Text Mining
Models and Applications

Road Map

1. Basics
2. Named Entity Recognition
3. Opinion Mining

Text Analysis

- The goal is to understand textual content
- Abstract concepts are difficult to represent: similar representations (e.g., space ship, flying saucer, UFO)
- High dimensionality: tens or hundreds of thousands of features
- We can take advantage of the redundancy of data
- Perform small and simple tasks:
  - Find important phrases,
  - Find related words,
  - Create summaries from documents

How To Represent Text

- **Lexical:** Character, Words, Phrases, Part-of-speech, Taxonomies
- **Syntactic**: Vector-space model, Language models, Full parsing
- **Semantic**: Collaborative tagging, Ontologies

Lexical: Character Level

- Represent documents as a sequence of characters of length 1,2,3…

  Text: democracy is a government by the people

  Description: dem, emo, moc, ocr, cra, rac, acy, is, a, gov, ove, …

- Each character sequence represents a feature with its frequency
- Advantages
  - Very robust (e.g., useful for language detection)
  - Captures simple patterns (e.g., spam detection, copy detection)
  - Redundancy in data helps analytical tasks (learning, clustering, search)
- Disadvantages
  - Too weak for deeper semantic tasks

Lexical: Word Level

- The most common representation of text

Text: democracy is a government by the people

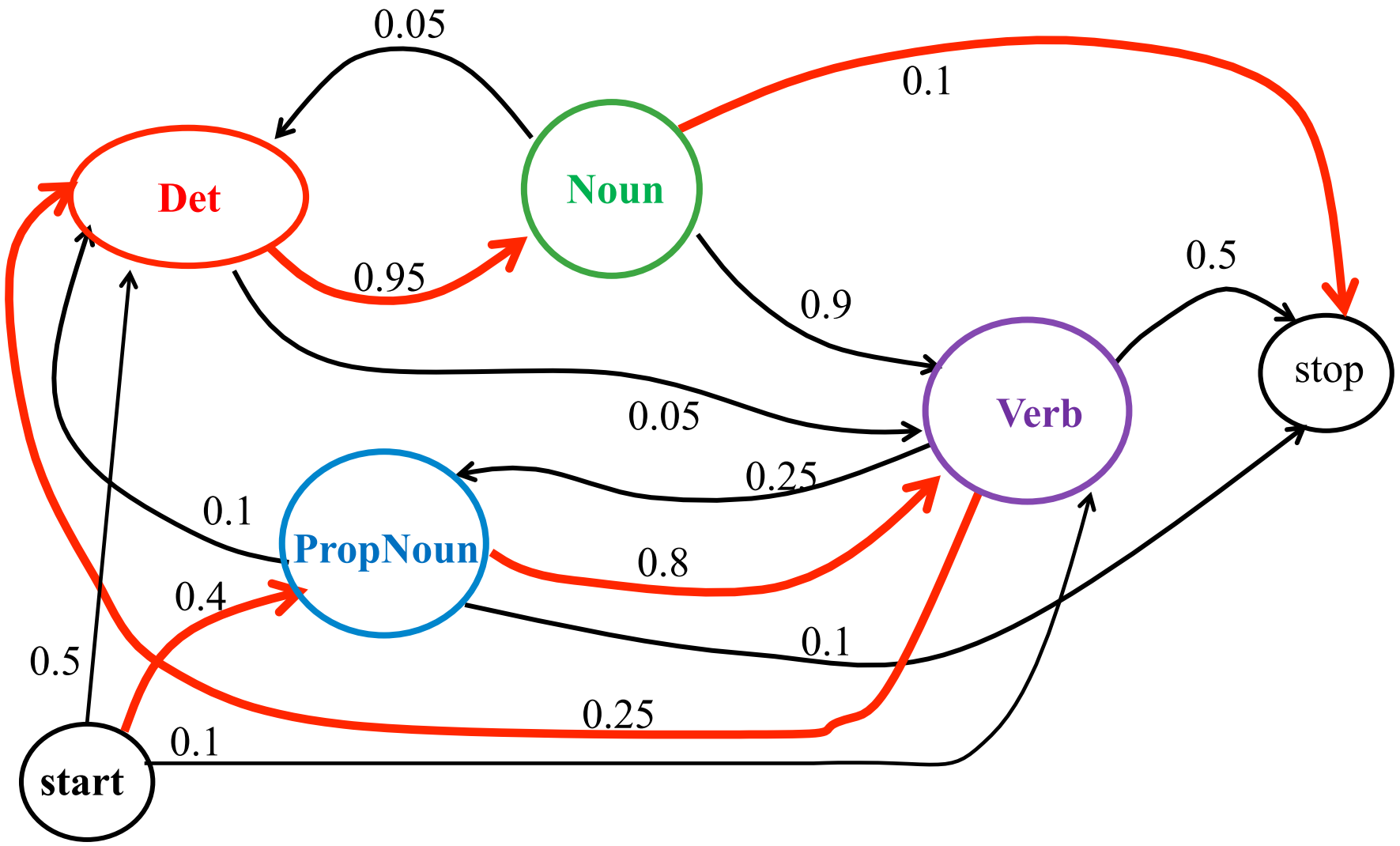
Description: **(1)** democracy, is, a, government, by, the, people
(2) democracy-is, is-a, a-government, government-by, by-the, the-people

- It is important to note that non-Western languages might have different notion of semantic unit (e.g. Chinese)
- Words are related depending on their forms and senses: *Homonymy, Polysemy, Synonymy, Hyponymy*
- Word frequencies have power distribution (small number of very frequent words and big number of words with low frequency)

