Overview

- NLP pipeline continued
- NLP applications
NLP pipeline continued
Basic NLP pipeline - Syntactic Analysis

**Input:** Apple took its annual spring event to Chicago this year.

**Tokenization**
Apple / took / its / annual / spring / event / to / Chicago / this / year

**Part-of-Speech Tagging**

```
NNP  VBD  PRP$  JJ  NN  NN  TO  NNP  DT  NN
```

Apple took its annual spring event to Chicago this year.

**Lemmatization**

```
Apple  take  its  annual  spring  event  to  Chicago  this  year .
```

Apple took its annual spring event to Chicago this year.

**Dependency Parsing**

Examples generated with the Stanford Core NLP toolset (http://corenlp.run/).

Dagmar Gromann, 23 April 2018

Semantic Computing
Basic NLP pipeline - Semantic Analysis

Input: Apple took its annual spring event to Chicago this year.

**Named Entity Recognition**
- Apple (ORGANIZATION)
- took (SET)
- its (DATE)
- annual (CITY)
- spring (DATE)
- event (DATE)
- to (DATE)
- Chicago (DATE)
- this (DATE)
- year (DATE)

**Relation Extraction**
- Apple (Entity) took (Relation) its (Entity) annual (Entity) spring (Entity) event (Entity) to (Entity) Chicago (Entity) this (Entity) year (Entity)

**Coreference Resolution**
- Apple (Mention) took (coref) its (Mention) annual (Entity) spring (Entity) event (Entity) to (Entity) Chicago (Entity) this (Entity) year (Entity)

**Sentiment Analysis**
- Apple (NEUTRAL) took (NEUTRAL) its (NEUTRAL) annual (NEUTRAL) spring (NEUTRAL) event (NEUTRAL) to (NEUTRAL) Chicago (NEUTRAL) this (NEUTRAL) year (NEUTRAL)

Examples generated with the Stanford Core NLP toolset (http://corenlp.run/).

Dagmar Gromann, 23 April 2018
Named Entity Recognition

Subtask of information extraction that locates and classifies named entities, i.e., a real-world object that can be denoted with a proper name - person, organization, location, products, etc.

```python
from nltk.tag.perceptron import PerceptronTagger
tagger = PerceptronTagger()

sent = "Apple took its annual spring event to Chicago this year."
tags = tagger.tag(nltk.word_tokenize(sent))
sent = nltk.ne_chunk(tags, binary=True) #
print(sent)
```

```
(S
(NE Apple/NNP)
took/VBD
its/PRP$
annual/JJ
spring/NN)
  event/NN
to/TO
  (NE Chicago/NNP)
this/DT
year/NN
./.)
```
Relation Extraction from Text

Also a subtask of information extraction with two main processes:

1. extraction of entities (NER)
   - People, organizations, locations, times, dates, prices, etc.

2. extraction of relations between those entities
   - Located in, employed by, part of, etc.

How?

- lexico-syntactic patterns (X is_a Y: “A dog is_a mammal.”)
- patterns and rules (PERSON [be]? (born) PREP PLACE, “Trump was born in New York City.”)
- Machine learning (supervised, unsupervision,...)
- Deep learning (all potential architectures)
Code Example Relex

Running Stanford CoreNLP from the command line.

```java
java -cp "*" -Xmx2g edu.stanford.nlp.pipeline.StanfordCoreNLP
-annotators tokenize,ssplit,pos,lemma,ner,parse,relation -file input.txt
Java 9: java --add-modules java.se.ee
Alternative: java -mx2g -cp "*" edu.stanford.nlp.naturalli.OpenIE
```

```xml
<MACHINE_READING>

<ENTITIES>

<ENTITY id="EntityMention-1">LOCATION
  <SPAN start="0" end="1"/>
  <PROBABILITIES/>
</ENTITY>

<ENTITY id="EntityMention-2">0
  <SPAN start="1" end="2"/>
  <PROBABILITIES/>
</ENTITY>

<ENTITY id="EntityMention-3">0
  <SPAN start="5" end="6"/>
  <PROBABILITIES/>
</ENTITY>
</ENTITIES>

</MACHINE_READING>
```

Alternative: TU Dresden is located in Germany

Dagmar Gromann, 23 April 2018
Semantic Computing

¹https://stanfordnlp.github.io/CoreNLP/cmdline.html
Coreference Resolution

Coreference resolution is the task of identifying all expressions (mentions) in a text that refer to the same real-world entity, such as

“She has not told her friend about that story because it is too embarrassing for her.”
Running StanfordCoreNLP from the command line ¹.

