Lecture 9 NER and Conference Resolution

Information Extraction

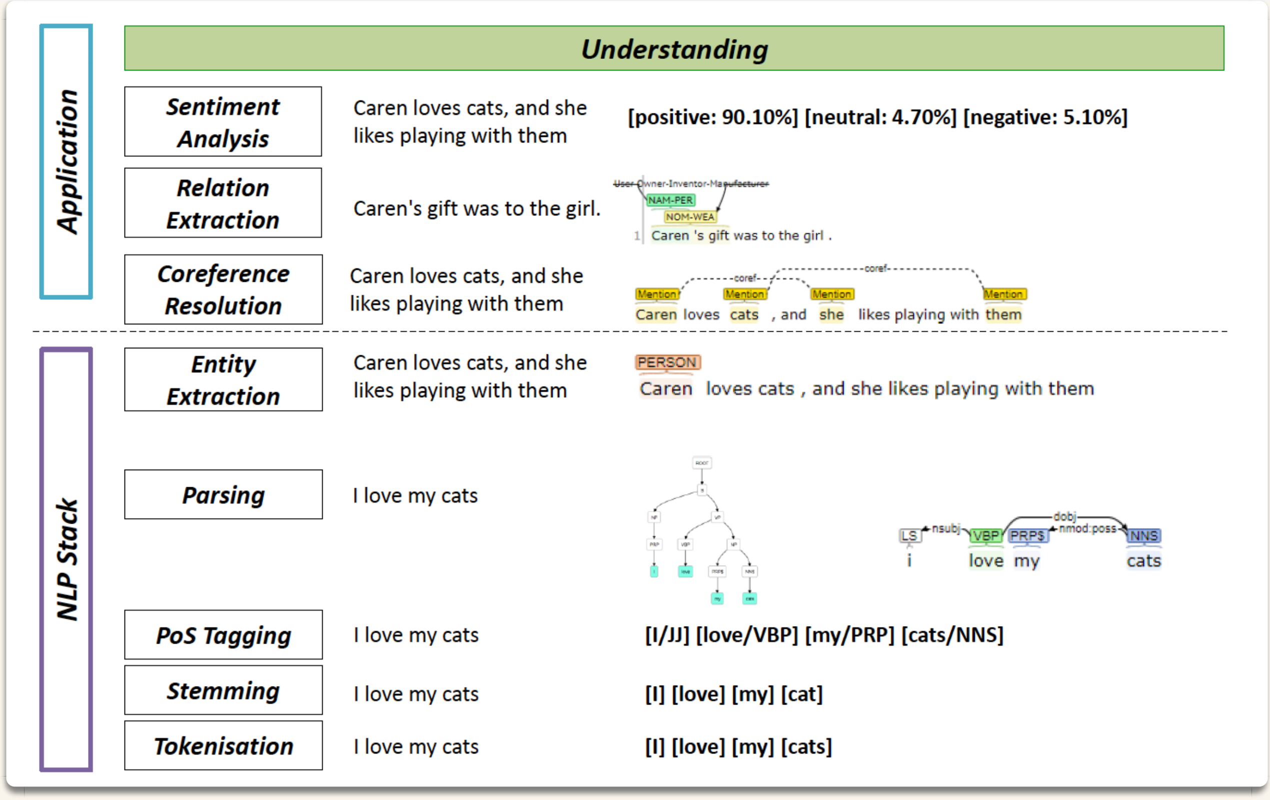
The task is to automatically extracting structured information for unstructured and/or semi structured machine-readable documents

- How to allow computation to be done on the unstructured data
- How to extract the structured clear, factual information
  - FInd and understand limited relevat parts of texts
  - Gather indormation from many pieces of text
  - produce a structured representation of relevant information

    —> Relations: database sense or a knowledge base

- How to put in a semantically prices form that allows further inferences to be made by computer algorithms

