



ULBS

Universitatea "Lucian Blaga" din Sibiu

FACULTATEA DE INGINERIE HERRMANN OBERTH
MASTER-PROGRAM „EMBEDDED SYSTEMS“

Machine Learning

PROFESSOR:

PROF. DR. ING. VOLOVICI DANIEL
SEF L. DR. CREȚULESCU RADU
CONF. DR. ING. MORARIU DANIEL

STUDENT:

STEFAN FEILMEIER

Text classification for German and Romanian language

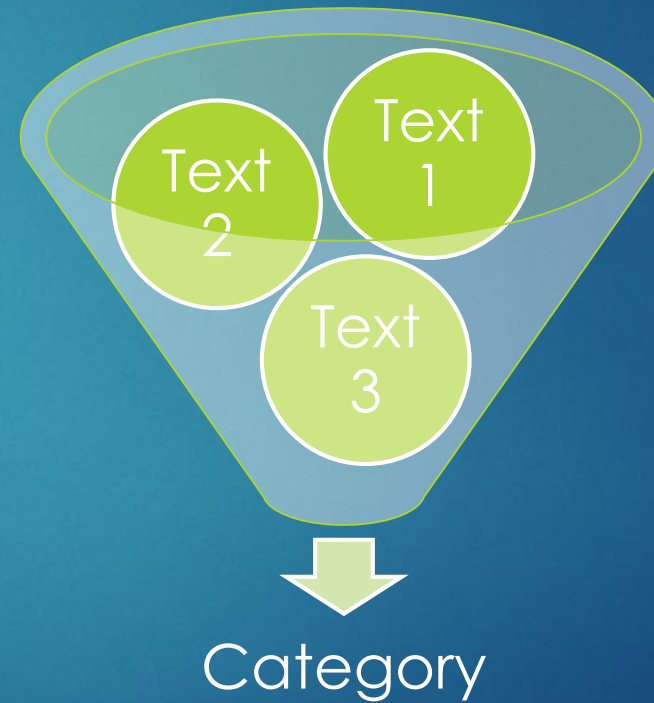
Agenda

1. What is **text classification**?
2. What is the **general approach**?
3. Why this project?
4. Generate **example data**
5. **Train and test** Naïve Bayes algorithm
6. **Performance** and **results**
7. **Conclusions**

1. What is **text classification**?

What is **text classification**?

- ▶ Find out, which **category** a given **text** belongs to?
- ▶ Used in
 - ▶ news aggregators,
 - ▶ libraries,
 - ▶ text mining,
 - ▶ document management,
 - ▶ ...



2. What is the **general approach**?

What is a **general approach**?

1. Prepare text


Example:

Wikipedia article about „**art**“

What is a **general approach**?

1. Prepare text
 1. Remove **markup code** and **punctuation**

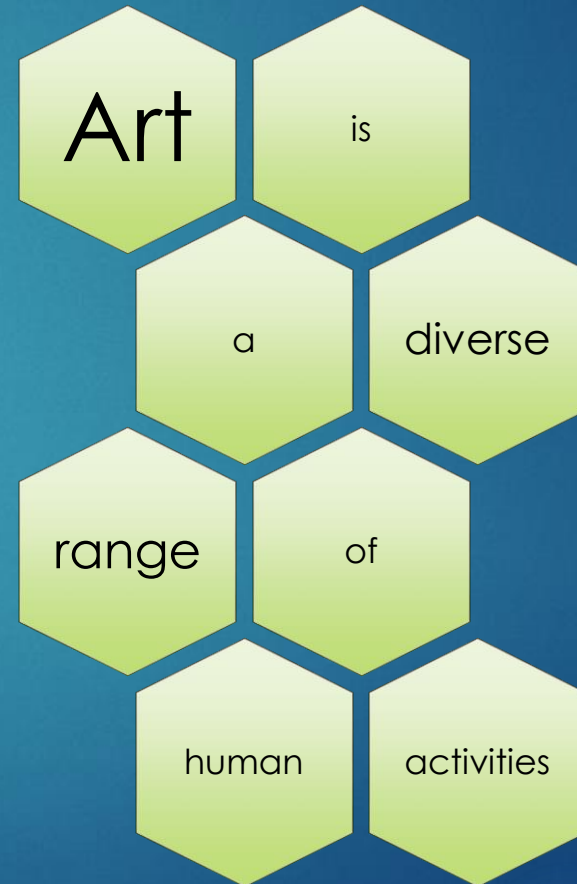
"Art" is a diverse
range of
[[human
behavior | huma
n activities]]



