NLP tasks for Data Science

CMPT 733
Steven Bergner
Suraj Swaroop (coop in Summer 2020)

What is NLP?

Natural Language

how humans communicate with each other via **speech and text**

Processing

- branch of AI to read, decipher, and make sense of human language
- Applications: information extraction, translation, personal assistants, word processors, spam detection, …

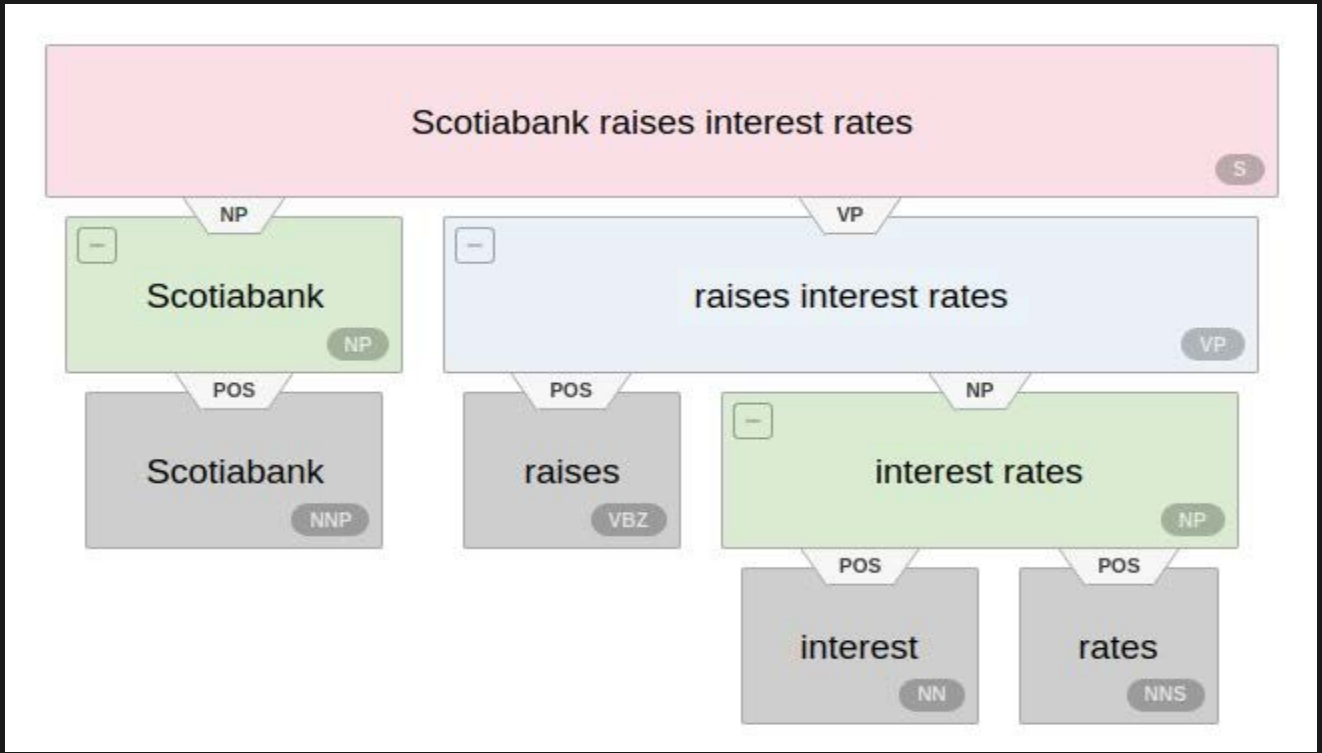
Techniques for NLP

Text Parsing

- Analyzing sentence structure and representing it according to syntactic formalism

