IN3120 week 10

Group 1

Agenda

- Assignment e-2
- Assignment e-1
  - Naive Bayes example
- Shoutout of the week

Assignment e-2

- Create your own embedding generation/ANN index
- Compare with similaritysearchengine and make a report
  - Report at least speed, scalability, search quality aspects
- No test suite
- A lot of freedom here, do what you think is cool!

Assignment e-1

- Implement a Naive bayes multinomial classifier for language prediction
- Classify documents as *no, da, en or de*
- «Naive» because it doesn’t consider the position of terms

Example: Oliver’s library

- Oliver is super rich and owns a library
- He picks up a book, but doesn’t understand the language
- He uses Naive Bayes to make a *prediction*
- Prediction is based on the books in the library (our training data)!

Preparations

- What do we need?
  - Vocabulary
  - Priors
  - Posteriors

The vocabulary

- First, Oliver needs to figure out what words he is working with
- He writes down all unique terms in the entire library

```
def __compute_vocabulary(self, training_set, fields) -> None
```

- ***Informatikk***, ***everyone***, ***pasta***, ***ciao***, ***Norge***

Starting the calculations

- Oliver needs some notion of the probabilities of languages occurring in his library

What are priors?

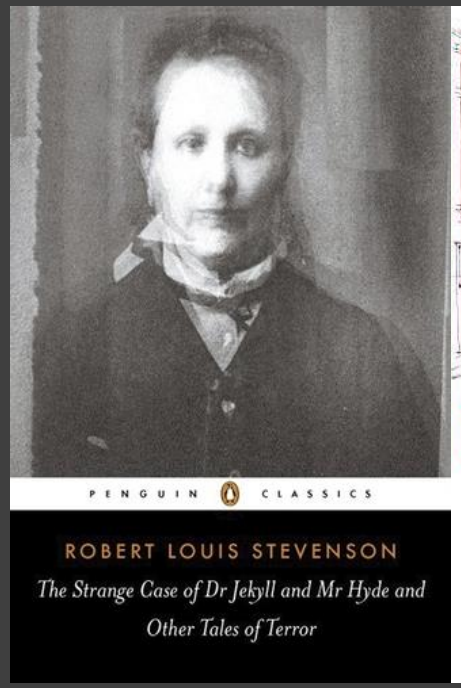
- We call them «priors» because we haven’t looked at the mystery book’s contents yet!
- «If i pick up a random book, without reading it, what are the odds it’s written in a specific language?»
- The probability of the class itself

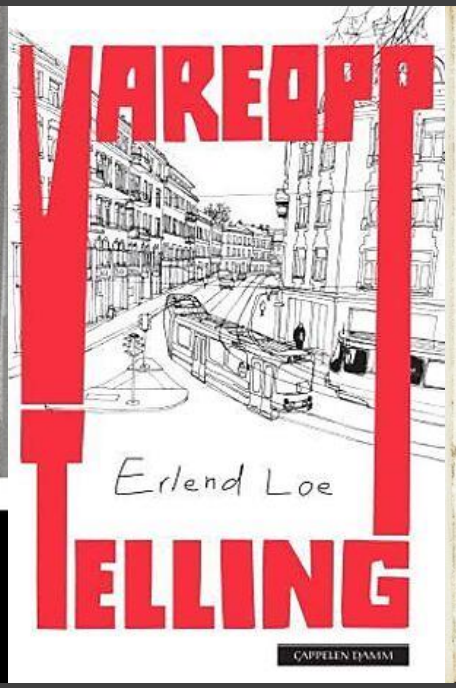
Priors

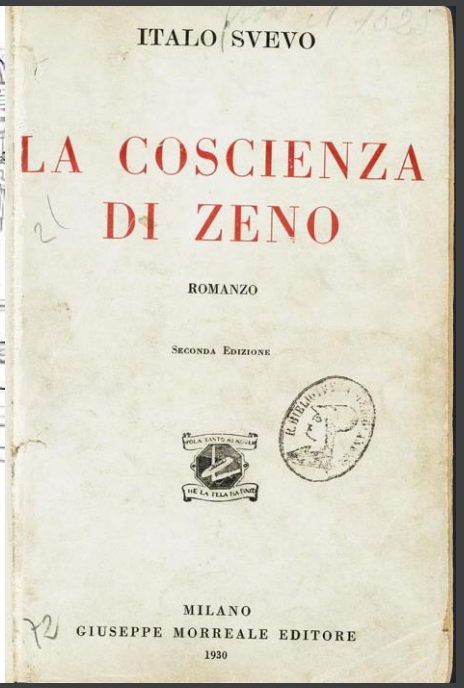
- Oliver’s library contains books of 3 languages!

