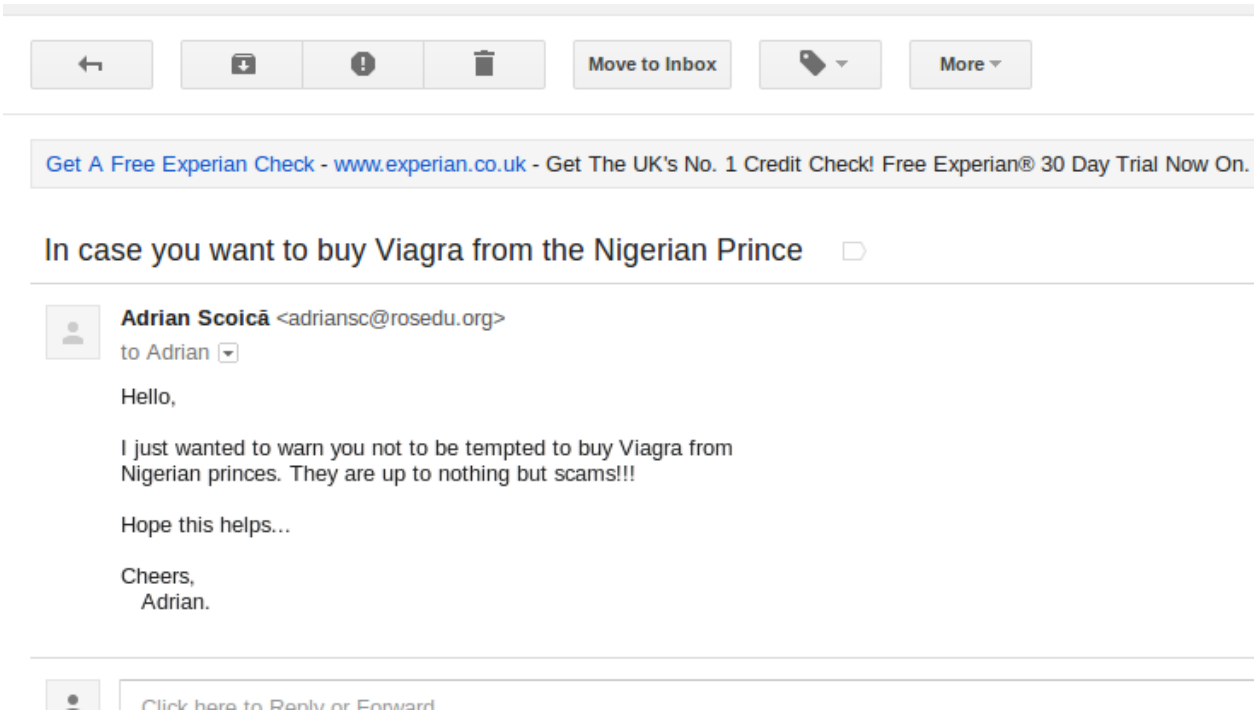
1. Problem Insights
2. Naive Bayes Classifiers
3. Results
4. Conclusions

Automatic Spam Detection

Is it a straightforward classification problem?

Automatic Spam Detection

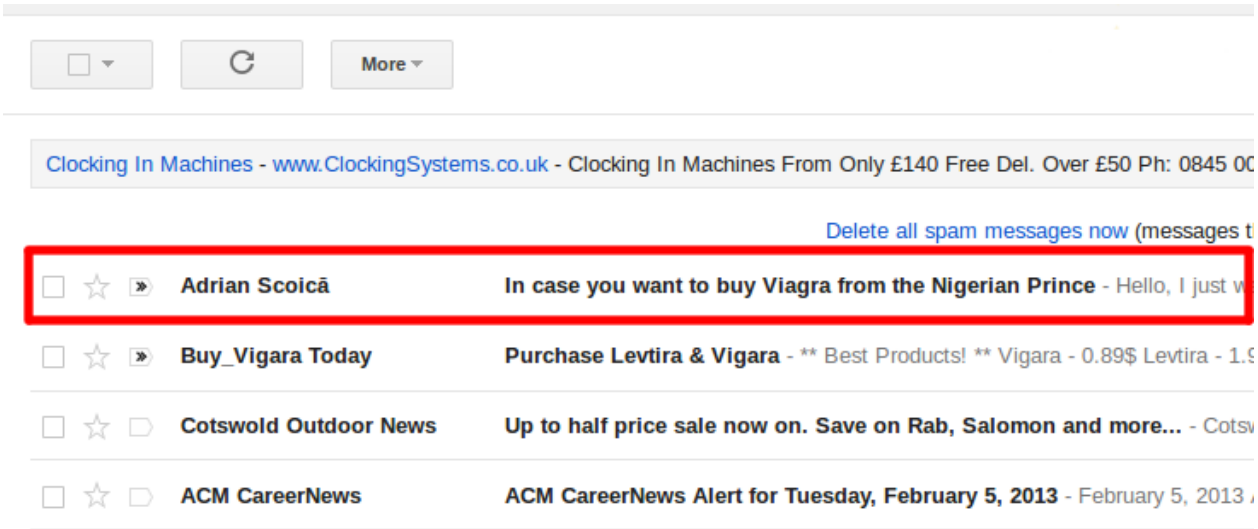
Is it a straightforward classification problem?



