TEXT MINING

L04. TEXT CATEGORIZATION

SUZAN VERBERNE 2022

Universiteit Leiden

TODAY'S LECTURE

- Quiz about week 3
- Text categorization introduction
- The text categorization process
  - Task definition
  - (Example data)
  - Pre-processing
  - Feature extraction
  - Classifier learning
  - Evaluation

QUIZ ABOUT WEEK 3

- What is the role of the distributional hypothesis in training word embeddings?
  a. Words that occur in similar contexts get similar representations
  b. The embeddings get updated while the collection is processed
  c. Randomly sampled non-context words are used as counter-examples
  d. The distributional hypothesis makes word2vec a highly effective model

QUIZ ABOUT WEEK 3

- What is the role of the distributional hypothesis in training word embeddings?
  a. Words that occur in similar contexts get similar representations
  b. The embeddings get updated while the collection is processed
  c. Randomly sampled non-context words are used as counter-examples
  d. The distributional hypothesis makes word2vec a highly effective model

QUIZ ABOUT WEEK 3

- Which of these statements are true about word2vec?
  a. The hidden layer typically has 100-1000 nodes
  b. The learned embedding space has interpretable dimensions
  c. The model is trained on a classification task with self-supervision
  d. It encodes syntactic and semantic relationships between words

QUIZ ABOUT WEEK 3

- Which of these statements are true about word2vec?
  a. The hidden layer typically has 100-1000 nodes
  b. The learned embedding space has interpretable dimensions
  c. The model is trained on a classification task with self-supervision
  d. It encodes syntactic and semantic relationships between words

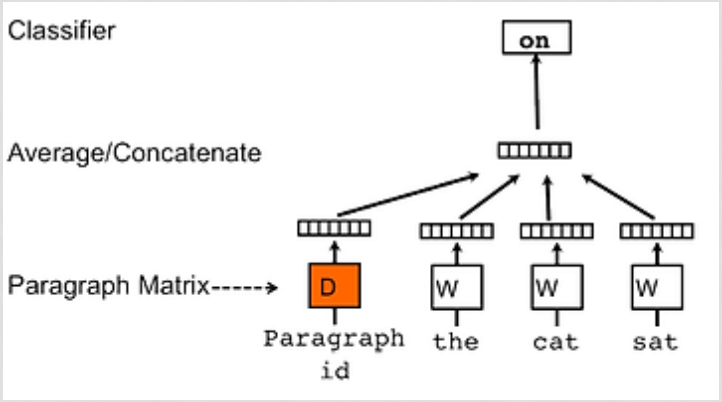
WEEK 3 FOLLOW-UP

- Do we need lemmatization for word2vec?
  - It depends on our application
  - For most applications: yes, because we don't want representations for both 'bicycle' and 'bicycles' in our vector space

WEEK 3 FOLLOW-UP

- How to go from word embeddings to document embeddings?
  - Taking the average vector over all words in the document
  - Use doc2vec

    https://radimrehurek.com/gensim/models/doc2vec.html



  - Or, with BERT embeddings, use SentenceBERT (highly efficient but only for short documents/passages)
    - https://www.sbert.net/

Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents.
Wieting et al. (2016). Towards universal paraphrastic sentence embeddings.

WEEK 3 FOLLOW-UP

- How is the input of word2vec encoded?
  - As one-hot vectors
  - https://towardsdatascience.com/word2vec-out-of-the-black-box-a404b4119681

QUIZ ABOUT WEEK 3

- Did you work on the exercise of week 3 (embeddings tutorial)
  a. Yes, I completed it
  b. Yes, I completed at least half of it
  c. Yes, I started it
  d. No

ABOUT TEXT CATEGORIZATION

(OR TEXT CLASSIFICATION)

TYPES OF CLASSIFICATION TASKS

- Binary classification

- Multi-class / one-of classification

- Multi-label classification

EXAMPLE APPLICATIONS

- Is this task binary, multi-class or multi-label?

SPAM CLASSIFICATION

SEO Important Ranking Factors?

Emma Smith <gstailoringeweb35seo@outlook.com>

Tuesday, 20 September 2022 at 08:40

This message appears to be Junk. Links and other functionality will not work. Mark as Not Spam

Retention: Junk Email Expires: 20/10/2022.

Dear Web Owner,

Want more clients and customers?

Please provide me your domain name which you want to optimize.

We will analysis your website and URL send full SEO proposal with plan and activities which will be implemented on your website.

We will help them find you by putting you on the 1st page of Google.

We have some special offers this season.