Library contents

- **13** italian books
- **12** norwegian books
- **9** english books







Computing priors

- Only based on the library, what language is the book?
- Total amount of books is 13 + 12 + 9 = 34

Computing priors

- 13 / 34 chance it is Italian, so 38%
- 12 / 34 chance it is Norwegian, so 35%
- 9 / 34 chance it is English, so 26%

Computing priors

- `self.__priors["Italian"] = 0.38`
- `self.__priors["Norwegian"] = 0.35`
- `self.__priors["English"] = 0.26`

Computing priors

```
def __compute_priors(self, training_set) -> None:
    # Find the total amount of entries in training_set
    # Calculate the prior of each category
    # Add { category: prior } to self.__priors
```

Computing priors

```
def __compute_priors(self, training_set) -> None:
    total = 34
    # Calculate the prior of each category
    # Add { category: prior } to self.__priors
```

Computing priors

```
def __compute_priors(self, training_set) -> None:
    total = 34
    prior = 13/34
    # Add { category: prior } to self.__priors
```

Computing priors

```
def __compute_priors(self, training_set) -> None:
    total = 34
    prior = 13/34
    self.__priors["it"] = prior
```

Posteriors

- Ok, now Oliver knows the distribution of languages. What's next?
- We need to figure out some correlation between words and classes
- Classifying book based on other books is pointless if we never regard the content

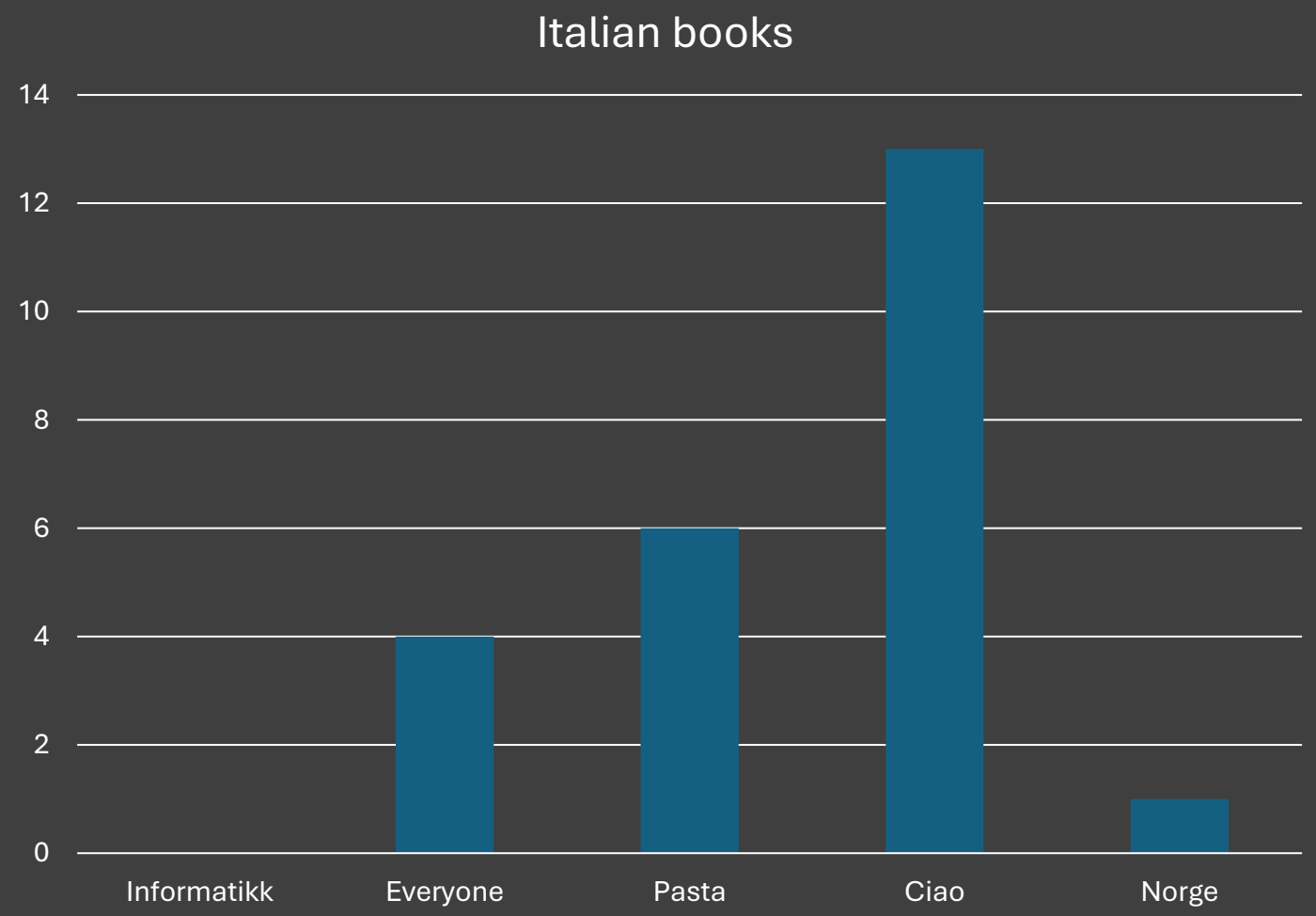
What are posteriors?