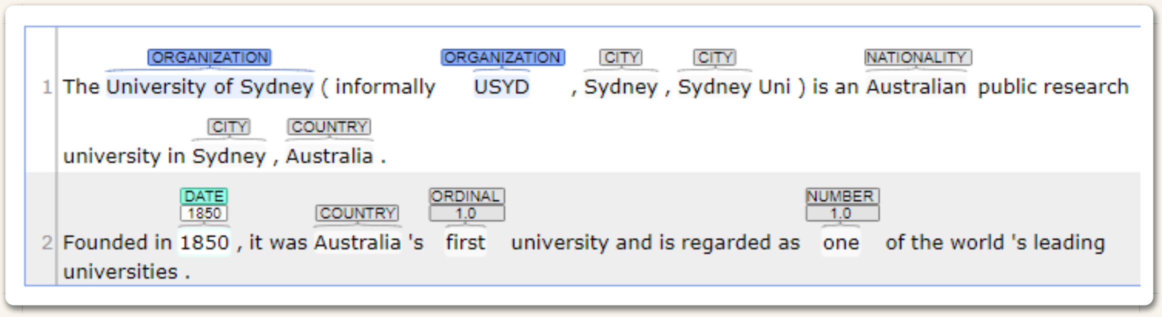
IE Pipline with NLP



NER

The subtaks of IE that seeks to locat and classify named entity mentions in unstructured text into predefined categories

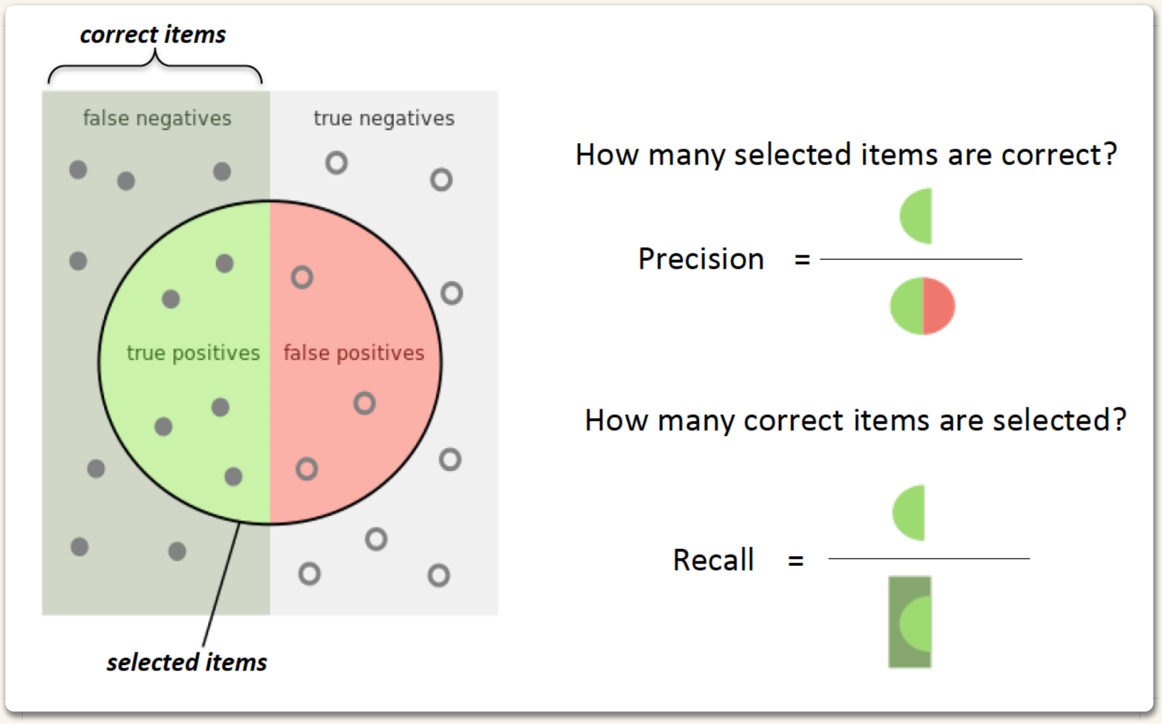
- Why NER ?
  - named entities can be indexed and linked off
  - Sentiment can be attributed to companies or products
  - A lot of relations are associations between named entities
  - For QA system, answers are often named entities
- How to recognize?
  - identify and classify names in text



- How to evaluate?