- Two views of syntactic structure
  - Constituency
  - Dependency

Constituency Structure Example

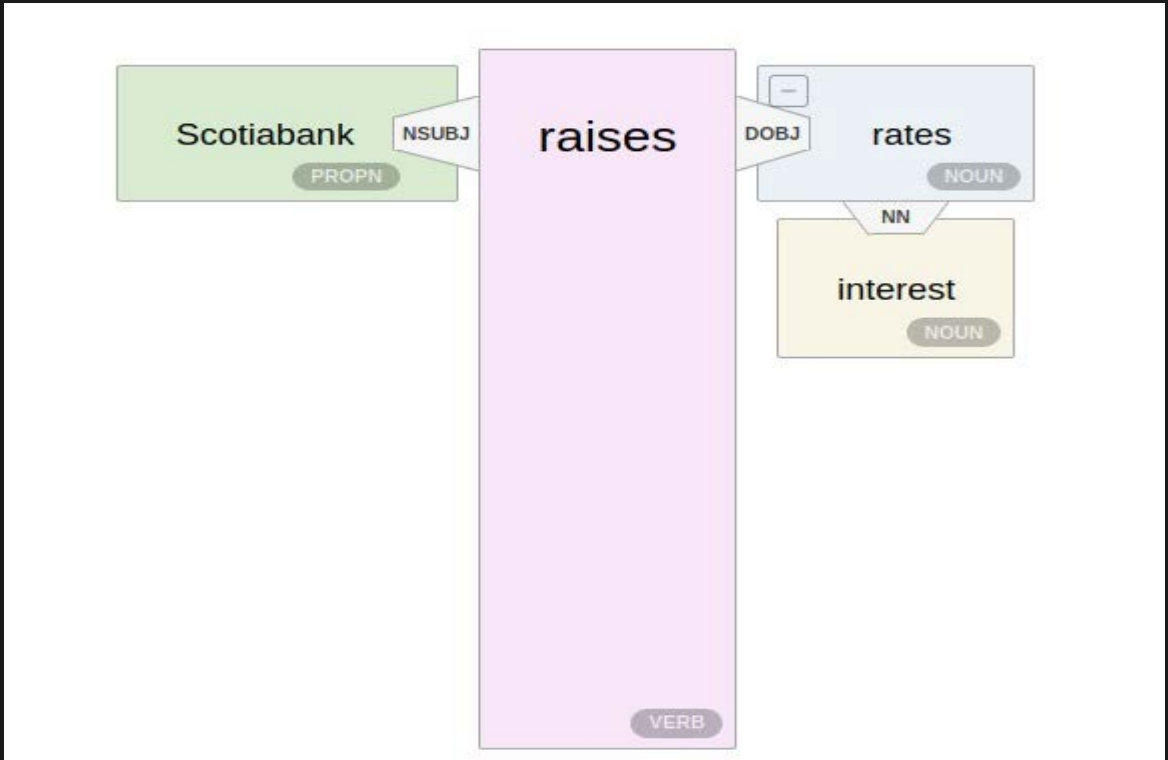


Constituency Parsing Implementation

```
[7] from allennlp.predictors.predictor import Predictor
    import allennlp_models.structured_prediction
    predictor = Predictor.from_path("https://storage.googleapis.com/allennlp-public-models/elmo-constituency-parser-2020.02.10.tar.gz")
    x = predictor.predict(sentence="Scotiabank raises interest rates")
    print(x['trees'])

    (S (NP (NNP Scotiabank)) (VP (VBZ raises) (NP (NN interest) (NNS rates))))
```

