Automatic Text Categorization: The Case of Authorship Detection

A gentle introduction to Machine Learning for students of Social Studies and Digital Humanities

Martin Holub
holub@ufal.mff.cuni.cz

Jakub Genči
genci.jakub@gmail.com

Charles University in Prague, Faculty of Mathematics and Physics,
Institute of Formal and Applied Linguistics

Motivation of the talk

- To demonstrate elementary principles of Machine Learning (ML) — practical procedures
- Intuitively clear examples without technical details
- A typical Natural Language Processing task
- Students' work in an introductory ML course — first experience with a "real" ML task
- Follow-up to the previous introductory lesson on natural language processing and machine learning — Class #8 (April 5, 2022)

Outline of the talk — logical parts

I. Elementary principles of Machine Learning

- Classification tasks in ML, training and test data, feature vectors, learning process, confusion matrices, evaluation, overfitting

II. General remarks on Text Categorization tasks

- Typical ML approach: models based on n-gram features
- Examples: Topic Detection, Native Language Identification

III. Authorship recognition

- Definition of the task, available data, authorship vs. authorial style
- Data preprocessing, development data sets
- Experiments and results

IV. Recap

- What we learned about automatic authorship recognition
- What we learned about general machine learning principles

Authorship recognition — motivation example

A randomly chosen passage from a classical Czech novel

Uchopil ji za ni a vtiskl na růžové prsty žhavý polibek.

»Ano, ruku . . . I o mou i o tvou jde.«

»Nastaly snad překážky?«

»Nikoliv . . . Zůstává při tom, že máš v neděli dopoledne přijít a rodiče o mne požádat.«

»Proč má být tedy zle? . . . «

»Dověděla jsem se zrazeným tajemstvím, že ti budou položeny podmínky, které prý nebudeš ani chtít a snad ani moci vyplnit.«

Can you recognize the author?

Authorship recognition — motivation example

Can you recognize the author?

Uchopil ji za ni a vtiskl na růžové prsty žhavý polibek.

»Ano, ruku … I o mou i o tvou jde.«

»Nastaly snad překážky?«

»Nikoliv … Zůstává při tom, že máš v neděli dopoledne přijít a rodiče o mne požádat.«

»Proč má být tedy zle? …«

»Dověděla jsem se zrazeným tajemstvím, že ti budou položeny podmínky, které prý nebudeš ani chtít a snad ani moci vyplnit.«

Possible authors:

(1) A. Stašek (2) J. Neruda
(3) J. Arbes (4) K. Klostermann
(5) F. X. Šalda (6) T. G. Masaryk

Authorship recognition — motivation example

Can you recognize the author?

```
Possible authors:
    (1) A. Stašek            (2) J. Neruda
    (3) J. Arbes             (4) K. Klostermann
    (5) F. X. Šalda          (6) T. G. Masaryk
```

How to prepare for such a task?

- Read a lot of texts by given authors
- Try to recognize their "characteristic styles"
- Keep this knowledge in your mind
- Compare the given passage with the different styles of the authors
- Predict the author with the most similar style

Authorship recognition — motivation example

How to recognize the author?

- Compare the given passage with the different styles of the authors
- Predict the author with the most similar style

In fact, you do NOT recognize authorship,
BUT author's style!

Problems

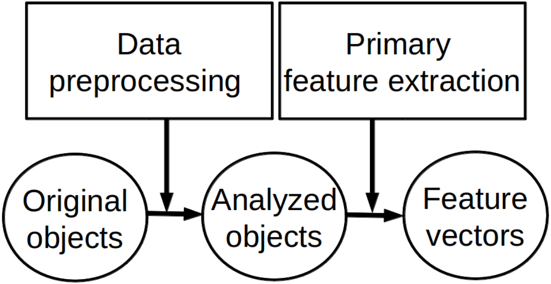
- What if the styles of different authors are NOT different?
  ... or, only a little bit?
- What if the given passage does NOT reflect its author's typical style?

$\longrightarrow$ **Authorship recognition cannot be always certain,**
especially if the given passage is short

Machine Learning works similar!

- Computer learns from **training data set**
- It needs a large number of **training examples**
  = a lot of example passages with known authors
- It creates a **model** with encoded "knowledge" of different authors
  = "training procedure" or "learning process"
- Then the model can be used to predict even authors of new passages that were not seen during the learning process
- Different authors are interpreted as different **categories**, also called **"target values"**, which are assigned to passages

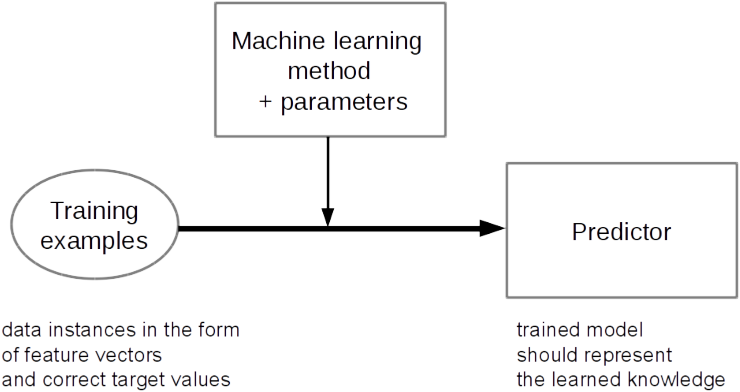
Each training example should be first analyzed . . . and represented as a feature vector