- The goal: predicting entities in a text
- Standard evaluation is per entity,not per token
- Metrics: Precision and Recall

  — straightforward for text categorization or web search, where there is only one grain size (document)



Tranditional NER

- **Rule-based NER**
  - Entity references have internal and external language cues
  - Can recognise names using lists

    — Personal tiltles: Mr., Miss, Dr., President

    — Given names: Scott,David,Jamse

    — Corporate suffixes: & Co., Corp., Ltd

    — Orgnisation: Microsoft, IBM, Telstra
  - Cna recogniase names using rules

    — personal tiltle X => per

    — X,location => loc or org

    — travel verb to X => loc

    - Effectively regular expressions

- **Statistical approaches (more portable)**

    - Leran NER from annotated text

      — weights $\approx$ rules calculated from the corpus

      — same machine learner,different language or domain

    - Token-by-token classification
    - Each token may be:

      — not part of an entity (tag o)

      — beginning an entity (tag b-per, b-org, etc.)

      — continuing an entity (tag i-per, i-org, etc.)

    - N-gram model:

      $t_n = argmax\, p(t|w_n, wn-1, w_{n-2}) \quad t \in T$

- **Comparison (rule-based v.s statistical)**

    - Rule-based

      — can be high-performing and efficient

      — require experts to make rules

      — rely heavily on gazetteers that are always incomplete

      — not robust to new domains and languages

    - Statistical approaches

      — require(expert-) annotated training data

      — may identify unseen patterns

      — robust for experimentation with new features

      — largely portable to new languages and domains

Sequence model

- IOB tagging v.s IO tagging

  — computation time: IOB > IO

  As I is a token inside a chunk, O is a token outside a chunk and B is the beginning of chunk immediately following another chunk of the same Named Entity, the IOB tagging need more time to computing each chunks

  — efficiency: IOB > IO

  Here, only the I and O labels are used. This therefore cannot distinguish between adjacent chunks of the same named entity.

- Features for sequence labeling
  - Words

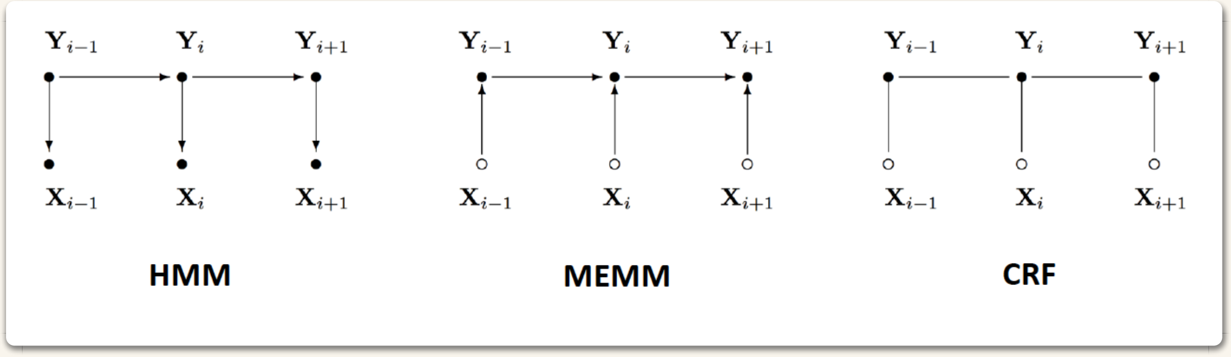
    — current word (like a learned dictionary)