- If you already know the language of the book, what is the probability of seeing each word?
- «If i pick a random term from a specific class, what is the probability of picking *that* specific term»
- Number of term occurrences in a class, divided by total number of terms in the class

Computing posteriors

- Oliver’s books arevery short, no more than 5 words
- ***Informatikk***, ***everyone***, ***pasta***, ***ciao***, ***Norge***

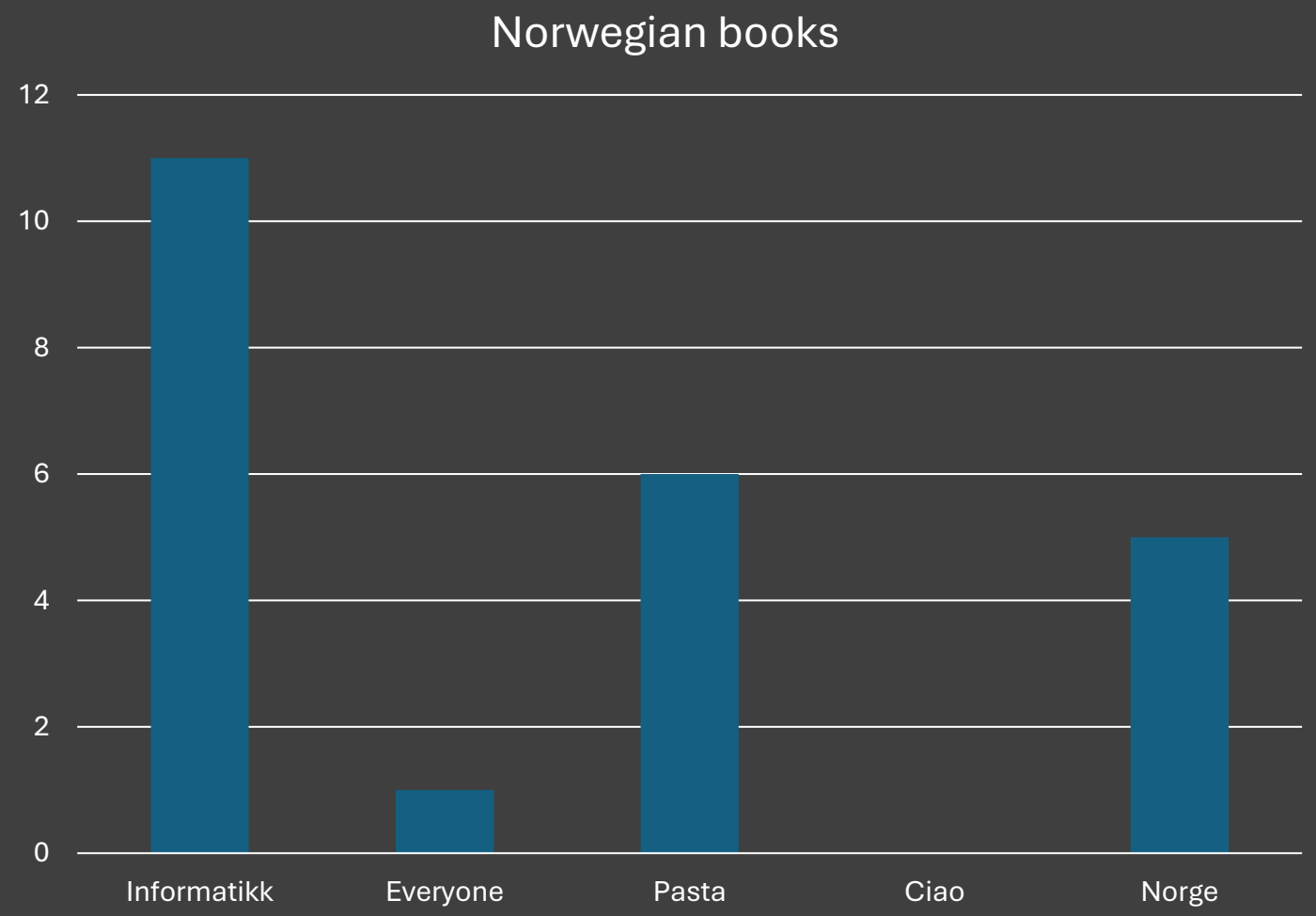
Computing posteriors



Computing posteriors

- p(Informatikk | IT) = 0 / 24 = 0
- p(Everyone | IT) = 4 / 24 = 0.17
- p(Pasta | IT) = 6 / 24 = 0.25
- p(Ciao | IT) = 13 / 24 = 0.54
- p(Norge | IT) = 1 / 24 = 0.04