Dependency Structure Example

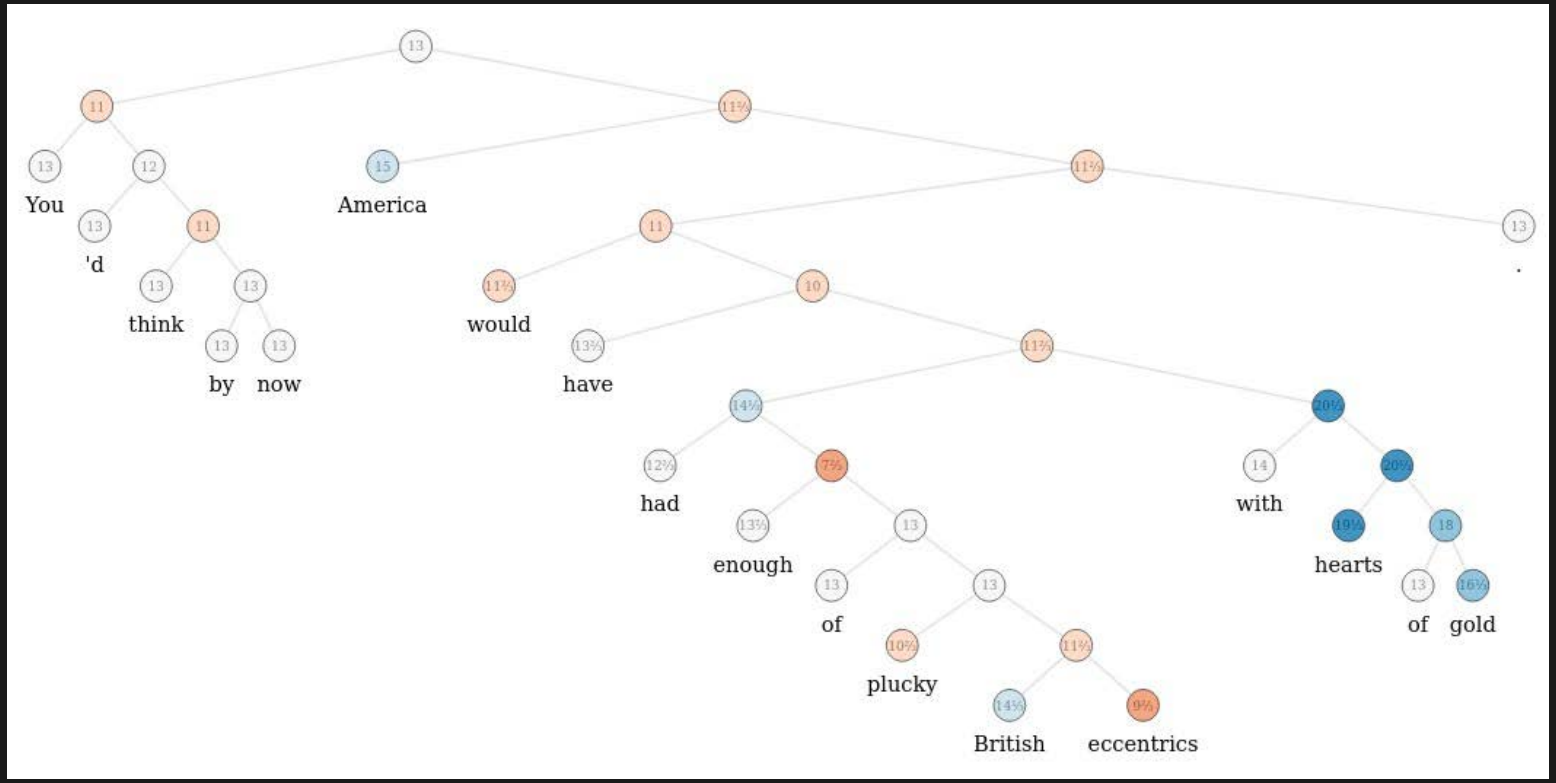


Dependency Parsing Implementation

```
from allennlp.predictors.predictor import Predictor
import allennlp_models.structured_prediction
predictor = Predictor.from_path("https://storage.googleapis.com/allennlp-public-models/biaffine-dependency-parser-ptb-2020.04.06
x = predictor.predict(sentence="Scotiabank raises interest rates")
print(x['hierplane_tree'])
```

```
{'text': 'Scotiabank raises interest rates', 'root': {'word': 'raises', 'nodeType': 'root', 'attributes': ['VERB'], 'link': 'root', 'spans':
```

Tree Example



Information Extraction

- Automatic extraction of structured and unstructured information

- Various modules
  - POS Tagging
  - Entity Recognition
  - Relation extraction
  - Sentiment Analysis

Named Entity Recognition

- Classify named entities into categories
- NER Techniques
  - Lexicon approach
  - Rule-based systems
  - ML based systems
  - Hybrid approach

NER Implementation

```
[17] from allennlp.predictors.predictor import Predictor
     import allennlp_models.tagging
     predictor = Predictor.from_path("https://storage.googleapis.com/allennlp-public-models/ner-model-2020.02.10.tar.gz")
     x= predictor.predict(sentence="Barack Obama went to Paris")
```

```
[18] print(x['words'])
     print(x['tags'])
```

```
['Barack', 'Obama', 'went', 'to', 'Paris']
['B-PER', 'L-PER', 'O', 'O', 'U-LOC']
```

Sentiment Analysis

- Determine if an opinion is positive, negative or neutral

- Techniques for Sentiment Analysis
  - Lexical Methods
  - Machine Learning methods