Each passage is analyzed and represented as an exactly organized set of characteristic properties (= feature vector)

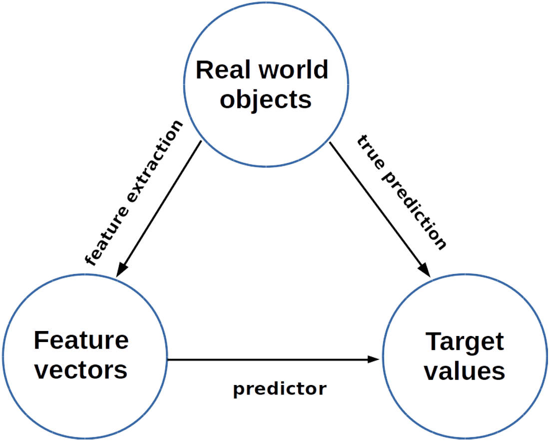
Supervised learning process: From training examples to a trained model/predictor

A predictor is the output of machine learning process



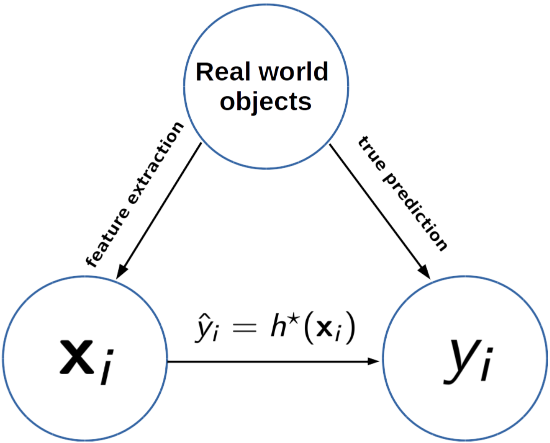
If target values are **discrete**, the predictor can be also called **classifier**

Predictor is a function



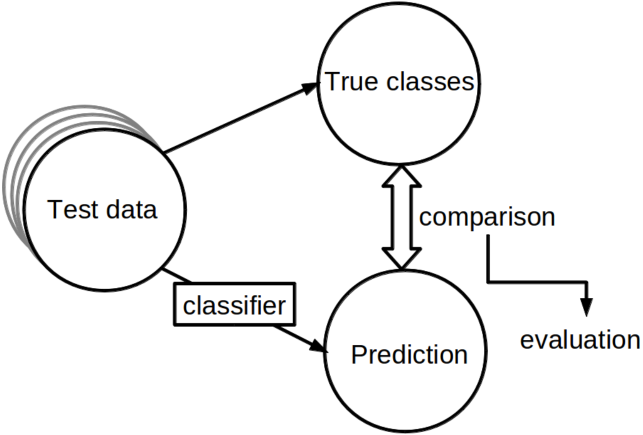
Categories that should be predicted are called **target values**

Prediction function should be found and chosen as a "best/optimal" hypothesis



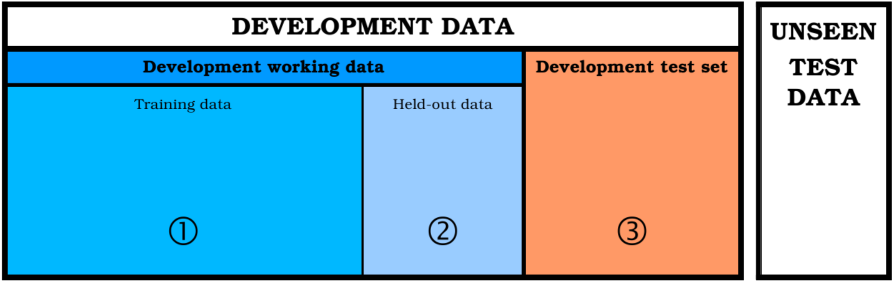
From the mathematical point of view, the learning process is an **optimization problem**

Finally, predictor should be evaluated



Test data set is a set of example passages with known authors that were **NOT used during the learning process**

Working with available data



Development data set is the only data that the developer can use

Development test set should simulate the “real” test set

Basic evaluation measures: accuracy and error rate

- **Accuracy** is the proportion of corretly predicted examples in a test set
- **Error rate** $= 1 -$ accuracy

Examples

- There are 200 test examples. In the evaluation experiment, 180 of them were correctly predicted. What is the accuracy? What is the error rate?
- A model was improved and accuracy increased from 95 % to 96 %.
  – Is it a good/great improvement?

A detailed view on evaluation results: Confusion Matrix (CM)

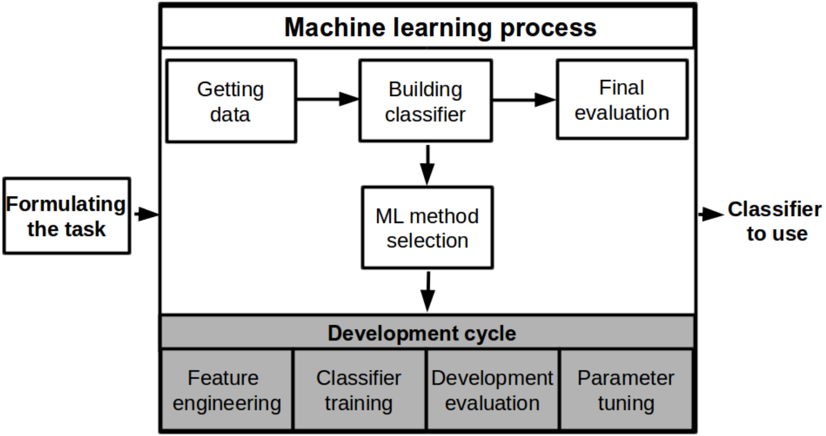
Example of evaluation results

— 6 target categories to be predicted, 431 test examples

predicted \ true	01	02	03	04	05	06
01	85	3	7	0	0	0
02	0	64	18	14	1	1
03	0	1	41	1	0	0
04	2	1	0	57	1	1
05	1	1	4	1	53	5
06	0	1	16	0	0	51

