ΔΙΑΧΕΙΡΙΣΗ ΠΕΡΙΕΧΟΜΕΝΟΥ ΠΑΓΚΟΣΜΙΟΥ ΙΣΤΟΥ ΚΑΙ ΓΛΩΣΣΙΚΑ ΕΡΓΑΛΕΙΑ

Information Extraction

Information Extraction

- Μορφή της πληροφορίας
  - Δομημένα δεδομένα
    - Relational Databases (SQL)
    - XML markup
  - Μη-δομημένα δεδομένα
    - Ελεύθερο κείμενο
    - Multimedia (images/audio/video)
  - Ημι-δομημένα δεδομένα
    - Πίνακες μέσα στο κείμενο
    - Σχόλια
- Information Extraction:
  - Το πρόβλημα της μετατροπής της πληροφορίας σε δομημένα δεδομένα

Information Extraction – Παράδειγμα

- Μη δομημένη μορφή

  «Η Μαρία σπούδασε στην Ιατρική του Πανεπιστημίου Πατρών. Ο Γιώργος σπούδασε στη Νομική του ΑΠΘ. Εκεί σπούδασε και η Χριστίνα.»

- Μετατροπή σε δομημένη
  - Αναγνώριση οντοτήτων
    - Με επίλυση αναφορών
  - Αναγνώριση σχέσεων
  - Αναπαράσταση σε δομημένη πληροφορία
    - XML
    - Relational

Information Extraction – Παράδειγμα

- XML
  - <info>
  - <university name='Πανεπιστήμιο Πατρών'>
  - <department name='Ιατρική'>
  - <student='Μαρία'/>
  - </department>
  - </university>
  - <university name='ΑΠΘ'>
  - <department name='Νομική'>
  - <student='Γιώργος'/>
  - <student='Χριστίνα'/>
  - </department>
  - </university>
  - </info>
- Relational
  - σπούδασε(Μαρία, Ιατρική)
  - ανήκει(Ιατρική, Πανεπιστήμιο Πατρών)
  - σπούδασε(Γιώργος, Νομική)
  - ανήκει(Νομική, ΑΠΘ)
  - σπούδασε(Χριστίνα, Νομική)

