



Statistical Natural Language Processing

Lecture 7: Text Classification

Dr. Momtazi

Amirkabir University of Technology

Outline

2

- ➊ Applications
- ➋ Task
- ➌ Naïve Bayes Classification
Smoothing
Language Modeling
- ➍ Evaluation

Outline

3

- 1 Applications
- 2 Task
- 3 Naïve Bayes Classification
Smoothing
Language Modeling
- 4 Evaluation

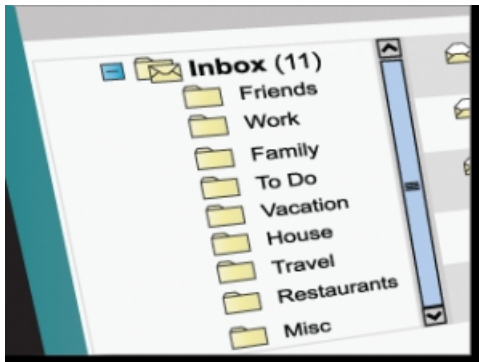
Spam Mail Detection

4



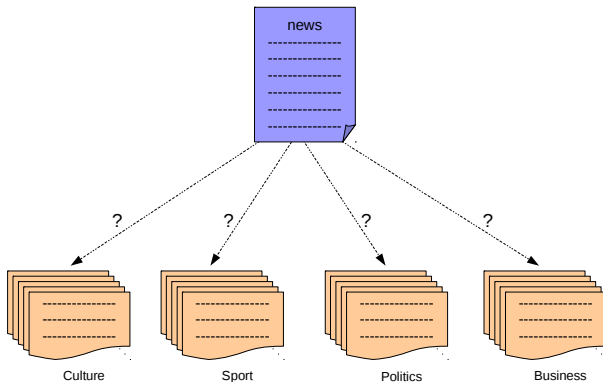
Email Foldering

5



News Classification

6



Language Identification

7

The screenshot displays the Google Translate web interface. At the top, the Google logo is on the left, and the word "Translate" is in red. Below the logo, there are two dropdown menus: "From: English - detected" and "To: German". A blue "Translate" button is to the right of these menus. Below the menus, there are two tabs: "English" (selected) and "Spanish". The input text area contains the sentence "This is a sample sentence in English which is translated to German." with a close button (X) and a speaker icon. The output text area shows the German translation: "Dies ist ein Beispielsatz in englischer Sprache, die auf deutsch übersetzt wird." with a speaker icon and a checkmark. At the bottom, there is a red "New!" label followed by the text "Click the words above to view alternate translations." and a blue "Dismiss" link.

Google

Translate

From: English - detected To: German Translate

English Spanish French

This is a sample sentence in English which is translated to German.

English Persian German

Dies ist ein Beispielsatz in englischer Sprache, die auf deutsch übersetzt wird.

New! Click the words above to view alternate translations. [Dismiss](#)

Sentiment Analysis

"The song was good."



"I hate the song."



Outline

9

- 1 Applications
- 2 Task
- 3 Naïve Bayes Classification
Smoothing
Language Modeling
- 4 Evaluation

Task

10

■ Input

- A document d
- A fixed set of classes $C = c_1, c_2, \dots, c_n$

■ Output

- A predicted class $\hat{c} \in C$

Variations

- Binary vs. Multiclass
- Flat vs. Hierarchical
- Hard vs. Soft (Multi-label)

Supervised Categorization

12

- Using a training set of m manually labeled documents

$d_1 \rightarrow c_1$

$d_2 \rightarrow c_2$

...

$d_m \rightarrow c_m$

- Applying any kinds of classifiers

- K Nearest Neighbor
- Support Vector Machines
- Naïve Bayes
- Maximum Entropy
- Logistic Regression
- ...