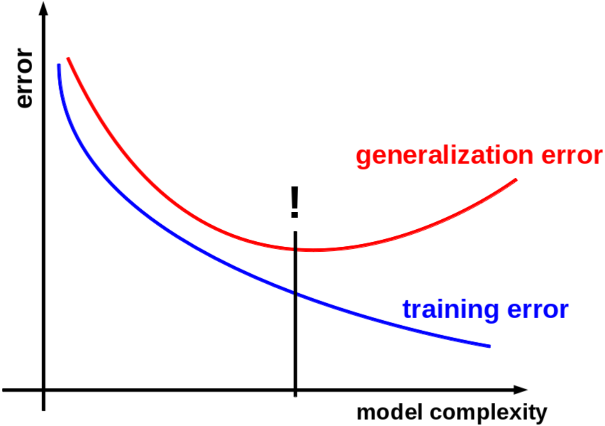
```
Total predictions      = 431
Correct predictions    = 351
Accuracy               = 81.44 %
```

Typical machine learning development cycle

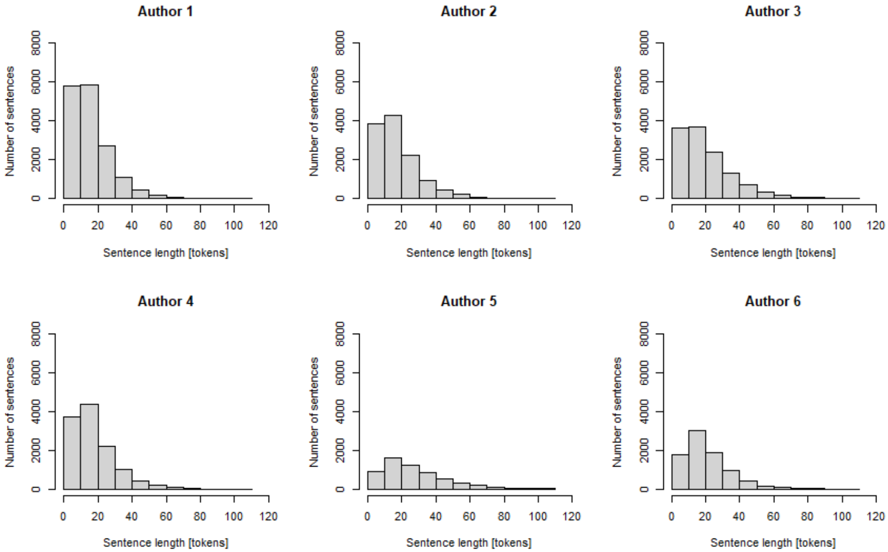


The central problem of Machine Learning (both theoretical and practical)

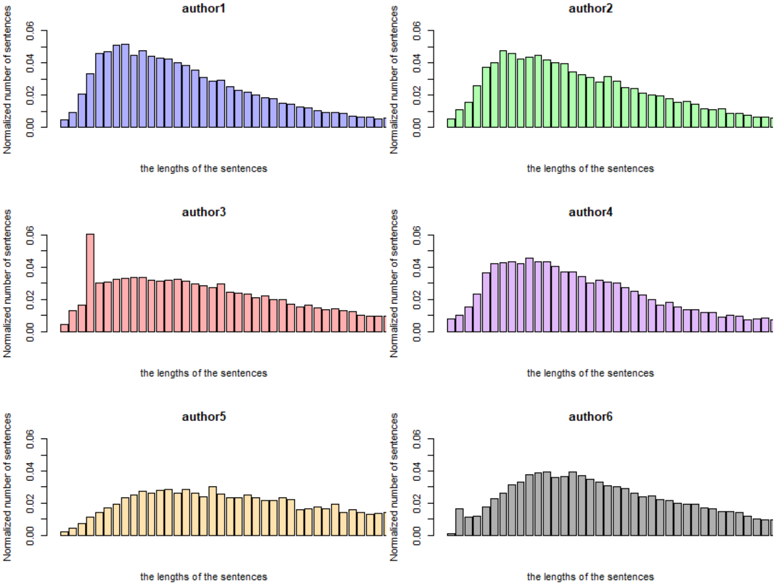
How to minimize the generalization error?



Sentence length distribution – does it differ?



Sentence length distribution – normalized!