Εφαρμογή στο Διαδίκτυο

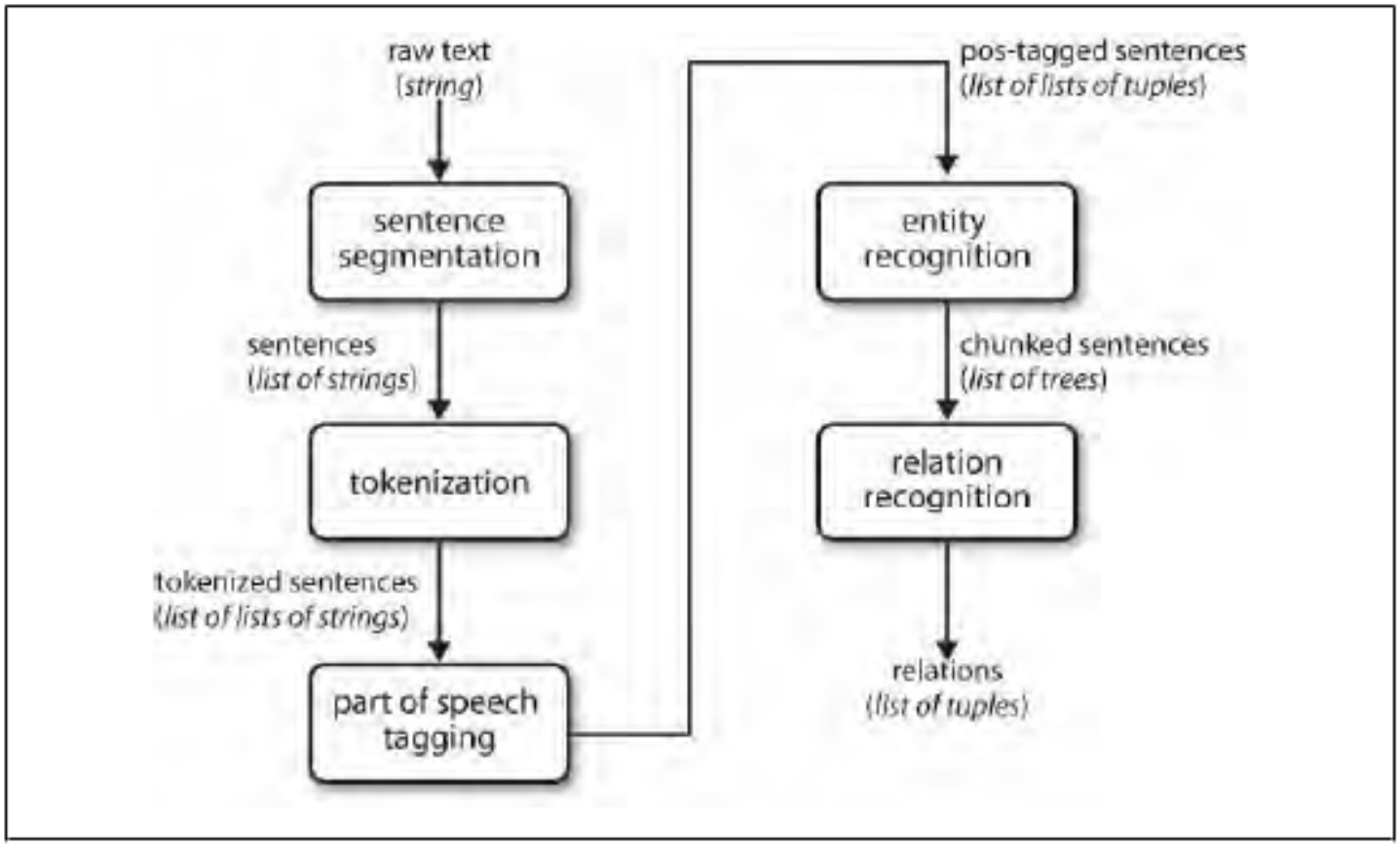
- Διαδίκτυο: εκρηκτική αύξηση της μη- δομημένης πληροφορίας
- Για να εκμεταλλευτούμε την πληροφορία πλήρως πρέπει να μετατραπεί σε δομημένη.
  - Tim Berners-Lee:
    - Transformation of “Web of Documents” to “Web of Data”.
- Απαιτήσεις για το web:
  - Χαμηλό κόστος
  - Προσαρμοστικότητα σε πολλά πεδία
  - Ευκολία ανάπτυξης
  - Χρήση του HTML/XML markup

Προσεγγίσεις

- Wrappers:
  - Σύνολα κανόνων μεγάλης ακρίβειας
  - Εξάγουν συγκεκριμένη πληροφορία από σελίδες
  - Εφαρμόζονται σε σελίδες με πολύ συγκεκριμένη δομή
  - Αποτυγχάνουν σε πιο ελεύθερα δομημένη πληροφορία
- Adaptive Information Extraction:
  - Συστήματα που εφαρμόζονται σε διαφορετικά δομημένη πληροφορία
  - Εφαρμόζονται τεχνικές NLP

