Statistical Natural Language Processing

Lecture 7: Text Classification

Dr. Momtazi
Amirkabir University of Technology

Outline

1. Applications

2. Task

3. Naïve Bayes Classification
   - Smoothing
   - Language Modeling

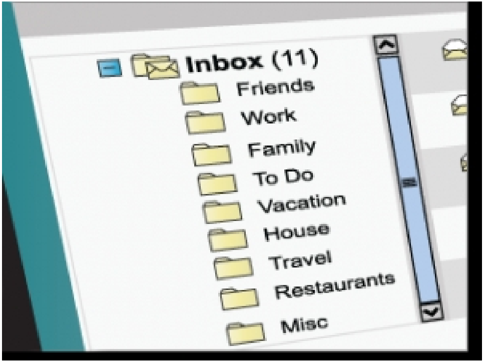
4. Evaluation

Outline

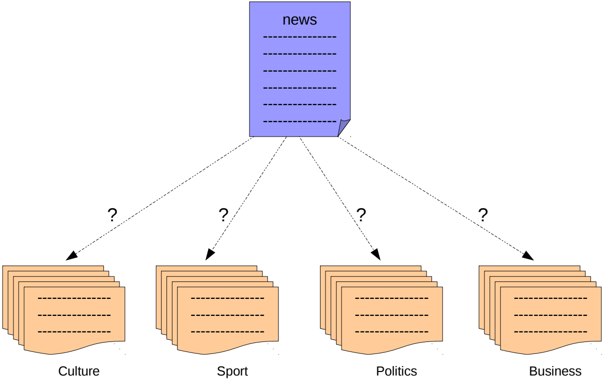
1. Applications
2. Task
3. Naïve Bayes Classification
   - Smoothing
   - Language Modeling
4. Evaluation

Spam Mail Detection

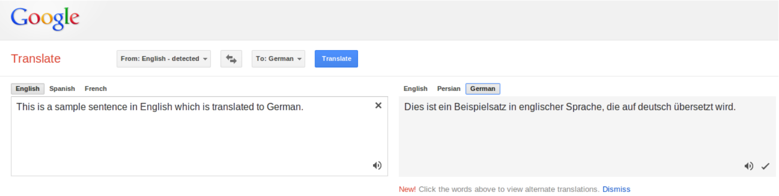
Email Foldering



News Classification



Language Identification



Sentiment Analysis

"The song was good." 

"I hate the song." 

Outline

Task

- Input
  - A document d
  - A fixed set of classes $C = c_1, c_2, ..., c_n$

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

Variations

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

Supervised Categorization

- Using a training set of *m* manually labeled documents

$$d_1 \rightarrow c_1$$
$$d_2 \rightarrow c_2$$
$$\ldots$$
$$d_m \rightarrow c_m$$

- Applying any kinds of classifiers
  - *K* Nearest Neighbor
  - Support Vector Machines
  - Naïve Bayes
  - Maximum Entropy
  - Logistic Regression
  - ...

Outline

Naïve Bayes

- Selecting the class with highest probability
  $\Rightarrow$ Minimizing the number of items with wrong labels

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

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

$P(d)$ has no effect

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

Naïve Bayes

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

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?
  $\Rightarrow$ 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

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

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

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

- Laplace (add-one) smoothing

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

Smoothing

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

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

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

Outline

Naïve Bayes Classifier

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

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

Naïve Bayes Classifier

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

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

Naïve Bayes Classifier

- 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

- 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

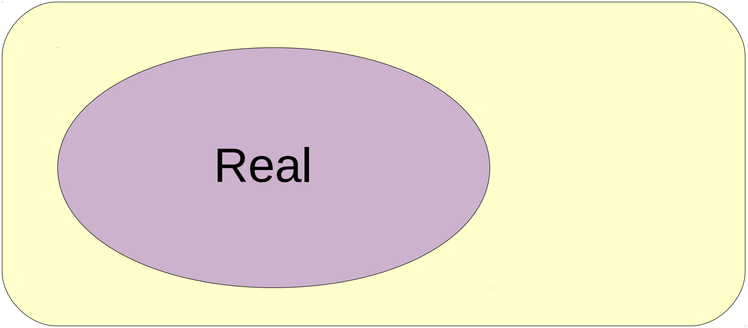
- Trigram

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

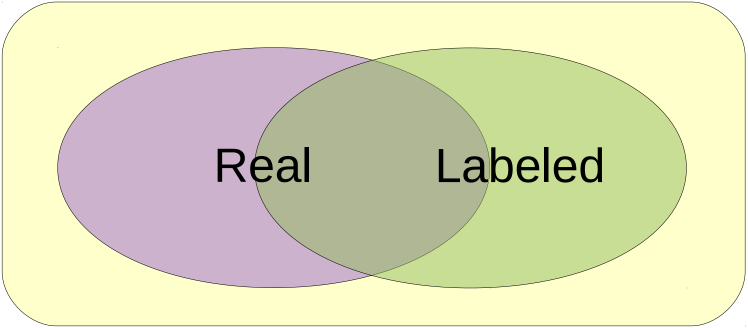
$$P(w_j|w_{j-2}w_{j-1}, c_i) = \frac{\#(w_{j-2}w_{j-1}w_j, c_i)}{\#(w_{j-2}w_{j-1}, c_i)}$$

Outline

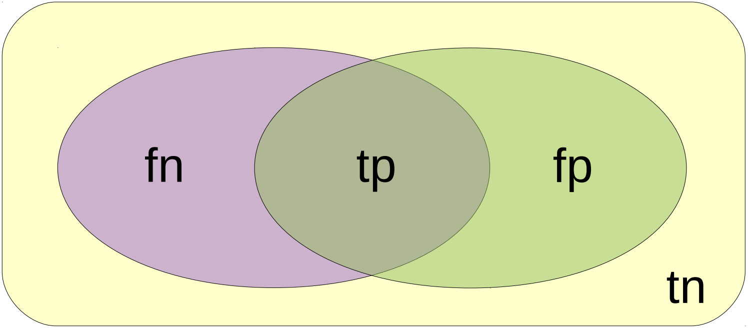
Precision and Recall



Precision and Recall



Precision and Recall



Precision and Recall

- 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

- 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

- 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

- 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

- 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