Lexical: Phrase Level

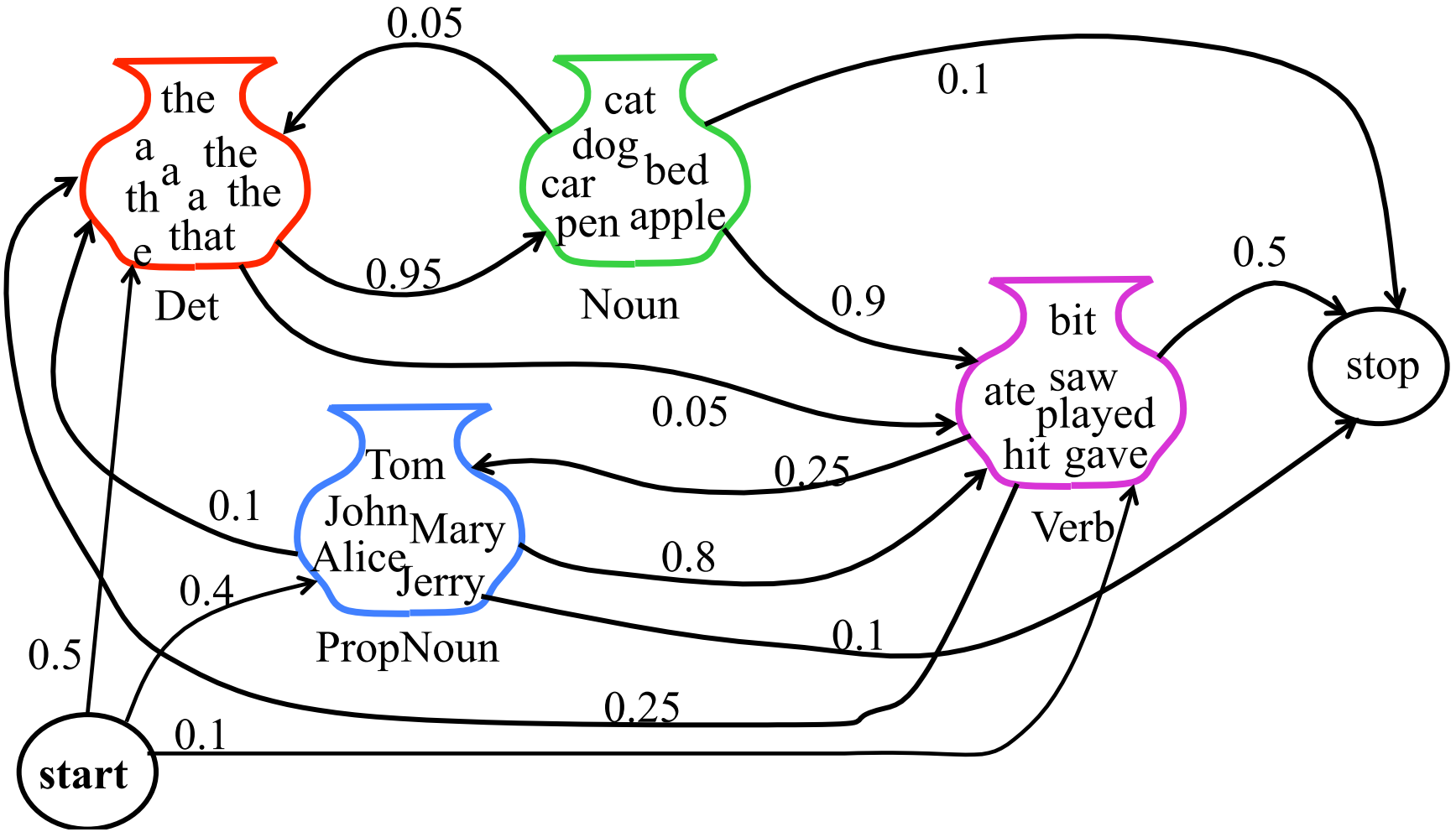
- Phrases can be used to represent text
  - Phrases as frequent contiguous word sequences
  - Phrases as frequent non-contiguous word sequences
- The main effect of using phrases is to more precisely identify sense
- Google use n-grams for different purposes

Lexical: Part-of-speech tags



Lexical: Part-of-speech tags

- Named Entity Extraction (noun phrases)
- Vocabulary Reduction (only nouns)



Lexical: Taxonomies

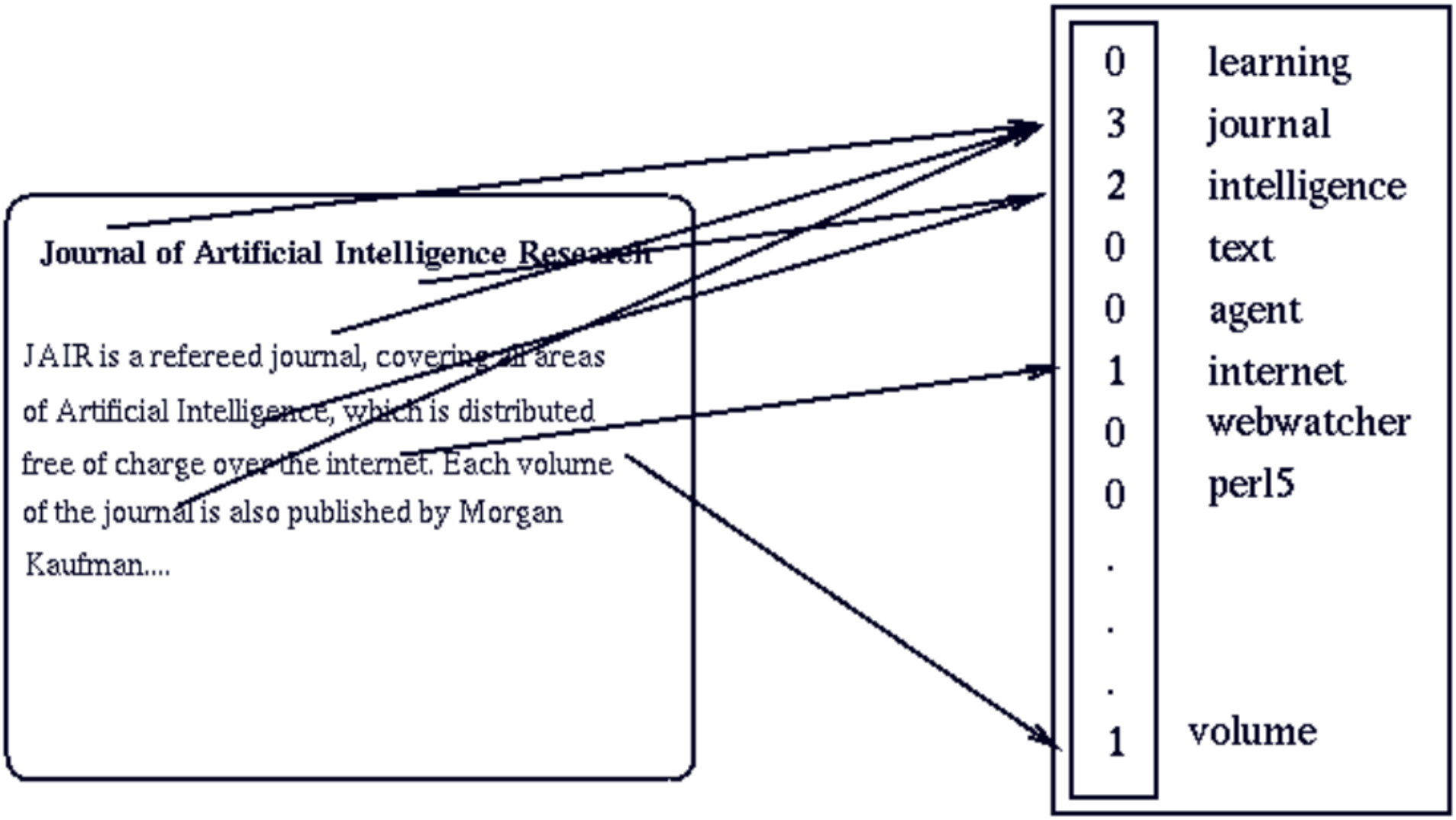
- WordNet: Each entry is connected with other entries in the graph through relations
- Relations in the database of nouns

Relation	Definition	Example
Hypernym	From lower to higher concepts	breakfast -> meal
Hyponym	From concepts to subordinates	meal -> lunch
Has-Member	From groups to their members	faculty -> professor
Member-Of	From members to their groups	copilot -> crew
Has-Part	From wholes to parts	table -> leg
Part-Of	From parts to wholes	course -> meal
Antonym	Opposites	leader -> follower

Syntactic: Vector Space Model

- Transform documents into sparse numeric vectors
- Forget everything about linguistic structure within the text
- It is referred to also as "**Bag-Of-Words**"
- Typical tasks are classification, clustering, visualization, etc

Syntactic: Bag-Of-Words Model



Syntactic: Bag-Of-Words Model

- **Word Weighting**: each word is represented as a separate variable having numeric weight (importance)
- The most popular weighting schema is normalized word frequency TFIDF

$$tfidf(word) = tf(word) \times idf(word)$$

tf: term frequency
The word is more important
If it appears several times
in a target document

idf: inverted document frequency
The word is more important
if it appears in less documents

Syntactic: Bag-Of-Words Model

$$tfidf(word) = tf(word) \times idf(word)$$

- Assume the following quotes of Albert Einstein

D1:"The difference between stupidity and genius is that genius has its limits.