Information Extraction Architecture

- Είσοδος: ελεύθερο κείμενο
- Έξοδος: σχέσεις μεταξύ οντοτήτων



Προεπεξεργασία

- Βήμα 1: Sentence Segmentations
- Βήμα 2: Tokenization
- Βήμα 3: Tagging

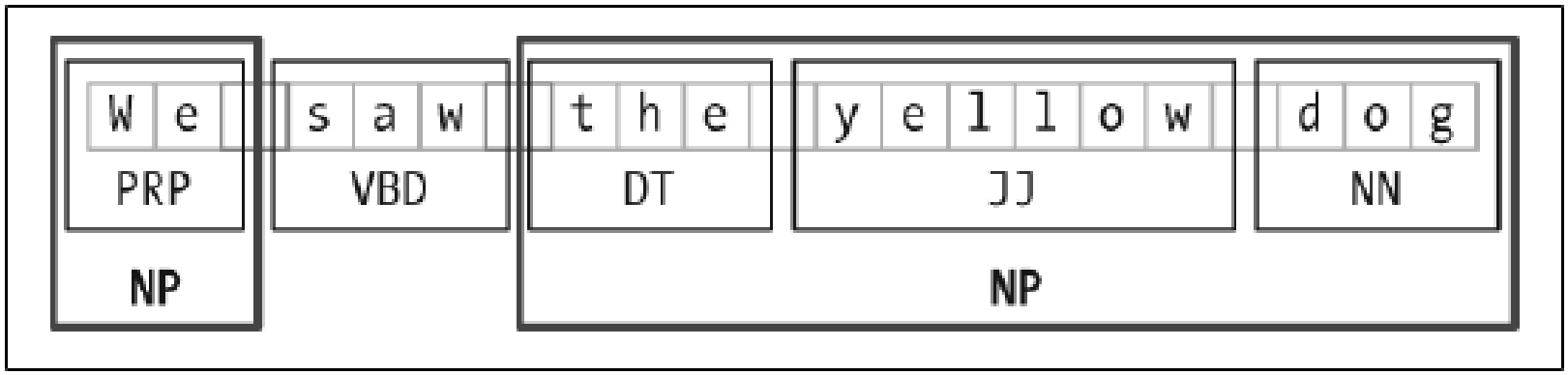
```
>>> import nltk
>>> raw='The Fulton County Grand Jury said Friday an investigation of Atlanta\'s
 recent primary election produced no evidence that any irregularities took place
 . The jury further said in term-end presentments that the City Executive Commit
tee , which had over-all charge of the election , deserves the praise and thanks
 of the City of Atlanta for the manner in which the election was conducted .'
>>> sentences=nltk.sent_tokenize(raw)
>>> sentences[0]
"The Fulton County Grand Jury said Friday an investigation of Atlanta's recent p
rimary election produced no evidence that any irregularities took place ."
>>> tokenized_sents=[nltk.word_tokenize(s) for s in sentences]
>>> tokenized_sents[0]
['The', 'Fulton', 'County', 'Grand', 'Jury', 'said', 'Friday', 'an', 'investigat
ion', 'of', 'Atlanta', "'s", 'recent', 'primary', 'election', 'produced', 'no',
'evidence', 'that', 'any', 'irregularities', 'took', 'place', '.']
>>> tagged_sents=[nltk.pos_tag(s) for s in tokenized_sents]
>>> tagged_sents[0]
[('The', 'DT'), ('Fulton', 'NNP'), ('County', 'NNP'), ('Grand', 'NNP'), ('Jury',
 'NNP'), ('said', 'VBD'), ('Friday', 'NNP'), ('an', 'DT'), ('investigation', 'NN
'), ('of', 'IN'), ('Atlanta', 'NNP'), ("'s", 'POS'), ('recent', 'JJ'), ('primary
', 'JJ'), ('election', 'NN'), ('produced', 'VBN'), ('no', 'DT'), ('evidence', 'N
N'), ('that', 'IN'), ('any', 'DT'), ('irregularities', 'NNS'), ('took', 'VBD'),
('place', 'NN'), ('.', '.')]
```

Entity Recognition

- Βήμα 4: Αναγνώριση οντοτήτων
  - Οντότητες που εκφράζονται από:
    - Κύρια ονόματα
      - **Ο Γιάννης** πήγε στην παραλία.
    - Ονοματικές φράσεις
      - Ο Γιάννης πήγε **στην παραλία.**
    - Αναφορές ή συνώνυμα με προηγούμενη αναφορά στις οντότητες
      - **Εκεί** ήπιε καφέ.

