Amirkabir University of Technology
(Tehran Polytechnic)

Natural Language Processing

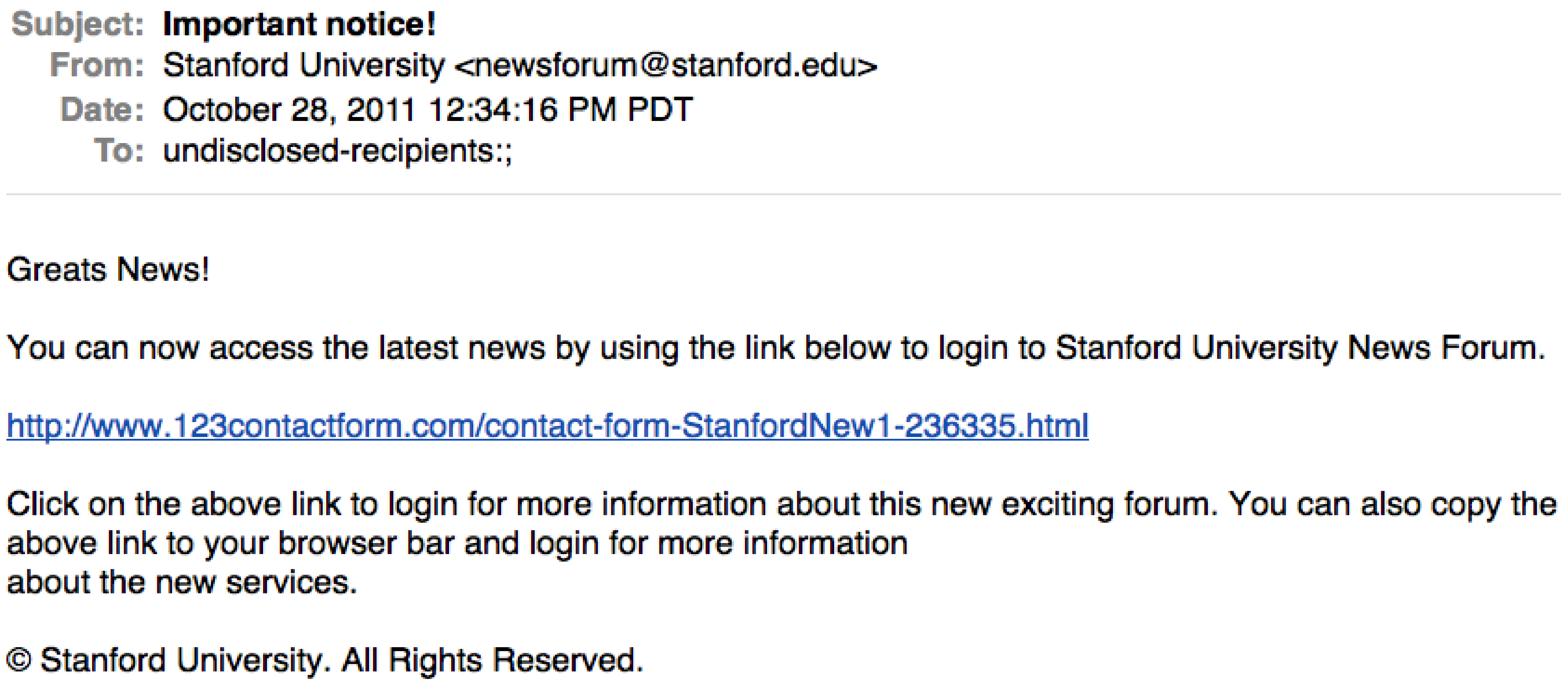
Lecture 9: Text Classification

Amirkabir University of Technology
Dr Momtazi

Outline

- **Applications**
- Task
- Naïve Bayes Classification
- Convolutional Neural Network
- Evaluation

Spam Mail Detection



Subject: **Important notice!**
From: Stanford University <newsforum@stanford.edu>
Date: October 28, 2011 12:34:16 PM PDT
To: undisclosed-recipients:;

Greats News!

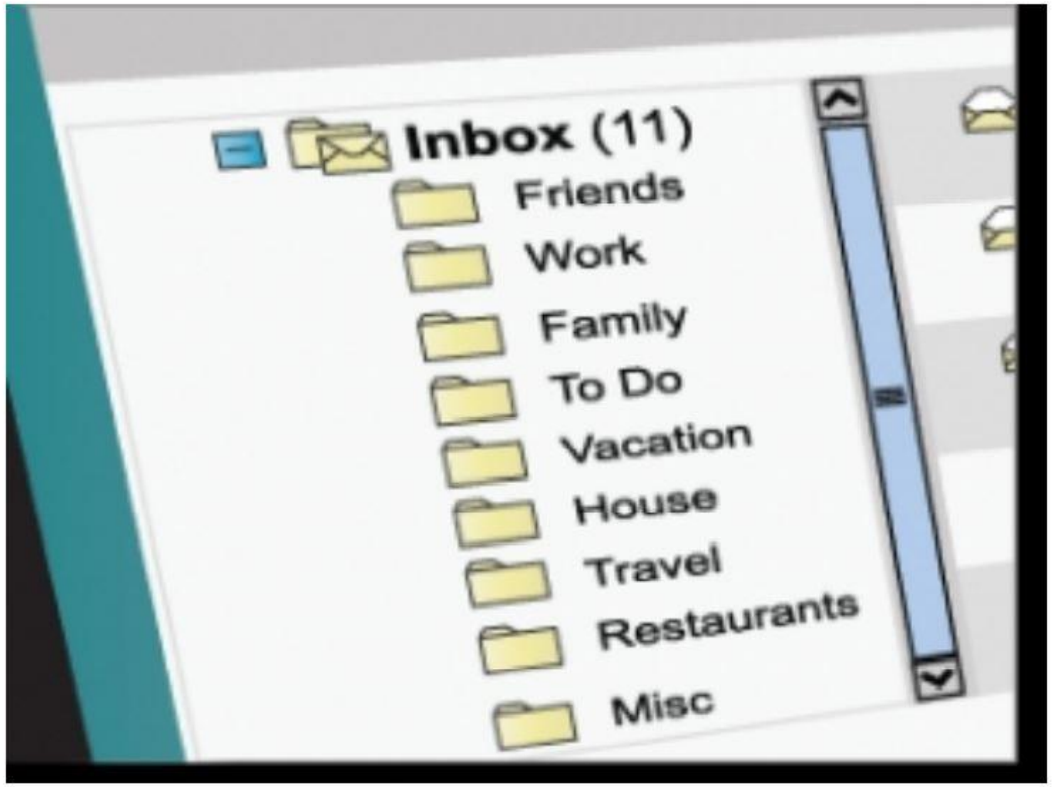
You can now access the latest news by using the link below to login to Stanford University News Forum.