Outline

- 1 Applications
- 2 Task
- 3 Naïve Bayes Classification**
 - Smoothing
 - Language Modeling
- 4 Evaluation

Naïve Bayes

14

- Selecting the class with highest probability
⇒ Minimizing the number of items with wrong labels

$$\hat{c} = \operatorname{argmax}_{c_i} P(c_i|d)$$

$$\hat{c} = \operatorname{argmax}_{c_i} \frac{P(d|c_i) \cdot P(c_i)}{P(d)}$$

$P(d)$ has no effect

$$\hat{c} = \operatorname{argmax}_{c_i} P(d|c_i) \cdot P(c_i)$$

Naïve Bayes

15

$$\hat{c} = \operatorname{argmax}_{c_i} P(d|c_i) \cdot P(c_i)$$

Likelihood
Probability

Prior
Probability

Prior Probability

$$P(c_i)$$

- How much the class c_i is important disregarding the document?

$$P(c_i) = \frac{\#(c_i)}{N}$$

Likelihood Probability

$$P(d|c_i)$$

- How likely the document d is selected, if we know c_i is the correct class?
⇒ How likely each of the words from document d will be selected if we know c_i is the correct class?

$$P(d|c_i) = \prod_{w \in d} P(w|c_i)$$

$$P(w|c_i) = \frac{\#(w, c_i)}{\sum_{w'} \#(w', c_i)}$$

Outline

- 1 Applications
- 2 Task
- 3 Naïve Bayes Classification
Smoothing
Language Modeling
- 4 Evaluation

Smoothing

$$P(d|c_i) = \prod_{w \in d} P(w|c_i)$$
$$P(w|c_i) = \frac{\#(w, c_i)}{\sum_{w'} \#(w', c_i)}$$

■ Shortcomings

- Words that are not available in the training data produce zero probability
- Even one zero probability makes the whole result zero

■ Solution

- Using a smoothing method to avoid zero probability

Smoothing

20

$$P(d|c_i) = \prod_{w \in d} P(w|c_i)$$

$$P(w|c_i) = \frac{\#(w, c_i)}{\sum_{w'} \#(w', c_i)}$$

- Laplace (add-one) smoothing

$$P(w|c_i) = \frac{\#(w, c_i) + 1}{\sum_{w'} \#(w', c_i) + |V|}$$

Smoothing

21

$$P(d|c_i) = \prod_{w \in d} P(w|c_i)$$

$$P(w|c_i) = \frac{\#(w, c_i)}{\sum_{w'} \#(w', c_i)}$$

- Advanced smoothing methods
 - Bayesian smoothing with Dirichlet prior
 - Absolute discounting
 - Kneser-Ney smoothing

Outline

22

- 1 Applications
- 2 Task
- 3 Naïve Bayes Classification
Smoothing
Language Modeling
- 4 Evaluation

Naïve Bayes Classifier

23

$$P(d|c_i) = \prod_{w \in d} P(w|c_i)$$

- Using words of a document as a bag-of-word model
- Similar to the unigram model in language modeling

Naïve Bayes Classifier

24

$$P(d|c_i) = \prod_{w \in d} P(w|c_i)$$

- Shortcoming
 - Considering no dependencies between words
- Solution
 - Using higher order n -grams
⇒ it is not "naïve" any more!

Naïve Bayes Classifier

25

- Unigram

$$P(d|c_i) = \prod_{j=1}^n P(w_j|c_i)$$

$$P(w_j|c_i) = \frac{\#(w_j, c_i)}{\sum_{w'} \#(w', c_i)}$$

Bayes Classifier

26

- Bigram

$$P(d|c_i) = \prod_{j=1}^n P(w_j|w_{j-1}, c_i)$$

$$P(w_j|w_{j-1}, c_i) = \frac{\#(w_{j-1} w_j, c_i)}{\#(w_{j-1}, c_i)}$$

Bayes Classifier

27

■ Trigram

$$P(d|c_i) = \prod_{j=1}^n P(w_j|w_{j-2}w_{j-1}, c_i)$$

$$P(w_j|w_{j-2}w_{j-1}, c_i) = \frac{\#(w_{j-2}w_{j-1}w_j, c_i)}{\#(w_{j-2}w_{j-1}, c_i)}$$