Intention is more important than content...

Automatic Spam Detection

Is it a straightforward classification problem?



... but language is a good predictor.

Building a Corpus

Authors compiled the publicly-available **PU1** corpus:

- 618 legitimate messages
- 481 spam messages

Real emails

Building a Corpus

Authors compiled the publicly-available **PU1** corpus:

- 618 legitimate messages
- 481 spam messages

Real emails, but **encrypted** due to privacy issues:

From: spammer@spamcompany.com *To: spamtarget@provider.com* *Subject: Get rich now !* *Click here to get rich ! Try it now !*	⇒	*Subject: 1 2 3 4* *5 6 7 1 2 4 8 9 3 4*

⇒ Paper focuses on **word-features** only.

Building the Classifier

Documents are modelled with vectors

$$\vec{x}_{Doc} = \langle x_1, x_2, x_3, \ldots x_n \rangle$$

Building the Classifier

Documents are modelled with vectors

$$\vec{x}_{Doc} = \langle x_1, x_2, x_3, \ldots x_n \rangle$$

using only binary, word-only attributes $x_i = \begin{cases} 1 & word_i \in Doc \\ 0 & \text{otherwise} \end{cases}$

Building the Classifier

Documents are modelled with vectors

$$\vec{x}_{Doc} = \langle x_1, x_2, x_3, ...x_n \rangle$$

using only binary, word-only attributes $x_i = \begin{cases} 1 & word_i \in Doc \\ 0 & \text{otherwise} \end{cases}$

which have the highest mutual information with the class variable, C

$$\sum_{x \in \{0,1\}, c \in \{spam, legitimate\}} P(X = x, C = c) log \frac{P(X = x, C = c)}{P(X = x)P(C = c)}$$

Building the Classifier

Documents are modelled with vectors

$$\vec{x}_{Doc} = \langle x_1, x_2, x_3, ...x_n \rangle$$

using only binary, word-only attributes $x_i = \begin{cases} 1 & word_i \in Doc \\ 0 & \text{otherwise} \end{cases}$

which have the highest mutual information with the class variable, C

$$\sum_{x \in \{0,1\}, c \in \{spam, legitimate\}} P(X = x, C = c) log \frac{P(X = x, C = c)}{P(X = x)P(C = c)}$$

Using the Naive Bayes assumption, we can compute:

$$P(spam|\vec{x}_{Doc}) = \frac{P(spam) * \prod_{i=1}^{n} P(X_i = x_i|spam)}{\sum_{c \in \{spam, legitimate\}} P(C = c) \prod_{i=1}^{n} P(X_i = x_i|C = c)}$$

Evaluating Spam Filters

In spam filtering, **precision** if more important than **recall**.

Blocking a legitimate message is λ times more expensive than letting a spam message pass, so when **do** we block?

Evaluating Spam Filters

In spam filtering, **precision** if more important than **recall**.

Blocking a legitimate message is λ times more expensive than letting a spam message pass, so when **do** we block?

$$\frac{P(spam|\vec{x}_{Doc})}{P(legitimate|\vec{x}_{Doc})} > \lambda \Leftrightarrow P(spam|\vec{x}_{Doc}) > t = \frac{\lambda}{\lambda+1}$$

The paper uses threshold t to analyze three scenarios:

$$t = 0.999 \qquad t = 0.9 \qquad t = 0.5$$

Evaluating Spam Filters

In spam filtering, **precision** if more important than **recall**.