D2:"The true sign of intelligence is not knowledge but imagination.

D3:"Logic will get you from A to B. Imagination will take you everywhere"

D4:"To raise new questions, new possibilities, to regard old problems from a new angle, requires creative imagination and marks real advance in science.

- **Task**:
  - compute the tfidf score of the word **"genius"** in document **D1**
  - compute the tfidf score of the word **"imagination"** in document **D2**

Syntactic: Bag-Of-Words Model

- The most popular weighting schema is normalized word frequency

$$tfidf(word) = tf(word) \times \log \frac{N}{df(word)}$$

tf(word): term frequency in the document
df (word): number of documents containing the word
N: the total number of documents
Tfidef(word): relative importance of the word in the document

- In Information Retrieval, the most popular weighting schema is:

$$BM25(word, D) = idf(word, D) \times \frac{tf(word, D) \times (k_1 + 1)}{tf(word, D) + k1 \times \left(1 - b + b \frac{|D|}{avgldl}\right)}$$

K1 belongs to [1.2 - 2.0] and b=0.75

Syntactic: Bag-Of-Words Model (Example)

TRUMP MAKES BID FOR CONTROL OF RESORTS Casino owner and real estate Donald Trump has offered to acquire all Class B common shares of Resorts International Inc, a spokesman for Trump said. The estate of late Resorts chairman James M. Crosby owns 340,783 of the 752,297 Class B shares. Resorts also has about 6,432,000 Class A common shares outstanding. Each Class B share has 100 times the voting power of a Class A share, giving the Class B stock about 93 pct of Resorts' voting power.

Original text

[RESORTS:0.624] [CLASS:0.487] [TRUMP:0.367] [VOTING:0.171] [ESTATE:0.166] [POWER:0.134] [CROSBY:0.134] [CASINO:0.119] [DEVELOPER:0.118] [SHARES:0.117] [OWNER:0.102] [DONALD:0.097] [COMMON:0.093] [GIVING:0.081] [OWNS:0.080] [MAKES:0.078] [TIMES:0.075] [SHARE:0.072] [JAMES:0.070] [REAL:0.068] [CONTROL:0.065] [ACQUIRE:0.064] [OFFERED:0.063] [BID:0.063] [LATE:0.062] [OUTSTANDING:0.056] [SPOKESMAN:0.049] [CHAIRMAN:0.049] [INTERNATIONAL:0.041] [STOCK:0.035] [YORK:0.035] [PCT:0.022] [MARCH:0.011]

Bag-of-Words representation (high dimensional sparse vector)

Syntactic: Language Models

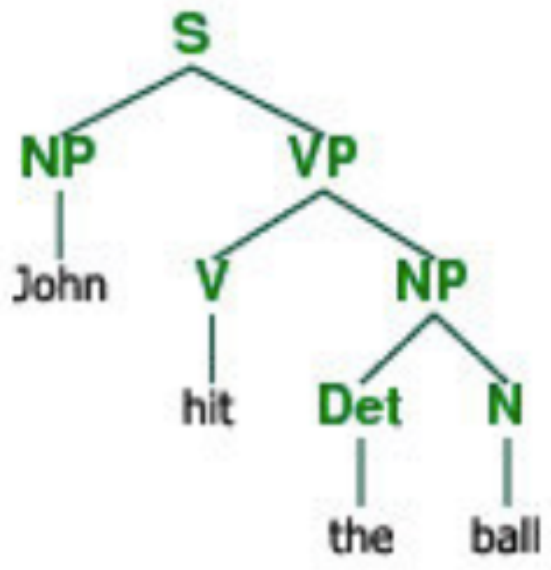
- Language modeling is about determining probability of a sequence of words
- The task typically gets reduced to the estimating probabilities of a next word given two previous words (trigram model):

$$P(w_3 \mid w_1, w_2) = \frac{P(w_1, w_2, w_3)}{P(w_2, w_3)}$$

- It has many applications including speech recognition, handwriting recognition, machine translation and spelling correction
- **Task**: take the Christmas wishes of lab5, and explain how would you implement:
  - A spell checker
  - A text word suggestion tool. Once it is done, what is the next word to suggest when typing "wish you"? And the next word when typing "full of"

Syntactic: Full- Parsing

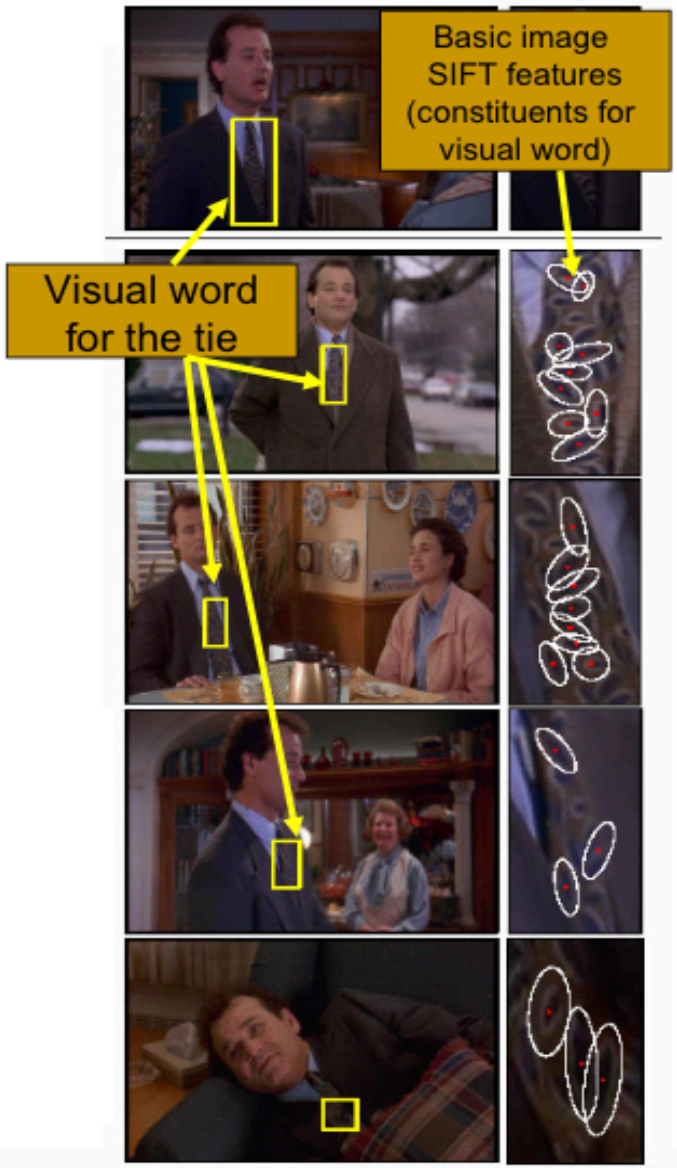
- Parsing provides maximum structural information per sentence
- On the input we get a sentence, on the output we generate a parse tree
- For most of the methods dealing with text data the information in parse trees is too complex