http://www.123contactform.com/contact-form-StanfordNew1-236335.html

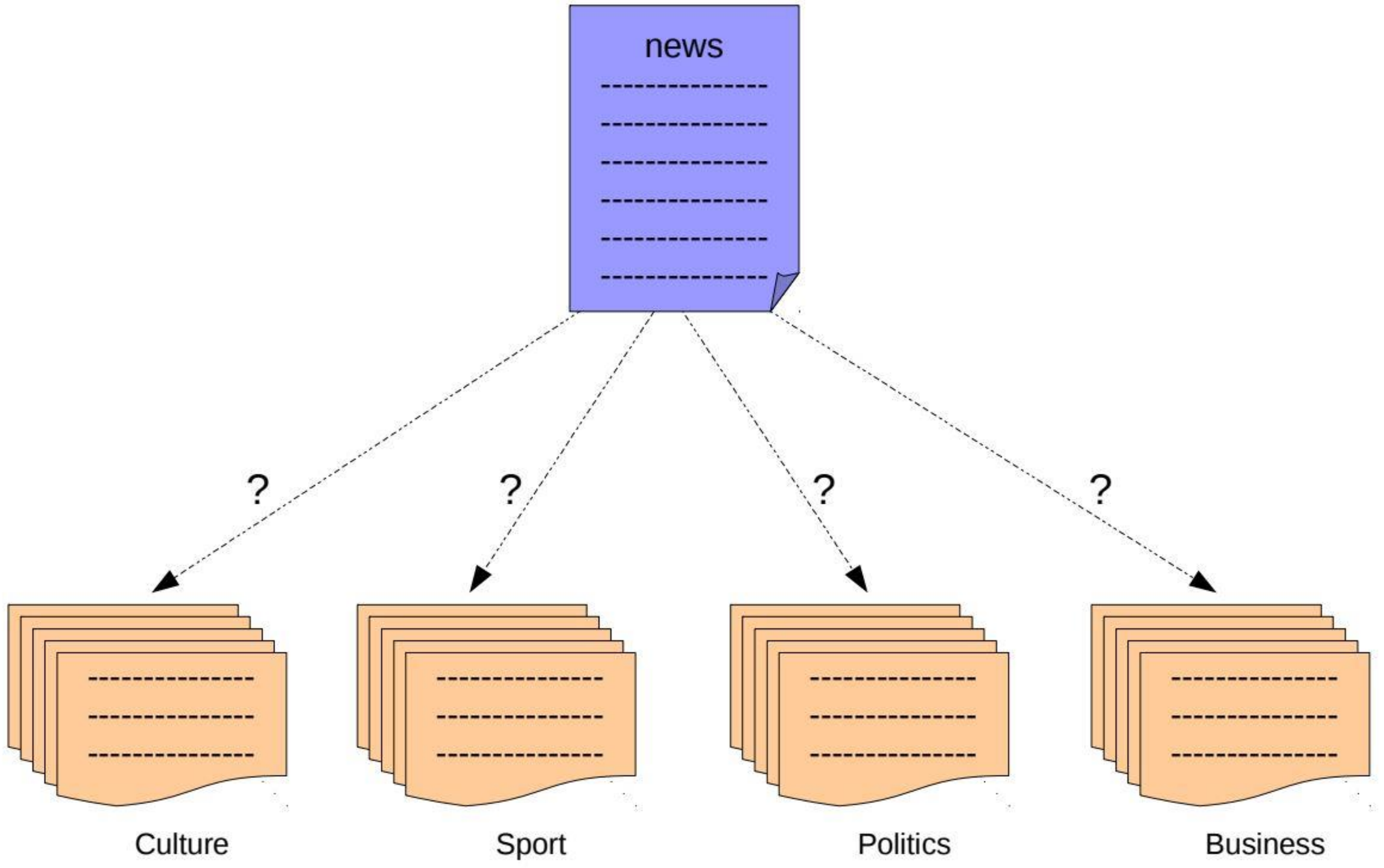
Click on the above link to login for more information about this new exciting forum. You can also copy the above link to your browser bar and login for more information
about the new services.

© Stanford University. All Rights Reserved.