Blocking a legitimate message is λ times more expensive than letting a spam message pass, so when **do** we block?

$$\frac{P(spam|\vec{x}_{Doc})}{P(legitimate|\vec{x}_{Doc})} > \lambda \Leftrightarrow P(spam|\vec{x}_{Doc}) > t = \frac{\lambda}{\lambda+1}$$

The paper uses threshold t to analyze three scenarios:

$t = 0.999$		$t = 0.9$		$t = 0.5$
cautious	$\leftarrow$	average	$\rightarrow$	aggressive

How do we evaluate performance?

Evaluating Spam Filters

In spam filtering, **precision** if more important than **recall**.

Blocking a legitimate message is λ times more expensive than letting a spam message pass, so when **do** we block?

$$\frac{P(spam|\vec{x}_{Doc})}{P(legitimate|\vec{x}_{Doc})} > \lambda \Leftrightarrow P(spam|\vec{x}_{Doc}) > t = \frac{\lambda}{\lambda+1}$$

The paper uses threshold t to analyze three scenarios:

$t = 0.999$		$t = 0.9$		$t = 0.5$
cautious	$\leftarrow$	average	$\rightarrow$	aggressive

How do we evaluate performance? $WErr = \frac{\lambda n_{L\rightarrow S} + n_{S\rightarrow L}}{\lambda N_L + N_S}$

Comparing to the baseline (no filter): $TCR = \frac{WErr^{baseline}}{WErr} = \frac{N_S}{\lambda n_{L\rightarrow S} + n_{S\rightarrow L}}$

Results

Filter used	t	No. attr.	TCR
(a) NB (bare)	0.5	50	4.90
(b) NB (stop-list)		50	4.95
(c) NB (lemmatizer)		100	4.29
(d) NB (stop-list + lemmatizer)		100	4.53
Keyword patterns		-	2.01

Results

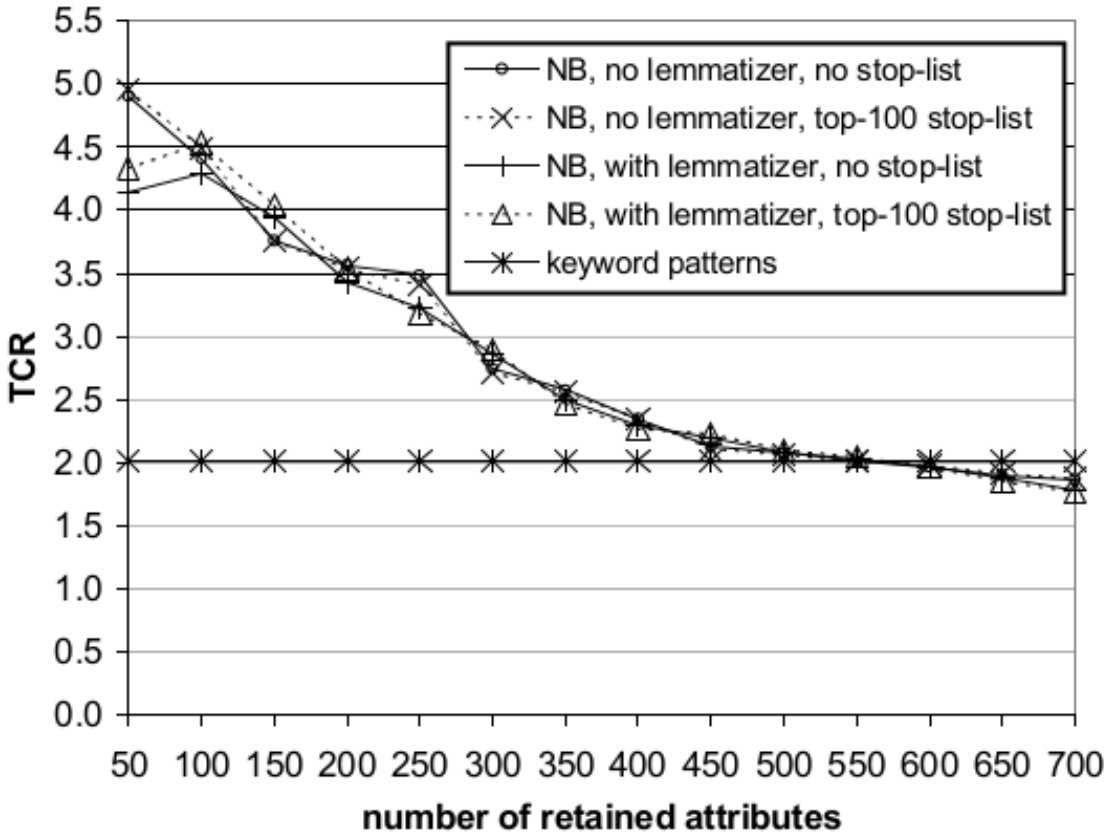
Filter used	t	No. attr.	TCR
(a) NB (bare)	0.5	50	4.90
(b) NB (stop-list)		50	4.95
(c) NB (lemmatizer)		100	4.29
(d) NB (stop-list + lemmatizer)		100	4.53
Keyword patterns		-	2.01
(a) NB (bare)	0.9	100	2.20
(b) NB (stop-list)		150	2.28
(c) NB (lemmatizer)		100	2.83
(d) NB (stop-list + lemmatizer)		100	2.56
Keyword patterns		-	1.40