Art is a diverse
range of human
activities

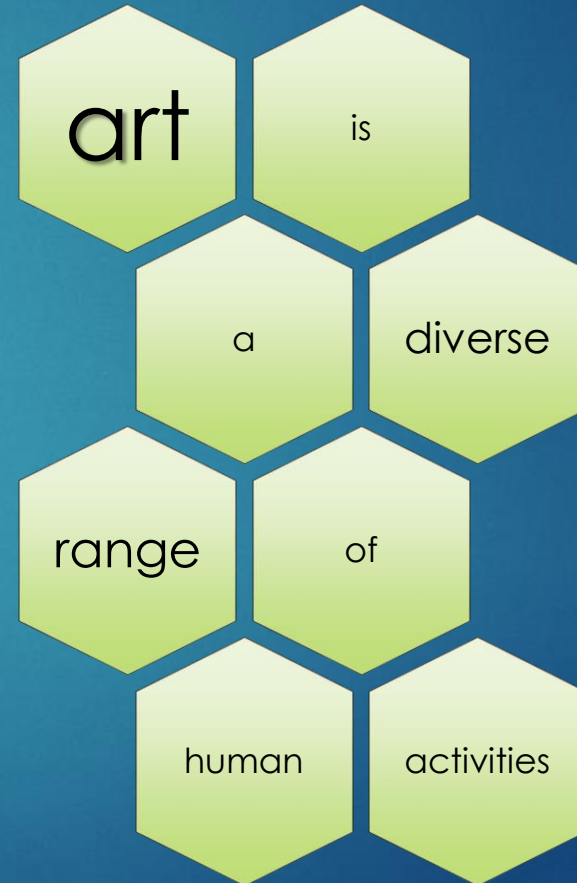
What is a **general approach**?

1. Prepare text
 1. Remove **markup code** and **punctuation**
 2. Handle **each word** separately



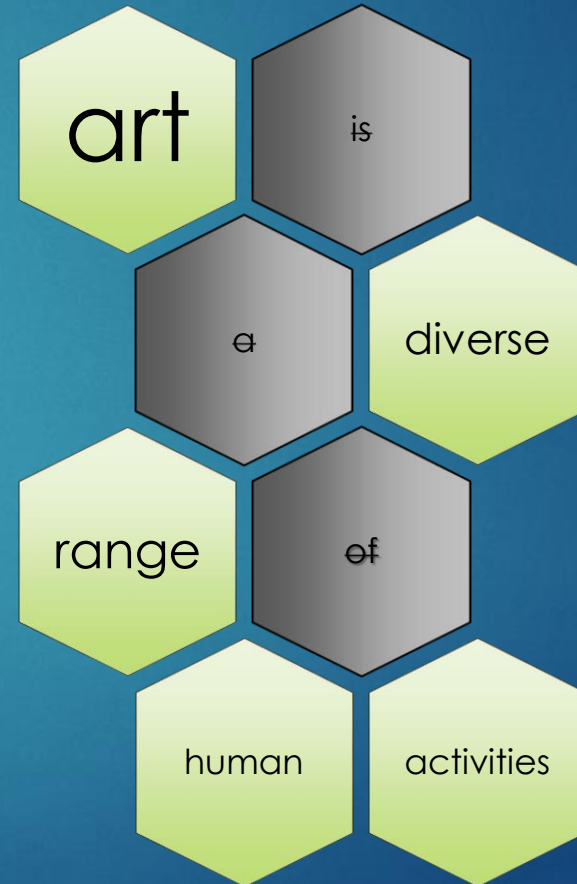
What is a **general approach**?