Syntactic: Cross Modality

- It is very often the case that objects are represented with different data types: Text documents, Multilingual texts documents, Images, Video, Social networks , Sensor networks

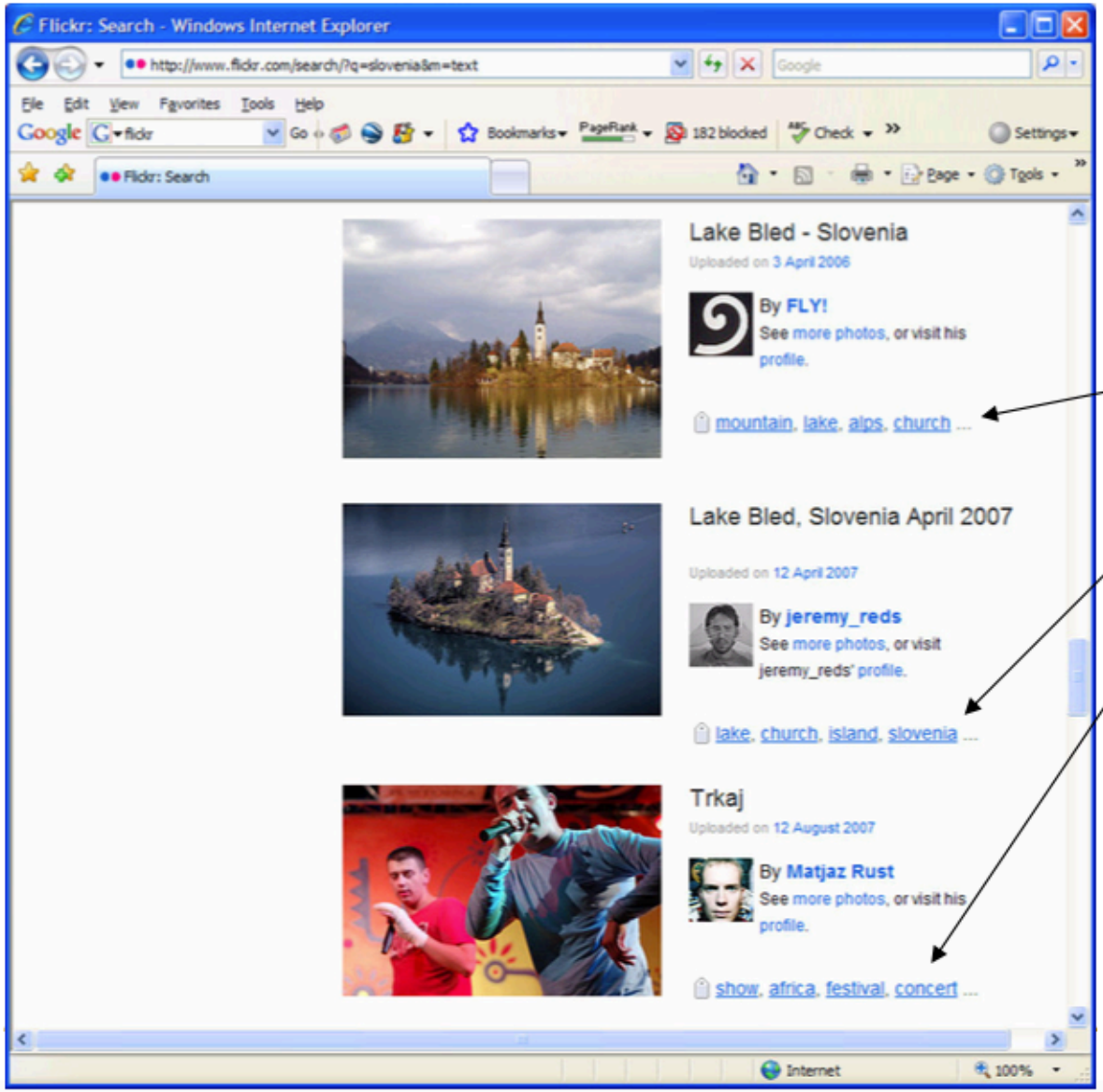
- The question is how to create mappings between different representation so that we can benefit using more information about the same objects



Semantic: Collaborative Tagging/Web2.0

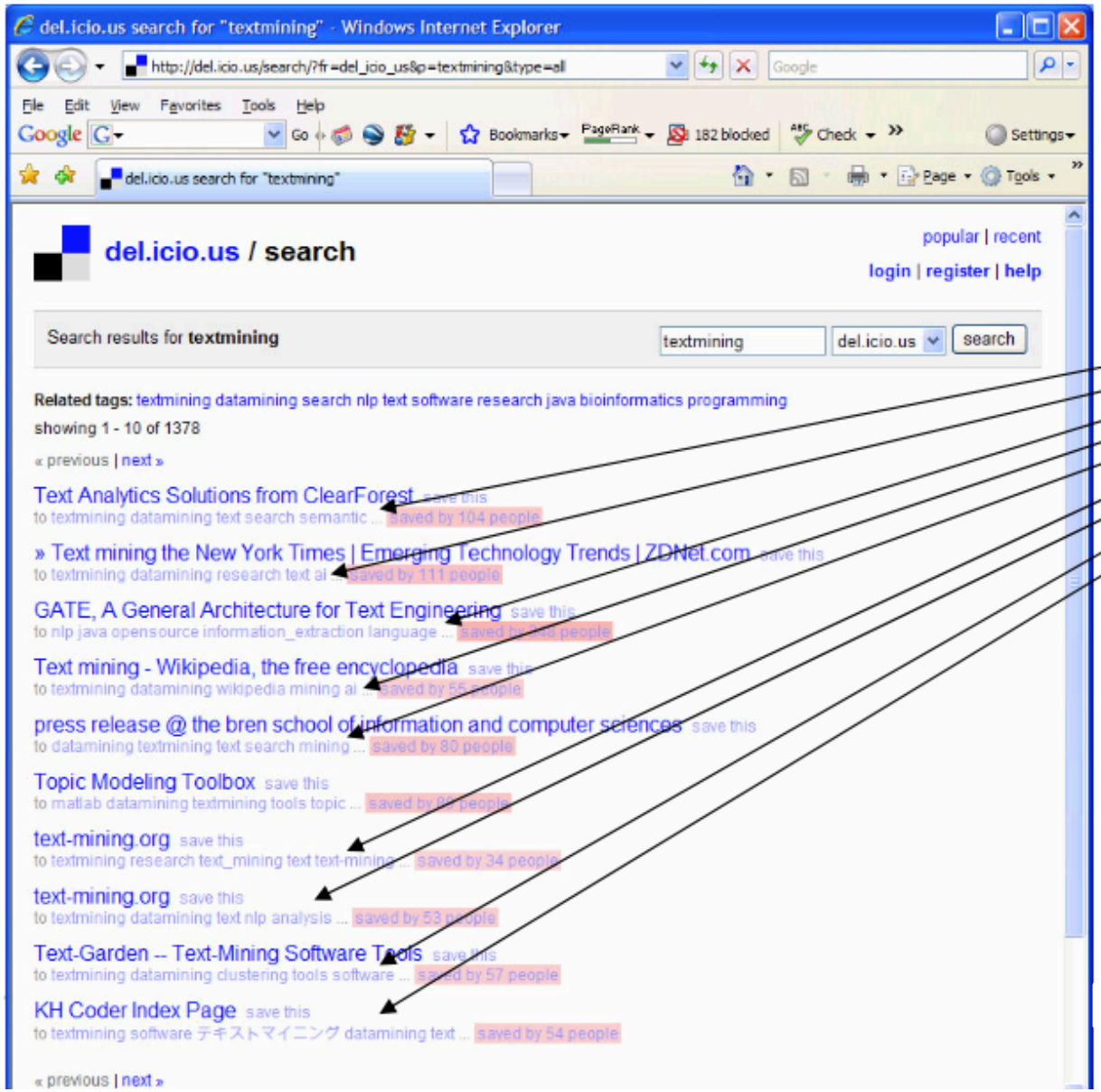
- Collaborative tagging is a process of adding metadata to annotate content (e.g. documents, web sites, photos)
- Metadata is typically in the form of keywords
- This is done in a collaborative way by many users from larger community collectively having good coverage of many topics
- As a result we get annotated data where tags enable comparability of annotated data entries