Results

Filter used	t	No. attr.	TCR
(a) NB (bare)	0.5	50	4.90
(b) NB (stop-list)		50	4.95
(c) NB (lemmatizer)		100	4.29
(d) NB (stop-list + lemmatizer)		100	4.53
Keyword patterns		-	2.01
(a) NB (bare)	0.9	100	2.20
(b) NB (stop-list)		150	2.28
(c) NB (lemmatizer)		100	2.83
(d) NB (stop-list + lemmatizer)		100	2.56
Keyword patterns		-	1.40
(a) NB (bare)	0.999	700	0.15
(b) NB (stop-list)		700	0.15
(c) NB (lemmatizer)		50	0.11
(d) NB (stop-list + lemmatizer)		600	0.11
Keyword patterns		-	0.04

Results

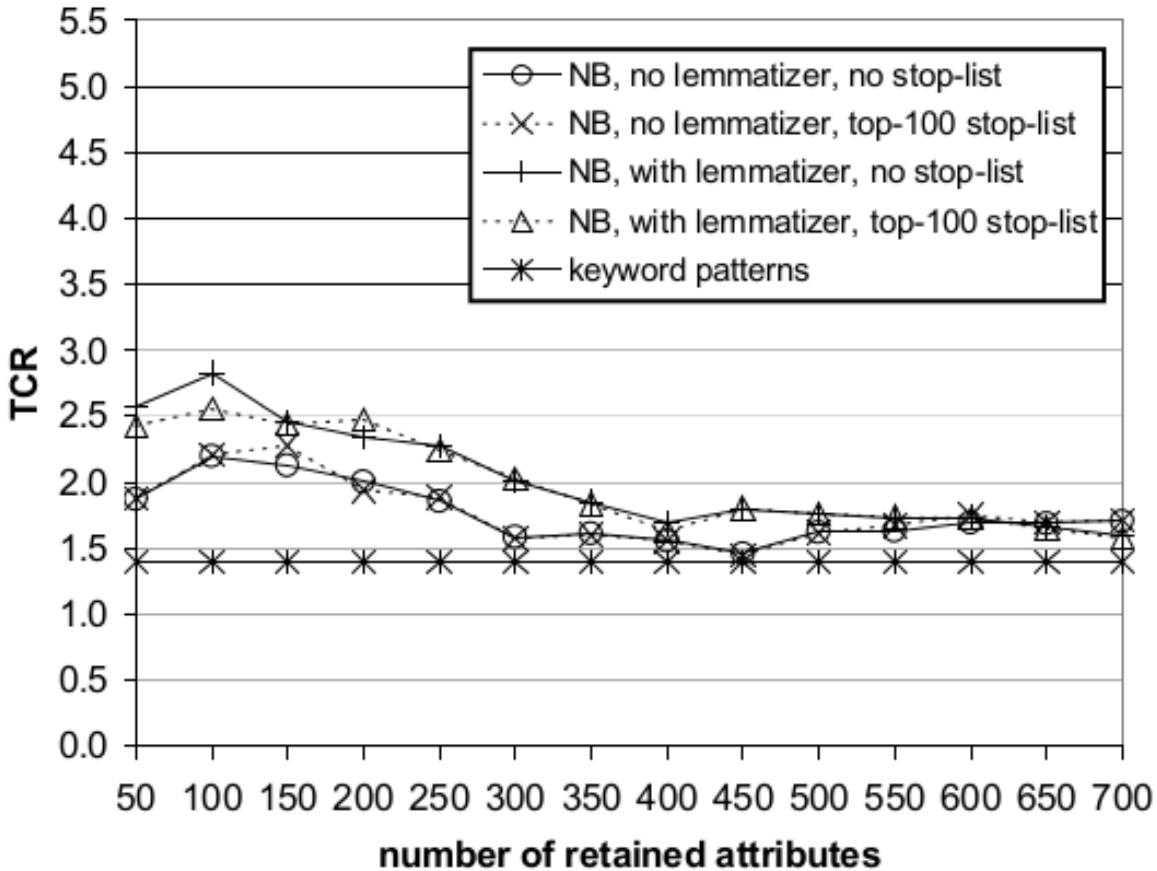
How many attributes do we need?