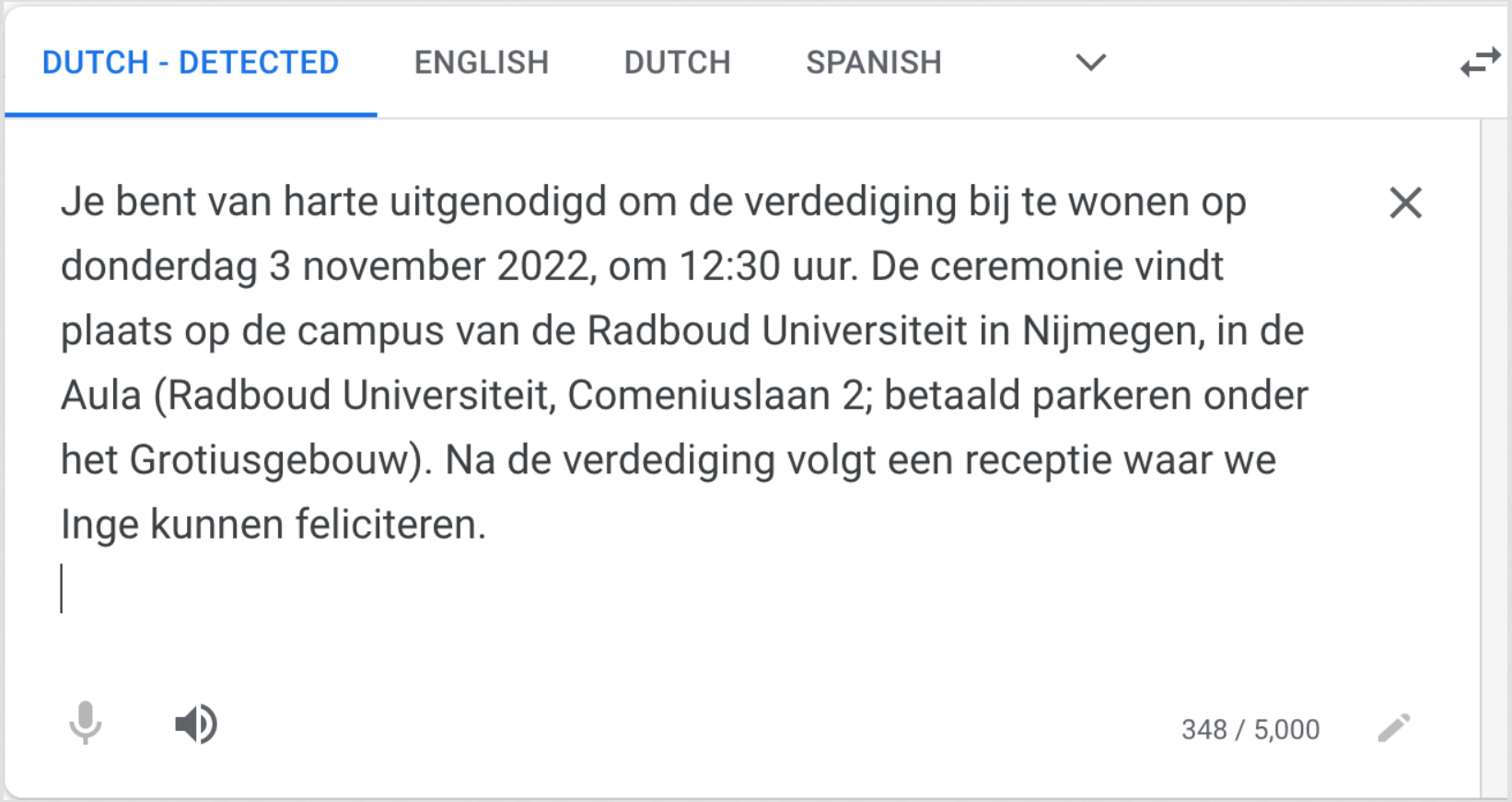
Email us back to get a full proposal.

