Text Preprocessing II

From textual information to numerical vector

Syntactic Analysis

Syntactic Analysis

- Part-of-Speech Tagging
- Word Sense Disambiguation
- Parsing

Part-of-Speech Tagging

- If no further linguistic analysis is necessary, one might proceed directly to feature generation, in which the features will be obtained from the tokens (E.g., *linguistic* and *analysis* from this sentence).
- However, if the goal is more specific, say recognizing names of people, places, and organizations, it is usually desirable to perform additional linguistic analyses of the text and extract more sophisticated features.
    - E.g., San Francisco
- In English, some analyses may use as few as six or seven categories and others nearly one hundred.
- Most English grammars would have a minimum noun, verb, adjective, adverb, preposition, and conjunction.

Part-of-Speech Tagging

- POS can be used for feature reduction, e.g., use only verb, adjective, and adverb for sentiment classification.

- Distribution of POS can be used for author, gender, and document genre (formal vs. informal) classification

Part-of-Speech Tagging

- A set of 36 categories is used in the **PennTree Bank** (https://catalog.ldc.upenn.edu/docs/LDC95T7/cl93.html) constructed from the Wall Street Journal corpus (see next page)
  - A **tree bank** is a parsed text corpus that annotates sentence structure, such as POS and phrases.
- Almost all POS taggers have been trained on the Wall Street Journal corpus available from LDC (linguistic Data Consortium, www.ldc.upenn.edu)
  - E.g., *I love you* -> *I* (**personal pronoun**) *love* (**verb**, not noun)
- The Brill tagger is in the public domain and is in wide use.
  - Online Brill tagger: https://nlpweb01.nors.ku.dk/online/pos_tagger/uk/index.html
- The Stanford Parser: a statistical parser
  - An implementation in Java: https://nlp.stanford.edu/software/lex-parser.shtml

Penn Tree Bank POS set

Tag	Description
CC	Coordinating conjunction
CD	Cardinal number
DT	Determiner
EX	Existential there
FW	Foreign word
IN	Preposition or subordinating conjunction
JJ	Adjective
JJR	Adjective, comparative
JJS	Adjective, superlative
LS	List item marker
MD	Modal
NN	Noun, singular or mass
NNS	Noun, plural
POS	Possessive ending
UH	Interjection
VB	Verb, base form
VBD	Verb, past tense
VBG	Verb, gerund or present participle
VBN	Verb, past participle
VBP	Verb, non-3rd person singular present
WDT	Wh-determiner

All the POS categories:
http://cs.nyu.edu/grishman/jet/guide/PennPOS.html

Part-of-Speech Tagging

- The Stanford Parser: online parser
  - http://nlp.stanford.edu:8080/parser/

Note:
PRP$: Possessive pronoun
RB: Adverb
VBZ: Verb, 3rd person singular present

Part-of-Speech Tagging

- The Stanford Parser: online parser
  - http://nlp.stanford.edu:8080/parser/

Note:
PRP$: Possessive pronoun
RB: Adverb
VBZ: Verb, 3rd person singular present

Word Sense Disambiguation

- Let's disambiguate "bank" in this sentence:
  - The bank can guarantee deposits will eventually cover future tuition costs because it invests in adjustable-rate mortgage securities.
- Given the following two WordNet senses:

bank^1	Gloss:	a financial institution that accepts deposits and channels the money into lending activities
	Examples:	"he cashed a check at the bank", "that bank holds the mortgage on my home"
bank^2	Gloss:	sloping land (especially the slope beside a body of water)
	Examples:	"they pulled the canoe up on the bank", "he sat on the bank of the river and watched the currents"

WSD: The Simplified Lesk Algorithm

- Choose sense with **most word overlap** between gloss and context (not counting stop words)
  - The **bank** can guarantee deposits will eventually cover future tuition costs because it invests in adjustable-rate mortgage securities.

bank[1]	Gloss:	a financial institution that accepts deposits and channels the money into lending activities
	Examples:	"he cashed a check at the bank", "that bank holds the mortgage on my home"
bank[2]	Gloss:	sloping land (especially the slope beside a body of water)
	Examples:	"they pulled the canoe up on the bank", "he sat on the bank of the river and watched the currents"