TCR of the filters for $t = 0.5$ $(\lambda = 1)$

Results

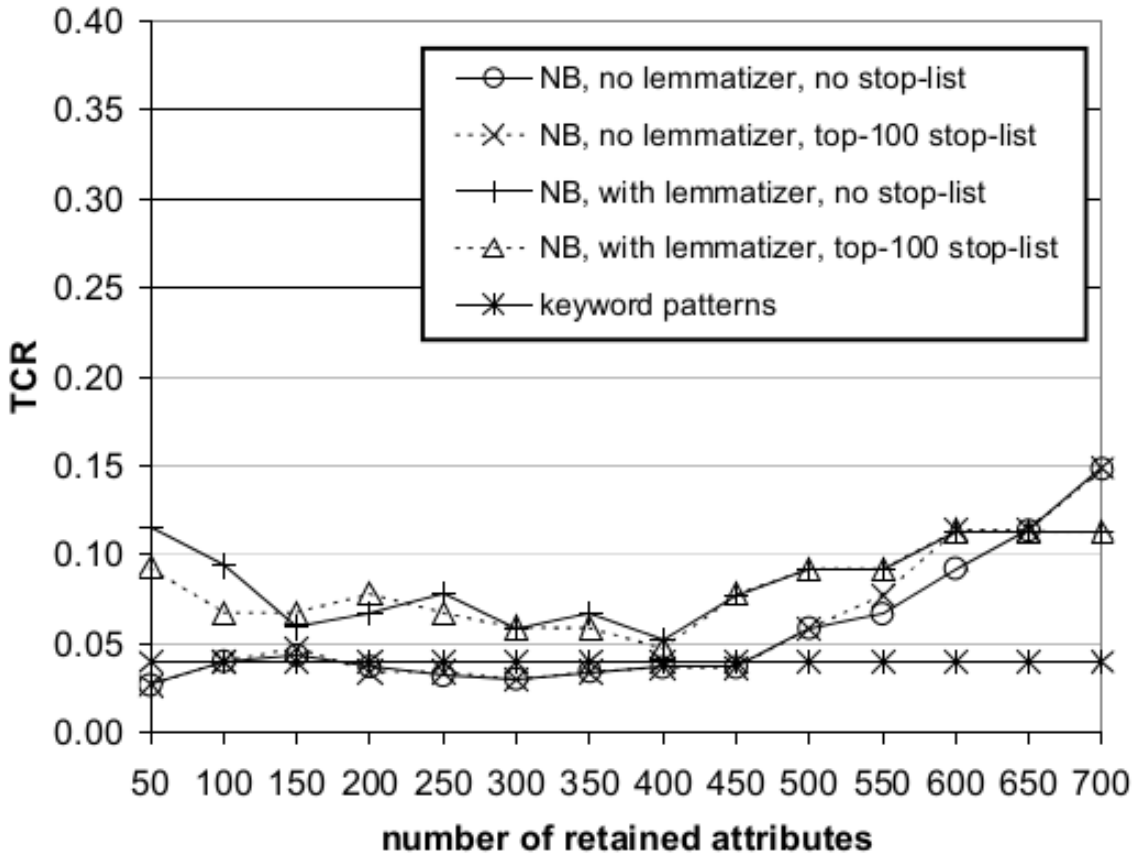
How many attributes do we need?



