Lecture 4
Text classification

LTAT.01.001 – Natural Language Processing
Kairit Sirts (kairit.sirts@ut.ee)
06.03.2020
Plan for today

- Text classification tasks
- Text representations for classification
- Generative vs discriminative models
- Text classification with feed-forward neural networks
- Evaluating text classifiers
Text classification tasks
Text classification
Text/document classification

- Spam detection
- Topic classification (sport/finance/travel etc)
- Genre classification (news/sports/fiction/social media etc)
- Sentiment analysis
- Emotion prediction
- Dialogue intent detection
Authorship attribution

- Native language identification
- Clinical text classification
  - diagnosing psychiatric or cognitive impairments
  - Predicting diagnosis from clinical notes
- Identification of gender, dialect, educational background, level of language proficiency etc
Types of classification tasks

- Binary classification (true/false, 1/0, 1/-1)
- Multi-class classification (text genre: politics/finance/travel)
- Multi-label classification (image captioning)
- Clustering: mostly unsupervised
  - Topic modeling – important but will not talk about it today
Multi-label classification

- What if each document can have multiple labels?

- Turn the multi-label classification problem into several binary classification problems

- Form “super-classes” from label combinations and perform multi-class classification
Classification vs regression

- What if the labels are ordered?
- For instance sentiment scores: -3, -2, -1, 0, 1, 2, 3
  - -3 – the sentiment/attitude is extremely negative
  - 0 – the sentiment/attitude is neutral
  - 3 – the sentiment/attitude is very positive
- With multi-class classification all scores are similarly dissimilar
  - Here 2 and 3 are more similar than -3 and 3
- It might be good idea to try regression instead of classification to predict the continuous score
Text representation for classification
Preprocessing: text normalization

- Stemming/Lemmatization
- Remove stop words
- Remove infrequent words
- Replace certain word tokens with special entities
  - Replace all numeric tokens with NUM
- Lowercase the text
- Social media - handle emoticons
- Sentence and word/token segmentation
How to represent document $d$?

- **Bag of Words (BOW)**
  - Easy, no effort required
  - Ignores sentence structure

- **Hand-crafted features**
  - Can use NLP pipelines, class-specific features
  - Incomplete, makes use of NLP pipelines, need to do feature engineering

- **Learned feature representations**
  - Can learn to contain task-relevant information
  - Need to be learned
BOW representations

- Binary BOW

<table>
<thead>
<tr>
<th>the</th>
<th>your</th>
<th>model</th>
<th>cash</th>
<th>Viagra</th>
<th>class</th>
<th>account</th>
<th>orderz</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

- Multinominal BOW

<table>
<thead>
<tr>
<th>the</th>
<th>your</th>
<th>model</th>
<th>cash</th>
<th>Viagra</th>
<th>class</th>
<th>account</th>
<th>orderz</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

- Tf-idf
**Tf-idf**

- **Tf-idf** = \( tf \) (term frequency) × \( idf \) (inverse document frequency)

- **Term frequency:**
  - Raw frequency
  - Binary
  - Raw frequency normalized by the document length

- **Inverse document frequency**
  - \( N \) - the number of documents
  - \( n_t \) – the number of documents containing term \( t \)

\[
idf = \log \frac{N}{n_t}
\]
Tf-idf example

- $\text{Tf-idf}(\text{doc4, the}) = \frac{n_{d,t} \cdot \log \frac{N}{n_t}}{\text{Tf-idf}(\text{doc4, your}) = \frac{n_{d,t} \cdot \log \frac{N}{n_t}}{\text{Tf-idf}(\text{doc4, model}) = \frac{n_{d,t} \cdot \log \frac{N}{n_t}}{\text{Tf-idf}(\text{doc4, cash}) = \frac{n_{d,t} \cdot \log \frac{N}{n_t}}{\text{Tf-idf}(\text{doc4, Viagra}) = \frac{n_{d,t} \cdot \log \frac{N}{n_t}}{\text{Tf-idf}(\text{doc4, class}) = \frac{n_{d,t} \cdot \log \frac{N}{n_t}}{\text{Tf-idf}(\text{doc4, account}) = \frac{n_{d,t} \cdot \log \frac{N}{n_t}}{\text{Tf-idf}(\text{doc4, orderz}) = \frac{n_{d,t} \cdot \log \frac{N}{n_t}}{\text{Table}}$

<table>
<thead>
<tr>
<th></th>
<th>the</th>
<th>your</th>
<th>model</th>
<th>cash</th>
<th>Viagra</th>
<th>class</th>
<th>account</th>
<th>orderz</th>
</tr>
</thead>
<tbody>
<tr>
<td>doc1</td>
<td>12</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>doc2</td>
<td>10</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>doc3</td>
<td>25</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>doc4</td>
<td>14</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>doc5</td>
<td>17</td>
<td>5</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Tf-idf example