Thanks,

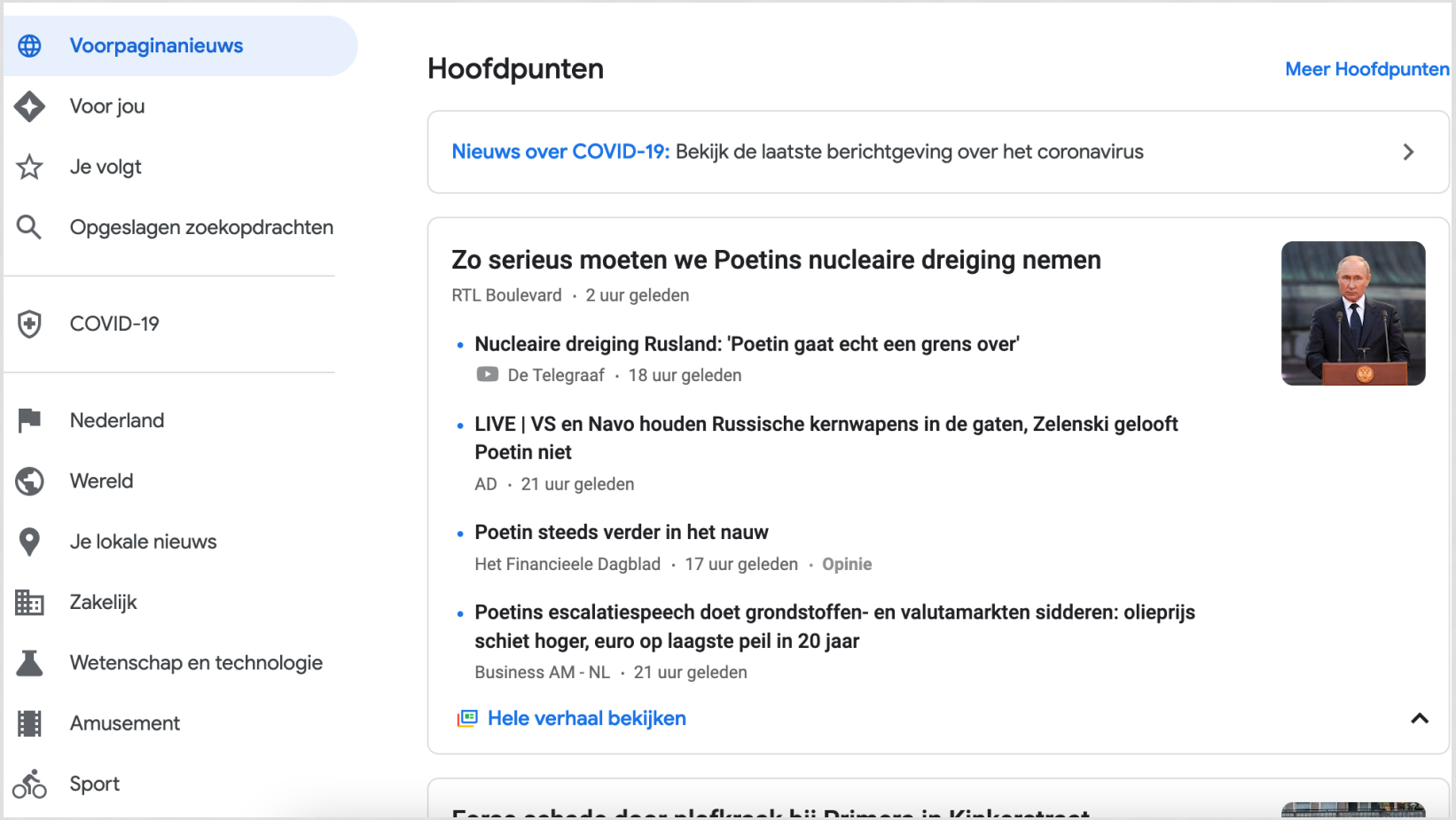
Emma

SEO Manager,

LANGUAGE IDENTIFICATION



NEWS SECTIONING



ADDING KEYWORDS TO ARCHIVED DOCUMENTS

THE TEXT CATEGORIZATION PROCESS

WHAT IS NEEDED FOR TEXT CLASSIFICATION

- A definition of the task
- Example data
- Pre-processing
- Feature extraction
- Classifier learning
- Evaluation

WHAT IS NEEDED FOR TEXT CLASSIFICATION

- A definition of the task
- Example data
- Pre-processing
- Feature extraction
- Classifier learning
- Evaluation

TASK DEFINITION

- What is the text unit (a.k.a. 'document')?
  - Complete documents (emails, scientific articles)
  - Sections (minutes, speeches)
  - Sentences? (language identification, sentiment classification)

TASK DEFINITION

- What are the categories?
  - Spam/no spam
  - Language
  - Positive/negative/neutral (sentiment)
  - Agree/disagree/unrelated (stance)
  - Topic
  - News section
  - …

Lecture 9

WHAT IS NEEDED FOR TEXT CLASSIFICATION

- A definition of the task
- Example data — Lecture 5
- Pre-processing
- Feature extraction
- Classifier learning
- Evaluation

PRE-PROCESSING AND FEATURE EXTRACTION

WORDS AS FEATURES

Doc id	Content	Class
1	request urgent interest urgent	Spam
2	assistance low interest deposit	Spam
3	symposium defense june	No spam
4	notas symposium deadline june	No spam
5	registration assistance symposium deadline	?

WORDS AS FEATURES

Doc id	Content	Class
1	request urgent interest urgent	Spam
2	assistance low interest deposit	Spam
3	symposium defense june	No spam
4	notas symposium deadline june	No spam
5	registration assistance symposium deadline	?

- Each item in the vocabulary becomes a feature/dimension in the vector space
- Each document x_i is represented as a vector in this space
- Each document has a label y_i

What is the vocabulary size of this training set?

11

PRE-PROCESSING

- We need at least tokenization to get the words
- Or we create even lower-level features: character k-grams
  - E.g. 4-grams:
  - “you will. To” →

```
you_  ou_w  u_wi  _wil  will  ill.  ll._  l._T  ._To
```

- For most applications, we apply lowercasing and removal of punctuation

PRE-PROCESSING

- Additional pre-processing steps might include:
  - Remove stopwords
  - Lemmatization or stemming
  - Add phrases as features (e.g. “PhD defense”, “not good”)
    - Phrases: statistical (bigrams, trigrams) or linguistic (noun phrases)
    - The countvectorizer in sklearn has options for character n-grams and word n-grams

