CPE 372/641 Natural Language Processing

Applications in NLP (using Python and related resources)

Asst. Prof. Dr. Nuttanart Muansuwan

Recap

- Now we passed the mid semester. What we have learned up to midterm covers the theoretical aspects of NLP.
  - Starting from words and components of words (morphology)
  - Tools for morphological analysis: RE, Finite State Machines
  - Part-of-speech tagging, n-gram
  - Syntax, grammar, sentence structure
  - Semantic representations
  - Lexical semantics
  - Word vectors
  - Parser

What else?

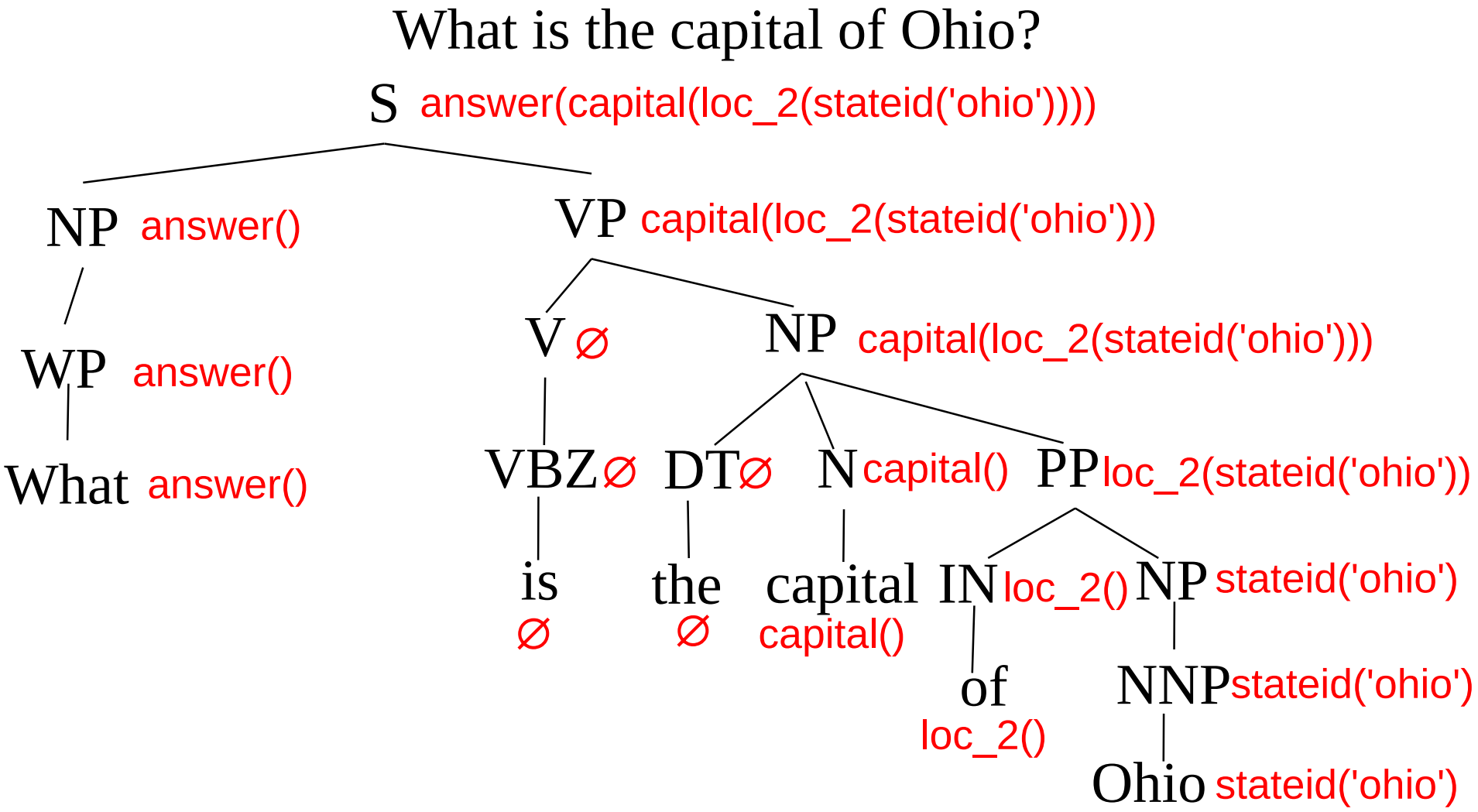
- We haven’t seen how to match the sentence structure to meaning. This could benefit question answering. So we will look at this issue more when we study about question answering systems.