1. Prepare text
 1. Remove **markup code** and **punctuation**
 2. Handle **each word** separately
 3. Convert to **lower case**



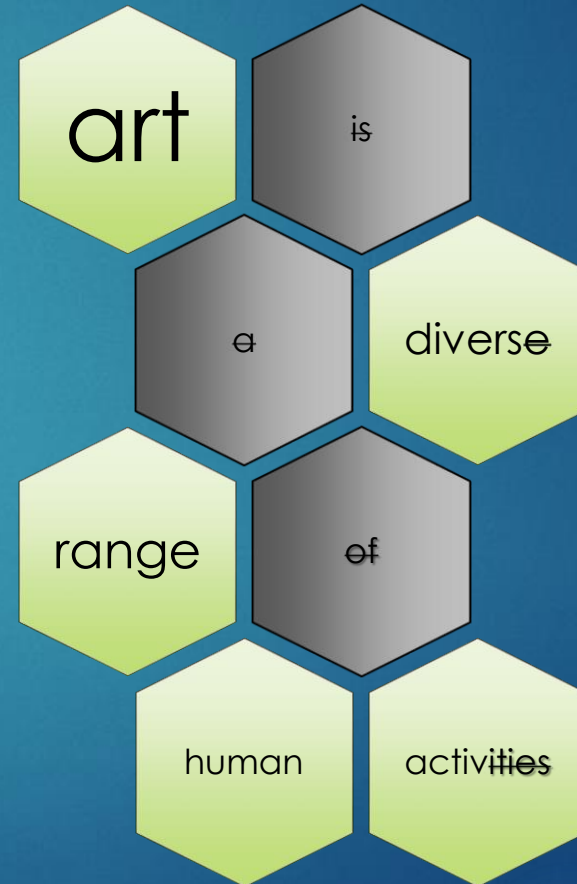
What is a **general approach**?

1. Prepare text
 1. Remove **markup code** and **punctuation**
 2. Handle **each word** separately
 3. Convert to **lower case**
 4. Remove **stop-words**



What is a **general approach**?

1. Prepare text
 1. Remove **markup code** and **punctuation**
 2. Handle **each word** separately
 3. Convert to **lower case**
 4. Remove **stop-words**
 5. Apply **stemming** algorithm



What is a **general approach**?

1. Prepare text
 1. Remove **markup code** and **punctuation**
 2. Handle **each word** separately
 3. Convert to **lower case**
 4. Remove **stop-words**
 5. Apply **stemming** algorithm
 6. Create **word counter list**

Word	Count
art	1
divers	1
range	1
human	1
activ	1

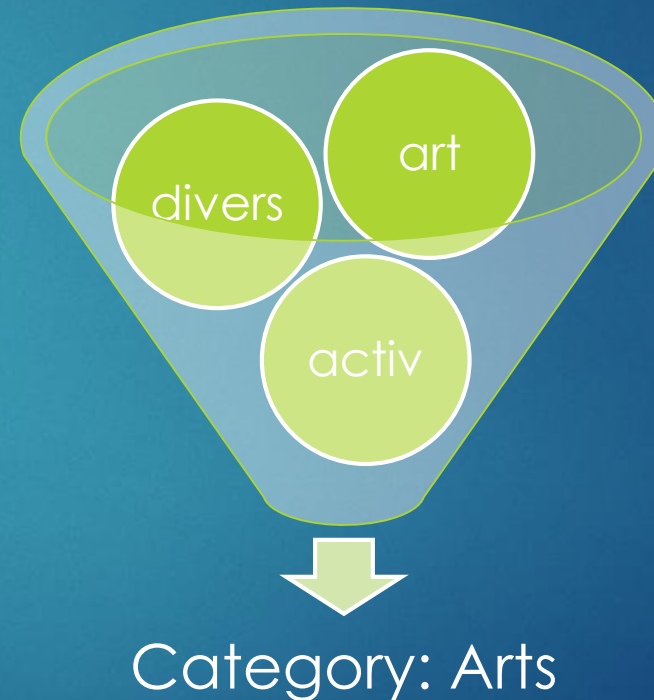
What is a **general approach**?