WSD

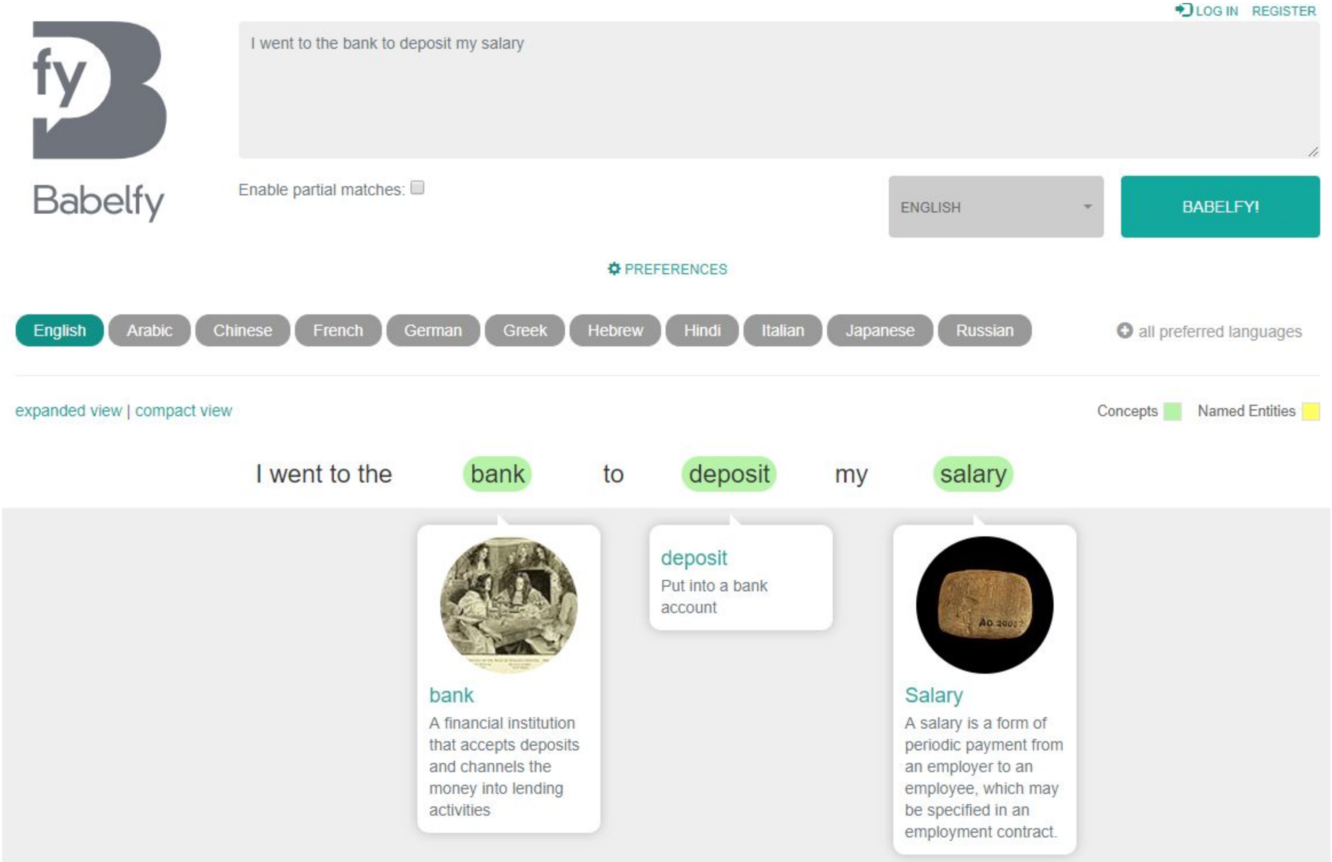
- Performs the classic Lesk algorithm for Word Sense Disambiguation (WSD)
  - Given an ambiguous word and the context in which the word occurs, Lesk returns a Synset with the highest number of overlapping words between the context sentence and different definitions from each Synset.
  - http://www.nltk.org/howto/wsd.html

```
>>> from nltk.wsd import lesk
>>> sent = ['I', 'went', 'to', 'the', 'bank', 'to', 'deposit', 'money', '.']

>>> print(lesk(sent, 'bank', 'n'))
Synset('savings_bank.n.02')

>>> print(lesk(sent, 'bank'))
Synset('savings_bank.n.02')
```

```
>>> from nltk.corpus import wordnet as wn
>>> for ss in wn.synsets('bank'):
...     print(ss, ss.definition())
...
Synset('bank.n.01') sloping land (especially the slope beside a body of water)
Synset('depository_financial_institution.n.01') a financial institution that accepts deposits and channels the money into lending activities
Synset('bank.n.03') a long ridge or pile
Synset('bank.n.04') an arrangement of similar objects in a row or in tiers
Synset('bank.n.05') a supply or stock held in reserve for future use (especially in emergencies)
Synset('bank.n.06') the funds held by a gambling house or the dealer in some gambling games
Synset('bank.n.07') a slope in the turn of a road or track; the outside is higher than the inside in order to reduce the effects of centrifugal force
Synset('savings_bank.n.02') a container (usually with a slot in the top) for keeping money at home
Synset('bank.n.09') a building in which the business of banking transacted
Synset('bank.n.10') a flight maneuver; aircraft tips laterally about its longitudinal axis (especially in turning)
Synset('bank.v.01') tip laterally
Synset('bank.v.02') enclose with a bank
Synset('bank.v.03') do business with a bank or keep an account at a bank
Synset('bank.v.04') act as the banker in a game or in gambling
Synset('bank.v.05') be in the banking business
Synset('deposit.v.02') put into a bank account
Synset('bank.v.07') cover with ashes so to control the rate of burning
Synset('trust.v.01') have confidence or faith in
```