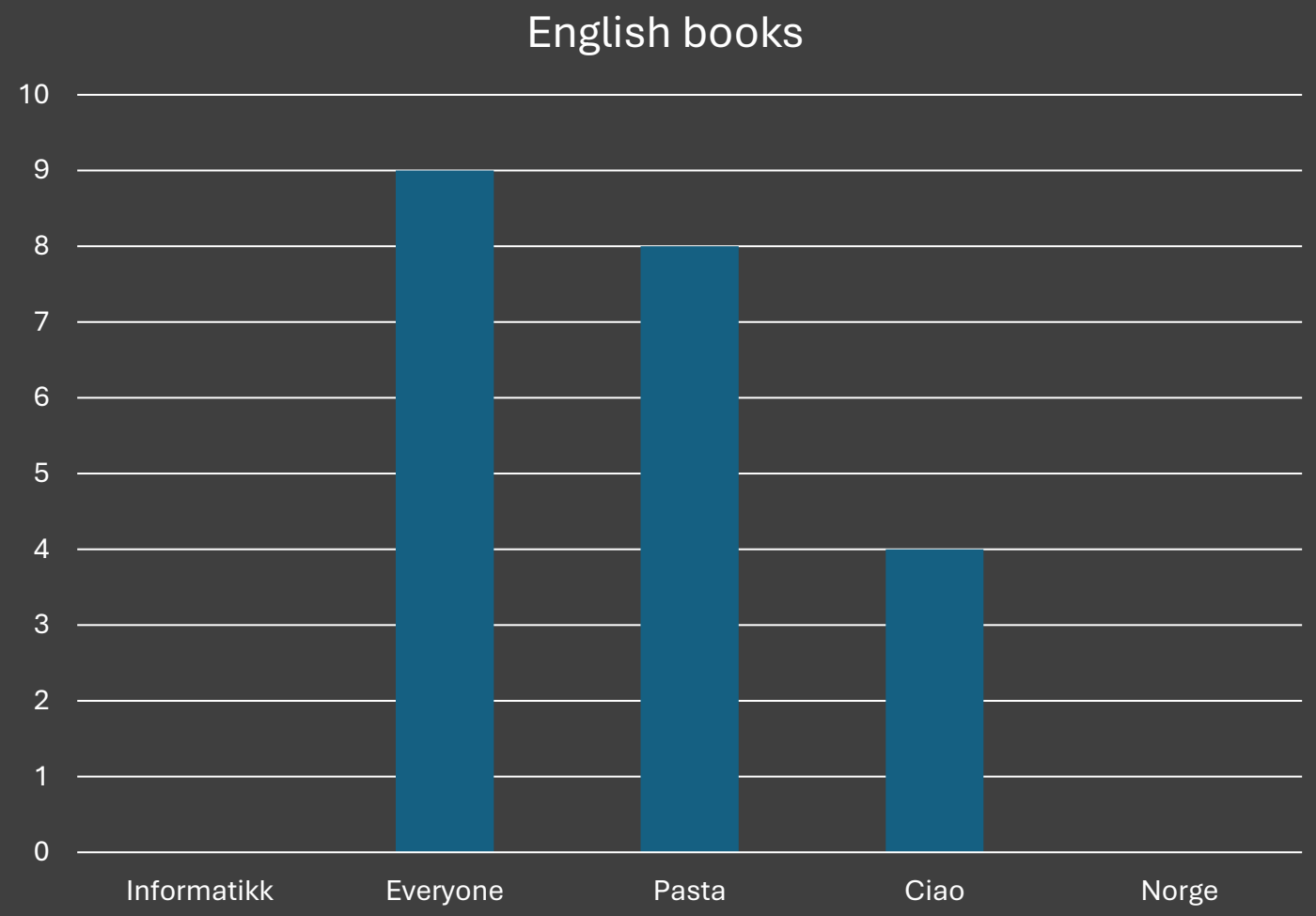
Computing posteriors



Computing posteriors

- p(Informatikk | NO) = 11 / 23 = 0.48
- p(Everyone | NO) = 1 / 23 = 0.04
- p(Pasta | NO) = 6 / 23 = 0.26
- p(Ciao | NO) = 0 / 23 = 0
- p(Norge | NO) = 5 / 23 = 0.22

Computing posteriors



Computing posteriors

- p(Informatikk | GB) = 0 / 21 = 0
- p(Everyone | GB) = 9 / 21 = 0.43
- p(Pasta | GB) = 8 / 21 = 0.38
- p(Ciao | GB) = 4 / 21 = 0.19
- p(Norge | GB) = 0 / 21 = 0

Computing posteriors (it)

```
def get_posterior(self, category: str, term: str) -> float:
    # Iterate each category and corpus in training_set
    # Get all terms per category
    # Add total term freq to self.__denominators
    # Add { term: probability } to self.__conditionals
```

Computing posteriors (it)

```
def get_posterior(self, category: str, term: str) -> float:
    # Iterate each category and corpus in training_set
    # Everyone⁴, Pasta⁶, Ciao¹³, Norge¹
    # Add total term freq to self.__denominators
    # Add { term: probability } to self.__conditionals
```

Computing posteriors (it)

```
def get_posterior(self, category: str, term: str) -> float:
    # Iterate each category and corpus in training_set
    # Everyone⁴, Pasta⁶, Ciao¹³, Norge¹
    # self.__denominators["It"] = 24
    # Add { term: probability } to self.__conditionals
```

Computing posteriors (it)

```
def get_posterior(self, category: str, term: str) -> float:
    # Iterate each category and corpus in training_set
    # Everyone⁴, Pasta⁶, Ciao¹³, Norge¹
    # self.__denominators["It"] = 24