1. Prepare text
 1. Remove **markup code** and **punctuation**
 2. Handle **each word** separately
 3. Convert to **lower case**
 4. Remove **stop-words**
 5. Apply **stemming** algorithm
 6. Create **word counter list**
7. Train **text classification algorithm** (Naïve Bayes)



What is a **general approach**?

1. Prepare text
 1. Remove **markup code** and **punctuation**
 2. Handle **each word** separately
 3. Convert to **lower case**
 4. Remove **stop-words**
 5. Apply **stemming** algorithm
 6. Create **word counter list**
7. Train **text classification algorithm** (Naïve Bayes)
8. **Apply** learnt algorithm (using test data)



3. Why this project?

Text classification for German and Romanian language. **Why?**

- ▶ Based on Lab @ Dr. MORARIU:
 - ▶ MACHINE LEARNING – Advanced group
 - ▶ Topic 2: Text Document classification using Naïve Bayes
- ▶ Well researched for **English language**, but not for **German and Romanian**



4. Generate **example data**

Generate example data: a supervised learning problem

- ▶ In the Lab: **Reuters** dataset of **English** language texts

- ▶ Requirements for project:

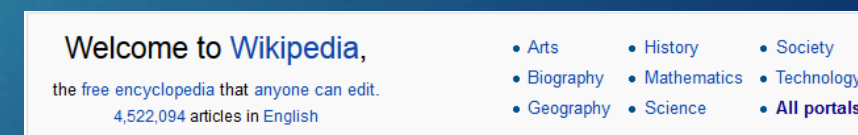
- ▶ Example dataset of **classified** texts
- ▶ Available in German and Romanian **language**
- ▶ Similar **data structure**
- ▶ Decent **number** of examples



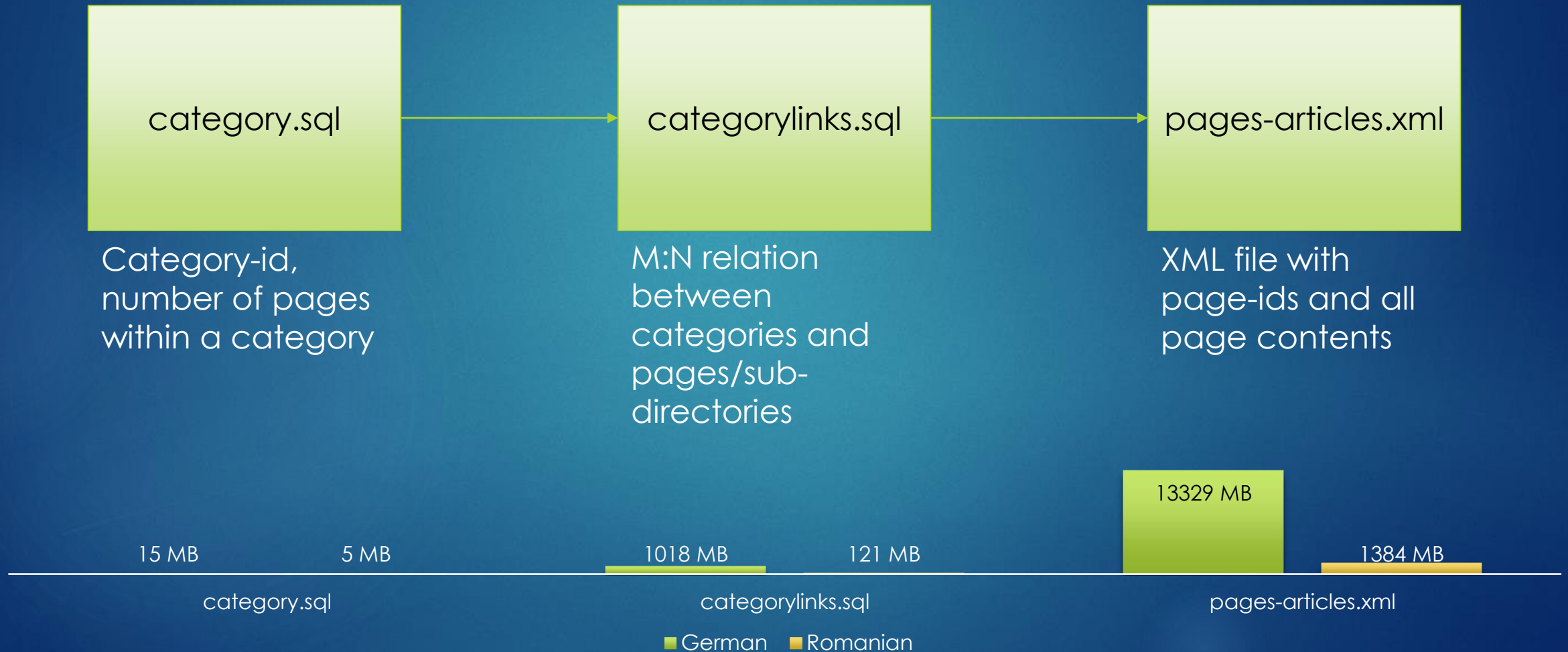
Generate example data. Idea: Wikipedia articles and categories