Exploiting sentence length distributions — Part I

1) Predictions based on Kolmogorov-Smirnov statistic: **35 %** accuracy

2) Predictions using SVM learning algorithm: **40 %** accuracy

Exploiting sentence length distributions — Part II
Recognizing one author vs. others

A binary classification problem

Accuracy

- Author 1: 77 %
- Author 2: 78 %
- Author 3: 78 %
- Author 4: 75 %
- Author 5: 85 %
- Author 6: 78 %

Our first model

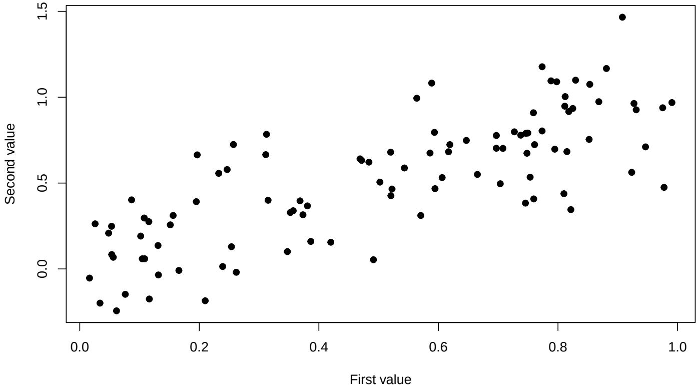
We used a learning method called **Support Vector Machines** (SVM)

- originally created by Vapnik and Chervonenkis in 1963
- further developed in 90s
- still commonly used (not only) with NLP tasks

Let us introduce it briefly.

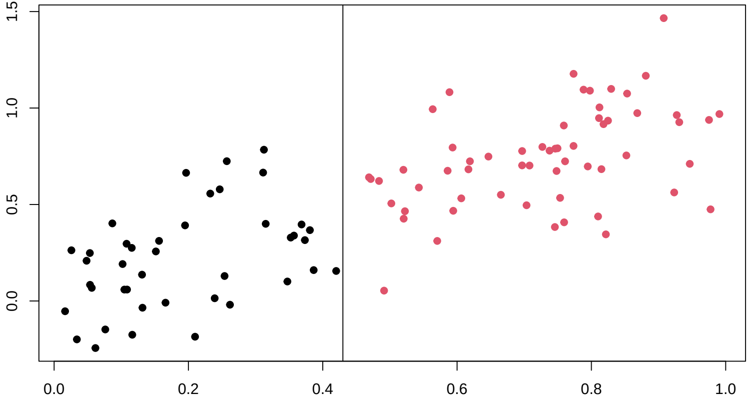
Firstly, what are vectors?

Each point in this graph visualizes one feature vector and represents one passage



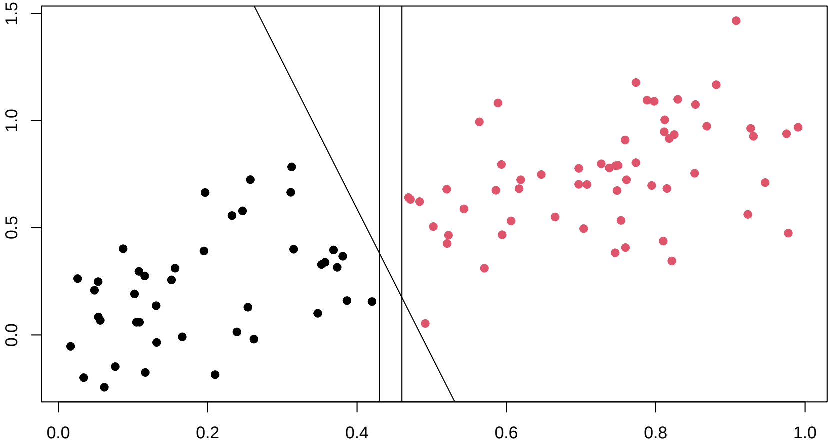
How do SVMs work?

They allow us to differentiate between two groups by looking for a *separating hyperplane*



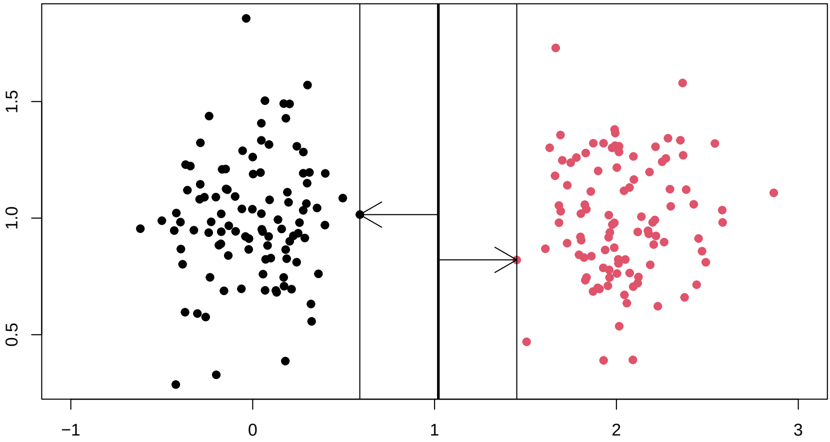
How do SVMs work?

However, there are many hyperplanes...



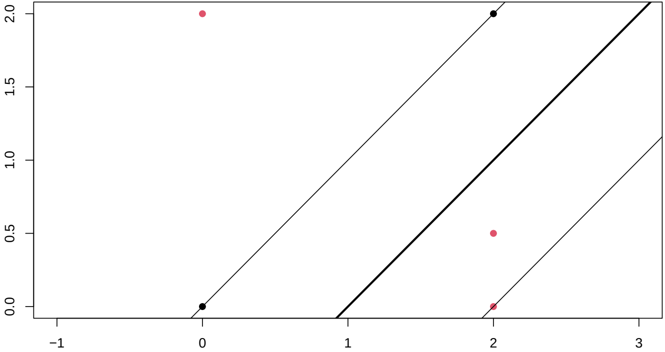
Which hyperplane is the best?