- Tf-idf(doc4, the) = 0
- Tf-idf(doc4, your) = 0
- Tf-idf(doc4, model) = 0
- Tf-idf(doc4, cash) = $\log(5/3) = 0.51$
- Tf-idf(doc4, Viagra) = $3\log(5) = 4.83$
- Tf-idf(doc4, class) = 0
- Tf-idf(doc4, account) = $\log(5/4) = 0.22$
- Tf-idf(doc4, orderz) = $\log(5/2) = 0.92$

\[
\text{tf-idf} = n_{d,t} \cdot \log \frac{N}{n_t}
\]

<table>
<thead>
<tr>
<th></th>
<th>the</th>
<th>your</th>
<th>model</th>
<th>cash</th>
<th>Viagra</th>
<th>class</th>
<th>account</th>
<th>orderz</th>
</tr>
</thead>
<tbody>
<tr>
<td>doc1</td>
<td>12</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>doc2</td>
<td>10</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>doc3</td>
<td>25</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td><strong>doc4</strong></td>
<td><strong>14</strong></td>
<td><strong>2</strong></td>
<td><strong>0</strong></td>
<td><strong>1</strong></td>
<td><strong>3</strong></td>
<td><strong>0</strong></td>
<td><strong>1</strong></td>
<td><strong>1</strong></td>
</tr>
<tr>
<td>doc5</td>
<td>17</td>
<td>5</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Tf-idf example

\[ \text{tf-idf} = n_{d,t} \cdot \log \frac{N}{n_t} \]

<table>
<thead>
<tr>
<th></th>
<th>the</th>
<th>your</th>
<th>model</th>
<th>cash</th>
<th>Viagra</th>
<th>class</th>
<th>account</th>
<th>orderz</th>
</tr>
</thead>
<tbody>
<tr>
<td>doc1</td>
<td>0</td>
<td>0</td>
<td>1.61</td>
<td>0</td>
<td>0</td>
<td>1.83</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>doc2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2.04</td>
<td>0</td>
<td>0</td>
<td>0.45</td>
<td>0</td>
</tr>
<tr>
<td>doc3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.92</td>
<td>0.22</td>
<td>0</td>
</tr>
<tr>
<td><strong>doc4</strong></td>
<td><strong>0</strong></td>
<td><strong>0</strong></td>
<td><strong>0</strong></td>
<td><strong>0.51</strong></td>
<td><strong>4.83</strong></td>
<td><strong>0</strong></td>
<td><strong>0.22</strong></td>
<td><strong>0.92</strong></td>
</tr>
<tr>
<td>doc5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.02</td>
<td>0</td>
<td>0</td>
<td>0.22</td>
<td>0.92</td>
</tr>
</tbody>
</table>
Variations of BOW – word n-grams

new price update for the cake decoration tools from Alick
Variations of BOW – character n-grams

el gato se sentó en la alfombra
BOW representations in scikit-learn

- `sklearn.feature_extraction.text.CountVectorizer`
- `sklearn.feature_extraction.text.TfidfVectorizer`
Generative vs discriminative classifiers
Generative task formulation

Given document $d$ and a set of class labels $C$, assign to $d$ the most probable label $\hat{c}$.

$$
\hat{c} = \arg \max_{c \in C} P(c|d)
$$

Bayes rule

$$
= \arg \max_{c \in C} \frac{P(d|c)P(c)}{P(d)}
$$

$\text{Naïve Bayes}$
Discriminative task formulation

- Given document $d$ and a set of class labels $C$, assign to $d$ the most probable label $\hat{c}$.

$$\hat{c} = \arg \max_{c \in C} P(c|d)$$

Logistic regression
Generative vs discriminative models

● Generative (joint) models – P(c, d)
  ● Model the probability of both input document and output label
  ● Can be used to generate a new document with a particular label
  ● N-gram models, Naïve Bayes

● Discriminative (conditional) models - P(c|d)
  ● Learn boundaries between classes. Input data is taken as given.
  ● Logistic regression, SVM, most neural models
Text classification with feed-forward neural networks
Feed-forward sentence classification

\[ h = g(W^{(i)}x + b^{(i)}) \]

\[ z = W^{(o)}h + b^{(o)} \]

\[ p(c|x) = \text{softmax}(z) \]

\[ x = \sum_{i=1}^{t} v_{wi} \]

Embed Embed Embed Embed Embed Embed

The cat sat on the mat
FastText text classification

- Very quick and simple text classification baseline, often with good performance
- Joulin et al., 2017. Bag of Tricks for Efficient Text Classification
- C++ code available: https://github.com/facebookresearch/fastText
- Trains the sub-word-based word embeddings along with the classifier
- Can also use pre-trained FastText embeddings
FastText text classification