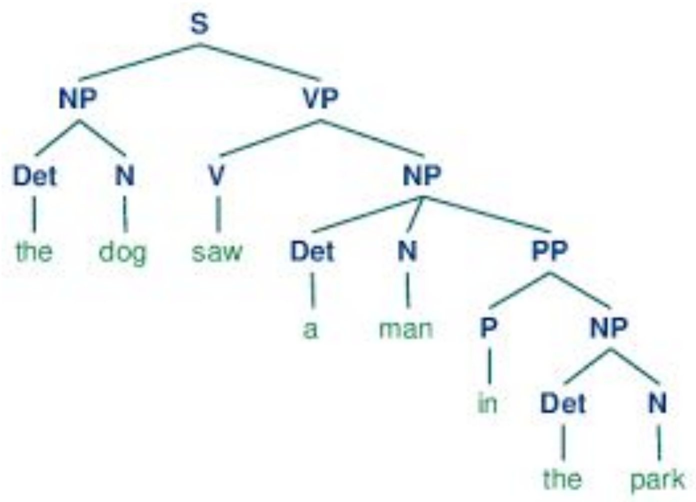
WSD

- Use Babelfy for Word Sense Disambiguation (WSD)
  - Considered as a state-of-the-art system based on BabelNet Multilingual Semantic network for multilingual Word Sense Disambiguation and Entity Linking.
  - http://babelfy.org/



Parsing

- Is the step of producing **a full parse of a sentence**.
- Each word in a sentence is connected to a single structure, usually a tree.
- Considerable research has been done on constructing parasers from a statistical analysis of tree banks of sentences parsed by hand.
- The reason for considering such a comparatively expensive process is that it provides **detailed syntactic relationships information** that phrase identification cannot provide.



Parsing

- Consider a sentence such as “Johnson was replaced at XYZ Corp. by Smith” for which a simple parse tree is shown in the below.

- **Phrase structure tree**

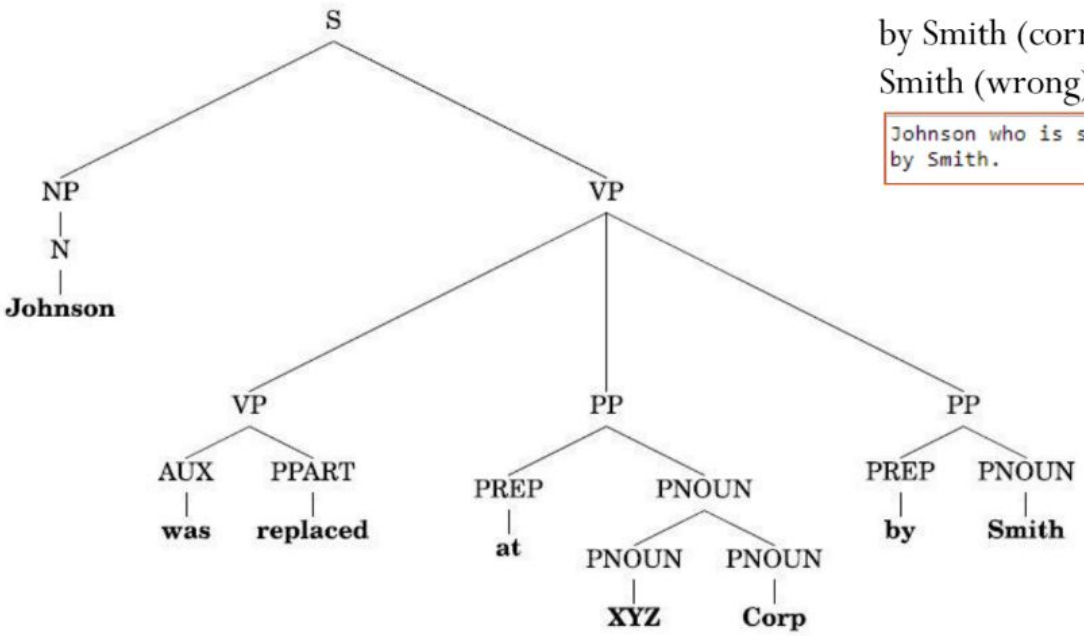


Fig. 2.9 Simple parse tree

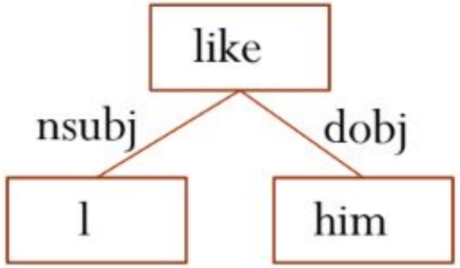
For instance, by looking at the Parse Tree, machine can infer that Johnson was replaced by Smith (correct); Steve was replaced by Smith (wrong).