PRE-PROCESSING

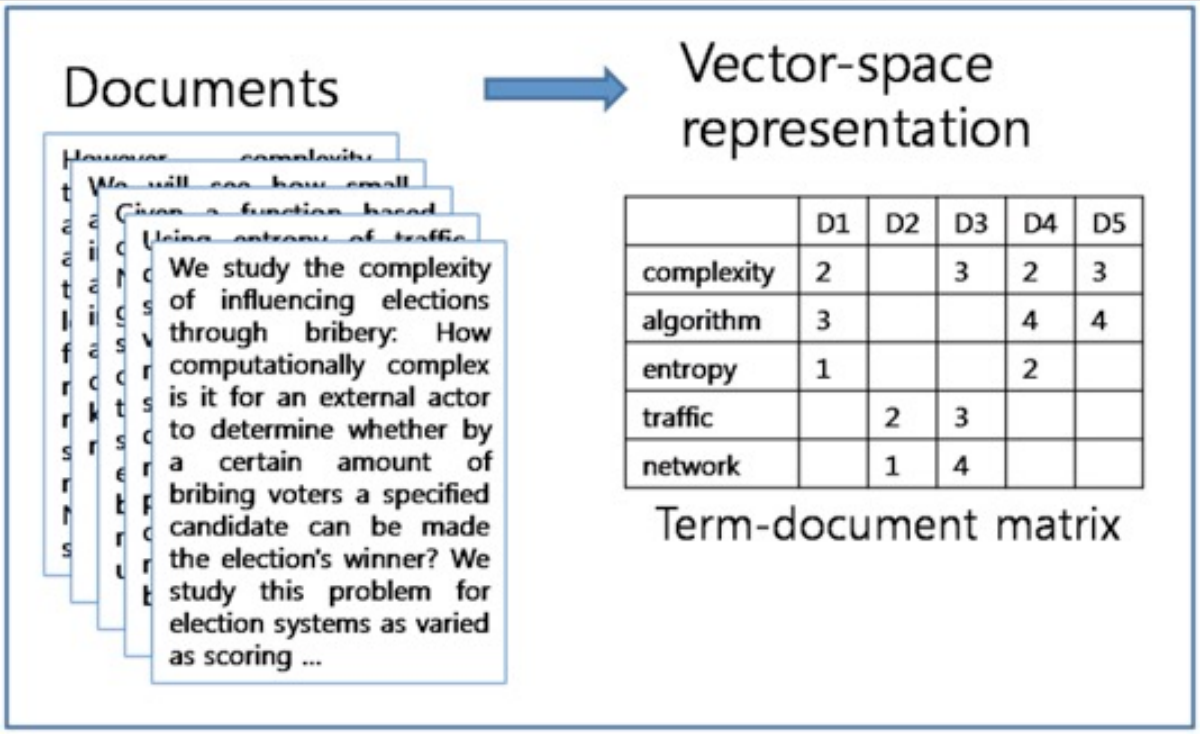
- Decide on vocabulary size (number of dimensions / number of distinct terms)
  - Feature selection

FEATURE SELECTION

- Goals:
  - Dimensionality reduction
  - Reduce overfitting
- Global term selection: overall term frequency (e.g. 3) is used as cut-off (remove rare terms, the long tail)
- Local term selection: Each term is scored by a scoring function that captures its degree of correlation with each class it occurs in (for example using the χ^2 test)
- Only the top-n terms are used for classifier training

TERM-DOCUMENT MATRIX

- Once we have defined the terms in our vocabulary (the features), we assign them weights (feature values)



	D1	D2	D3	D4	D5
complexity	2		3	2	3
algorithm	3			4	4
entropy	1			2	
traffic		2	3		
network		1	4		

Term-document matrix

TERM WEIGHTING

TERM WEIGHTING

- Compute term weights

  - Binary: occurrence of term $w_{t,d} \in \{0,1\}$
  - Integer: term count
  - Real-valued: more advanced term weighting

  - Most used weighting scheme: TF-IDF
    (see also J&M Section 6.5)

TERM WEIGHTING

- Term frequency
  - The term count $tc_{t,d}$ of term t in document d is defined as the number of times that t occurs in d.
  - Raw term count is not what we want because:
    - A document with tc = 10 is more relevant than a document with tc = 1
    - But not 10 times more relevant
    - Relevance does not increase proportionally with term frequency.
  - → log frequency is more useful:

TERM WEIGHTING

- Term frequency
  - The term count $tc_{t,d}$ of term t in document d is defined as the number of times that t occurs in d.
  - Raw term count is not what we want because:
    - A document with tc = 10 is more relevant than a document with tc = 1
    - But not 10 times more relevant
    - Relevance does not increase proportionally with term frequency.
  - → log frequency is more useful: $$tf_{t,d} = \begin{cases} 1 + \log_{10} tc_{t,d} & \text{if } tc_{t,d} > 0 \\ 0 & \text{otherwise} \end{cases}$$

TERM WEIGHTING

- Inverse document frequency:
  - Intuition: the most frequent terms are not very informatve
  - df_t is the document frequency, the number of documents that t occurs in
  - df_t is an inverse measure of the informativeness of term t
  - We define the idf weight of term t as follows:

TERM WEIGHTING

- Inverse document frequency:
  - Intuition: the most frequent terms are not very informatve
  - df_t is the document frequency, the number of documents that t occurs in
  - df_t is an inverse measure of the informativeness of term t
  - We define the idf weight of term t as follows:

    (N is the number of documents in the collection)

$$\text{idf}_t = \log_{10} \frac{N}{\text{df}_t}$$

- Tf-idf = tf * idf

EXERCISE

$$\mathrm{tf}_{t,d} = \begin{cases} 1 + \log_{10} \mathrm{tc}_{t,d} & \text{if } \mathrm{tc}_{t,d} > 0 \\ 0 & \text{otherwise} \end{cases} \qquad \mathrm{idf}_t = \log_{10} \frac{N}{\mathrm{df}_t}$$

1. We have a collection of 1 Million documents. The term ‘computer’ occurs in 100 documents. What is the inverse document frequency for ‘computer’?