- It is essentially a CBoW model
- Form a text representation by averaging the embeddings of all words in text

\[
\text{new price update for the cake decoration tools from Alick}
\]

\[
v_1 \quad v_2 \quad v_3 \quad v_4 \quad v_5 \quad v_6 \quad v_7 \quad v_8 \quad v_9 \quad v_{10}
\]

\[
h = \frac{1}{N} \sum_{i=1}^{N} v_i \quad \quad p(c|h) = \text{softmax}(hW)
\]

\[W \in \mathbb{R}^{d \times |C|}\]
Evaluating text classification
Evaluation measures

- **Accuracy**: not suitable when the classes are unbalanced
- **Precision**: also called positive predictive value
- **Recall**: also called true positive rate or sensitivity
- **F1-score**: combines precision and recall
Accuracy

- Accuracy = \[ \frac{\#Correct}{\#Total} \]

<table>
<thead>
<tr>
<th>Gold</th>
<th>Predicted</th>
<th>Document about Sports?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>

- Correct =
- Total =
- Accuracy =
Accuracy

\[ \text{Accuracy} = \frac{\#\text{Correct}}{\#\text{Total}} \]

<table>
<thead>
<tr>
<th>Gold</th>
<th>Y</th>
<th>Y</th>
<th>N</th>
<th>N</th>
<th>Y</th>
<th>N</th>
<th>N</th>
<th>N</th>
<th>Y</th>
<th>N</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predicted</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>

- Correct = 8
- Total = 11
- Accuracy = 72.7%
Accuracy – another example

- Accuracy = \[\frac{\#Correct}{\#Total}\]

<table>
<thead>
<tr>
<th></th>
<th>Document about Sports?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gold</td>
<td>N N N N N Y N N N N N</td>
</tr>
<tr>
<td>Predicted</td>
<td>N N N N N N N N N N</td>
</tr>
</tbody>
</table>

- Correct =
- Total =
- Accuracy =
Accuracy – another example

- Accuracy = \( \frac{\text{#Correct}}{\text{#Total}} \)

<table>
<thead>
<tr>
<th></th>
<th>Document about Sports?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gold</td>
<td>Y</td>
</tr>
<tr>
<td>Predicted</td>
<td></td>
</tr>
</tbody>
</table>

- Correct = 10
- Total = 11
- Accuracy = 90.9%
Precision, recall and F1-score

- TP – true positives – correctly predicted positives
- FP – false positives – negatives that were predicted as positives
- FN – false negatives – positives that were predicted as negatives

\[
\text{Precision} = \frac{TP}{TP + FP} \quad \text{Recall} = \frac{TP}{TP + FN}
\]

\[
F1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}
\]
Precision and recall

Precision and recall: example

<table>
<thead>
<tr>
<th>Gold</th>
<th>Y</th>
<th>Y</th>
<th>N</th>
<th>N</th>
<th>Y</th>
<th>N</th>
<th>N</th>
<th>N</th>
<th>Y</th>
<th>N</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predicted</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>

- True Positives =
- False Positives =
- False Negatives =
- Precision =
- Recall =
- F1-score =
Precision and recall: example

<table>
<thead>
<tr>
<th>Document about Sports?</th>
<th>Gold</th>
<th>Predicted</th>
</tr>
</thead>
<tbody>
<tr>
<td>True Positives = 2</td>
<td>Y Y</td>
<td>N Y N N Y N N N Y N N</td>
</tr>
<tr>
<td>False Positives = 1</td>
<td>N Y</td>
<td>N Y N N N N Y N N</td>
</tr>
<tr>
<td>False Negatives = 2</td>
<td>N Y</td>
<td>N Y N N N N Y N N</td>
</tr>
</tbody>
</table>

- True Positives = 2
- False Positives = 1
- False Negatives = 2
- Precision = 2/3
- Recall = 2/4
- F1-score = 4/7
Precision, recall and F1-score in multi-class setting

- Precision, recall and F1-score assume binary classification

- In the multi-class setting, for each class c:
  - treat the class c as the positive class
  - treat all other classes as negative classes
  - Now can compute precision, recall and F1-score with respect to every class

- Combine the class-specific scores using either micro-averaging or macro-averaging
Micro- and macro-averaging

- **Micro-averaging**
  - Count TP, FP and FNs for all classes and sum them up
  - Compute the precision, recall and F1-score
  - Takes class imbalances more into account

- **Macro-averaging**
  - Compute precision, recall and F1-score with respect to every class
  - Average the scores
  - The scores for each class make an equal contribution

- **Weighted macro-averaging**
  - Weight each class score according by the relative number of instances of that class