Semantic: Collaborative Tagging/Web2.0



Tags entered by users annotating photos

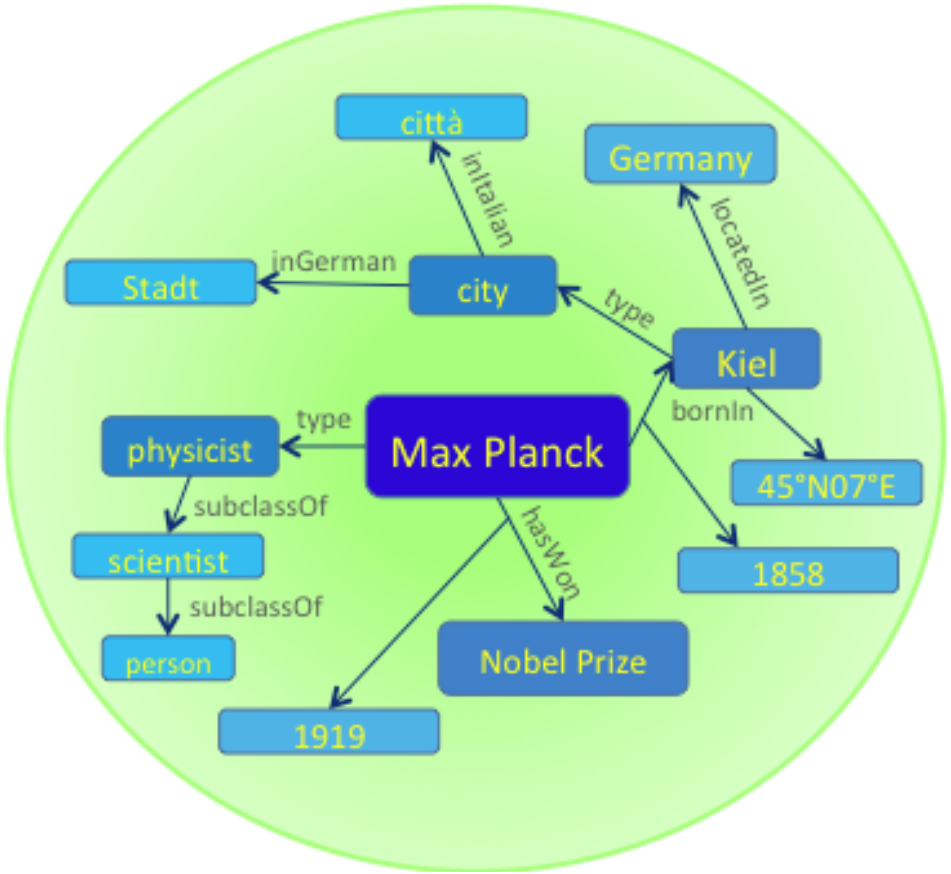
Semantic: Collaborative Tagging/Web2.0



Tags entered by users annotating Web sites

Semantic: Ontologies

- Ontologies are the most general formalism for describing data objects
- Ontologies represent very generic data-models where we can store extracted information from text



Road Map

1. Basics
2. Named Entity Recognition
3. Opinion Mining

Named Entity Recognition (NER)

- Identify mentions in text and classify them into a predefined set of categories of interest:
  - Person Names: **Prof. Jerry Hobbs**, **Jerry Hobbs**
  - Organizations: **Hobbs corporation**, **FbK**
  - Locations: **Ohio**
  - Date and time expressions: **February 2010**
  - E-mail: **mkg@gmail.com**
  - Web address: **www.usc.edu**
  - Names of drugs: **paracetamol**
  - Names of ships: **Queen Marry**
  - Bibliographic references:
  - …

First Approach: Rule-based NER

- Create rules to extract named entities
- Example of rules:
    1. Telephone number: blocks of digits separated by hyphens
    2. Email: contains "@"
    3. Location: Capitalized Word + city
    4. Person: Capitalized Word
- Why simple rules do not work?
    - First word of a sentence or words in a title are capitalized
    - New proper names constantly emerge
    - Ambiguity: Jordan can be the name of a person or a location
    - Various names for the same entity: Mouna Kacimi, Dr. Kacimi

Second Approach: Learning-based NER

Adam_B-PER Smith_I-PER works_O for_O **IBM_B-ORG** ,_O **London_B-LOC** ._O

- **NED**: Identify named entities using BIO tags
  - **B** beginning of an entity
  - **I** continues the entity
  - **O** word outside the entity

- **NEC:** Classify into a predefined set of categories
  - Person names
  - Organizations (companies, governmental organizations, etc.)
  - Locations (cities, countries, etc.)
  - Miscellaneous (movie titles, sport events, etc.)

Example of NER Tools

- Stanford NER

http://nlp.stanford.edu:8080/ner/

- Aida

https://gate.d5.mpi-inf.mpg.de/webaida/

Road Map

1. Basics
2. Named Entity Recognition
3. Opinion Mining

Facts and Opinions

- Two main types of textual information on the Web
  - Facts and Opinions

- Current search engines search for facts (assume they are true)
  - Facts can be expressed with topic keywords.
  - **Example**: "World cup 2014"

- Search engines do not search for opinions
  - Opinions are hard to express with a few keywords
  - **Example:** "How do people think of iPhones?"
  - Current search ranking strategy is not appropriate for opinion retrieval/search.

User Generated Content

- **Word-of-mouth on the Web**
  - One can express personal experiences and opinions on almost anything, at review sites, forums, discussion groups, blogs ... (called the user generated content.)
  - They contain valuable information
  - Web/global scale: No longer – one's circle of friends

- **Important to mine opinions** expressed in user- generated content
  - An intellectually very challenging problem.
  - Practically very useful.

Why Are Opinions Important?