2. We have a document with 10 times the word ‘computer’ What is the tf-idf weight for the term ‘computer’ in this document?

EXERCISE

$$\mathrm{tf}_{t,d} = \begin{cases} 1 + \log_{10} \mathrm{tc}_{t,d} & \text{if } \mathrm{tc}_{t,d} > 0 \\ 0 & \text{otherwise} \end{cases} \qquad \mathrm{idf}_t = \log_{10} \frac{N}{\mathrm{df}_t}$$

1. We have a collection of 1 Million documents. The term 'computer' occurs in 100 documents. What is the inverse document frequency for 'computer'?
   - idf = log (N/100) = log (1000000/100) = log (10000) = 4
2. We have a document with 10 times the word 'computer' What is the tf-idf weight for the term 'computer' in this document?
   - tf = 1 + log(10) = 1+1 = 2
   - tf-idf = 2*4 = 8

WHAT IS NEEDED FOR TEXT CLASSIFICATION

- A definition of the task
- Example data
- Pre-processing
- Feature extraction
- Classifier learning
- Evaluation

CLASSIFICATION METHODS

3 options:

1. If we use word features, we need an estimator that is well-suited for high-dimensional, sparse data:
   - Naïve Bayes (MultinomialNB in sklearn)
   - Support Vector Machines (LinearSVC in sklearn)
   - Random Forest (RandomForestClassifier in sklearn)
2. When text is represented as dense embeddings vectors, we can use neural network architectures to train classifiers
3. Or we use transfer learning from pre-trained contextual embeddings with transformers (lecture 7)

NAIVE BAYES

J&M CHAPTER 4

NAIVE BAYES CLASSIFICATION

- Learning and classification method based on probability theory
- Uses prior probability of each category given no information about an item
- Classification produces a posterior probability distribution over the possible categories given a description of an item

NAIVE BAYES CLASSIFICATION

- Naive Bayes classification method

- Based on Bayes' theorem:

$$P(c \mid d) = \frac{P(d \mid c)P(c)}{P(d)}$$

Prior

Evidence

NAIVE BAYES CLASSIFICATION

- Task: classify a new document d based on its feature representation

$$X(d): \{t_1, t_2, \ldots, t_k\}$$

Maximum A Posteriori

$$c_{MAP}(d) = \underset{c \in C}{\operatorname{argmax}}\, P(c \mid d)$$

$$c_{MAP}(d) = \underset{c \in C}{\operatorname{argmax}}\, \frac{P(d \mid c)P(c)}{P(d)}$$

$$c_{MAP}(d) = \underset{c \in C}{\operatorname{argmax}}\, P(d \mid c)P(c)$$

NAIVE BAYES CLASSIFICATION

- Learning the model: use the frequencies in the training data

$$P(c \mid d) = \frac{P(d \mid c)P(c)}{P(d)}$$

Prior

$$P(c) = \frac{|D_c|}{|D|}$$

the number of documents for class c

the total number of documents

EXAMPLE

Doc id	Content	Class
1	request urgent interest urgent	Spam
2	assistance low interest deposit	Spam
3	symposium defence june	No spam
4	siks symposium deadline june	No spam
5	registration assistance symposium deadline	?

- P(spam) = ?
- P(no spam) = ?

EXAMPLE

Doc id	Content	Class
1	request urgent interest urgent	Spam
2	assistance low interest deposit	Spam
3	symposium defence june	No spam
4	siks symposium deadline june	No spam
5	registration assistance symposium deadline	?

- P(spam) = 2/4
- P(no spam) = 2/4

NAIVE BAYES CLASSIFICATION

- Learning the model: Use the frequencies in the data

$$P(c \mid d) = \frac{P(d \mid c)P(c)}{P(d)}$$

$$P(d \mid c) = P(t_1, t_2, \ldots, t_k \mid c)$$

$$P(t|c) = \frac{T_{c,t}}{\sum_{t' \in V} T_{c,t'}}$$

the number of occurrences of t in training documents from class c

total number of term occurrences in training documents from class c

EXAMPLE

Doc id	Content	Class
1	request urgent interest urgent	Spam
2	assistance low interest deposit	Spam
3	symposium defence june	No spam
4	siks symposium deadline june	No spam
5	registration assistance symposium deadline	?

- P(spam) = 2/4
- P(registration|spam) = ?
- P(assistance|spam) = ?
- P(symposium|spam) = ?
- P(deadline|spam) =?

- P(no spam) = 2/4
- P(registration|no spam) = ?
- P(assistance|no spam) = ?
- P(symposium|no spam) = ?
- P(deadline|no spam) = ?

EXAMPLE

Doc id	Content	Class
1	request urgent interest urgent	Spam
2	assistance low interest deposit	Spam
3	symposium defence june	No spam
4	siks symposium deadline june	No spam
5	registration assistance symposium deadline	?

- P(spam) = 2/4
- P(registration|spam) = 0/8
- P(assistance|spam) = 1/8
- P(symposium|spam) = 0/8
- P(deadline|spam)=0/8

- P(no spam) = 2/4
- P(registration|no spam) = ?
- P(assistance|no spam) = ?
- P(symposium|no spam) = ?
- P(deadline|no spam) = ?

