Introduction to

Information Retrieval

Text Classification

Standing queries

- The path from IR to text classification:
  - You have an information need to monitor, say:
    - Natural Language Processing
  - You want to rerun an appropriate query periodically to find new news items on this topic
  - You will be sent new documents that are found
    - I.e., it's not ranking but classification (relevant vs. not relevant)
- Such queries are called **standing queries**
  - Long used by "information professionals"
  - A modern mass instantiation is **Google Alerts**
- Standing queries are (hand-written) text classifiers

From: Google Alerts
Subject: **Google Alert - stanford -neuro-linguistic nlp OR "Natural Language Processing" OR parser OR tagger OR ner OR "named entity" OR segmenter OR classifier OR dependencies OR "core nlp" OR corenlp OR phrasal**
Date: May 7, 2012 8:54:53 PM PDT
To: Christopher Manning

Web

3 new results for **stanford -neuro-linguistic nlp OR "Natural Language Processing" OR parser OR tagger OR ner OR "named entity" OR segmenter OR classifier OR dependencies OR "core nlp" OR corenlp OR phrasal**

Twitter / **Stanford NLP** Group: @Robertoross If you only n ...
@Robertoross If you only need tokenization, java -mx2m edu.**stanford.nlp**. process.PTBTokenizer file.txt runs in 2MB on a whole file for me.... 9:41 PM Apr 28th ...
twitter.com/stanfordnlp/status/196459102770171905

[Java] LexicalizedParser lp = LexicalizedParser.loadModel("edu ...
loadModel("edu/**stanford**/**nlp**/models/lexparser/englishPCFG.ser.gz");. String[] sent = { "This", "is", "an", "easy", "sentence", "." };. Tree parse = lp.apply(Arrays.
pastebin.com/az14R9nd

More Problems with Statistical **NLP** || kuro5hin.org
Tags: nlp, ai, coursera, **stanford**, **nlp**-class, cky, nltk, reinventing the wheel, ... Programming Assignment 6 for **Stanford's nlp**-class is to implement a CKY parser .
www.kuro5hin.org/story/2012/5/5/11011/68221

Tip: Use quotes ("like this") around a set of words in your query to match them exactly. Learn more.

Delete this alert.
Create another alert.
Manage your alerts.

Another text classification task: Email spam filtering

```
From: ''' <takworlld@hotmail.com>
Subject: real estate is the only way... gem oalvgkay
Anyone can buy real estate with no money down
Stop paying rent TODAY !
There is no need to spend hundreds or even thousands for
similar courses
I am 22 years old and I have already purchased 6 properties
using the
methods outlined in this truly INCREDIBLE ebook.
Change your life NOW !
=============================================
Click Below to order:
http://www.wholesaledaily.com/sales/nmd.htm
=============================================
```

Most Common Methods:

1.Collect and preprocess email data

2. Split data into training and testing sets

3. Train a machine learning model, such as a Naive Bayes classifier or a Support Vector Machine (SVM).

4. Test the model

5. Evaluate the precision, recall, and F1 score of the model to determine its effectiveness.

More examples of text classification

- Language identification (classes: English vs. French etc.)
- Personal email sorting
- The automatic detection of sexually explicit content (sexually explicit vs. not)
- Sentiment detection: is a movie or product review positive or negative (positive vs. negative)
- Profanity & abuse detection: is used for keeping communications safe from insults and for detecting bullying on social networks and online communities.
- E-commerce support ticket classification (missing item, shipping problem, product availability, etc.)

Classification methods: 1. Manual

- Manual classification was used by Yahoo in the beginning of the web. Also: ODP, PubMed
- Very accurate if job is done by experts
- Consistent when the problem size and team is small
- Scaling manual classification is difficult and expensive.
- → We need automatic methods for classification.

Classification methods: 2. Rule-based

- Our Google Alerts example was rule-based classification.
- There are IDE-type development environments for writing very complex rules efficiently. (e.g., Verity)
- Often: Boolean combinations (as in Google Alerts)
- Accuracy is very high if a rule has been carefully refined over time by a subject expert.
- Building and maintaining rule-based classification systems is cumbersome and expensive.

Classification methods: 3. Statistical

- The set of rules is learned automatically from training data.
- Statistical text classification, particularly when the learning method is statistical in nature.
- In statistical text classification, a crucial component is the training documents, for each class.
- Manual classification is not eliminated but transformed.

Classification methods: 3. Statistical (Cont.)