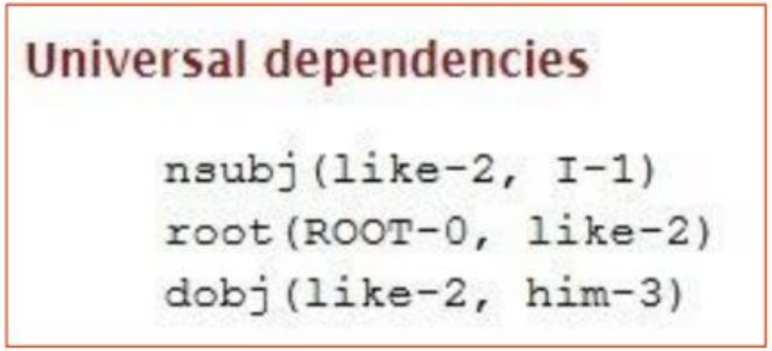
```
Johnson who is son of Steve was replaced at XYZ Corp.
by Smith.
```

```
(ROOT
  (S
    (NP
      (NP (NNP Johnson))
      (SBAR
        (WHNP (WP who))
        (S
          (VP (VBZ is)
            (NP
              (NP (NN son))
              (PP (IN of)
                (NP (NNP Steve))))))))
    (VP (VBD was)
      (VP (VBN replaced)
        (PP (IN at)
          (NP (NNP XYZ) (NNP Corp.)))
        (PP (IN by)
          (NP (NNP Smith)))))
    (. .)))
```

Parsing

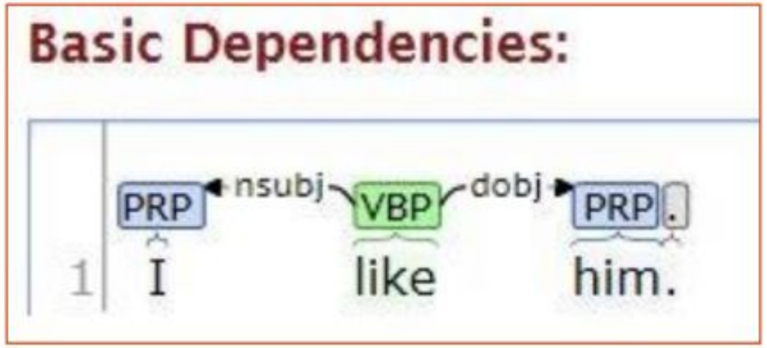
- Universal dependencies (i.e. grammatical relations; evolved out of Stanford Dependencies) from Stanford Parser: "I like him".




Universal dependencies

```
nsubj(like-2, I-1)
root(ROOT-0, like-2)
dobj(like-2, him-3)
```

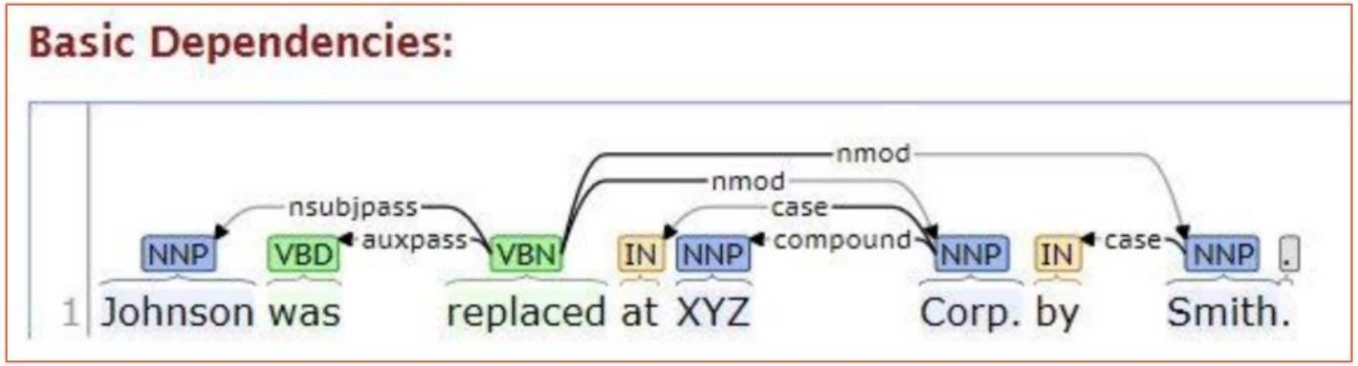
Output from Stanford Parser



Output from Stanford CoreNLP

Parsing

- Universal dependencies (i.e. grammatical relations; evolved out of Stanford Dependencies) from Stanford CoreNLP: “Johnson was replaced at XYZ Corp. by Smith”.



Universal dependencies

```
nsubjpass(replaced-3, Johnson-1)
auxpass(replaced-3, was-2)
root(ROOT-0, replaced-3)
case(Corp.-6, at-4)
compound(Corp.-6, XYZ-5)
nmod(replaced-3, Corp.-6)
case(Smith-8, by-7)
nmod(replaced-3, Smith-8)
```

Bag-of-Words: counting is everything

Vector Representation for Documents

- Without any deep analysis of the linguistic content of the documents, we can describe each document by features that represent the most frequent tokens.
- Each row is a document, and each column represents a feature.
- Thus, a cell in the csv/excel file is a measurement of a feature (corresponding to the column) for a document (corresponding to a row).

Bag-of-Words

- Steps
  - Build vocab i.e., set of all the words in the corpus
  - Count the occurrence of words in each document

The cat and the dog play
The cat is on the mat

corpus

and, the, cat, dog, play, on, mat, is

vocab.