EXAMPLE

Doc id	Content	Class
1	request urgent interest urgent	Spam
2	assistance low interest deposit	Spam
3	symposium defence june	No spam
4	siks symposium deadline june	No spam
5	registration assistance symposium deadline	?

- P(spam) = 2/4
- P(registration|spam) = 0/8
- P(assistance|spam) = 1/8
- P(symposium|spam) = 0/8
- P(deadline|spam)=0/8

- P(no spam) = 2/4
- P(registration|no spam) = 0/7
- P(assistance|no spam) = 0/7
- P(symposium|no spam) = 2/7
- P(deadline|no spam)=1/7

NAIVE BAYES CLASSIFICATION

$$P(c \mid d) = \frac{P(d \mid c)P(c)}{P(d)}$$

$$P\left(d \mid c\right) = P(t_1, t_2, \ldots, t_k \mid c) =$$

$$P\left(d \mid c\right) = P(t_1, t_2, \ldots, t_k \mid c) = P(t_1 \mid c) \cdot P(t_2 \mid c) \cdot \ldots \cdot P(t_k \mid c)$$

EXAMPLE

Doc id	Content	Class
1	request urgent interest urgent	Spam
2	assistance low interest deposit	Spam
3	symposium defence june	No spam
4	siks symposium deadline june	No spam
5	registration assistance symposium deadline	?

- P(spam) = 2/4
- P(registration|spam) = 0/8
- P(assistance|spam) = 1/8
- P(symposium|spam) = 0/8
- P(deadline|spam)=0/8

- P(no spam) = 2/4
- P(registration|no spam) = 0/7
- P(assistance|no spam) = 0/7
- P(symposium|no spam) = 2/7
- P(deadline|no spam)=1/7

NAIVE BAYES CLASSIFICATION

$$P(c \mid d) = \frac{P(d \mid c)P(c)}{P(d)}$$

$$P(d \mid c) = P(t_1, t_2, \ldots, t_k \mid c) =$$

$$P(d \mid c) = P(t_1, t_2, \ldots, t_k \mid c) = P(t_1 \mid c) \cdot P(t_2 \mid c) \cdot \ldots \cdot P(t_k \mid c)$$

- P(spam|registration assistance symposium deadline) =
- 2/4 * 0/8 * 1/8 * 0/8 * 0/8 = 0
- P(no spam|registration assistance symposium deadline) =
- 2/4 * 0/7 * 0/7 * 2/7 * 1/7 = 0

NAIVE BAYES CLASSIFICATION

- Problem with Maximum Likelihood Estimate: outcome is zero for a term-class combination that did not occur in the training data
- In those cases, multiplication of probabilities for all terms will give a posterior probability of zero

NAIVE BAYES CLASSIFICATION

- Solution: add-one smoothing (Laplace smoothing)

  = a uniform prior: assumption that each term occurs one additional time for each class

$$P(t \mid c) = \frac{T_{ct} + 1}{\sum_{t' \in V} (T_{ct'} + 1)} = \frac{T_{ct} + 1}{(\sum_{t' \in V} T_{ct'}) + |V|}$$

Doc id	Content	Class
1	request urgent interest urgent	Spam
2	assistance low interest deposit	Spam
3	symposium defence june	No spam
4	siks symposium deadline june	No spam
5	registration assistance symposium deadline	?

- P(spam) = 2/4
- P(registration|spam) =
- P(assistance|spam) =
- P(symposium|spam) =
- P(deadline|spam) =

Doc id	Content	Class
1	request urgent interest urgent	Spam
2	assistance low interest deposit	Spam
3	symposium defence june	No spam
4	siks symposium deadline june	No spam
5	registration assistance symposium deadline	?

- P(spam) = 2/4
- P(registration|spam) = (0+1)/(8+11)
- P(assistance|spam) = (1+1)/(8+11)
- P(symposium|spam) = (0+1)/(8+11)
- P(deadline|spam) = (0+1)/(8+11)

Doc id	Content	Class
1	request urgent interest urgent	Spam
2	assistance low interest deposit	Spam
3	symposium defence june	No spam
4	siks symposium deadline june	No spam
5	registration assistance symposium deadline	?

- P(spam) = 2/4
- P(registration|spam) = (0+1)/(8+11)
- P(assistance|spam) = (1+1)/(8+11)
- P(symposium|spam) = (0+1)/(8+11)
- P(deadline|spam) = (0+1)/(8+11)

- P(spam|registration assistance symposium deadline) =
- $2/4 * 1/19 * 2/19 * 1/19 * 1/19 = 7.67 * 10^{-6}$

Doc id	Content	Class
1	request urgent interest urgent	Spam
2	assistance low interest deposit	Spam
3	symposium defence june	No spam
4	siks symposium deadline june	No spam
5	registration assistance symposium deadline	?

- P(no spam) = 2/4
- P(registration|no spam) =
- P(assistance|no spam) =
- P(symposium|no spam) =
- P(deadline|no spam) =

EXERCISE

Doc id	Content	Class
1	request urgent interest urgent	Spam
2	assistance low interest deposit	Spam
3	symposium defence june	No spam
4	siks symposium deadline june	No spam
5	registration assistance symposium deadline	?