- This was our definition of the classification problem – text classification as a learning problem
- (i) Supervised learning of a classification function Υ and

  (ii) its application to classifying new documents
- We will look at a couple of methods for doing this: Naive Bayes, kNN, SVMs, Decision Trees
- No free lunch: requires hand-classified training data
- But this manual classification can be done by non-experts.

Formal definition of TC: Training

Given:

- A document space X
  - Documents are represented in this space – typically some type of high-dimensional space.
- A fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$
  - The classes are human-defined for the needs of an application (e.g., relevant vs. nonrelevant).
- A training set D of labeled documents with each labeled document $\langle d, c \rangle \in X \times C$

Using a learning method or learning algorithm, we then wish to learn a classifier Υ that maps documents to classes:

$$\Upsilon : X \rightarrow C$$

Formal definition of TC: Application/Testing

Given: a description $d \in X$ of a document

Determine: $\Upsilon(d) \in C$,

that is, the class that is most appropriate for *d*

For example:
<d, c> = <Beijing joins the World Trade Organization, China>

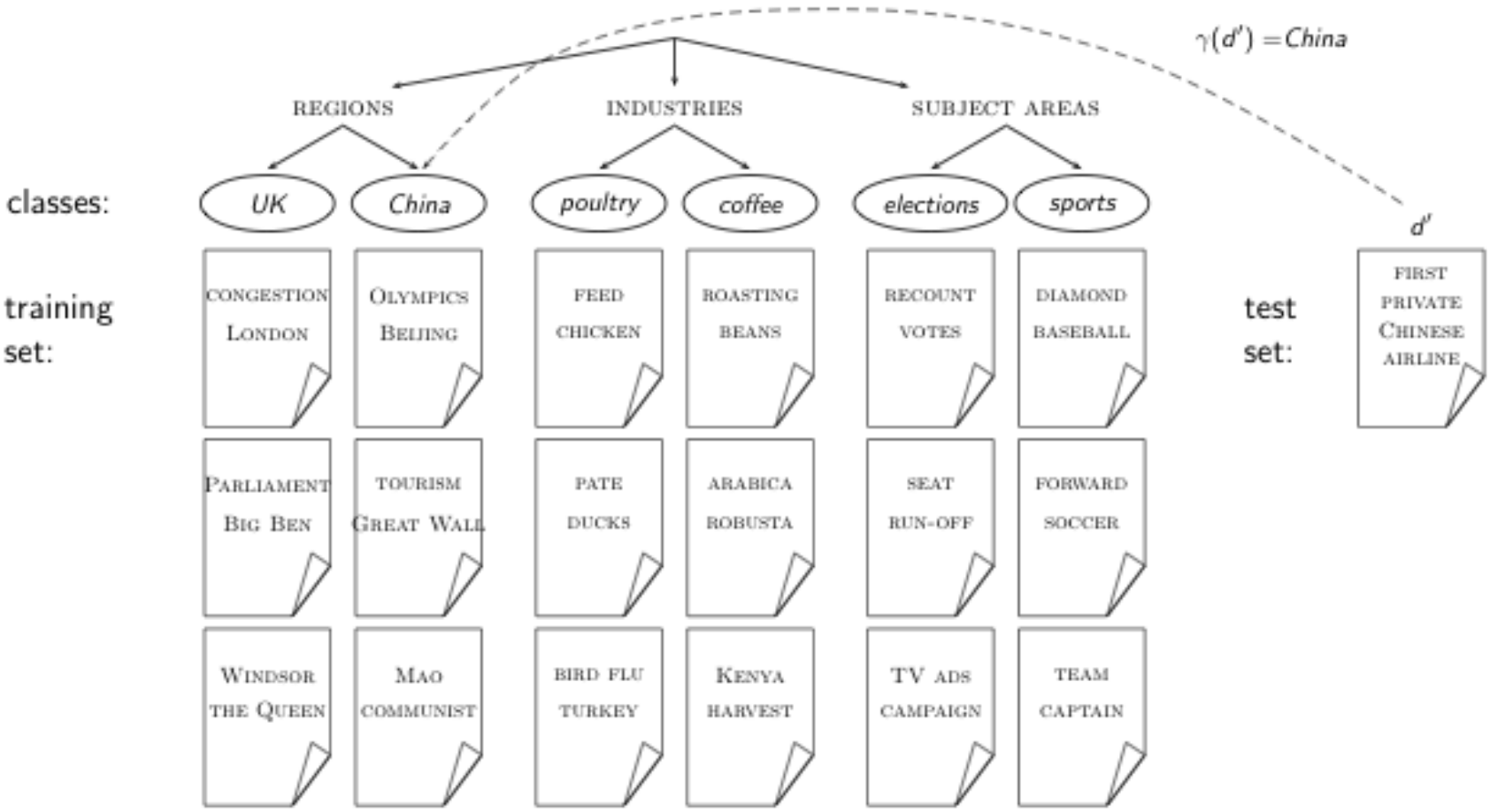
- for the one-sentence document *Beijing joins the World Trade Organization* and the class (or label) *China*.