- ▶ Example dataset of **classified** texts ✓
- ▶ Available in German and Romanian **language** ✓
- ▶ Similar **data structure** ✓
- ▶ Decent **number** of examples ✓



Wikipedia dump **data structure** (and size)



Generate example data

Pseudo code

1. Define **top categories**
"Geographie, Geschichte, Gesellschaft,.../ Geografie, Istorie, Societate,..."
2. Read meta data about **categories** from **category.sql**-file
3. Read **matching page-ids** from **categorylinks.sql**
Remove markup code and punctuation
Handle each word separately
Convert to lower case
Remove stop-words
Apply stemming algorithm
Create word counter list
4. Write **arff-file(s)**
Standardized format to be read by Naïve Bayes algorithm developed in Machine Learning lab

5. **Train and test** Naïve Bayes algorithm

Train and test Naïve Bayes algorithm

Pseudo code

- ▶ Read **arff-files** with training and testing examples
- ▶ For **training** examples:
 - ▶ Apply word-counter list
 - ▶ Reduce features using **feature selection** algorithm
keep only features („words“) with high „mutual information“
- ▶ For **testing** examples:
 - ▶ Calculate **probability** for **each category**
 - ▶ Category with **highest probability** = **prediction**

6. Performance and results

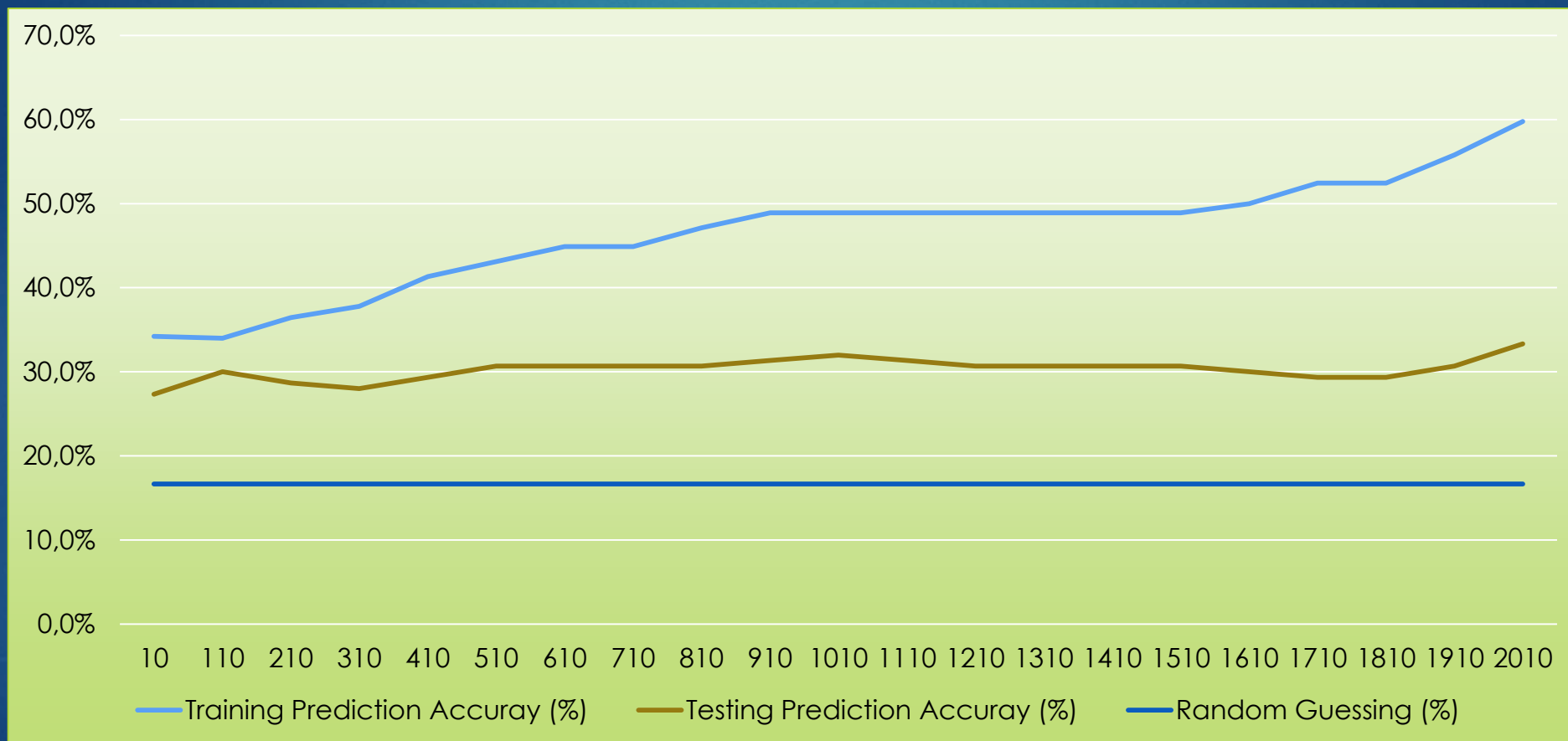
Performance and results

Test 1: German language

- ▶ Parameters:
 - ▶ Categories: **6**
Geographie, Geschichte, Gesellschaft, Religion, Sport, Technik
 - ▶ Pages per Category: **100**
 - ▶ Training documents: **450**
 - ▶ Testing documents: **150**
- ▶ Statistics
 - ▶ Total number of distinct words: 79.905