1	2	1	1	1	0	0
1	2	0	0	1	1	1

countVec

Document Features

- How to define document features (i.e., entry value in the matrix)
    - Presence (0 or 1)
    - Frequencies (0,1,2,3)
    - Thresholding frequencies - three values
        - 0 (do not exit), 1 (occurred once), and 2 (occurred 2 or more times)

Term Frequency-Inverse Document Frequency

- Tf-id(w): tf(w) * idf(w), where $idf(w) = log(1 + \frac{N}{df(w)})$
    - The tf-idf weight assigned to word w is the **term frequency** (i.e., the word count) modified by a scale factor for the importance of the word.
    - The scale factor is called the **inverse document frequency**, which checks the number of documents containing word w (i.e., *df(w)*) and reverses the scaling.
    - The N is the number of documents.

Term Frequency-Inverse Document Frequency

- Intuitive logic:
    - Capture the importances of a word to document in a corpus
    - Importance of words is proportionally to the number of times a word appears
    - Importance of words is inversely proportionally to the document containing the word
    - Thus, when a word appears in many documents, it is considered unimportant and the scale is lowered, perhaps near zero, e.g., “the”, “I”, “on”, “document”, etc.

Term Frequency-Inverse Document Frequency

- When prepare the feature matrix, most of the entries will be zero.
- Most documents contain a small subset of the vocab's words
- Rather than storing all the zeros, it may be better to represent the matrix as a set of sparse vectors, where a row is represented by a list of paris, one element of the pair being a column number and the other element being the corresponding nonzero feature value.

0	15	0	3
12	0	0	0
8	0	5	2

(2,15) (4,3)
(1,12)
(1,8) (3,5) (4,2)

Multiword Features

- A variety of measures can be used for this purpose.
    - E.g., frequent n-grams, such as “text mining”, “hip hop”
- As another method, an Association Measure AM for the multiword T, is used for evaluation multiword features, where size(T) is the number of words in phrase T and freq(T) is the number of times phrases T occurs in the document collection.

$$AM(T) = \frac{size(T)log_{10}(freq(T))freq(T)}{\sum_{word_i \in T} freq(word_i)}$$

- Generally, multiword features are not found too frequently in a document collection, but when they do occur they are often high predictive.

Bag-of-Words

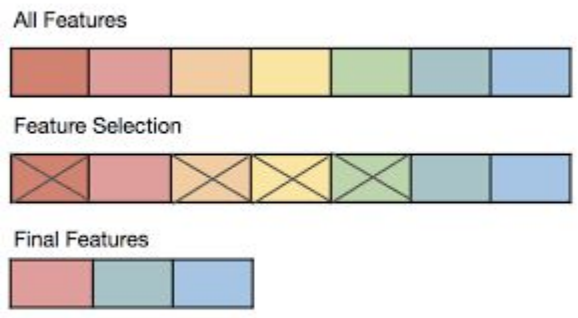
- Pros
  - Simple
  - Surprisingly effective
  - Fast

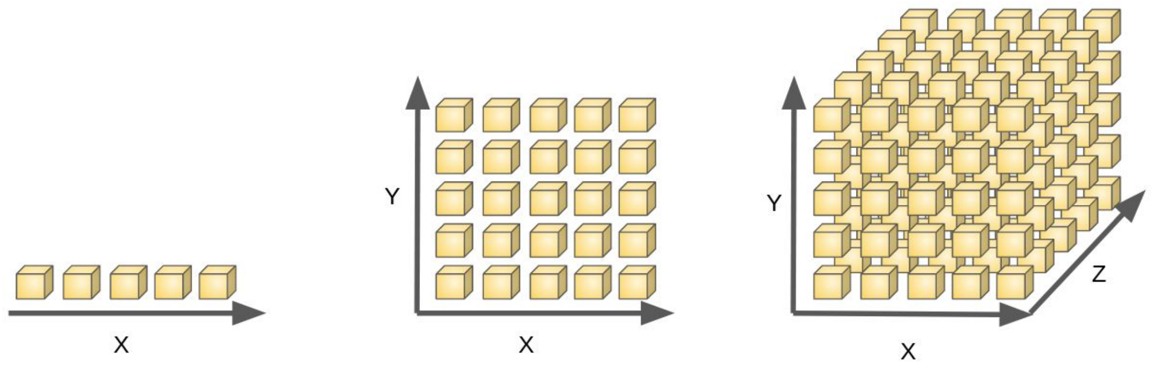
- Cons
  - Order of words does not matter
  - Cannot capture syntactic/semantic information
  - High dimensionality