- Opinions are key influencers of behaviors
- Our beliefs and perceptions of reality are largely conditioned on how others see the world
- Whenever we need to make a decision we often seek out the opinions of others:
  - **Individuals**: ask opinions from friends and family
  - **Organizations**: use surveys, focus groups, opinion polls, consultants

Applications

- **Businesses and organizations**: product and service benchmarking. Market intelligence
  - Business spends a huge amount of money to find consumer sentiments and opinions
  - Consultants, surveys and focused groups, etc
- **Individuals**: interested in other's opinions when
  - Purchasing a product or using a service
  - Finding opinions on political topics
- **Ads placements**: Placing ads in user-generated content
  - Place an ad when one praises a product
  - Place an ad from a competitor if one criticizes a product
- **Opinion retrieval/search**: providing general search for opinions

Typical Opinion Search Queries

- Find the opinion of a person or organization (opinion holder) on a particular object or a feature of the object
  - **E.g., what is Trump's opinion on abortion?**
- Find positive and/or negative opinions on a particular object (or some features of the object), e.g.,
  - **customer opinions on a digital camera**
  - **public opinions on a political topic**
- Find how opinions on an object change over time
- How object A compares with Object B?
  - **E.g., iPhones vs. Android**

How to Search Opinions

- **Opinion of a person on X**
  - E.g., : Trump's opinion on abortion
  - Can be handled using keyword search

- **Find Opinions on an object**
  - E.g., product reviews
  - **General Web search (for a fact)**: rank pages according to some authority and relevance scores- The user views the first page (if the search is perfect). **One fact = Multiple facts**
  - **Opinion search:** Opinion search: rank is desirable, however reading only the review ranked at the top is not appropriate because it is only the opinion of one person. **One opinion ≠ Multiple opinions**

Search Opinions

- **Produce two rankings**
  - Positive opinions and negative opinions
  - Some kind of summary of both, e.g., # of each
- **Or, one ranking but**
  - The top (say 30) reviews should reflect the natural distribution of all reviews (assume that there is no spam), i.e., with the right balance of positive and negative reviews
- **Questions**
  - Should the user read all the top reviews?
  - OR Should the system prepare a summary of the reviews?

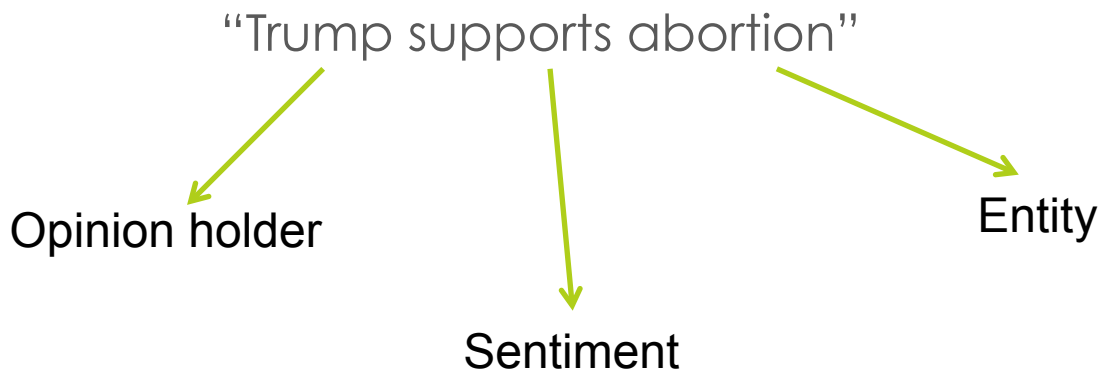
More on Opinion Mining

1. Opinion mining – the abstraction
2. Sentiment classification
3. Feature-based opinion mining

Basic Components of an Opinion

- **Opinion holder**: The person or organization that holds a specific opinion on a particular object.
- **Object/Entity**: on which an opinion is expressed
- **Sentiment**: a view, attitude, or appraisal on an object from the opinion holder

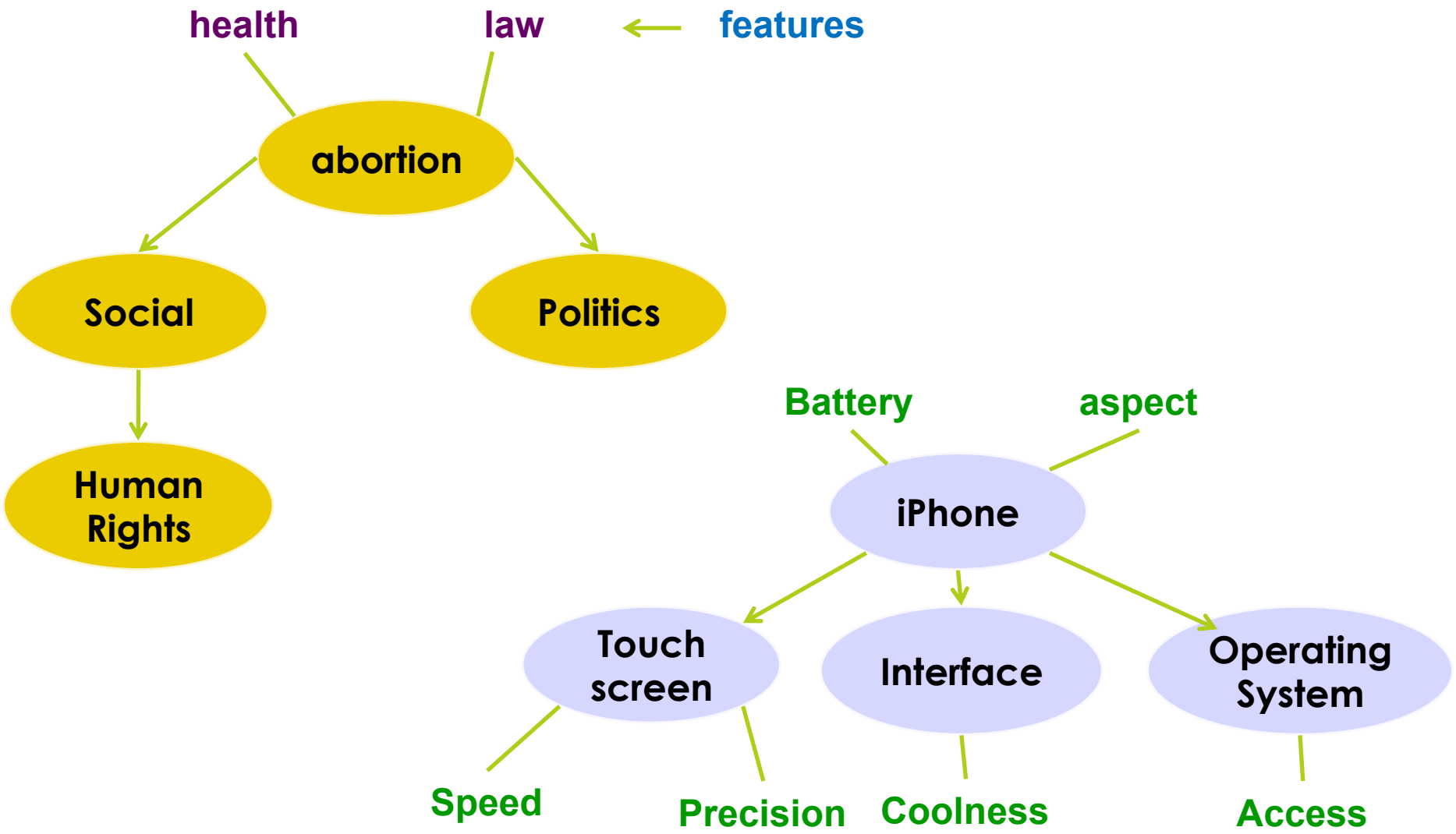
- **Example**:



Object or Entity

- An object ***O*** is an entity which can be a product, person, event, organization, or topic.
- ***O*** is represented as
  - a hierarchy of components, sub-components, and so on.
  - Each node represents a component and is associated with a set of **attributes** of the component.
  - *O* is the root node (which also has a set of attributes)
  - An opinion can be expressed on any node or attribute of the node.
- To simplify our discussion, we use "**features**" to represent both components and attributes.

