Step 0: is it easy?

	INPUT	OUTPUT	LEARNING	ASSESSMENT
1	A TWEET	{GOOD, BAD}	Y	· USE COLLECTED DATA · COMPARE W/ SURVEYS · MEASURE MONEY
2	CORPUS	SET OF $\overbrace{\text{SETS OF WORDS}}^{\text{TOPIC}}$ WORDS	N	"MANUAL" READING (POSSIBLY SAMPLE)

1 ▸ **Q:** for "sentiment on brands"?

2 ▸ **Q:** for "topics in letters"?

3 ▸ **Q:** for "relevance of citations"?

Natural language and ambiguity

- Text is (usually) natural language
- Natural means "as humans naturally express" $\implies$ ambiguity!

Expect "raw results" to be worse than in normal ML!

Subsection 1

Sentiment analysis (and text categorization)

Problem formalization

- Input: a piece of text (*document*)
- Output?
  - a numeric value in $[-1, 1]$ (positivity)
  - a categorical value in $\{\text{Pos}, \text{Neg}\}$
  - a categorical value in $\{\text{Pos}, \text{Neutral}, \text{Neg}\}$
- **Q:** learning data?

Regression, multiclass classification, or binary classification (possibly with confidence).

We can use the techniques we already know (e.g., RF)!

Which are the features?

```
Good coffee.  Great for families.  Always had good
service.  We go early so pretty empty.  Flexible with
menus.  Wish they would remove service charge.
```

Need to transform a document into an numerical vector!

Text to features

- Not only for sentiment analysis
- Many options
- Options can be combined

Bag of words

- One dimension (feature, dependent variable) for each word
- Value of $x_{i,j}$ is the number of occurrences of j-th word in the i-th document.

$$
\begin{aligned}
t_1 &= \texttt{the cat is on the table} \\
x_{1,\text{the}} &= 2 \\
x_{1,\text{cat}} &= 1 \\
x_{1,\text{is}} &= 1 \\
x_{1,\text{mouse}} &= 0 \\
\ldots
\end{aligned}
$$

Bag of which words?

- One dimension (feature, dependent variable) for each word

Which words? How big is x_i? $p = ?$

- common solution: the most k frequent words in the corpus

"Interesting" words not frequent enough in the corpus may be lost

Stop words

- Some words may be very frequent, but useless for specific task (e.g., sentiment analysis)
  - `a`, `an`, `the`, `are`, ... (**stop words**)
- Just remove them!

Stop words are language dependent!

Stemming

- There are variants for many words:
  - `drink`, `drinks`, `drinking`
  - `happy`, `happier`
- Even more in other languages:
  - `mangio`, `mangia`, `mangi`, `mangiai`, ...
- **Stemming**: reduce word to its word stem (the morphological root)
  - `drinking` → `drink`
  - `argued` → `argu`

Stemming are language dependent!

A typical workflow

- Preprocessing ($d \to d'$)
  1. remove punctuation
  2. to lowercase
  3. remove stop words
  4. stemming
- Learning
  1. preprocess each d in corpus
  2. find most frequent k words in preprocessed corpus
  3. compute $\mathbf{X}$
  4. learn a classifier
- Predicting
  1. preprocess input d
  2. predict based on preprocessed d

Limitations and caveat: punctuation

- remove punctuation

It has been show that often punctuation matter (e.g., Twitter sentiment analysis):

- `I just saw Alice.`
- `I just saw Alice!!!!!`
- `I just saw Alice!!!  :-))))`

Case

- to lowercase

Case may be relevant in some case (e.g., music genre preferences classification):

- `I like the Take That and I hate The Who.`
- `Who likes to take that song of Hate?  Me!`

Goal, context, hypothesis

Twitter profiling: predict age and gender of user from his/her tweets. (**Q:** what kind of problem/problems?)

- people of different ages differently use case
- people of different age/gender differently use punctuation

A step in the workflow corresponds to an (implicit) hypothesis:

- remove stop words $\rightarrow$ stop words frequencies is not useful for predicting X

Words that matter

Word count may be too coarse to capture desired information:

- documents with very different lengths
- irrelevant terms with general high frequencies

Use frequency or more complex variants

tf-idf

t → term → word

$$x_{i,j} = x_{d,t} = \mathrm{tf}(t, d)\mathrm{idf}(t, D)$$

- $\mathrm{tf}(t, d) = f_{t,d}$, term frequency
  - the more important the term t in document d, the larger
- $\mathrm{idf}(t, D) = \log \frac{|D|}{|\{d \in D : f_{t,d} > 0\}|}$, inverse document frequency
  - the more common t in the corpus, the lower

RARENESS

Bag of words and ordering

Sentiment analysis of restaurant reviews:

$t_1 =$ `The beer was good and the pub was not too noisy.`

$t_2 =$ `The beer was not good and the pub was too noisy.`

$x_1 = x_2$

- fundamental problem: ordering is lost
- even more fundamental: natural language can be hard to algorithmically understand (irony, sarcasm, ...)

Solutions:

- ngrams
- text parsing (NLP)

Aside: collecting data for text classification

Example: irony detection

- Pavel Savov and Radoslaw Nielek. "Ridiculously Expensive Watches and Surprisingly Many Reviewers: A Study of Irony". In: *Web Intelligence (WI), 2016 IEEE/WIC/ACM International Conference on*. IEEE. 2016, pp. 725–729

ngrams

Instead of counting words, count short (up to n) sequences of words:

- $x_{1,\text{dog,eat,cat}}$ instead of $x_{1,\text{dog}}$
- size of data (p) grows dramatically (and is sparser)
- useful in general for manipulating sequences

Generality of a sentiment classifier

How many sentiment classifier should exist? Words that matter in sentiment should be predefined

- predefined list of opinion words (positive, negative), i.e., features are those words
- but context often matter
  - `predictable` is good for a car and bad for a movie
  - features for sentiment analysis in Twitter are likely different than features from sentiment analysis of a early '900 writer's corrispondence

There are many pre-trained tool, often with more complex outcome than positive/negative.

Out-of-the-box sentiment analysis

Should I use a pre-trained tool or build my own?

It depends:

- is sentiment analysis just a piece of a more complex ML system?
- which is my budget?
- is learning data easily available?

Parsing: POS tagging

Assign a role to part of speech (**POS**):

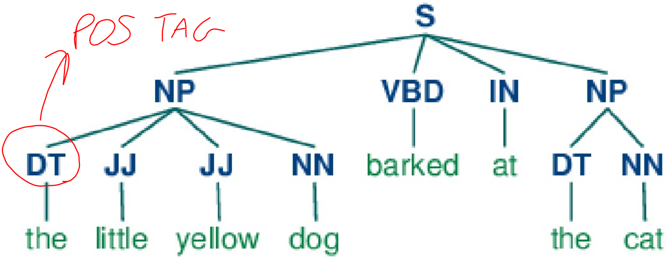

Image from `http://www.nltk.org/`

Lab: Text categorization: sport vs. politics (4h)

Build a binary classifier for tweets: sport vs. politics

1. decide input, output
2. decide solution assessment
3. decide (if any) how to obtain learning data
4. decide workflow and ML technique

ON PAPER!