    # self.__conditionals["it"] = { everyone : 4/24 }
    # self.__conditionals["it"] = { pasta : 6/24 }
    # self.__conditionals["it"] = { ciao : 13/24 }
    # self.__conditionals["it"] = { norge : 1/24 }
```

Classification

- All the preparations are finished
- We can finally start classifying books!

Oliver’s book

Calculations

- self.__priors["Italian"] = 0.38
- p(Ciao | IT) = 0.54
- p(Everyone | IT) = 0.17
- p(Pasta | IT) = 0.25

Calculations

- Italian probability = 0.38 x 0.54 x 0.17 x 0.25 = **0.0087**

Calculations

- self.__priors["Norwegian"] = 0.35
- p(Ciao | NO) = 0
- p(Everyone | NO) = 0.04
- p(Pasta | NO) = 0.26

Calculations

- Norwegian probability = 0.35 x 0 x 0.04 x 0.26 = **0**

Calculations

- self.__priors["English"] = 0.26
- p(Ciao | GB) = 0.19
- p(Everyone | GB) = 0.43
- p(Pasta | GB) = 0.38

Calculations

- English probability = 0.26 x 0.19 x 0.43 x 0.38 = **0.008**

And the winner is...

- Italian probability = 0.38 x 0.54 x 0.17 x 0.25 = 0.0087
- English probability = 0.26 x 0.19 x 0.43 x 0.38 = 0.0080
- Norwegian probability = 0.35 x 0 x 0.04 x 0.26 = 0.0000

Extra requirements

- Add one/plus one/laplace smoothing
  - Score for Norwegian was 0.0 on previous slide
- Log-probabilities
- Neither are covered in this example

Need a better explanation?

- Last week's shoutout: https://www.youtube.com/watch?v=O2L2Uv9pdDA&ab_channel=StatQuestwithJoshStarmer

Agenda

- Assignment e-2
- Assignment e-1
  - Naive Bayes example
- Shoutout of the week

Shoutout

15 min break