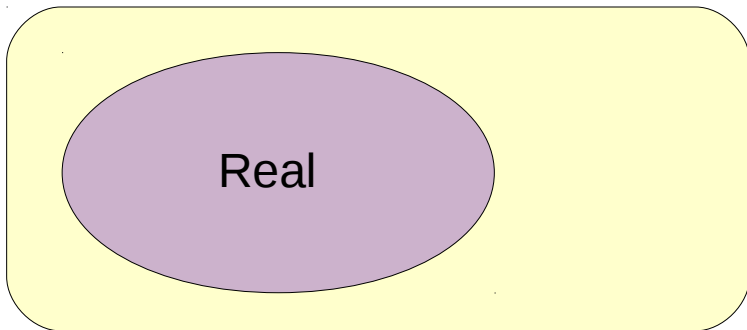
Outline

28

- ① Applications
- ② Task
- ③ Naïve Bayes Classification
 - Smoothing
 - Language Modeling
- ④ Evaluation

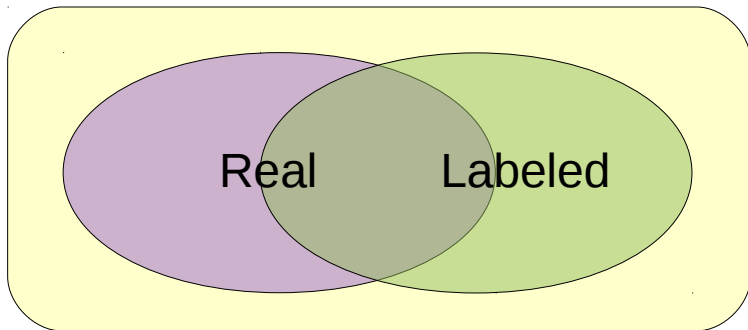
Precision and Recall

29



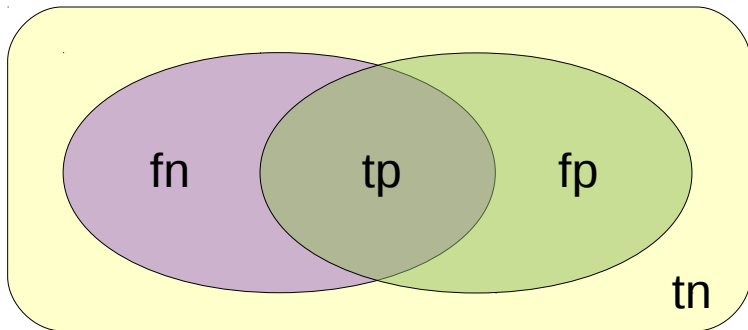
Precision and Recall

30



Precision and Recall

31



Precision and Recall

32

■ Confusion matrix:

	real positive	real negative
labeled positive	tp	fp
labeled negative	fn	tn

■ Precision:

- Amount of labeled item that are correct

$$Precision = \frac{tp}{tp + fp}$$

■ Recall:

- Amount of correct item that are labeled

$$Recall = \frac{tp}{tp + fn}$$

Precision/Recall Trade off

33

- There is a strong anti-correlation between precision and recall
- Having a trade off between these two metrics
 - Achieving higher recall ends to lower precision
 - Achieving higher precision results lower recall

F-measure

34

- Using F -measure to consider both metrics together
- F -measure is a weighted harmonic mean of precision and recall

$$F = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

- $\beta < 1$ gives a higher priority to precision
- $\beta > 1$ gives higher priority to recall
- $\beta = 1$ gives the same priority to both precision and recall

$$F_1 = \frac{2PR}{P + R}$$

Evaluating Multi-label Classifiers

35

- Creating a separate confusion matrix for each label
 - Positive: the target label
 - Negative: the rest of labels
- Calculating precision, recall, and f-measure for each label
- Taking the average of values
 - Macro-averaging: giving equal weight to each class
 - Micro-averaging: considering the weight of classes

Evaluating Multi-label Classifiers

36

■ Macro- vs. Micro-averaging

label	tp	fp	fn	precision	recall
c_1	10	10	10	0.5	0.5
c_2	90	10	10	0.9	0.9
total	100	20	20		
macro-averaged				0.7	0.7
micro-averaged				0.83	0.83