- P(no spam) = 2/4
- P(registration|no spam) = (0+1)/(7+11)
- P(assistance|no spam) = (0+1)/(7+11)
- P(symposium|no spam) = (2+1)/(7+11)
- P(deadline|no spam) =(1+1/7+11)

- P(no spam|registration assistance symposium deadline) =
- 2/4 * 1/18 * 1/18 * 3/18 * 2/18 = $2.86 * 10^{-5}$

EXAMPLE

- P(spam|registration assistance symposium deadline) = $7.67 * 10^{-6}$
- P(no spam|registration assistance symposium deadline) = $2.86 * 10^{-5}$

EXAMPLE

- P(spam|registration assistance symposium deadline) = $7.67 * 10^{-6}$
- P(no spam|registration assistance symposium deadline) = $2.86 * 10^{-5}$

- P(no spam|registration assistance symposium deadline) > P(spam|registration assistance symposium deadline)
- → This message is no spam

NAIVE BAYES CLASSIFICATION

- Assumptions:
  - Conditional Independence Assumption: features are independent of each other given the class:

$$P(d \mid c) = P(x_1, x_2, \ldots, x_k \mid c) = P(x_1 \mid c) \cdot P(x_2 \mid c) \cdot \ldots \cdot P(x_k \mid c)$$

  - Positional Independence Assumption: the conditional probabilities of a term are the same, independent of the position in the document

NAIVE BAYES CLASSIFICATION

- Classification results of Naive Bayes (the class with maximum posterior probability) are usually fairly accurate
- However, due to the inadequacy of the conditional independence assumption, the actual posterior-probability numerical estimates are not
- Correct estimation $\Rightarrow$ accurate prediction, but correct probability estimation is NOT necessary for accurate prediction (just need right ordering of probabilities)
- A good baseline for text classification

WHAT IS NEEDED FOR TEXT CLASSIFICATION

- A definition of the task
- Example data
- Pre-processing
- Feature extraction
- Classifier learning
- Evaluation

CLASSIFIER EVALUATION

J&M SECTION 4.7, 4.8

TEST DATA

- We use a held-out test set for evaluation.
  - Prevents overfitting on the training set
  - Split the example data in a training set and a test set
  - E.g. 80% as train set and 20% as test set

- For hyperparameter tuning, we use a validation set
  - This is typically part of the train set
  - Often using cross validation

https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_validate.html

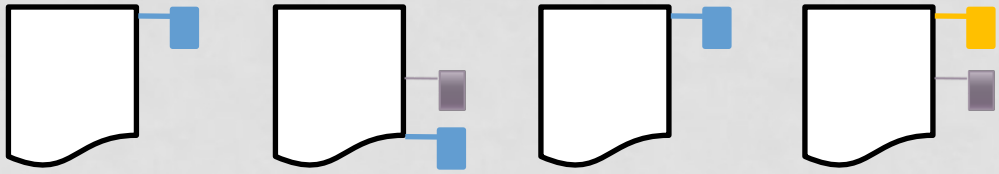
EVALUATION METRICS

- Why is accuracy often not suitable?
  - The 2 classes are often unbalanced. High accuracy in one class might mean low accuracy in the other class
  - Also, we might be more interested in correctness of the labels than in completeness of the labels, or vice versa
- Alternative: use precision and recall

PRECISION AND RECALL

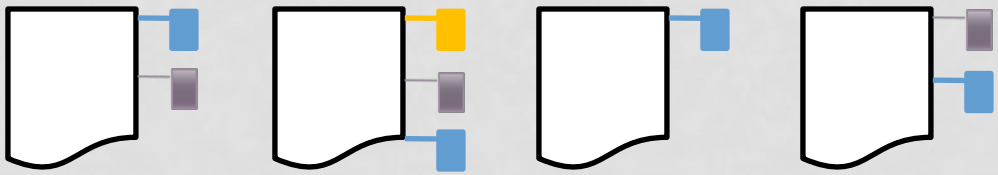
- Human classification = reference (true)

- Automatic classification

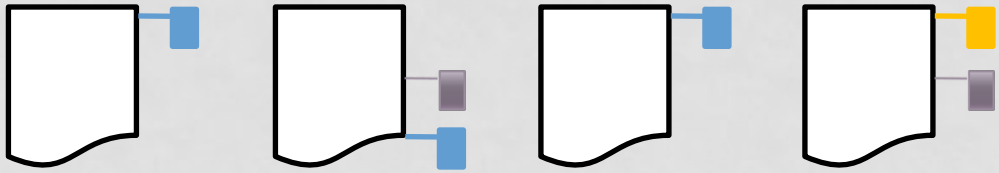


PRECISION AND RECALL

- Human classification = reference (true)