Example: The Reuters collection

symbol	statistic	value
N	documents	800,000
L	avg. # word tokens per document	200
M	word types	400,000
	avg. # bytes per word token (incl. spaces/punct.)	6
	avg. # bytes per word token (without spaces/punct.)	4.5
	avg. # bytes per word type	7.5
	non-positional postings	100,000,000

type of class	number	examples
region	366	UK, China
industry	870	poultry, coffee
subject area	126	elections, sports

Example: Topic classification on Reuters



Evaluating classification

- Evaluation must be done on test data that are independent of the training data.
- No information of the test data should be used to train the classifier.
- It's easy to get good performance on a test set that was available to the learner during training.
- Instead, set aside a development set for testing while training the model and select the parameter values that give best results on the development set.
- Measures: Precision, recall, F_1, classification accuracy

Precision P, recall R and accuracy A

	in the class	not in the class
predicted to be in the class	true positives (TP)	false positives (FP)
predicted to not be in the class	false negatives (FN)	true negatives (TN)

$$P = TP / (TP + FP)$$

$$R = TP / (TP + FN)$$

$$A = (TP + TN) / (TP + TN + FP + FN)$$

A combined measure: *F*

- F_1 allows us to trade off precision against recall.
- $$F_1 = \frac{1}{\frac{1}{2}\frac{1}{P} + \frac{1}{2}\frac{1}{R}} = \frac{2PR}{P+R}$$
- This is the harmonic mean of *P* and *R*: $\frac{1}{F} = \frac{1}{2}\left(\frac{1}{P} + \frac{1}{R}\right)$

Averaging: Micro vs. Macro

- We now have an evaluation measure (F_1) for one class.
- But we also want a single number that measures the aggregate performance over all classes in the collection.
- Macroaveraging
  - Compute F_1 for each of the C classes
  - Average these C numbers
- Microaveraging
  - Compute TP, FP, FN for each of the C classes
  - Sum these C numbers (e.g., all TP to get aggregate TP)
  - Compute F_1 for aggregate TP, FP, FN

Issues in the classification of text documents

Many commercial applications

- "There is no question concerning the commercial value of being able to classify documents automatically by content. There are many potential applications of such a capability for corporate Intranets, government departments, and Internet publishers."

Often greater performance gains from exploiting domain-specific text features than from changing from one machine learning method to another.

- "Understanding the data is one of the keys to successful categorization, yet this is an area in which most categorization tool vendors are extremely weak. Many of the 'one size fits all' tools on the market have not been tested on a wide range of content types."

Choosing what kind of classifier to use

When building a text classifier, first question: how much training data is there currently available?

Practical challenge: creating or obtaining enough training data

Hundreds or thousands of examples from each class are required to produce a high performance classifier and many real world contexts involve large sets of categories.

- None?
- Very little?
- Quite a lot?
- A huge amount, growing every day?

If you have no labeled training data

Use hand-written rules

Example

IF (wheat OR grain) AND NOT (whole OR bread) THEN
c = grain

- Complex rules, beyond Boolean expressions.
- High accuracy: 90%+ precision, 80%+ recall.
- Requires substantial initial effort.
- Ongoing maintenance for evolving content.

If you have fairly little data and you are going to train a supervised classifier

Work out how to get more labeled data as quickly as you can.

- Best way: insert yourself into a process where humans will be willing to label data for you as part of their natural tasks.

Example

Often humans will sort or route email for their own purposes, and these actions give information about classes.

If you have labeled data

Reasonable amount of labeled data

Use everything that we have presented about text classification.
Preferably hybrid approach (overlay Boolean classifier)

Huge amount of labeled data

Choice of classifier probably has little effect on your results.
Choose classifier based on the scalability of training or runtime efficiency.
Rule of thumb: each doubling of the training data size produces a linear increase in classifier performance, but with very large amounts of data, the improvement becomes sub-linear.

Large and difficult category taxonomies

- If small number of well-separated categories, then many classification algorithms are likely to work well.
- But often: very large number of very similar categories.

Example

Web directories (e.g. the Yahoo! Directory consists of over 200,000 categories or the Open Directory Project), library classification schemes (Dewey Decimal or Library of Congress), the classification schemes used in legal or medical applications.

Accurate classification over large sets of closely related classes is **inherently difficult.**