Example


health
law
features
abortion
Social
Politics
Human Rights
Battery
aspect
iPhone
Touch screen
Interface
Operating System
Speed
Precision
Coolness
Access

Opinion Model

- An object ***O*** is represented by a finite set of features, $F = \{f_1, f_2, \ldots, f_n\}$.

- Each feature f_i in ***F*** can be expressed with a finite set of words or phrases W_i, which are **synonyms**. (we have a set of corresponding synonym sets $W = \{W_1, W_2, \ldots, W_n\}$ for the features)

- An opinion holder ***j*** comments on a subset of the features $Sj \subseteq F$ of object O. For each feature $f_k \in S_j$ that *j* comments on, he/she
  - chooses a word or phrase from W_k to describe the feature
  - expresses a positive, negative or neutral opinion on f_k

Opinion Mining Tasks

At document level

- **Sentiment classification**
  - **Classes**: positive, negative, and neutral
  - **Assumption:** each document (or review) focuses on a single object (not true in many discussion posts) and contains opinion from a single opinion holder

At sentence level

- **Identifying subjective/opinionated sentences**
  - Classes: objective and subjective (opinionated)
  - E.g, iPhone interface is cool (subjective or objective?)
  - Android is cheaper than iPhone ((subjective or objective?)
- **Sentiment classification of sentences**
  - **Classes**: positive, negative and neutral.
  - **Assumption**: a sentence contains only one opinion (not true in many cases).

Opinion Mining Tasks

At feature level

- Identify and extract object features that have been commented on by an opinion holder (e.g., a reviewer)
- Determine whether the opinions on the features are positive, negative or neutral
- Group feature synonyms.
- Produce a feature-based opinion summary of multiple reviews

Others

- **Opinion holders**: identify holders is also useful, e.g., in news articles, etc, but they are usually known in the user generated content, i.e., authors of the posts.

More on Opinion Mining

1. Opinion mining – the abstraction
2. Sentiment classification
3. Feature-based opinion mining

Sentiment Classification

- Classify documents (e.g., reviews) based on the overall sentiments expressed by opinion holders (authors),
  - Positive, negative, and (possibly) neutral

- What is the difference between topic-based text classification and sentiment classification?

- Similar but different from topic-based text classification.
  - In topic-based text classification, topic words are important.
  - In sentiment classification, sentiment words are more important, e.g., great, excellent, horrible, bad, worst, etc.

Unsupervised Sentiment Classification

- **Step 1**
  - Part of Speech Tagging (identification of words as nouns, verbs, adjectives, adverbs, etc)
  - The Stanford NLP
  - Extract two consecutive words (two-word phrases) from reviews if their tags conform to some given patterns, e.g., (1) JJ, (2) NN.
- **Sep2**
  - Compute phrases scores using Pointwise mutual information

$$PMI(word_1, word_2) = \log_2 \left(\frac{P(word_1 \wedge word_2)}{P(word_1)P(word_2)} \right)$$

(Turney, ACL-02)

Unsupervised Sentiment Classification

- **Step 3**
  - Estimate the semantic orientation (SO) of the extracted phrases
  - Semantic orientation (SO):

SO(phrase) = PMI(phrase, "excellent") - PMI(phrase, "poor")

  - "poor" and "excellent" correspond to the rates users give to the product together with the review

- **Step4**
  - Compute the average SO of all phrases
  - Classify the review as recommended if average SO is positive, not recommended otherwise.

(Turney, ACL-02)

Unsupervised Sentiment Classification

- Example of recommended review (Bank of America)

Extracted Phrase	Part-of-Speech Tags	Semantic Orientation
online experience	JJ NN	2.253
low fees	JJ NNS	0.333
local branch	JJ NN	0.421
small part	JJ NN	0.053
online service	JJ NN	2.780
printable version	JJ NN	-0.705
direct deposit	JJ NN	1.288
well other	RB JJ	0.237
inconveniently located	RB VBN	-1.541
other bank	JJ NN	-0.850
true service	JJ NN	-0.732
Average Semantic Orientation		0.322

Unsupervised Sentiment Classification

- Example of non-recommended review (Bank of America)

Extracted Phrase	Part-of-Speech Tags	Semantic Orientation
little difference	JJ NN	-1.615
clever tricks	JJ NNS	-0.040
programs such	NNS JJ	0.117
possible moment	JJ NN	-0.668
unethical practices	JJ NNS	-8.484
low funds	JJ NNS	-6.843
old man	JJ NN	-2.566
other problems	JJ NNS	-2.748
probably wondering	RB VBG	-1.830
virtual monopoly	JJ NN	-2.050
other bank	JJ NN	-0.850
extra day	JJ NN	-0.286
direct deposits	JJ NNS	5.771
online web	JJ NN	1.936
cool thing	JJ NN	0.395
very handy	RB JJ	1.349
lesser evil	RBR JJ	-2.288
Average Semantic Orientation		-1.218

Unsupervised Sentiment Classification

- **Data**: reviews from epinions.com on automobiles, banks, movies, and travel destinations.

- **Final classification accuracy:**
  - automobiles - 84%
  - banks - 80%
  - movies - 65.83
  - travel destinations - 70.53%

Supervised Sentiment Classification

- This approach directly applied several machine learning techniques to classify movie reviews into positive and negative

- Three classification techniques were tried:
  - Naïve Bayes
  - Maximum entropy
  - Support vector machine

- Pre-processing settings: negation tag, unigram (single words), bigram, POS tag, position.

- SVM: the best accuracy 83% (unigram)

(Pang et al, EMNLP-02)

Supervised Sentiment Classification

- A second approach classify reviews by scoring features
- It first selects a set of features **$F = f_1, f_2, \ldots\ldots$** (product features)
- Score the features
  - C and C' are classes

$$score(f_i) = \frac{P(f_i \mid C) - P(f_i \mid C')}{P(f_i \mid C) + P(f_i \mid C')}$$

- Classification of a review *dj* (using sign):

$$class(d_j) = \begin{cases} C & \text{eval}(d_j) > 0 \\ C' & \text{eval}(d_j) <= 0 \end{cases}$$

$$eval(d_i) = \sum_i score(f_i)$$

- Accuracy of 84-88%.