We want to maximize distance between the hyperplane and the vectors

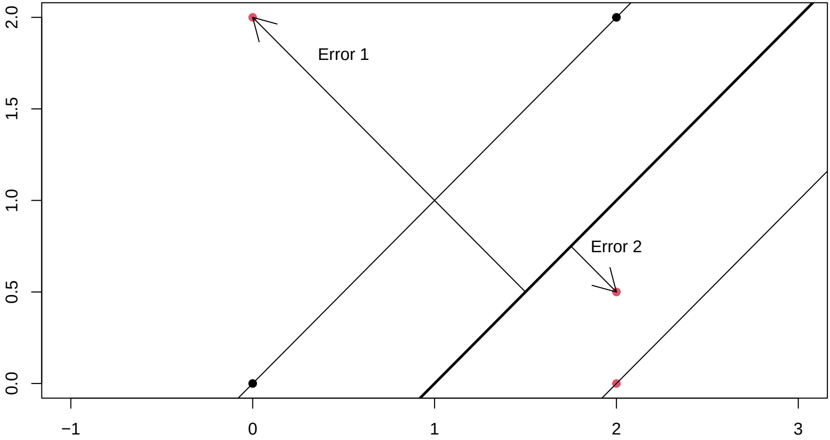


When this does not work?

We assumed that we can divide our vectors into two groups. That does not happen every time...



How to fix this problem?



Non-separating hyperplane

How is it handled?

- we assign an error to each point which is inside the margin or incorrectly classified
- we want to minimize the sum of errors during the learning
- usually, sum of errors has some weight called **cost**

Cost is a **hyper parameter** - we have to find its optimal value during the training. Non-optimal values of cost lead to overfitting.

- small value of cost $\implies$ large margin
- high value of cost $\implies$ no points inside the margin

From binary to multi-class classification

We only did binary classification. What if we have more than 2 classes?

We use so called **one-vs-all** approach

- for n classes, we train n binary classifiers
- each classifier must give us some probability
- we choose the class with the highest probability

Back to the authorship detection

We want to get from the **n-grams in the text** to some **numeric feature vectors**

- one vector is one passage
- we start with number of occurences of an n-gram in a passage
- re-scaling to a value between 0 and 1
  - relative term frequency
  - weighted term frequency

Training the model (part 1)

We start with a table of n-gram frequencies:

	author	X.	můj	být	jenž	samý	na	s	ten	v	všechen
1	1	0.004975124	0.014925373	0.089552239	0.009950249	0.004975124	0.009950249	0.004975124	0.014925373	0.019900498	0.009950249
2	1	0.000000000	0.000000000	0.044776119	0.014925373	0.000000000	0.004975124	0.009950249	0.024875622	0.014925373	0.009950249
3	1	0.004761905	0.000000000	0.052380952	0.009523810	0.000000000	0.004761905	0.000000000	0.004761905	0.004761905	0.004761905
4	1	0.000000000	0.004950495	0.059405941	0.004950495	0.000000000	0.009900990	0.004950495	0.039603960	0.019801980	0.000000000
5	1	0.000000000	0.000000000	0.036269430	0.005181347	0.000000000	0.005181347	0.005181347	0.015544041	0.010362694	0.000000000
6	1	0.000000000	0.000000000	0.009216590	0.013824885	0.000000000	0.004608295	0.018433180	0.004608295	0.013824885	0.000000000
7	1	0.005235602	0.000000000	0.036649215	0.005235602	0.000000000	0.010471204	0.015706806	0.020942408	0.010471204	0.000000000
8	1	0.030769231	0.000000000	0.025641026	0.005128205	0.000000000	0.015384615	0.015384615	0.015384615	0.005128205	0.000000000
9	1	0.015000000	0.000000000	0.040000000	0.010000000	0.000000000	0.020000000	0.010000000	0.000000000	0.005000000	0.000000000
10	1	0.010309278	0.015463918	0.030927835	0.010309278	0.000000000	0.015463918	0.015463918	0.030927835	0.000000000	0.005154639
11	1	0.000000000	0.000000000	0.009569378	0.023923445	0.000000000	0.004784689	0.009569378	0.033492823	0.000000000	0.004784689
12	1	0.010000000	0.005000000	0.020000000	0.000000000	0.000000000	0.010000000	0.010000000	0.020000000	0.030000000	0.000000000
13	1	0.030612245	0.000000000	0.015306122	0.000000000	0.000000000	0.015306122	0.000000000	0.010204082	0.010204082	0.005102041
14	1	0.000000000	0.000000000	0.028846154	0.014423077	0.000000000	0.019230769	0.019230769	0.014423077	0.014423077	0.004807692
15	1	0.000000000	0.000000000	0.027472527	0.021978022	0.000000000	0.000000000	0.021978022	0.010989011	0.032967033	0.005494505
16	1	0.000000000	0.000000000	0.008849558	0.008849558	0.000000000	0.008849558	0.004424779	0.039823009	0.030973451	0.022123894
17	1	0.010471204	0.000000000	0.010471204	0.010471204	0.000000000	0.005235602	0.000000000	0.020942408	0.010471204	0.000000000
18	1	0.009852217	0.000000000	0.019704433	0.009852217	0.000000000	0.009852217	0.004926108	0.009852217	0.014778325	0.000000000
19	1	0.000000000	0.000000000	0.009756098	0.014634146	0.004878049	0.019512195	0.009756098	0.000000000	0.000000000	0.004878049