“She has not told her friend about that story because it is too embarrassing for her.”

```
java -cp "*" -Xmx3g edu.stanford.nlp.pipeline.StanfordCoreNLP
-annotators tokenize,ssplit,pos,lemma,ner,parse,dcoref -file input.txt
Java 9: java --add-modules java.se.ee
```

²https://stanfordnlp.github.io/CoreNLP/cmdline.html

Dagmar Gromann, 23 April 2018

Semantic Computing
Sentiment Analysis

Computational study of opinions, sentiments, evaluations, attitudes, affects, emotions, etc. found in text. Also called opinion mining.

- Polarity detection: positive, negative, neutral or on a scale of 1 to 5 how positive, negative or neutral
- Valence detection: valence is the "goodness" or "badness" of an emotion, which means it takes sentiment intensity into account (e.g. 0.83 negative on a scale from 0 to 1)
- Objectivity: how objective or subjective is a statement?
- Emotion classification: anger, fear, sadness, joy, etc.
- Stance classification: for or against a position
Sentiment Analysis - Example

Massive business value for all sentiment analysis applications - complaint management, product improvement, word-of-mouth marketing analysis, brand awareness, etc.

**Movie reviews**

- “Get off the screen.” 📽️
- “I watched the screening tonight and I really loved it.” 😊

**Product rating**

- ★★★☆☆☆“The echo dot turned Alexa into a douchebag salesman.”
- ★★★☆☆☆“A fun gadget, but the jury is still out on how useful it actually is.”
- ★★★★★★“The Smartest of Them All!!!”
Sentiment Analysis on Twitter

**Twitter analysis**

Measurement of the collective mood state based on large-scale Twitter feeds analysis and its correlation to the value of the Dow Jones Industrial Average (DJIA) over time.
Comparison: presidential election and Thanksgiving (as baseline)
SenticNet: Concept-Level Sentiment Analysis

Basic Code Example using NLTK Vader

VADER = Valence Aware Dictionary and sEntiment Reasoner

```python
from nltk.sentiment.vader import SentimentIntensityAnalyzer

sia = SentimentIntensityAnalyzer()

sentences = ['Get off the screen.', 'I watched the screening tonight and I really loved it.', 'The Smartest of Them All', 'Very bad movie!']

for sentence in sentences:
    print(sentence)
    ss = sia.polarity_scores(sentence)
    for k in sorted(ss):
        print('{0}: {1}, '.format(k, ss[k], end=''))

Get off the screen.
compound: 0.0, neg: 0.0, neu: 1.0, pos: 0.0
I watched the screening tonight and I really loved it.
compound: 0.6361, neg: 0.0, neu: 0.625, pos: 0.375
The Smartest of Them All
compound: 0.6124, neg: 0.0, neu: 0.5, pos: 0.5
Very bad movie!
compound: -0.623, neg: 0.671, neu: 0.329, pos: 0.0
```
NLP tasks

Each of the presented processing steps in the NLP pipeline is a whole research field in its own right with many different approaches to tackle its core problems. Some more:

- **Word Sense Disambiguation**: identify the correct sense of a word in a context, e.g. Tutorial 1 Exercise on WordNet
- **Semantic Role Labeling (shallow parsing)**: assigning labels to elements of a sentence that indicate their role, e.g. agent, goal, means. Demo: Curator
- **Spelling correction**: automatically correct spelling mistakes
- **Many more...**
Review of Lecture 3

- What is Named Entity Recognition?
- Which two processes are needed for relation extraction?
- What is sentiment analysis?
- What is the difference between emotion classification and polarity detection?