Sentiment Analysis Implementation

```
[8] import nltk
    nltk.download('vader_lexicon')
    from nltk.sentiment.vader import SentimentIntensityAnalyzer
    sid = SentimentIntensityAnalyzer()
    sid.polarity_scores("I am happy today")
```

```
{'compound': 0.5719, 'neg': 0.0, 'neu': 0.351, 'pos': 0.649}
```

Part of speech Tagging

- Tags each word with its corresponding part of speech
- Techniques of POS
  - Lexical Based Methods
  - Rule Based Method
  - Probabilistic Method
  - Deep learning models

POS Tagging Implementation

```
import nltk
from nltk import word_tokenize
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')
text = word_tokenize("He would not accept anything of value from those he was writing about")
nltk.pos_tag(text)
```

```
[('He', 'PRP'),
 ('would', 'MD'),
 ('not', 'RB'),
 ('accept', 'VB'),
 ('anything', 'NN'),
 ('of', 'IN'),
 ('value', 'NN'),
 ('from', 'IN'),
 ('those', 'DT'),
 ('he', 'PRP'),
 ('was', 'VBD'),
 ('writing', 'VBG'),
 ('about', 'IN')]
```

Semantic Role Labeling (SRL)

- Assigning labels to words or phrases in a sentence to indicate it's semantic role

- How it works:
  - Predicate identification
  - Predicate disambiguation
  - Argument identification
  - Argument classification

For Example

“He wouldn’t accept anything of value from those he was writing about”

The annotations of semantic roles for this sentence:

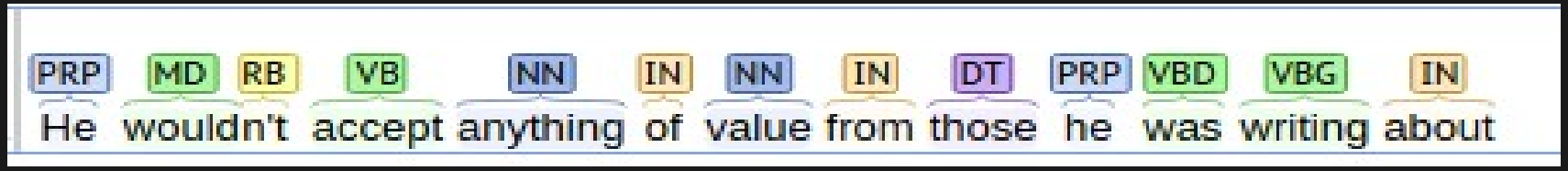
[$_{A0}$ He] [$_{AM\text{-}MOD}$ would] [$_{AM\text{-}NEG}$ n't] [$_{V}$ accept] [$_{A1}$ anything of value] from [$_{A2}$ those he was writing about] .

V: verb; A0: acceptor; A1: thing accepted; A2: accepted-from; A3: attribute;
AM-MOD: modal; AM-NEG: negation

Difference between POS and SRL

Sentence: “He wouldn’t accept anything of value from those he was writing about”

The annotations of POS Tagging:



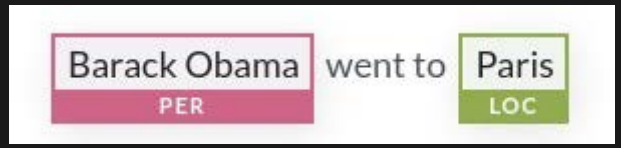
The annotations of semantic roles for this sentence:

[$_{A0}$ He] [$_{AM\text{-}MOD}$ would] [$_{AM\text{-}NEG}$ n't] [$_{V}$ accept] [$_{A1}$ anything of value] from [$_{A2}$ those he was writing about] .

NER and SRL

Sentence: “Barack Obama went to Paris “

The annotations of Entity Recognition Tagging:



The annotations of semantic roles for this sentence:

$[_{ARG0:}$ Barack Obama$]$ $[_{V:}$ went$]$ $[_{ARG4:}$ to Paris$]$

Combining ER and SRL

SA and NER

- Document-level sentiment analysis
  - Documents may have multiple topics
  - Not enough granularity

- Entity sentiment analysis identifies sentiment of each word
  - know how specific people, organizations, or things are being mentioned

Applications of SRL

- Question Answering system
- Summarization
- Information Extraction

Tools

Tools for NLP

- NLTK
- Spacy
  - Supports several different languages
- Gensim
  - Good word2vec and doc2vec implementation
- AllenNLP
- Huggingface transformers [github/demos]
  - Many state-of-the-art pre-trained models

Thank You