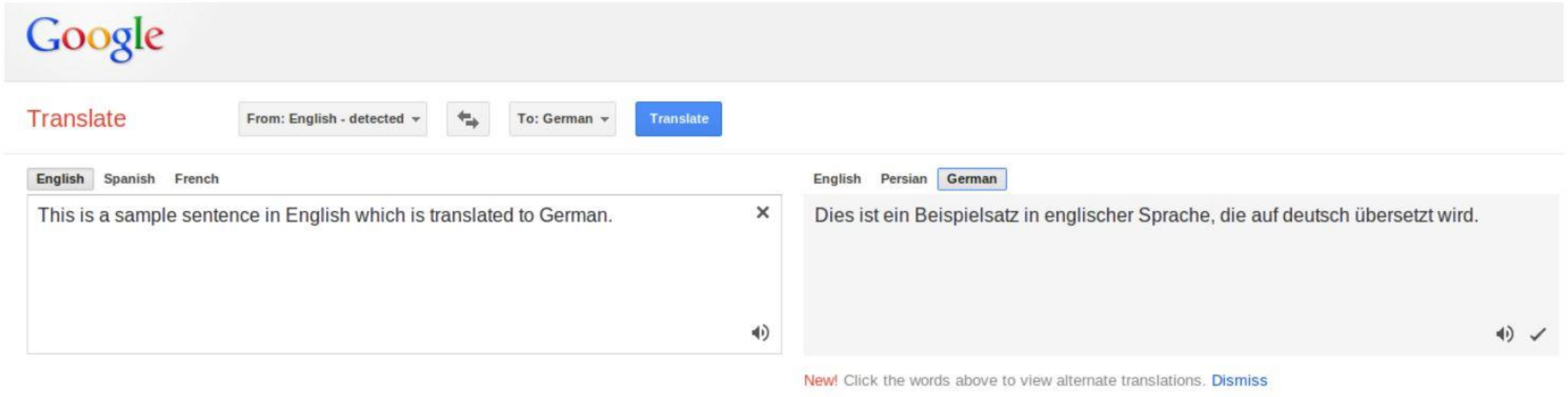
Email Foldering



News Classification



Language Identification



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

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

New! Click the words above to view alternate translations. Dismiss

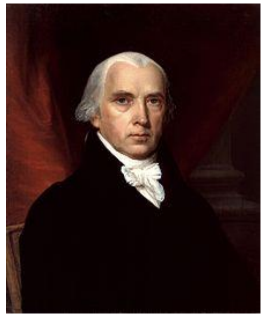
Sentiment Analysis

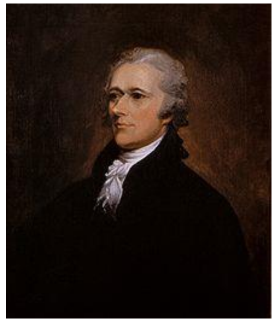
“The song was good.”

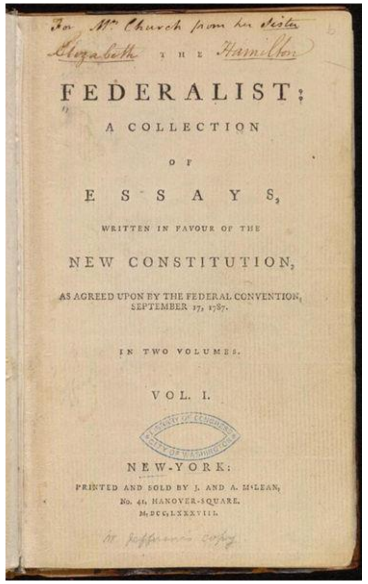
“I hate the song.”

Author Identification

- Who wrote which Federalist papers?


James Madison


Alexander Hamilton


For Mrs Church from her Sister
Elizabeth THE Hamilton
FEDERALIST:
A COLLECTION
OF
ESSAYS,
WRITTEN IN FAVOUR OF THE
NEW CONSTITUTION,
AS AGREED UPON BY THE FEDERAL CONVENTION, SEPTEMBER 17, 1787.
IN TWO VOLUMES.
VOL. I.
LIBRARY OF CONGRESS CITY OF WASHINGTON
NEW-YORK:
PRINTED AND SOLD BY J. AND A. M'LEAN,
No. 41, HANOVER-SQUARE.
M,DCC,LXXXVIII.
Mr. Jefferson's copy

What is the subject of this medical article?

MEDLINE Article

Brain and Cognition

Syntactic frame and verb bias in aphasia: Plausibility judgments of undergoer-subject sentences

Susanne Gahl, Lise Menn, Gail Ramsberger, Daniel S. Jurafsky, Elizabeth Elder, Molly Rewega, and L. Holland Audrey

[illegible]

?

MeSH Subject Category Hierarchy

- Antogonists and Inhibitors
- Blood Supply
- Chemistry
- Drug Therapy
- Embryology
- Epidemiology
- …

Outline

- Applications
- **Task**
- Naïve Bayes Classification
- Convolutional Neural Network
- Evaluation

Text Classification: definition

- *Input*:
  - a document d
  - a fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$

- *Output*: a predicted class $c \in C$

Variations

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

Classification Methods: Supervised Machine Learning

- *Input:*
  - a document d
  - a fixed set of classes $C = \{c_1, c_2,\ldots, c_J\}$
  - A training set of m hand-labeled documents $(d_1,c_1),\ldots,(d_m,c_m)$

- *Output:*
  - a learned classifier $y{:}d \rightarrow c$

Classification Methods: Supervised Machine Learning

- Any kind of classifier
  - Naïve Bayes
  - Logistic regression
  - Neural networks
  - k-Nearest Neighbors
  - …