Performance and results

Test 1: German language



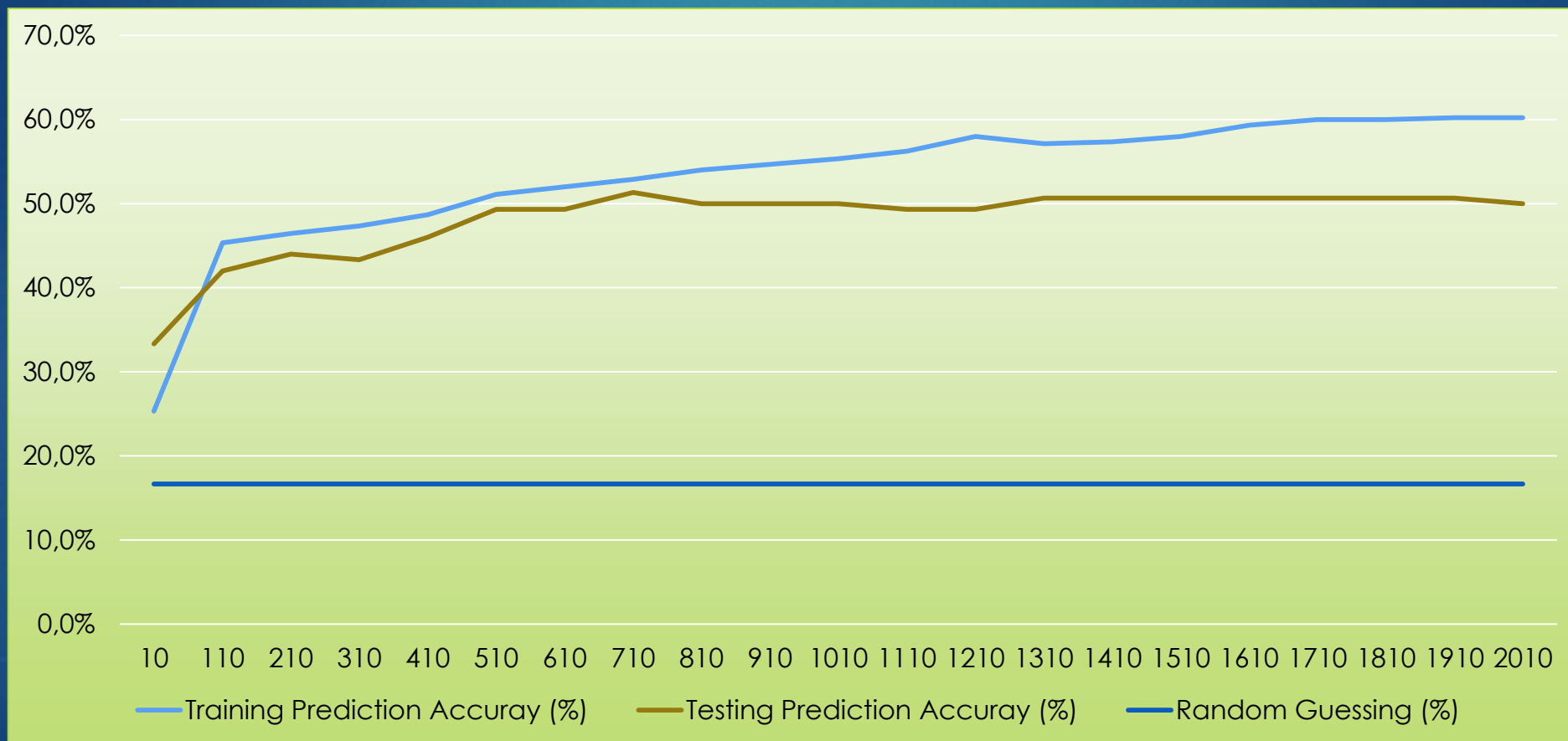
Performance and results

Test 2: Romanian language

- ▶ Parameters:
 - ▶ Categories: **6**
Artă, Cultură, Geografie, Istorie, Religie, Știință
 - ▶ Pages per Category: **100**
 - ▶ Training documents: **450**
 - ▶ Testing documents: **150**
- ▶ Statistics
 - ▶ Total number of distinct words: 34.538

Performance and results

Test 2: Romanian language



7. Conclusions

Prediction Accuracy

- ▶ Higher than **random guessing**
→ works in principle
- ▶ Still **not satisfying**

Problems with text classification approach

- ▶ Ignore word position in text
- ▶ Lost negations („not“)
„I do not love you, I hate you“ = „I do not hate you, I love you“
- ▶ Different meanings of words
„Jaguar“: Car? Animal?
- ▶ Stemming not perfect (→ better: dictionary-based)
In German: long, compound words („Donauschiffahrtsgesellschaft“)

Sources and more information

- ▶ **Machine Learning**

by Tom M. Mitchell

issued on 1st March 1997, chapter 6 „Bayesian Learning“, pages 154 to 200

ISBN: 0070428077

- ▶ **Text classification and Naïve Bayes**

Paper by Cambridge University Press

issued on 1st April 2009, chapter 13, pages 253 to 287

Download: <http://nlp.stanford.edu/IR-book/pdf/13bayes.pdf>

- ▶ **Snowball-Stemmer and stop-word lists**

OpenSource library mainly by Martin Porter and Richard Boulton

Download: <http://snowball.tartarus.org>

- ▶ **Source code and this presentation**

<https://github.com/sfeilmeier/TextClassification>

Questions?