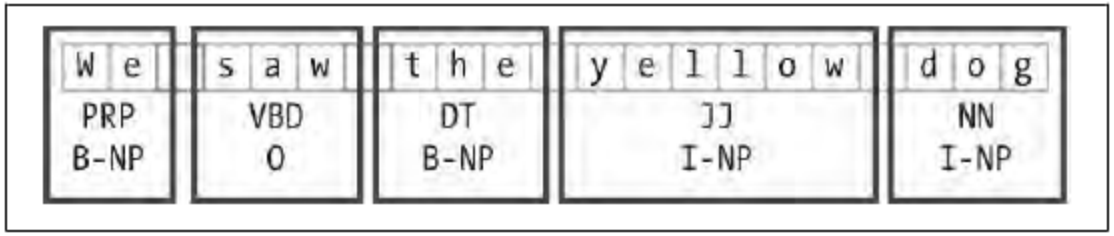
Entity Recognition - Chunking

- Η κύρια τεχνική αναγνώρισης οντοτήτων
- Ανάθεση ακολουθιών από tokens σε συντακτικές κατηγορίες πχ
  - Noun Phrases (Ονοματικές Φράσεις)
  - Verbal Phrases (Ρηματικές Φράσεις)
- Χρησιμοποιούμε κυρίως ονοματικές φράσεις

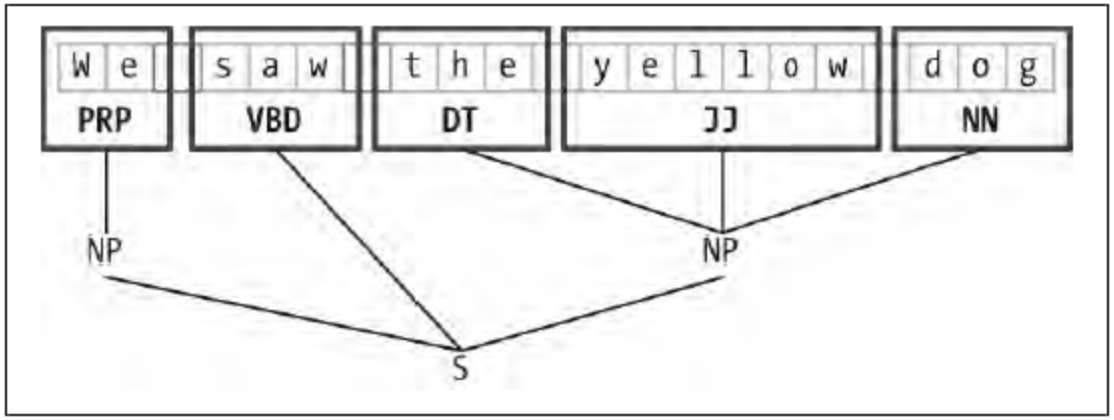


Chunking - Αναπαράσταση

- Διαφορετικές μορφές αναπαράστασης chunks
- Ανάλογα με την αναπαράσταση των εμφωλευμένων στοιχείων:
  - IOB tags: Input/Output/Begin tag σε κάθε chunk



  - Δέντρο



Chunking - Διαδικασία

- Δημιουργία ενός NP-chunker:
  - Ορισμός της γραμματικής για το chunking:
    - Κανόνες για το πως οι προτάσεις μπορούν να γίνουν chunked
  - Οι κανόνες ορίζουν tag patterns
    - Με σύνταξη παρόμοια με regular expressions, ορίζουν ακολουθίες από Part-Of-Speech tags.
      - πχ NP: {<DT|PP\$>?<JJ>*<NN>}
        {<NNP>+}

Chunking – Παράδειγμα

```
>>> grammar=r"""
...     NP: {<DT|PP\$>?<JJ>*<NN>}
...         {<NNP>+}
... """
>>> cp=nltk.RegexpParser(grammar)
>>> tagged_sents[0][0:10]
[('The', 'DT'), ('Fulton', 'NNP'), ('County', 'NNP'), ('Grand', 'NNP'), ('Jury',
'NNP'), ('said', 'VBD'), ('Friday', 'NNP'), ('an', 'DT'), ('investigation', 'NN')
, ('of', 'IN')]
>>> print cp.parse(tagged_sents[0])
(S
  The/DT
  (NP Fulton/NNP County/NNP Grand/NNP Jury/NNP)
  said/VBD
  (NP Friday/NNP)
  (NP an/DT investigation/NN)
  of/IN
  (NP Atlanta/NNP)
  's/POS
  (NP recent/JJ primary/JJ election/NN)
  produced/VBN
  (NP no/DT evidence/NN)
  that/IN
  any/DT
  irregularities/NNS
  took/VBD
  (NP place/NN)
  ./.)
>>>
```