Outline

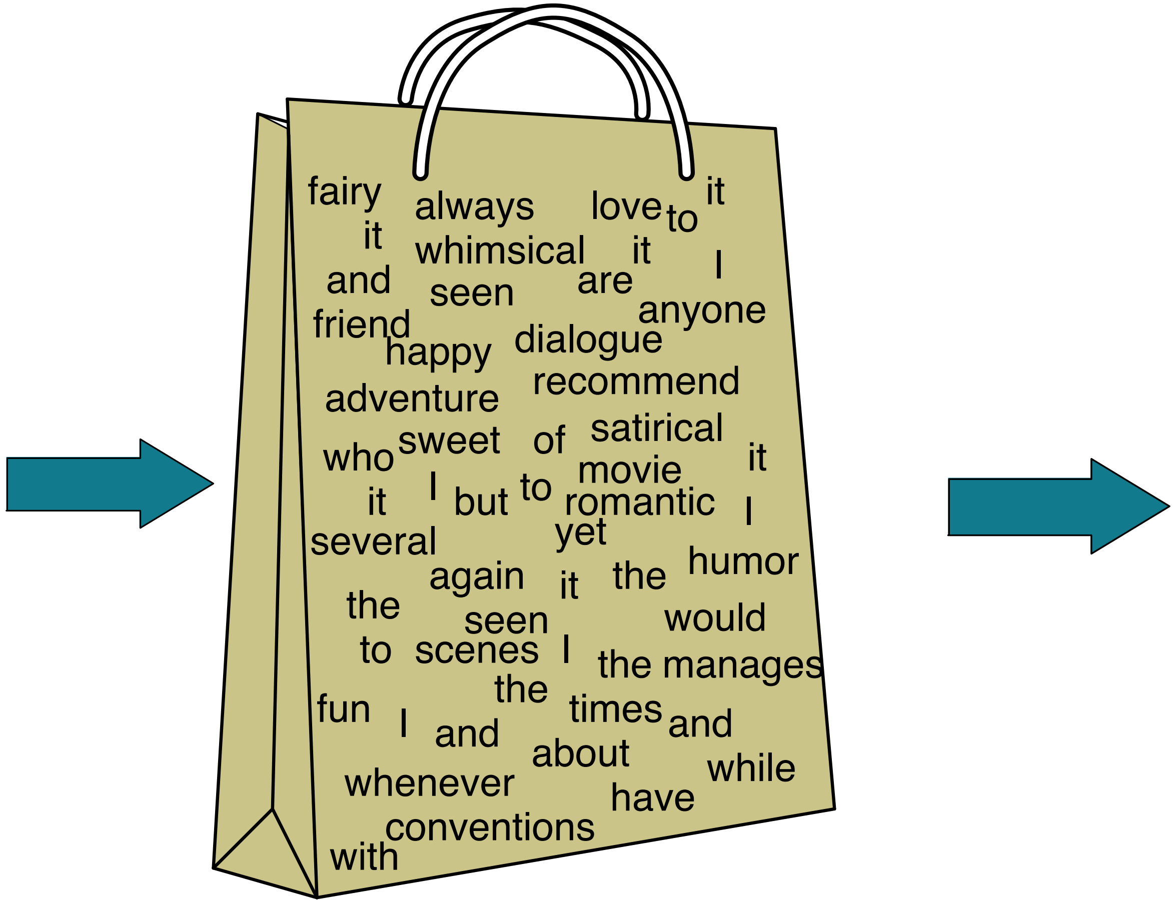
- Applications
- Task
- **Naïve Bayes Classification**
- Convolutional Neural Network
- Evaluation

Naive Bayes Intuition

- Simple ("naive") classification method based on Bayes rule
- Relies on very simple representation of document
  - **Bag of words**

The Bag of Words Representation

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!



it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
…	…

The bag of words representation

$\gamma($

seen	2
sweet	1
whimsical	1
recommend	1
happy	1
...	...

$)=c$

Bayes' Rule Applied to Documents and Classes

- For a document *d* and a class *c*

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

Naive Bayes Classifier (I)

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

MAP is "maximum a posteriori" = most likely class

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

Bayes Rule

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

Dropping the denominator

Naive Bayes Classifier (II)

"Likelihood"

"Prior"

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

$$= \underset{c \in C}{\operatorname{argmax}} P(x_1, x_2, \ldots, x_n \mid c) P(c)$$

Document d represented as features x1..xn

Naive Bayes Classifier (III)

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(x_1, x_2, \ldots, x_n \mid c) P(c)$$

$O(|X|^n \bullet |C|)$ parameters

How often does this class occur?

Could only be estimated if a very, very large number of training examples was available.

We can just count the relative frequencies in a corpus

Naive Bayes Independence Assumptions

$$P(x_1, x_2, \ldots, x_n \mid c)$$

- **Bag of Words assumption**: Assume position doesn't matter
- **Conditional Independence**: Assume the feature probabilities $P(x_i \mid c_j)$ are independent given the class *c.*

$$P(x_1, \ldots, x_n \mid c) = P(x_1 \mid c) \bullet P(x_2 \mid c) \bullet P(x_3 \mid c) \bullet \ldots \bullet P(x_n \mid c)$$

Multinomial Naive Bayes Classifier

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(x_1, x_2, \ldots, x_n \mid c) P(c)$$

$$c_{NB} = \underset{c \in C}{\operatorname{argmax}} P(c_j) \prod_{x \in X} P(x \mid c)$$

Applying Naive Bayes Classifiers to Text Classification

- positions $\leftarrow$ all word positions in test document

$$c_{NB} = \operatorname*{argmax}_{c_j \in C} P(c_j) \prod_{i \in positions} P(x_i \mid c_j)$$

Problems with multiplying lots of probs

- There's a problem with this:

$$c_{NB} = \underset{c_j \in C}{\operatorname{argmax}} P(c_j) \prod_{i \in positions} P(x_i \mid c_j)$$

- Multiplying lots of probabilities can result in floating-point underflow!

  .0006 * .0007 * .0009 * .01 * .5 * .000008….

- Idea: Use logs, because log(*ab*) = log(*a*) + log(*b*)
  - We'll sum logs of probabilities instead of multiplying probabilities!

Using log space

$$c_{NB} = \underset{c_j \in C}{\operatorname{argmax}} P(c_j) \prod_{i \in positions} P(x_i \mid c_j)$$

$$c_{\text{NB}} = \underset{c_j \in C}{\operatorname{argmax}} \left[\log P(c_j) + \sum_{i \in \text{positions}} \log P(x_i | c_j) \right]$$

- Notes:
  - Taking log doesn't change the ranking of classes!

    The class with highest probability also has highest log probability!
  - It's a linear model:

    Just a max of a sum of weights: a **linear** function of the inputs

    So naive bayes is a **linear classifier**

Learning the Multinomial Naive Bayes Model

- First attempt: maximum likelihood estimates
  - simply use the frequencies in the data

$$\hat{P}(c_j) = \frac{N_{c_j}}{N_{total}}$$

$$\hat{P}(w_i \mid c_j) = \frac{count(w_i, c_j)}{\sum_{w \in V} count(w, c_j)}$$

Parameter estimation

$$\hat{P}(w_i \mid c_j) = \frac{count(w_i, c_j)}{\sum_{w \in V} count(w, c_j)}$$

fraction of times word w_i appears among all words in documents of topic c_j

- Create mega-document for topic *j* by concatenating all docs in this topic
  - Use frequency of *w* in mega-document

Outline

- Applications
- Task
- **Naïve Bayes Classification**
  - **Smoothing**
  - **N-grams**
- Convolutional Neural Network
- Evaluation

Problem with Maximum Likelihood

- What if we have seen no training documents with the word ***fantastic*** and classified in the topic **positive** (***thumbs-up)***?