Dictionary Reduction

Dictionary Reduction

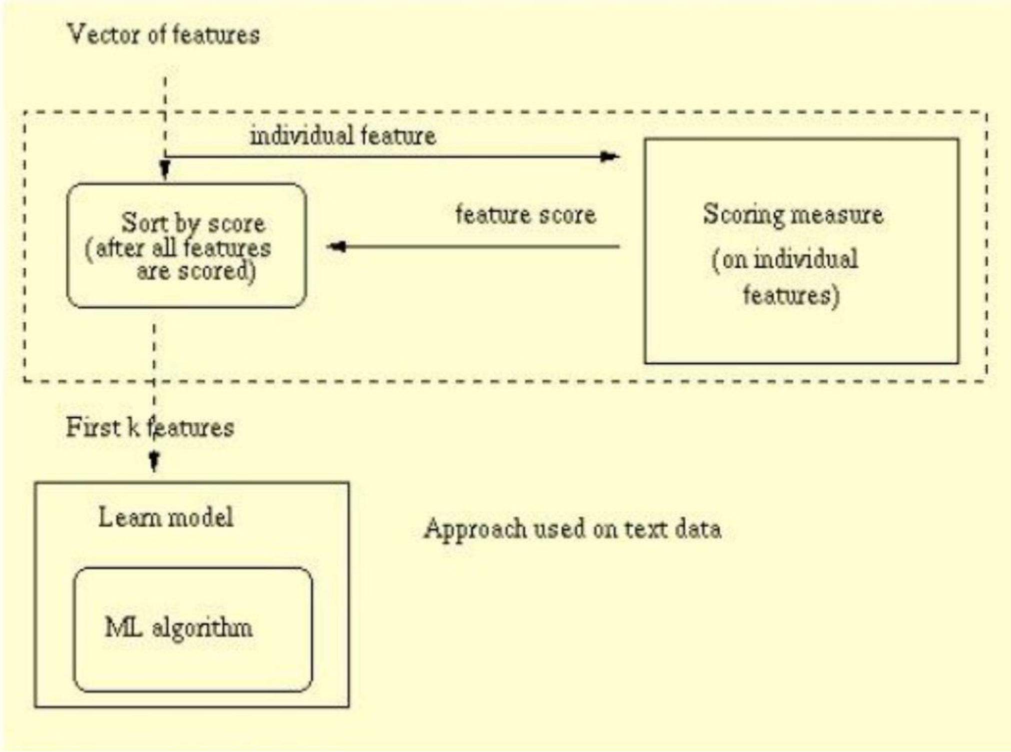
- Also called feature reduction techniques
- Due to **curse of high dimensionality**
- For BoW models:
  - Local dictionary
  - Removing Stopwords
  - Frequent Words
  - **Feature Selection**
  - Token reduction (stemming and synonyms)
  - Feature transformation (PCA, or Topic models)





Feature Selection by Attribute Ranking

- Can select a set of features (e.g., a set of words) to form a local dictionary.
- Raml feature attributes according to their predictive abilities for the category under consideration.
  - **Sports**: *soccer, football*, *etc*. **Travel**: *airport, cruise,etc*
- In this approach, simply select the top-ranking features.
- Feature Selection approaches:
  - Document Frequency
  - Information Gain
  - Mutual Information
  - CHI
  - [A survey]



Feature Selection based on Information Gain

On Widely Used Example

Outlook	Temperature	Humidity	Windy	Play
Sunny	Hot	High	False	***No***
Sunny	Hot	High	True	***No***
Overcast	Hot	High	False	***Yes***
Rainy	Mild	High	False	***Yes***
Rainy	Cool	Normal	False	***Yes***
Rainy	Cool	Normal	True	***No***
Overcast	Cool	Normal	True	***Yes***
Sunny	Mild	High	False	***No***
Sunny	Cool	Normal	False	***Yes***
Rainy	Mild	Normal	False	***Yes***
Sunny	Mild	Normal	True	***Yes***
Overcast	Mild	High	True	***Yes***
Overcast	Hot	Normal	False	***Yes***
Rainy	Mild	High	True	***No***

Features/ Attributes

Label

You may think the most important feature is the one that can be most related to the label.

Impurity of Splits

- S contain 20 occurrences of P and 20 of N.
- Assume each data has three binary features f1, f2, f3. Then, based on each feature, we are going to have three possible splits on the data.
- S1 means the feature is 0 and S2 means the feature is 1.
- For feature 1: S1 = 20P and S2 = 20N
- For feature 2: S1 = 10P, 10N and S2 = 10P, 10N
- For feature 3: S1 = 17P, 1N and S2 = 3P, 19N

Entropy

- Entropy is the measure of the information in a set of examples.

$$Entropy = -\sum_{i=1}^{K} p_i log_2 p_i$$

  - Where i={1,...,K}, K is the number of possible actions, pi is the proportion of each action i in the example set
  - For example: $Entropy([9*, 5+, 6-]) = -\frac{9}{20} log_2 \frac{9}{20} - \frac{5}{20} log_2 \frac{5}{20} - \frac{6}{20} log_2 \frac{6}{20}$

- High Entropy: more information
- Low Entropy: less information

Properties of Entropy

- Maximized when events are heterogeneous (impure):
  - A set of many mixed classes (say, rgb OOO) is unpredictable. High Entropy