Chunking – Classifier based chunkers

- Αρκεί μόνο το POS για την αναγνώριση των chunks?
  - Joey sold the farmer rice.
  - Nick broke my computer monitor.
  - Ανάγκη για εκμετάλλευση και της λέξης εκτός από το POS
- Εκπαίδευση ενός classifier (ανάλογα με τη δημιουργία του POS tagger)
  - Training set:
    - Ένα corpus με σημειωμένα τα IOB-tags
    - πχ στο NLTK το Wall Street Journal
  - Features:
    - POS της τρέχουσας λέξης
    - Η ίδια η τρέχουσα λέξη
    - POS από τις προηγούμενες
    - Οι προηγούμενες/επόμενες λέξεις

Chunking with NLTK

- Το NLTK έχει classifier-based chunker εκπαιδευμένο ήδη για να αναγνωρίζει Named Entities.

```
>>> sent=nltk.corpus.treebank.tagged_sents()[22]
>>> print nltk.ne_chunk(sent, binary=True)
(S
  The/DT
  (NE U.S./NNP)
  is/VBZ
  one/CD
  of/IN
  the/DT
  few/JJ
```

```
>>> print nltk.ne_chunk(sent)
(S
  The/DT
  (GPE U.S./NNP)
  is/VBZ
  one/CD
  of/IN
  the/DT
  few/JJ
```

Relation Extraction

- Βήμα 5: Αναγνώριση σχέσεων
  - Με δεδομένες τις οντότητες
  - Αναγνώριση τριάδων (X,α,Y) όπου X, Y οντότητες και α ακολουθία που εκφράζει τη σχέση.
  - Βασική προσέγγιση:
  - Επιλογή με regular expressions των λέξεων που εκφράζουν τη σχέση από τα tokens μεταξύ των οντοτήτων.

Relation Extraction – Παράδειγμα

```
>>> import re
>>>
>>> IN=re.compile(r'.*\bin\b(?!\b.+ing)')
>>> for doc in nltk.corpus.ieer.parsed_docs('NYT_19980315'):
...     for rel in nltk.sem.extract_rels('ORG','LOC',doc,corpus='ieer',pattern=IN
):
...             print nltk.sem.show_raw_rtuple(rel)
...
[ORG: 'WHYY'] 'in' [LOC: 'Philadelphia']
[ORG: 'McGlashan &AMP; Sarrail'] 'firm in' [LOC: 'San Mateo']
[ORG: 'Freedom Forum'] 'in' [LOC: 'Arlington']
[ORG: 'Brookings Institution'] ', the research group in' [LOC: 'Washington']
[ORG: 'Idealab'] ', a self-described business incubator based in' [LOC: 'Los Ange
les']
[ORG: 'Open Text'] ', based in' [LOC: 'Waterloo']
[ORG: 'WGBH'] 'in' [LOC: 'Boston']
[ORG: 'Bastille Opera'] 'in' [LOC: 'Paris']
[ORG: 'Omnicom'] 'in' [LOC: 'New York']
[ORG: 'DDB Needham'] 'in' [LOC: 'New York']
[ORG: 'Kaplan Thaler Group'] 'in' [LOC: 'New York']
[ORG: 'BBDO South'] 'in' [LOC: 'Atlanta']
[ORG: 'Georgia-Pacific'] 'in' [LOC: 'Atlanta']
>>>
```

Recommended Reading

- "Natural Language Processing with Python"
  - Chapter 7: Extracting Information from Text