$$\hat{P}(\text{"fantastic"} | \text{positive}) = \frac{count(\text{"fantastic"}, \text{positive})}{\sum_{w \in V} count(w, \text{positive})} = 0$$

- Zero probabilities cannot be conditioned away, no matter the other evidence!

$$c_{MAP} = \text{argmax}_c \, \hat{P}(c) \prod_i \hat{P}(x_i | c)$$

Smoothing for Naïve Bayes

- Laplace (add-1) smoothing

$$\hat{P}(w_i \mid c) = \frac{count(w_i, c)}{\sum_{w \in V} \left(count(w, c)\right)}$$

$$\hat{P}(w_i \mid c) = \frac{count(w_i, c) + 1}{\sum_{w \in V} \left(count(w, c) + 1\right)}$$

$$= \frac{count(w_i, c) + 1}{\left(\sum_{w \in V} count(w, c)\right) + |V|}$$

Smoothing for Naïve Bayes

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

Language Modeling in Naïve Bayes

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

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

Language Modeling in Naïve Bayes

- Unigram

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

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

Language Modeling in Naïve Bayes

- Bigram

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

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

Language Modeling in Naïve Bayes

- Trigram

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

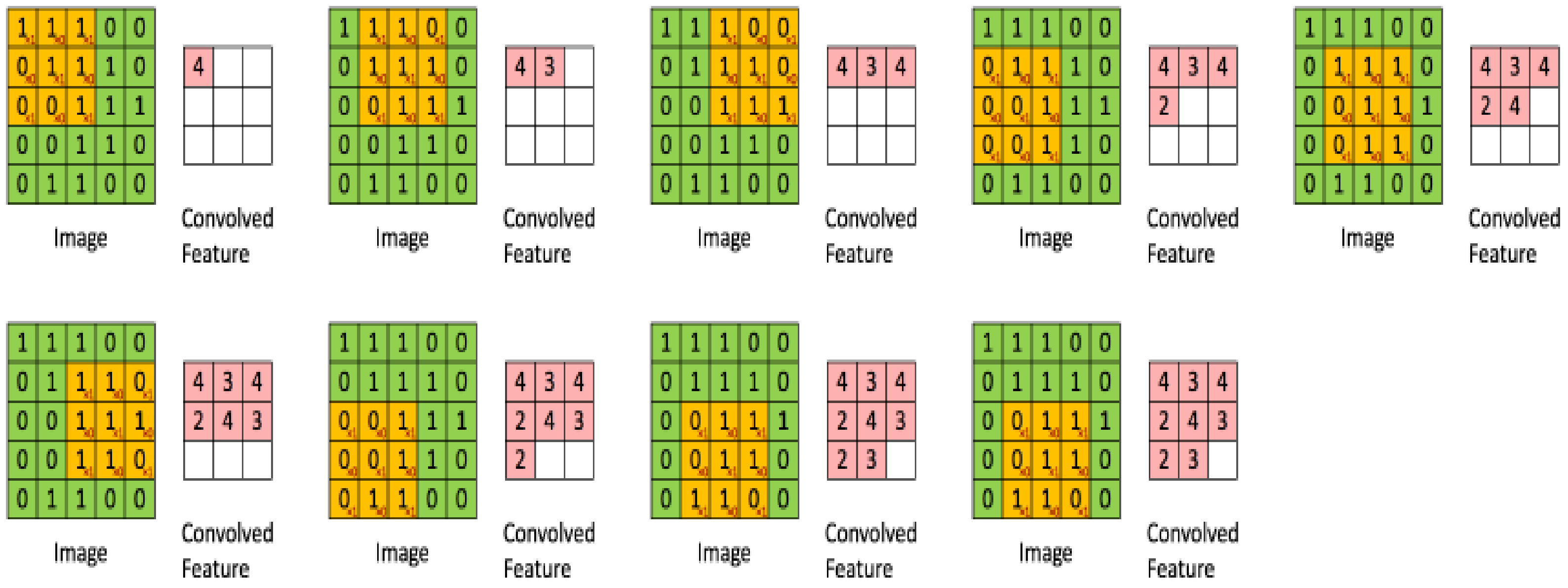
$$P(w_j \mid w_{j-2} w_{j-1}, c_i) = \frac{\#(w_{j-2} w_{j-1} w_j, c_i)}{\#(w_{j-2} w_{j-1}, c_i)}$$

Outline

- Applications
- Task
- Naïve Bayes Classification
- **Convolutional Neural Network**
- Evaluation

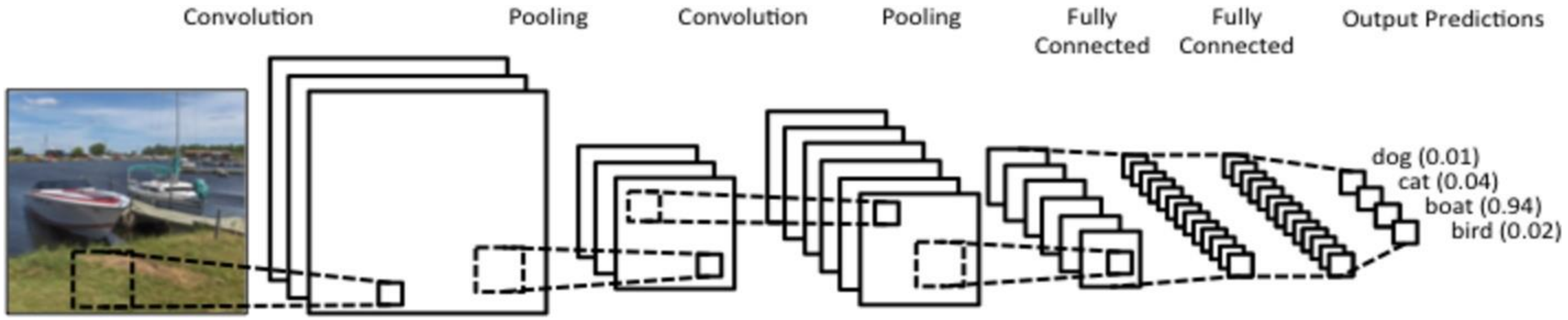
Convolutional Neural Networks (CNN)

- What is a convolution?
  - Intuition: sliding window function applied to a matrix Example: convolution with 3 × 3 filter
- Multiply values element-wise with original matrix, then sum. Slide over whole matrix.