    — previous/next word (context)
  - other kinds of inferred linguistic calssification

    — PoS tags
  - Label context

    —previous (and perhanps next) label

- HMM, MEMM, CRF sequence model



- Greedy Inference
  - Concept

— start at the left to assign a label by using our classifier at each position

— classifier can depend on previous labelling decisions as well as observed data

- Advantages

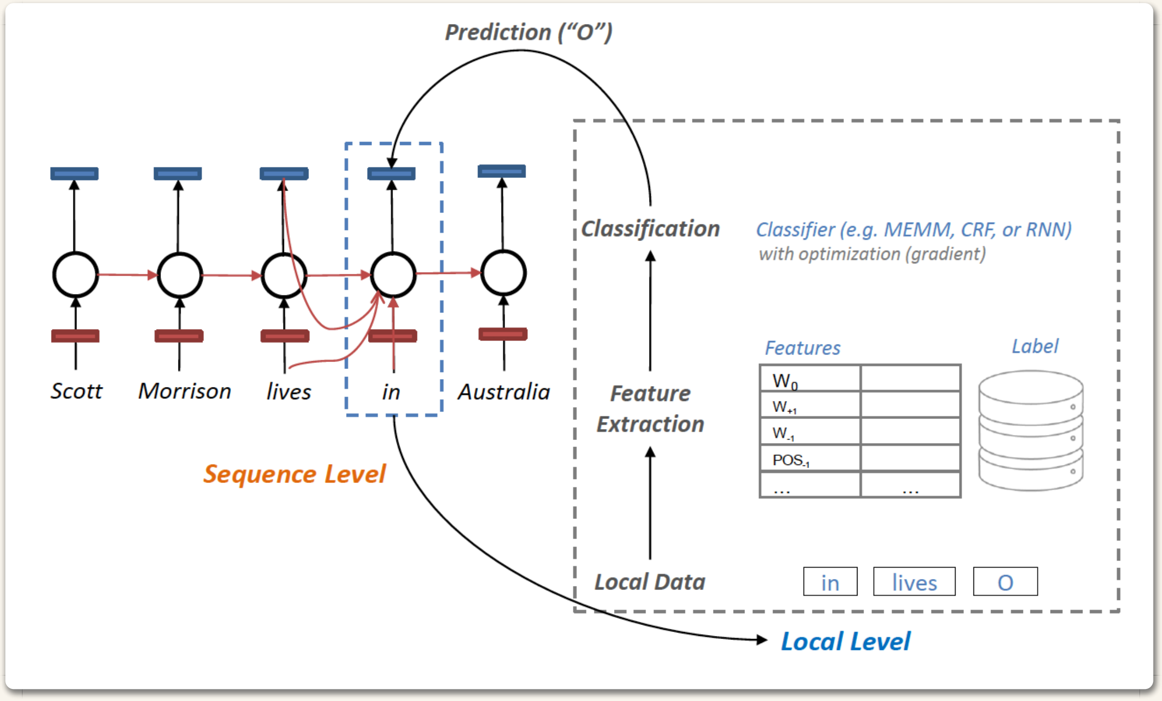
— no extra memory requirements, fast

— easy to implementation

— rich features including observations tot he right, may perform quite well

- Disadvantages

— Greedy, cannot recover from errors we make commit



- Beam Inference
  - Concept

— at each step keep top k complet sequence

— extend each sequence in each local way

— The extensions compete for the k slots at the next position

- Advantages

— Fast, beam-size 3-5

— Easy to implement (no dynamic programming required)

- Disadvantages

  — Inexact: globally best sequence can fall off the beam

- Viterbi Inference
  - Concept

    — Dynamic programming or memorisation

    — Requires small window of state influence (e.g past 2 states are relevant)
  - Advantage