Training the model (part 2)

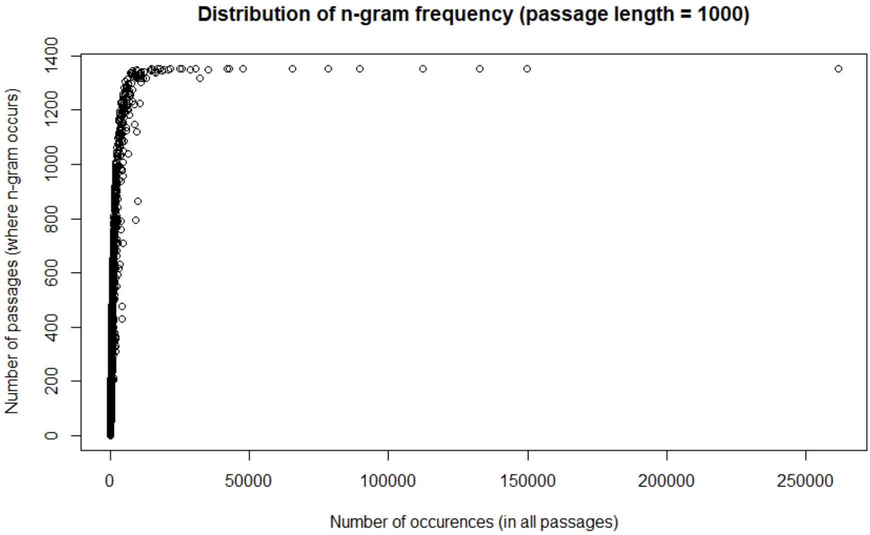
Next steps are:

1. estimating the value of cost (hyper parameter)
2. training the model
3. evaluate the model and (if needed) change the value of cost
    - we use the devel(opment) data set in this step

After this, we can evaluate the trained model using test passages and predict their authors

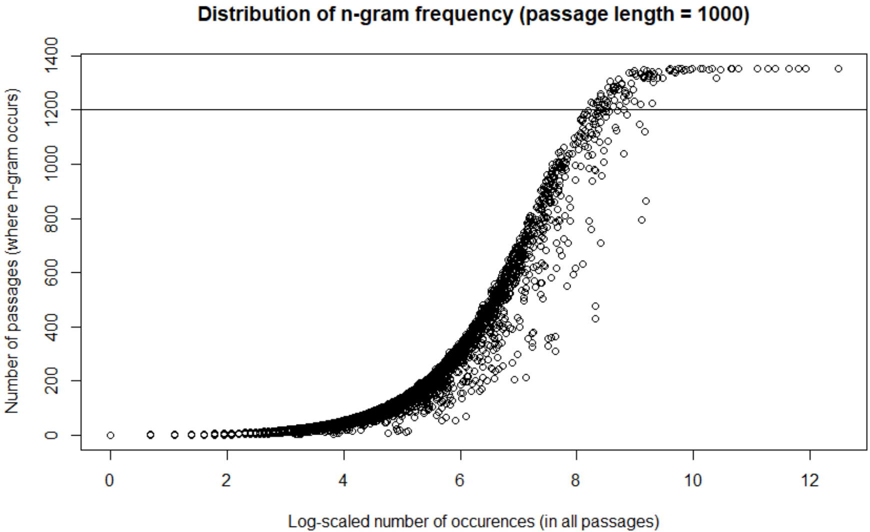
Which data did we use?

Let us look at the n-gram frequencies...



Which data did we use?

...or use a better way to visualize them:



And the results were...

Trained on train and devel data sets with average passage length of 1000

Tested on test data set with **average passage length of 1000**

```
        true
pred.       01      02      03      04      05      06
   01       88       0       0       0       0       0
   02        0      71       0       0       0       0
   03        0       0      86       0       0       0
   04        0       0       0      73       0       0
   05        0       0       0       0      55       0
   06        0       0       0       0       0      58
Total predictions         = 431
Correct predictions       = 431
Accuracy                  = 100.00 %
```

And the results were...

Trained on train and devel data sets with average passage length of 1000

Tested on test data set with **average passage length of 200**

true						
pred.	01	02	03	04	05	06
01	429	4	2	4	3	1
02	1	333	9	0	3	3
03	0	5	406	0	1	0
04	2	7	4	353	3	1
05	5	4	4	7	263	0
06	3	2	5	1	0	284

```
Total predictions         = 2152
Correct predictions       = 2068
Accuracy                  = 96.10 %
```

Are we done?

This model has some problems:

- it is a black-box - we do not know what is important to determine the authorship
- lower number of n-grams may give us the same results
- we want to discover more about the task

Analyzing n-grams: Which ones are most helpful?