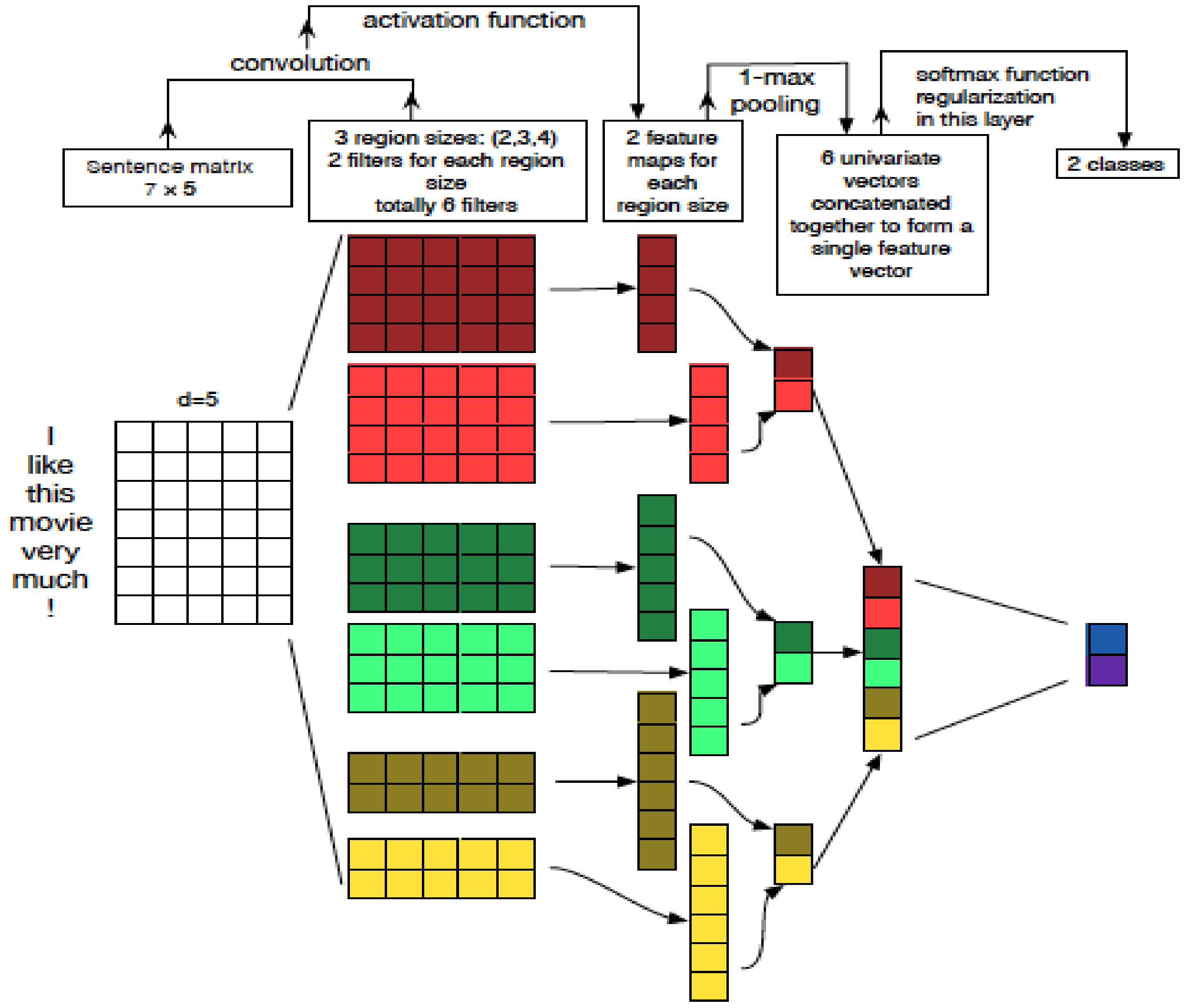
Convolutional Neural Networks (CNN)

- Each layer applies different filters and combines results
- Pooling (subsampling) layers
- During training, CNN learns values of filters
- First layer may learn to detect edges from raw pixels in first layer
- Use edges to detect simple shapes in second layer
- Use shapes to detect higher-level features, such as facial shapes in higher layers
- Last layer: classifier using high-level features



CNN for Text Classification

- Instead of image pixels, inputs typically are word embeddings.
- For a 10 word sentence using a 100-dimensional embedding we would have a 10 × 100 matrix as our input.
- That is our “image”
- Typically 1-dimensional convolutions are used; i.e., filters slide over rows of the matrix (words).