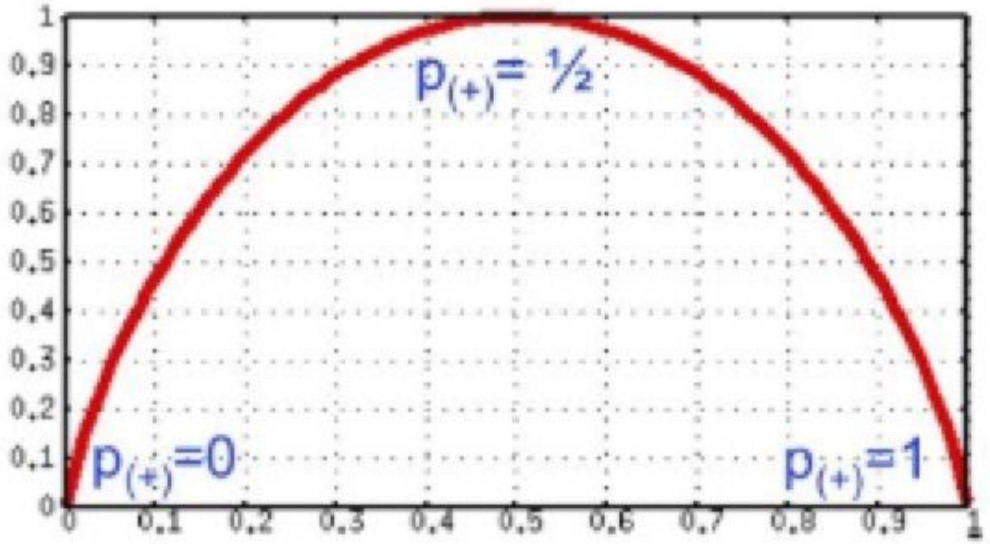
$$Entropy = log_2 K \quad \text{if all } p_i = \frac{1}{K}$$

- Minimized when events are homogenous (pure):
  - A set of only one class (say, blue OOO) is extremely predictable. Low entropy

$$Entropy = 0 \quad \text{if one } p_i = 1 \text{ the rest are zeros}$$

Entropy for binary case

- *S* is a sample of training examples
  - P+ is the proportion of positive examples in S
  - P- is the proportion of negative examples in S



- Entropy measures the impurity of S

$$Entropy(S) = -p_+ log_2 p_+ - p_- log_2 p_-$$

$$Entropy([9+, 5-]) = -\frac{9}{14} log_2(\frac{9}{14}) - \frac{5}{14} log_2(\frac{5}{14}) = 0.94$$

Information Gain

- Entropy:

$$E(X) = -\sum_{i=1}^{K} p(X = X_i) log_2 p(X = X_i)$$

  - **Intuition**: uncertainty of X, information contained in X, expected information bits required to represent X.
- Conditional Entropy

$$E(X|Y) = \sum_{i=1} p(Y = Y_i) E(X|Y = Y_i)$$

  - **Intuition**: given y, how much uncertainty remains in X
- **Mutual Information** (Information Gain)

$$I(X, Y) = E(X) - E(X|Y) = E(Y) - E(Y|X)$$

High IG, More Entropy Removed

  - **Intuition**: how much knowing Y reduces uncertainty about X, and vice versa.

IG: Example

Outlook	Temperature	Humidity	Windy	Play
Sunny	Hot	High	False	*No*
Sunny	Hot	High	True	*No*
Overcast	Hot	High	False	*Yes*
Rainy	Mild	High	False	*Yes*
Rainy	Cool	Normal	False	*Yes*
Rainy	Cool	Normal	True	*No*
Overcast	Cool	Normal	True	*Yes*
Sunny	Mild	High	False	*No*
Sunny	Cool	Normal	False	*Yes*
Rainy	Mild	Normal	False	*Yes*
Sunny	Mild	Normal	True	*Yes*
Overcast	Mild	High	True	*Yes*
Overcast	Hot	Normal	False	*Yes*
Rainy	Mild	High	True	*No*

$$E = -\sum_{i=1}^{K} p_k log_2 k$$

$$= -\frac{5}{14}\log_2\frac{5}{14} - \frac{9}{14}\log_2\frac{9}{14}$$
$$= 0.94$$

IG: Example

Outlook	Temperature	Humidity	Windy	Play
Sunny	Hot	**High**	False	***No***
Sunny	Hot	**High**	True	***No***
Overcast	Hot	**High**	False	***Yes***
Rainy	Mild	**High**	False	***Yes***
Rainy	Cool	**Normal**	False	***Yes***
Rainy	Cool	**Normal**	True	***No***
Overcast	Cool	**Normal**	True	***Yes***
Sunny	Mild	**High**	False	***No***
Sunny	Cool	**Normal**	False	***Yes***
Rainy	Mild	**Normal**	False	***Yes***
Sunny	Mild	**Normal**	True	***Yes***
Overcast	Mild	**High**	True	***Yes***
Overcast	Hot	**Normal**	False	***Yes***
Rainy	Mild	**High**	True	***No***

$$\Delta E(Humidity) = E - \frac{m_{i=H}}{m} E(i=H) - \frac{m_{i=N}}{m} E(i=N)$$

$$= 0.94 - \frac{7}{14} H_L - \frac{7}{14} H_R$$

IG: Example

Outlook	Temperature	Humidity	Windy	Play
Sunny	Hot	High	False	*No*
Sunny	Hot	High	True	*No*
Overcast	Hot	High	False	*Yes*
Rainy	Mild	High	False	*Yes*
Rainy	Cool	**Normal**	False	*Yes*
Rainy	Cool	**Normal**	True	*No*
Overcast	Cool	**Normal**	True	*Yes*
Sunny	Mild	High	False	*No*
Sunny	Cool	**Normal**	False	*Yes*
Rainy	Mild	**Normal**	False	*Yes*
Sunny	Mild	**Normal**	True	*Yes*
Overcast	Mild	High	True	*Yes*
Overcast	Hot	**Normal**	False	*Yes*
Rainy	Mild	High	True	*No*