(Dave, Lawrence and Pennock, WWW-03)

More on Opinion Mining

1. Opinion mining – the abstraction
2. Sentiment classification
3. Feature-based opinion mining

Let's go further

- Sentiment classification at both document and sentence (or clause) levels are useful, but they do not find what the opinion holder liked and disliked
- A negative sentiment on an object does not mean that the opinion holder dislikes everything about the object.
- A positive sentiment on an object does not mean that the opinion holder likes everything about the object
- **We need to go to the feature level**

Before we go further

- Let us discuss **Opinion Words or Phrases** (also called polar words, opinion bearing words, etc). E.g.,
  - **Positive**: beautiful, wonderful, good, amazing,
  - **Negative**: bad, poor, terrible, cost someone an arm and a leg (idiom).
- They are instrumental for opinion mining (obviously)
- Three main ways to compile such a list:
  - Manual approach: not a bad idea, only a one-time effort
  - Corpus-based approaches
  - Dictionary-based approaches
- **Important to note: Some opinion words are context independent (e.g., good). Some are context dependent (e.g., long)**.

Different Review Format

Format 1

My SLR is on the shelf

by camerafun4. Aug 09 '04

Pros: Great photos, easy to use, very small
Cons: Battery usage; included memory is stingy.

I had never used a digital camera prior to purchasing have always used a SLR ... **Read the full review**

Format 2

User rating
Perfect
10
out of 10

"It is a great digitbal still camera for this century"
September 1, 2004

Pros:
It's small in size, and the rotatable lens is great. It's very easy to use, and has fast response from the shutter. The LCD has increased from 1.5 in to 1.8, which gives bigger view. It has lots of modes to choose from in order to take better pictures.

Cons:
It almost has no cons, it would be better if the LCD is bigger and it's going to be best if the model is designed to a smaller size.

Format 3

GREAT Camera., Jun 3, 2004

Reviewer: **jprice174** from Atlanta, Ga.

I did a lot of research last year before I bought this camera... It kinda hurt to leave behind my beloved nikon 35mm SLR, but I was going to Italy, and I needed something smaller, and digital.

The pictures coming out of this camera are amazing. The 'auto' feature takes great pictures most of the time. And with digital, you're not wasting film if the picture doesn't come out.

Feature Extraction from Format1

- **Observation**: Each sentence segment in Pros or Cons contains only one feature. Sentence segments can be separated by commas, periods, semi-colons, hyphens, '&', etc.
- Pros in Format1 Example can be separated into 3 segments:
  - great photos <photo>
  - easy to use <use>
  - very small <small> ⇒ <size>
- Cons can be separated into 2 segments:
  - battery usage <battery>
  - included memory is stingy <memory>

(Liu et al WWW-03; Hu and Liu, AAAI-CAAW-05)

Extraction Using Label Sequential Rules

- Label sequential rules (LSR) are a special kind of sequential patterns, discovered from sequences.
- LSR Mining is supervised (Liu's Web mining book 2006)
- The training data set is a set of sequences, e.g.,

"*Included memory is stingy*"

is turned into a sequence with POS tags

⟨{included, VB}{memory, NN}{is, VB}{stingy, JJ}⟩

then turned into

⟨{included, VB}{$feature, NN}{is, VB}{stingy, JJ}⟩

Feature Extraction from Format2 and Format3

- Reviews of these formats are usually complete sentences

  e.g., "the pictures are very clear."

  - Explicit feature: picture

    "It is small enough to fit easily in a coat pocket or purse."

  - Implicit feature: size

- Extraction: Frequency based approach
  - Frequent features
  - Infrequent features

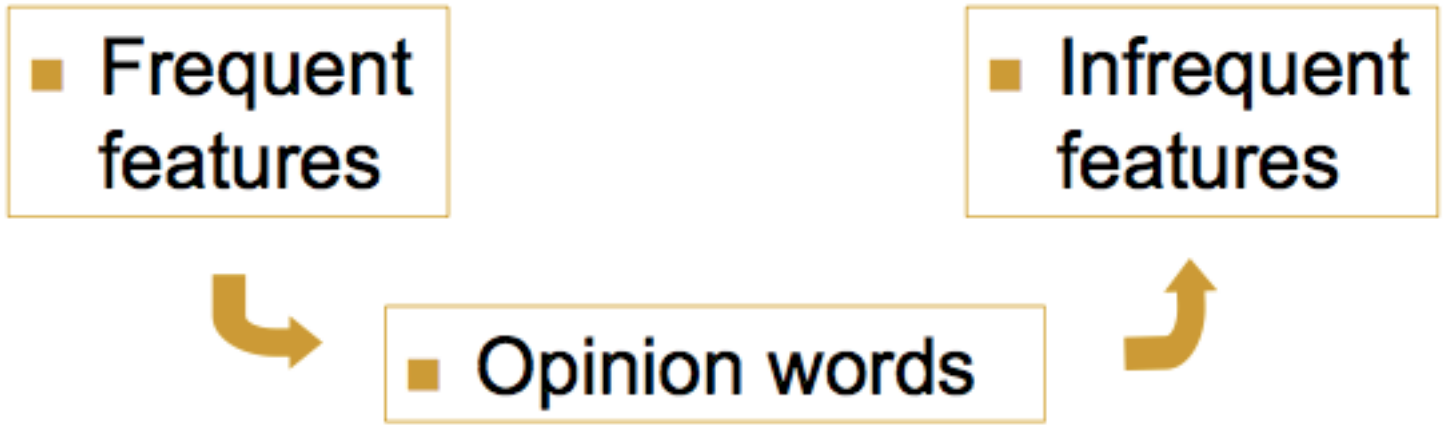
Frequency based approach

- **Frequent features**: those features that have been talked about by many reviewers.
  - Use sequential pattern mining
- Why the frequency based approach?
  - Different reviewers tell different stories (irrelevant)
  - When product features are discussed, the words that they use converge.
  - They are main features.
- Why sequential pattern mining?
  - Sequential pattern mining finds frequent phrases

(Hu and Liu, KDD-04; Liu, Web Data Mining book 2007)

Infrequent Features Extraction

- How to find the infrequent features?
- **Observation:** the same opinion word can be used to describe different features and objects
  - "The pictures are absolutely amazing."
  - "The software that comes with it is amazing."



Identify Opinion Orientation on Feature

- For each feature, identify the sentiment or opinion orientation expressed by a reviewer
- Analysis based on sentences, but also consider,
  - A sentence can contain multiple features.
  - Different features may have different opinions.

  E.g., The battery life and picture quality are *great* (+), but the view founder is *small* (-).

- Almost all approaches make use of opinion words and phrases. But notice again:
  - Some opinion words have context independent orientations, e.g., "great".
  - Some other opinion words have context dependent orientations, e.g., "small"
  - Many ways to use them.

Aggregation of Opinion Words

- **Input**: a pair (*f*, *s*), where *f* is a product feature and *s* is a sentence that contains *f*.
- **Output**: whether the opinion on *f* in *s* is positive, negative, or neutral.
- **Two steps**:
  - **Step 1**: split the sentence if needed based on BUT words (but, except that, etc).
  - **Step 2**: work on the segment s_f containing *f*. Let the set of opinion words in s_f be w_1, .., w_n. Sum up their orientations (1, -1, 0), and assign the orientation to (*f*, *s*) accordingly.