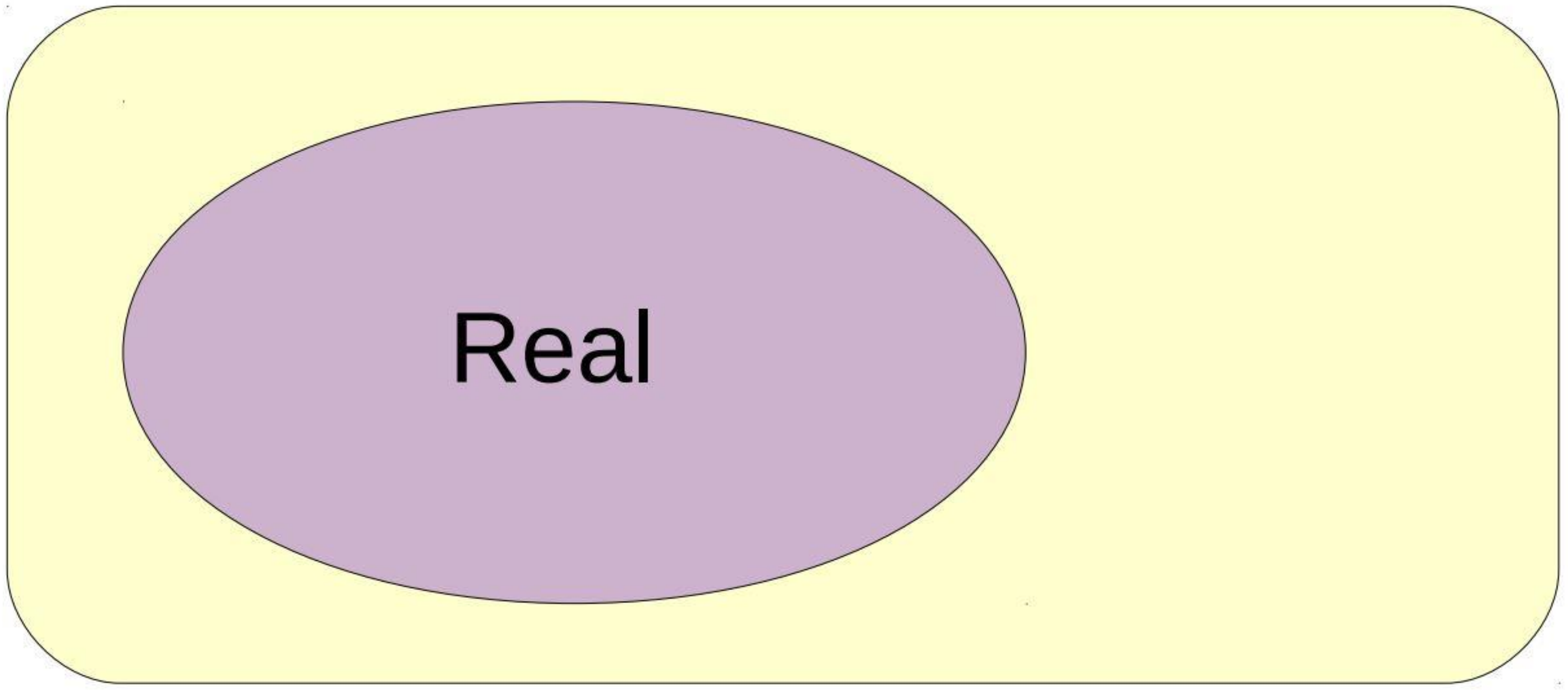


activation function
convolution
1-max pooling
softmax function regularization in this layer
Sentence matrix 7 x 5
3 region sizes: (2,3,4) 2 filters for each region size totally 6 filters
2 feature maps for each region size
6 univariate vectors concatenated together to form a single feature vector
2 classes
d=5
I
like
this
movie
very
much
!

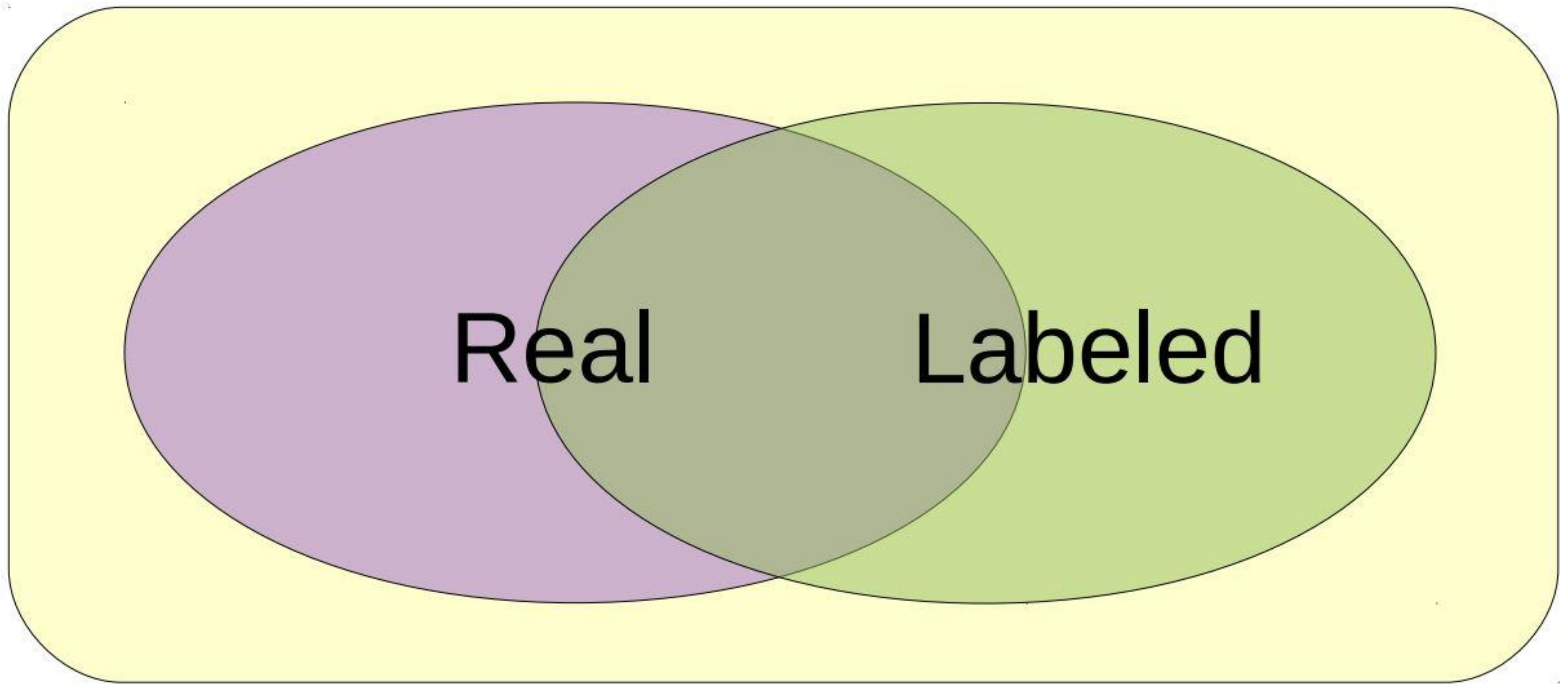
Outline

- Applications
- Task
- Naïve Bayes Classification
- Convolutional Neural Network
- **Evaluation**