    — Exact: the global best sequence is returned
  - Disadvantage

    — hard to implement long-distance step-state interactions

Corefernce Resolution

1. NER: task of poducing a list of entities in a text (How to trance?)

2. Coreference Resolution: finding expressions refer to the same entity in a text

- What is CR?
  - All mentions that refer to the same entities

- How conduct?
  - Detect mentions (= span of text refering to same entity)

    — Pronouns

    — Named Entities

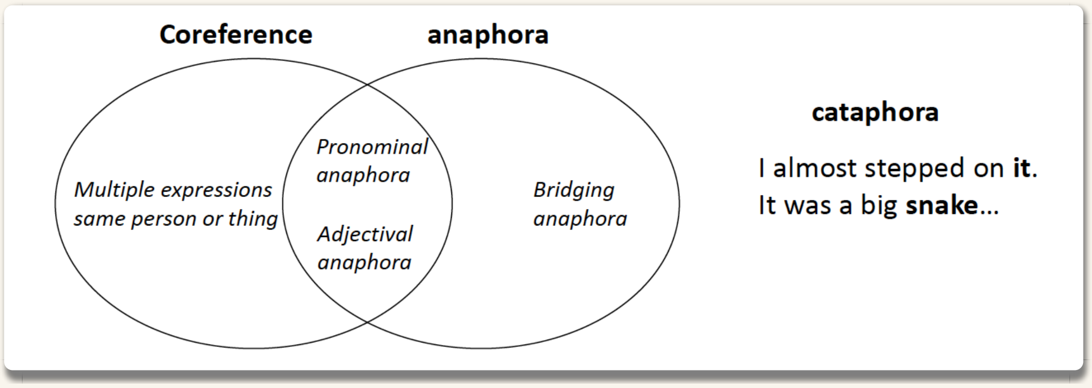
    — Noun phrases

    — Tricky mentions ==> classifier (e.g 'no staff', 'the best university in Australia')

- Cluster the mentions
  - — Coreference
    - occures when two or more expressions in a text refer to the same person or thing
  - — Anaphora
    - the use of a word referring back to a word used earlier in a text or conversation, usually nouns phrases
      - — a word(anaphor) referes to another word (antecedent)
  - — Not all anaphoric relations are coreferential
    - Not all NP have reference

  e.g Every student like his speech/No student like his speech
    - Not all anaphoric relations are co-referential (bridging anaphora)

      e.g I attended the meeting yesterday. The presentation was awesome



Coreference Model

Train a binary classifier that assigns every pair of mentions a probability of being coreferent: $p(m_i, m_j)$

Mention Pair training

1. N mentions in a document
2. y_{ij}=1 if mentions m_i and m_j are coreferent, -1 if otherwise
3. just train with regular cross-entropy loss (looks a bit differnet because it is binary classification)

$$J = -\sum_{i=2}^{N}\sum_{j=1}^{i} y_{ij} \log p(m_j, m_i)$$

Coreferent mentions pairs should get high probability, others should get low probability

Iterate through mentions

Iterate through candidate antecedents (previously occurring mentions)

4. Clustering based on pair score

   — pick uo some treshold (e.g., 0.5) and add coreference links mention pairs where $p(m_i, m_j)$ is above the threshold

   — take the transitive closure to get the clustering

5. Mention pair testing: Issue
   - Many mentions only have one clear antecedent, but are asking the model to predict all of them
   - Mention ranking: instead of predicting only one antecedent for each mention

Mention ranking

- Building calculation

  — Assign each mention its highest scoring candidate antecedent according to the model

  — Dummy NA mention allows model to decline linking the current mention to anything ("singleton" or "first" mention)
- Training

  — The current mention m_j should be linked to any one of the candidate antecedents it's corefernt with

  — maximize this probability: $\sum_{j=1}^{i-1} 1(y_{ij} = 1)p(m_j, m_i)$

$$\sum_{j=1}^{i-1} \mathbb{1}(y_{ij} = 1)p(m_j, m_i)$$

Iterate through candidate antecedents (previously occurring mentions)

For ones that are coreferent to m_j...