$$\Delta E(Humidity) = E - \frac{m_{i=H}}{m} E(i = H) - \frac{m_{i=N}}{m} E(i = N)$$

$$= 0.94 - \frac{7}{14} H_L - \frac{7}{14} H_R$$

$$H_L = -\frac{6}{7}\log_2\frac{6}{7} - \frac{1}{7}\log_2\frac{1}{7}$$

IG: Example

Outlook	Temperature	Humidity	Windy	Play
Sunny	Hot	**High**	False	***No***
Sunny	Hot	**High**	True	***No***
Overcast	Hot	**High**	False	***Yes***
Rainy	Mild	**High**	False	***Yes***
Rainy	Cool	Normal	False	***Yes***
Rainy	Cool	Normal	True	***No***
Overcast	Cool	Normal	True	***Yes***
Sunny	Mild	**High**	False	***No***
Sunny	Cool	Normal	False	***Yes***
Rainy	Mild	Normal	False	***Yes***
Sunny	Mild	Normal	True	***Yes***
Overcast	Mild	**High**	True	***Yes***
Overcast	Hot	Normal	False	***Yes***
Rainy	Mild	**High**	True	***No***

$$\Delta E(Humidity) = E - \frac{m_{i=H}}{m} E(i = H) - \frac{m_{i=N}}{m} E(i = N)$$

$$= 0.94 - \frac{7}{14} H_L - \frac{7}{14} H_R$$

$$H_L = -\frac{6}{7}\log_2\frac{6}{7} - \frac{1}{7}\log_2\frac{1}{7}$$
$$= 0.592$$

$$H_R = -\frac{3}{7}\log_2\frac{3}{7} - \frac{4}{7}\log_2\frac{4}{7}$$
$$= 0.985$$

IG: Example

Outlook	Temperature	Humidity	Windy	Play
Sunny	Hot	High	False	***No***
Sunny	Hot	High	True	***No***
Overcast	Hot	High	False	***Yes***
Rainy	Mild	High	False	***Yes***
Rainy	Cool	Normal	False	***Yes***
Rainy	Cool	Normal	True	***No***
Overcast	Cool	Normal	True	***Yes***
Sunny	Mild	High	False	***No***
Sunny	Cool	Normal	False	***Yes***
Rainy	Mild	Normal	False	***Yes***
Sunny	Mild	Normal	True	***Yes***
Overcast	Mild	High	True	***Yes***
Overcast	Hot	Normal	False	***Yes***
Rainy	Mild	High	True	***No***

$$\Delta E(Humidity) = E - \frac{m_{i=H}}{m} E(i = H) - \frac{m_{i=N}}{m} E(i = N)$$

$$= 0.94 - \frac{7}{14} H_L - \frac{7}{14} H_R$$

$$0.94 - \frac{7}{14} 0.592 - \frac{7}{14} 0.985$$

$$= 0.94 - 0.296 - 0.4925$$

$$= 0.1515$$

IG: Example

Outlook	Temperature	Humidity	Windy	Play
Sunny	Hot	High	False	***No***
Sunny	Hot	High	True	***No***
Overcast	Hot	High	False	***Yes***
Rainy	Mild	High	False	***Yes***
Rainy	Cool	Normal	False	***Yes***
Rainy	Cool	Normal	True	***No***
Overcast	Cool	Normal	True	***Yes***
Sunny	Mild	High	False	***No***
Sunny	Cool	Normal	False	***Yes***
Rainy	Mild	Normal	False	***Yes***
Sunny	Mild	Normal	True	***Yes***
Overcast	Mild	High	True	***Yes***
Overcast	Hot	Normal	False	***Yes***
Rainy	Mild	High	True	***No***

1. Compute the information gain for the rest three features:
   - outlook
   - temperature
   - windy
2. Should we select features with high IG or low IG?

When it comes to text mining

- Especially text categorization
- The previous features/attributes will be “words” or “terms”
- The information gain of a term measures:
    - The expected reduction in entropy caused by partitioning the sample documents according to the term:

$$IG(t) = -\sum_{i=1}^{m} p(c_i) log p(c_i) + p(t)\sum_{i=1}^{m} p(c_i|t) log p(c_i|t)) + p(\bar{t})\sum_{i=1}^{m} p(c_i|\bar{t}) log p(c_i|\bar{t}))$$

where
t is a term,
m is the total number of classes
$p(c_i)$ is the percentage of documents in category ci from total sample documents
$p(t)$ is the percentage of documents in which term t is present
$p(\bar{t})$ is the percentage of documents in which term t is absent
$p(c_i|t)$ is the conditional probability of category given term t
$p(c_i|\bar{t})$ is conditional probability of category given term t is absent