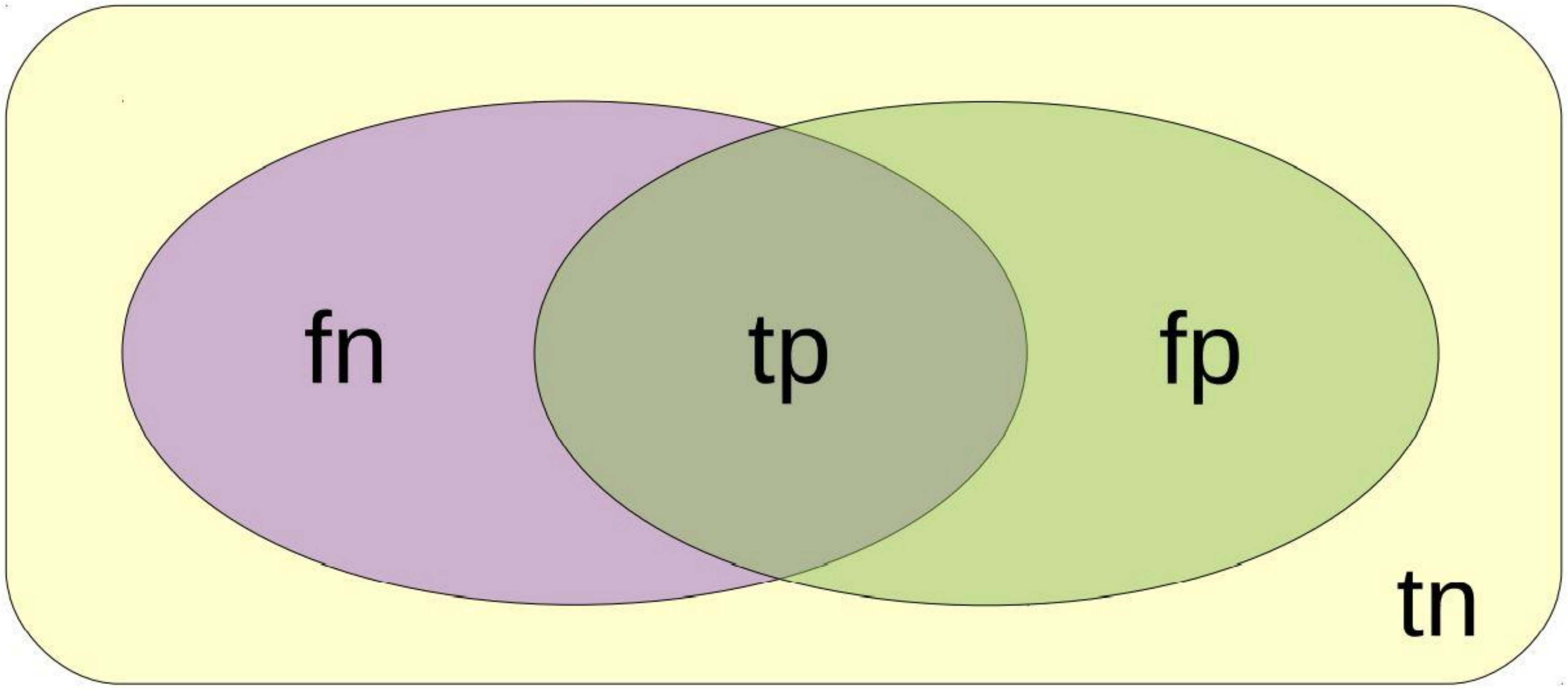
Precision and Recall



Precision and Recall



Precision and Recall



The 2-by-2 confusion matrix

		gold standard labels		
		gold positive	gold negative	
system output labels	system positive	**true positive**	**false positive**	**precision** $= \frac{\text{tp}}{\text{tp+fp}}$
	system negative	**false negative**	**true negative**	
		recall $= \frac{\text{tp}}{\text{tp+fn}}$		**accuracy** $= \frac{\text{tp+tn}}{\text{tp+fp+tn+fn}}$

Precision

- % of items the system detected (i.e., items the system labeled as positive) that are in fact positive (according to the human gold labels)

$$\textbf{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

Recall

- % of items actually present in the input that were correctly identified by the system.

$$\textbf{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

A combined measure: F

- F measure: a single number that combines P and R:

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

- We almost always use balanced F_1 (i.e., $\beta = 1$)

$$\mathrm{F}_1 = \frac{2PR}{P + R}$$

Confusion Matrix for 3-class classification

		gold labels			
		urgent	normal	spam	
system output	urgent	8	10	1	$\textbf{precision}_u = \frac{8}{8+10+1}$
	normal	5	60	50	$\textbf{precision}_n = \frac{60}{5+60+50}$
	spam	3	30	200	$\textbf{precision}_s = \frac{200}{3+30+200}$
		$\textbf{recall}_u = \frac{8}{8+5+3}$	$\textbf{recall}_n = \frac{60}{10+60+30}$	$\textbf{recall}_s = \frac{200}{1+50+200}$	

How to combine P/R from 3 classes?

- Macroaveraging:
  - compute the performance for each class, and then average over classes

- Microaveraging:
  - collect decisions for all classes into one confusion matrix
  - compute precision and recall from that table.

Macroaveraging and Microaveraging

Class 1: Urgent

	true urgent	true not
system urgent	8	11
system not	8	340

$$\text{precision} = \frac{8}{8+11} = .42$$

Class 2: Normal

	true normal	true not
system normal	60	55
system not	40	212

$$\text{precision} = \frac{60}{60+55} = .52$$

Class 3: Spam

	true spam	true not
system spam	200	33
system not	51	83

$$\text{precision} = \frac{200}{200+33} = .86$$

Pooled

	true yes	true no
system yes	268	99
system no	99	635

$$\text{microaverage precision} = \frac{268}{268+99} = \mathbf{.73}$$

$$\text{macroaverage precision} = \frac{.42+.52+.86}{3} = \mathbf{.60}$$

Further Reading

- Speech and Language Processing (3rd ed. draft)
  - Chapter 4