TCR of the filters for $t = 0.9$ ($\lambda = 9$)

Results

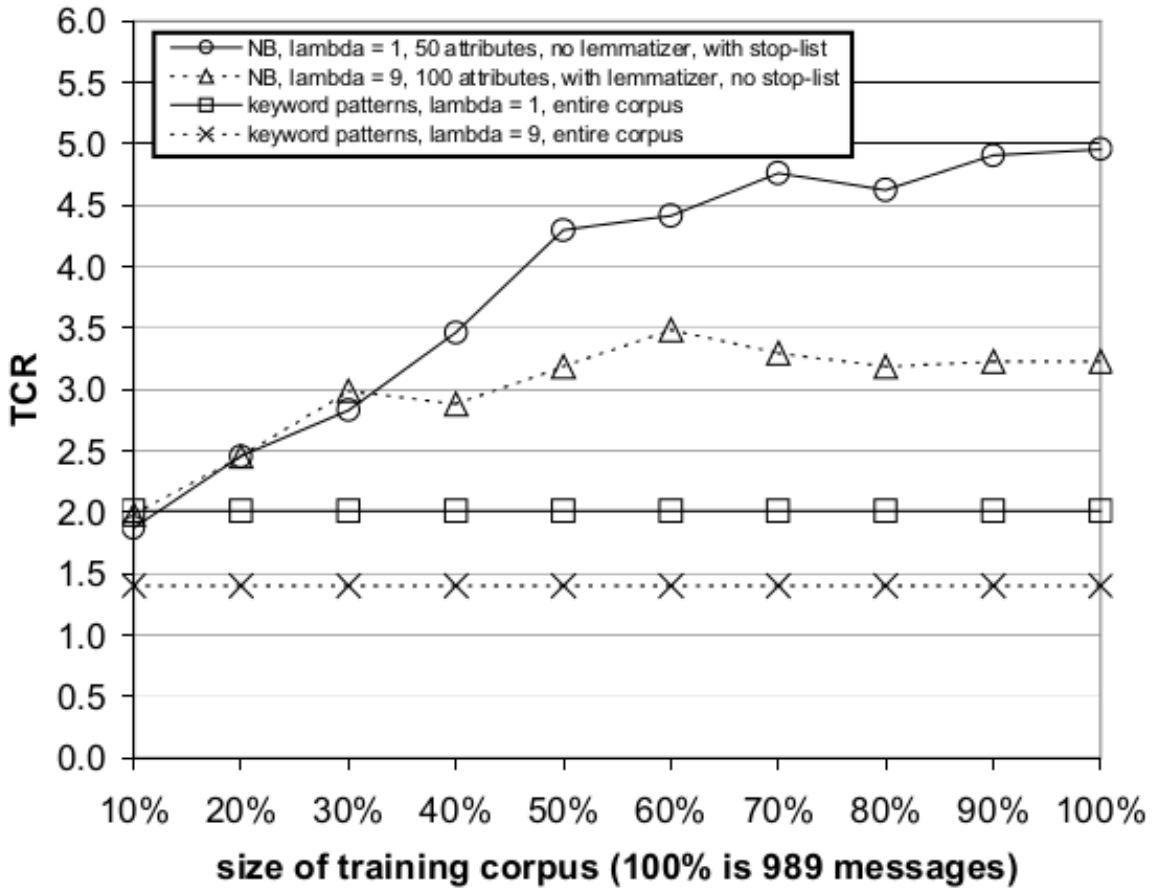
How many attributes do we need?



TCR of the filters for $t = 0.999$ $(\lambda = 999)$

Results

How much training data is enough?



TCR for variable size of training corpus

Conclusions

Paper key points and contributions:

- Introduces cost-sensitive evaluation.
- Proves effectiveness of automatic spam filtering.
- Proves stop-lists don't improve performance with MI attribute selection.
- Shows classifiers are trainable even with small amounts of data.

Thank you!