...we want the model to assign a high probability

- Test time

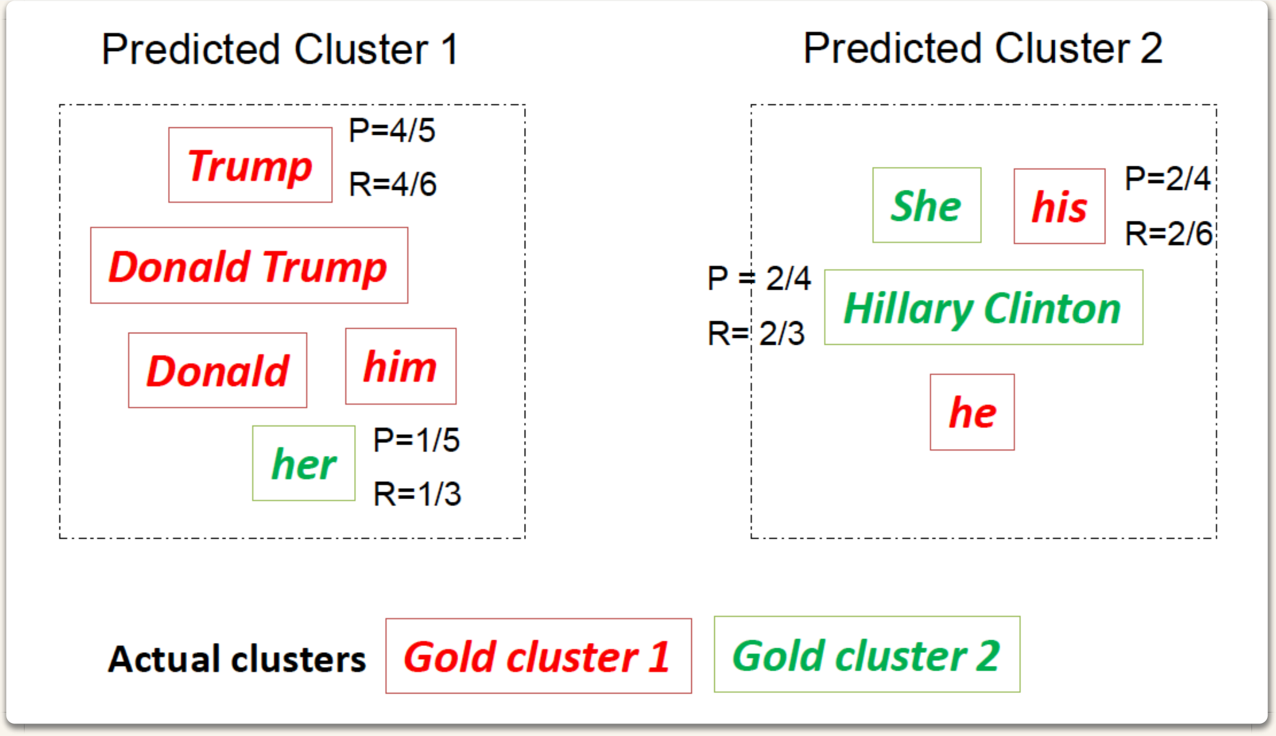
— similar to mention-pair model but each mention is assigned only one antecedent

— computation probabilities

  - Non-neural statistical classifier
  - Simple neural network
  - LSTMs,attention

Coreference Evaluation

- B-CUBED metrics
  - compute precision and recall for each mention
  - Average the individual Ps and Rs



Predicted Cluster 1
Trump
P=4/5
R=4/6
Donald Trump
Donald
him
her
P=1/5
R=1/3
Predicted Cluster 2
She
his
P=2/4
R=2/6
P = 2/4
R= 2/3
Hillary Clinton
he
Actual clusters
Gold cluster 1
Gold cluster 2