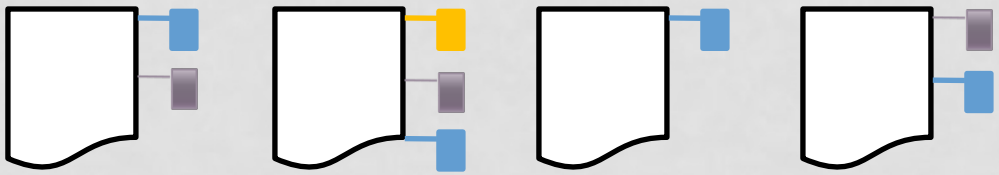
- Automatic classification



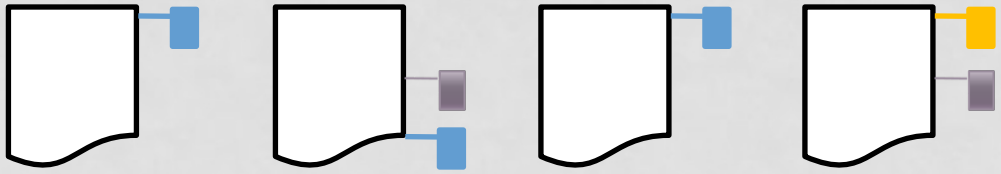
- 8 true categories
  - of which 5 assigned
  - Recall = 5/8
- 6 assigned categories
  - of which 5 correct
  - Precision = 5/6

PRECISION AND RECALL

- Human classification = reference (true)



- Automatic classification

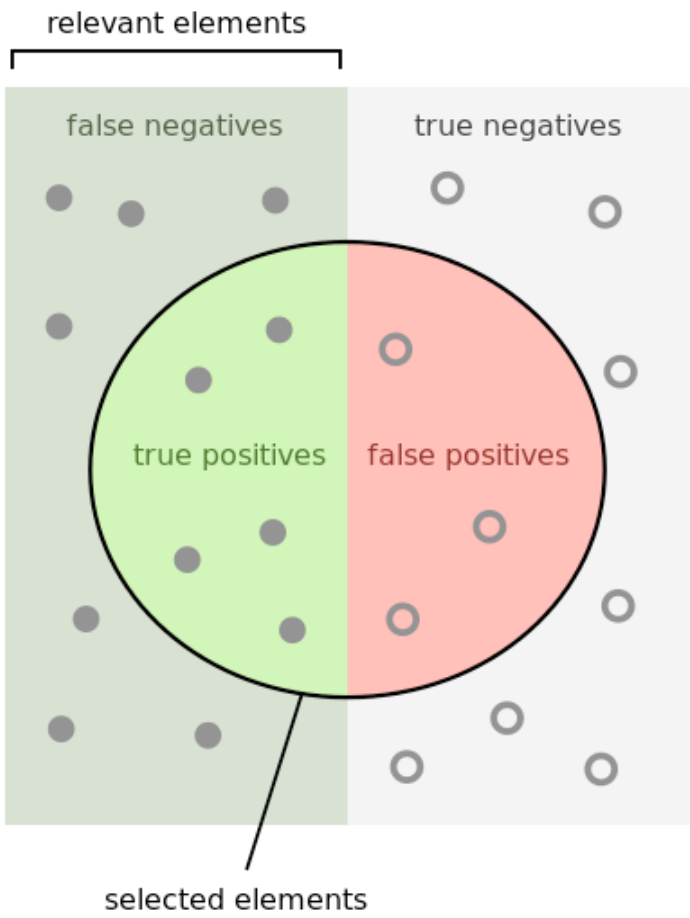


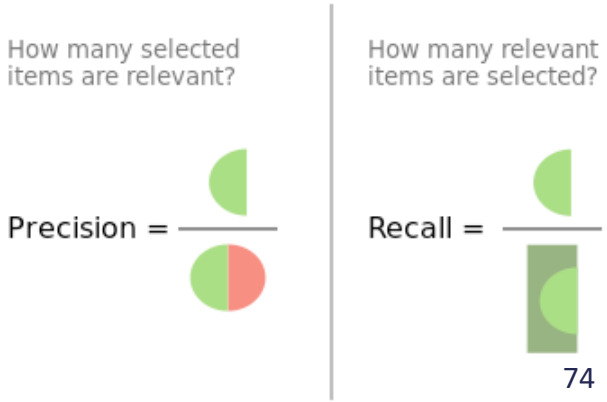
- 8 true categories
  - of which 5 assigned
  - Recall = 5/8
- 6 assigned categories
  - of which 5 correct
  - Precision = 5/6
- We also report the mean, the F1:

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

PRECISION AND RECALL

- $Precision = \frac{tp}{tp+fp}$
- $Recall = \frac{tp}{tp+fn}$





CONCLUSIONS

HOMEWORK

- Read:
  - Jurafsky & Martin chapter 4. Naive Bayes Classification
- Exercise week 4: text categorization with sklearn
  - Follow the tutorial on:http://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html (the exercises are not required)
  - Make sure you understand what the steps mean
  - This is **not** a hand-in assignment. After this exercise, you will complete the 1st hand-in assignment on this topic (deadline **17 October**)

HOMEWORK

- Form a group for the first assignment if you haven't done that yet
- Note that the assignment is only visible in Brightspace if you have enrolled in a group.

AFTER THIS LECTURE…

- you can define the difference between multi-class and multi-label classification
- you can describe the pre-processing needed for text categorization with word features
- you can compute the tf-idf weight for a term given its term count, document count, and corpus size
- you can compute a Naive Bayes classifier given a toy data set
- you can evaluate classification results using precision and recall