- Can we identify key n-grams that are the best indicators?
- Can we select an effective subset of all n-grams?
  $\longrightarrow$ "feature selection"
- Which n-grams are better — frequent, or rare?
  $\longrightarrow$ Is there any "measure" to recognize the good ones?
- A few experiments are shown

Reduced n-gram set – experiment 1

Using only first *m* most frequent n-grams for feature vectors

The value of m	SVM accuracy on the dev. data
20	62 %
40	75 %
60	85 %
80	93 %
100	94 %
300	99 %
1000	100 %

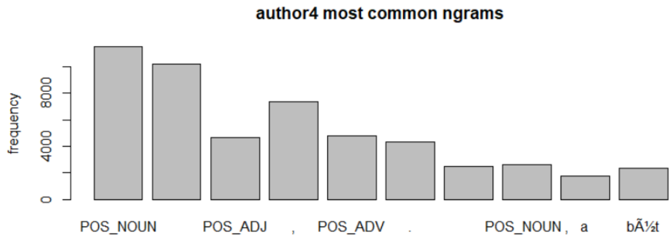
Analyzing n-grams: Can most frequent ones help?

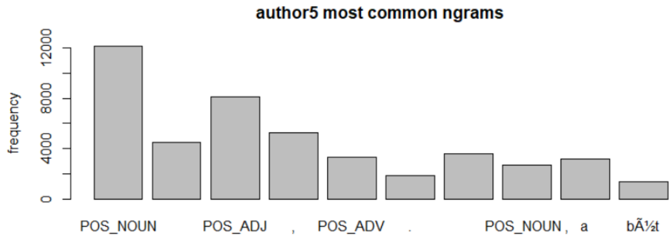
1	'POS_NOUN'	cf = 261382	df = 1352
2	'POS_VERB'	cf = 149630	df = 1352
3	'POS_ADJ'	cf = 132726	df = 1352
4	','	cf = 112382	df = 1352
5	'POS_ADV'	cf = 89798	df = 1352
6	'.'	cf = 78393	df = 1352
7	'POS_ADJ POS_NOUN'	cf = 65496	df = 1351
8	'POS_NOUN ,'	cf = 47716	df = 1352

Most frequent 20 n-grams

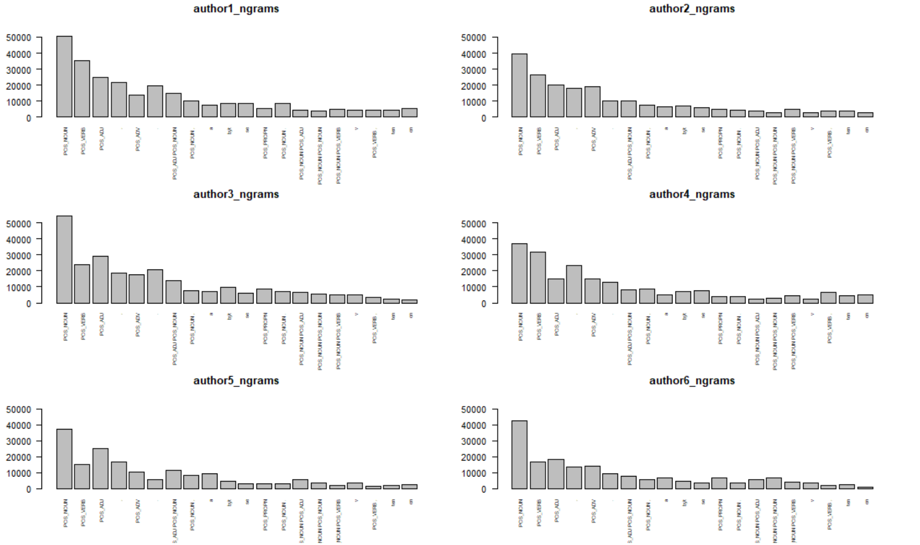
1 'POS_NOUN'
2 'POS_VERB'
3 'POS_ADJ'
4 ','
5 'POS_ADV'
6 '.'
7 'POS_ADJ POS_NOUN'
8 'POS_NOUN ,'
9 'a'
10 'být'
11 'se'
12 'POS_PROPN'
13 'POS_NOUN .'
14 'POS_NOUN POS_ADJ'
15 'POS_NOUN POS_NOUN'
16 'POS_NOUN POS_VERB'
17 'v
18 'POS_VERB ,'
19 'ten'
20 'on'

Most frequent 10 n-grams — different authors



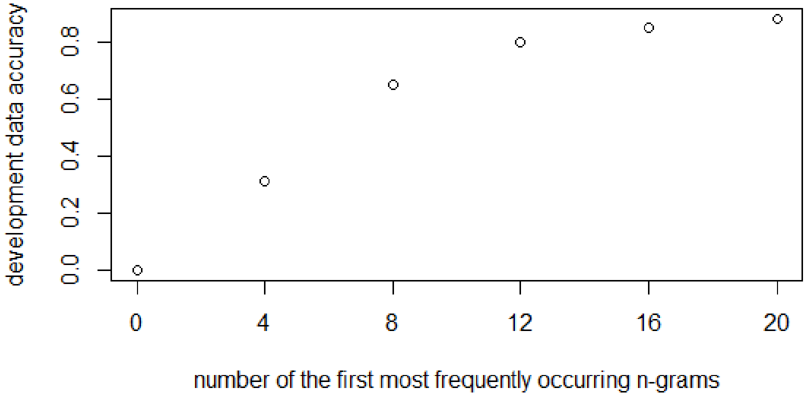


Most frequent 20 n-grams — distributions



Reduced n-gram set – experiment 2

Using only first 20 most frequent n-grams and combining their proportions to feature vectors



Dependency on the passage length

We decided that we will ignore n-grams only in the lowest and the highest 1% of passages.