Compositional Semantics

- Approach to semantic analysis based on building up a meaning representation (MR) compositionally based on the syntactic structure of a sentence.
- Build MR recursively bottom-up from the parse tree.

```
BuildMR(parse-tree)
    If parse-tree is a terminal node (word) then
        return an atomic lexical meaning for the word.
    Else
        For each child, subtree_i, of parse-tree
            Create its MR by calling BuildMR(subtree_i)
        Return an MR by properly combining the resulting MRs
        for its children into an MR for the overall parse-tree.
```

Composing MRs from Parse Trees



Leave that for now

What do we learn next?

Objectives

After this lesson, students should be able to:

1. Know the overview of applying what we have learned to NLP tasks using Python and other resources such as NLTK, textblob, etc.
2. Be able to code for simple text classification (homework 3)
3. Have some ideas and understanding of techniques to apply to your own projects

First step

- If you have not already done so, please download and install NLTK and its data

Text preprocessing

- Since, text is the most unstructured form of data, various types of noise are present in it and the data is not readily analyzable without any pre-processing.
- The entire process of cleaning and standardization of text, making it noise-free and ready for analysis is known as text preprocessing.

Text Processing Pipeline architecture

After preprocessing,

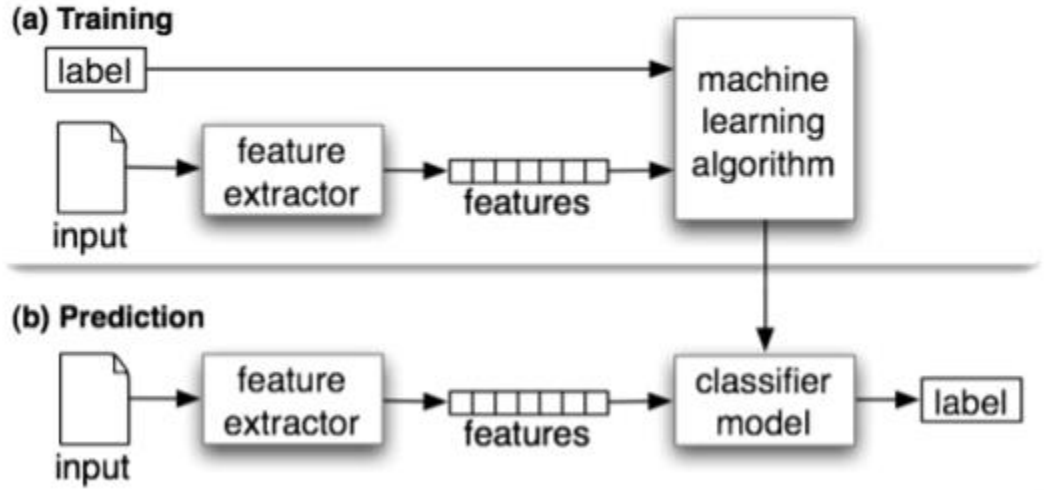
- We will process texts to acquire features of some sorts by using these techniques: part-of-speech tagging, syntactic parsing, semantic processing, n-gram, statistical features (TF-IDF), word embedding (Word2Vec). These are the techniques that we have learned.

Important NLP tasks
I. Text Classification

- Text classification is one of the classical problem of NLP. Examples include – Email Spam Identification, topic classification of news, sentiment classification and organization of web pages by search engines.
- Text classification is defined as a technique to systematically classify a text object (document or sentence) in one of the fixed categories. It is really helpful when the amount of data is too large, especially for organizing, information filtering, and storage purposes.

Natural language classifier

- Training
- Prediction



II. Text matching or text similarity

- Text matching is used for auto spelling correction, auto complete of search queries, for example
- You can use Minimum edit distance or **Levenshtein Distance**, cosine similarity or Soundex algorithm for this task
- For example: Soundex for names: *Michelle, Michael*

Other NLP tasks

- **Text Summarization** – Given a text article or paragraph, summarize it automatically to produce most important and relevant sentences in order.
- **Machine Translation** – Automatically translate text from one human language to another by taking care of grammar, semantics and information about the real world, etc.
- **Natural Language Generation and Understanding** – Convert information from computer databases or semantic intents into readable human language is called language generation. Converting chunks of text into more logical structures that are easier for computer programs to manipulate is called language understanding.
- **Information Extraction** – This involves parsing of textual data present in documents (websites, files, pdfs and images) to analyzable and clean format.

Important Libraries for NLP (Python)

- Scikit-learn: Machine learning in Python
- Natural Language Toolkit (NLTK): The complete toolkit for all NLP techniques.
- Pattern – A web mining module for the with tools for NLP and machine learning.
- TextBlob – Easy to use NLP tools API, built on top of NLTK and Pattern.
- spaCy – Industrial strength NLP with Python and Cython.
- Gensim – Topic Modelling for Humans
- Stanford Core NLP – NLP services and packages by Stanford NLP Group, Stanford University

For homework: Naïve Bayes Classifier

- https://medium.com/syncedreview/applying-multinomial-naive-bayes-to-nlp-problems-a-practical-explanation-4f5271768ebf