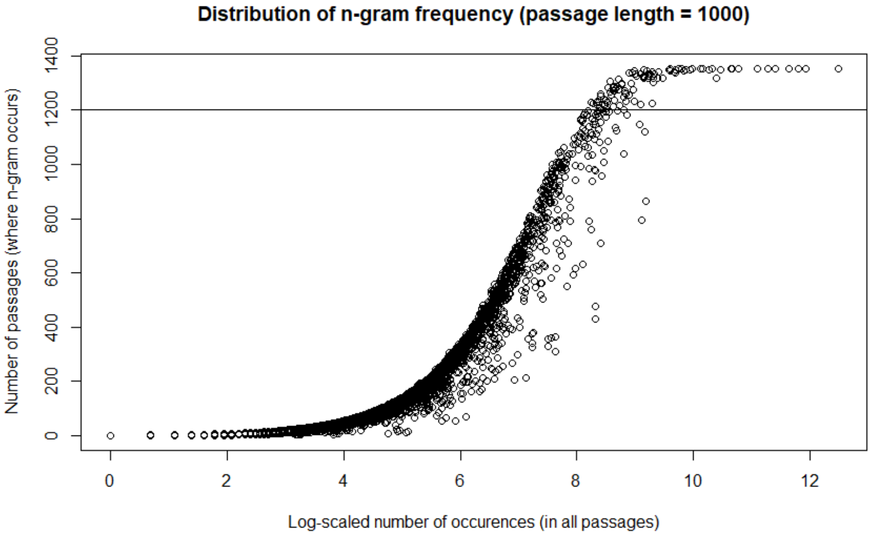
We had 4 options for training the SVM and 2 for predicting the authorship. So we did it:

Training data set	Test ($L = 1000$)	Test ($L = 200$)
Train ($L = 1000$)	99.07%	93.54%
Train + devel ($L = 1000$)	98.84%	94.28%
Train ($L = 200$)	99.30%	95.63%
Train + devel ($L = 200$)	99.30%	96.10%

Table: Table with accuracies for different predictions.

Lowering "number of passages" threshold

Reminder:



Lowering "number of passages" threshold

Trained on train and devel data sets with average passage length of 1000

Tested on test data set with average passage length of 1000

- started with n-grams which are in less than 1766 passages (99%)
- decreased the number of passages by 100
- stopped at 466

Lowering "number of passages" threshold

Results:

Max. number of passages	Correct predictions	Accuracy
466	429 / 431	99.54%
566	428 / 431	99.30%
...		
866	430 / 431	99.77%
...		
1466	427 / 431	99.07%
1566	427 / 431	99.07%
1666	371 / 431	86.08%
1766	419 / 431	97.22%

Lowering "number of occurrences" threshold

Unfortunately, we did not manage to analyze this in time. However, expectations are similar.

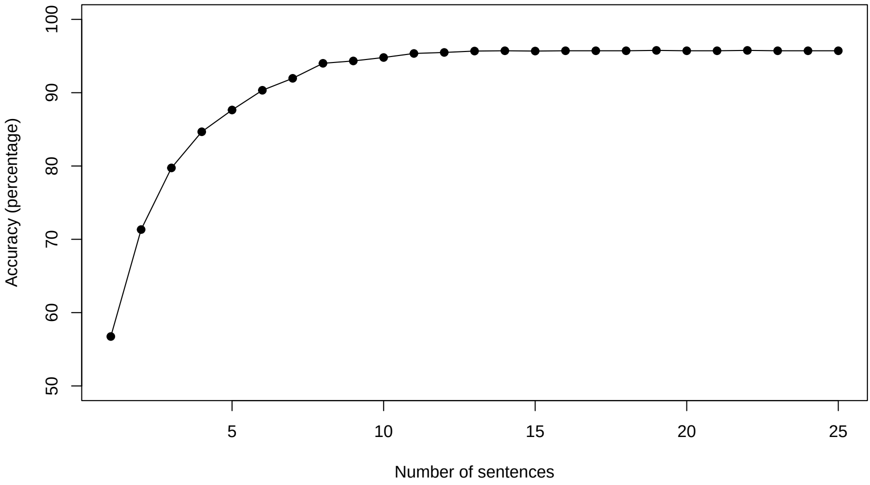
Can we identify author by just few sentences?

We trained this model on the train data with average passage length of 200 and tested it on the devel set with average passage length of 200

We took only the first sentence from each passage, then first two sentences, then first three, ..., until we used first 25 sentences from each passage (if possible)

- 26 was the maximum number of sentences in 1 passage

If we choose only first few sentences...



Recap on authorship recognition

What do our experiments tell us:

- **adding development data to the training set improves performance**
- **it is better to use longer passages for testing**
- **less frequent n-grams are probably the key to this problem**
- **few sentences are enough to predict authorship**

Recap on machine learning principles

- **Learning from the data**
- **Creating feature vectors**
- **Evaluation**
- **Generalization error**

Credits

- Data sets for this project were developed by the Digilab research group in the National Library of the Czech Republic (https://digilab.nkp.cz)
- Experiments presented here were done by students of the NPFL 054 course – "Introduction to machine learning in the R system" at Faculty of Mathematics and Physics, Charles University, namely Jakub Genči and Aibat Kossumov