Natural Language Processing

NLP Application

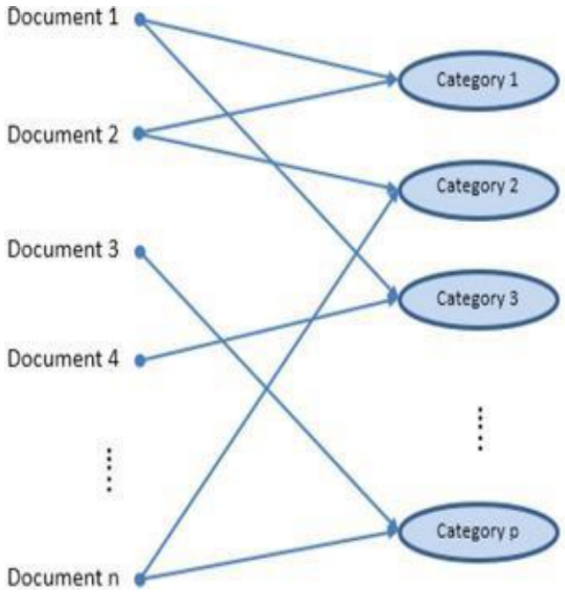
- Supervised:
    - Spam Detection
    - Sentiment Analysis
    - Intent Classification
    - Multi-Label, Multi-Class Text Classification

- Unsupervised:
    - Topic Modeling
    - Keyword Extraction
    - Trend/Outlier detection

Text Mining Applications –Supervised

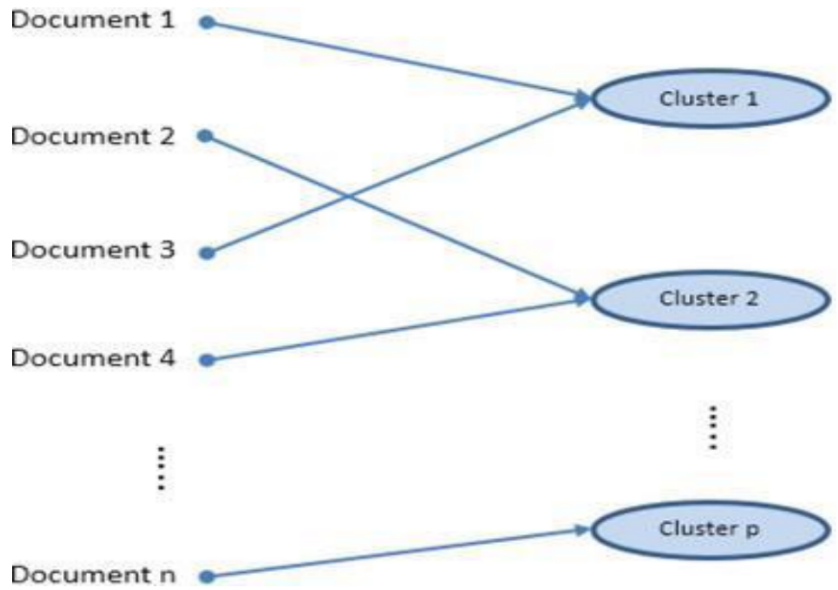
–Many typical predictive modeling or classification applications can be enhanced by incorporating textual data in addition to traditional input variables.

- churning propensity models that include customer center notes, website forms, e-mails, and Twitter messages
- hospital admission prediction models incorporating medical records notes as a new source of information
- insurance fraud modeling using adjustor notes
- sentiment categorization
- stylometry or forensic applications that identify the author of a particular writing sample



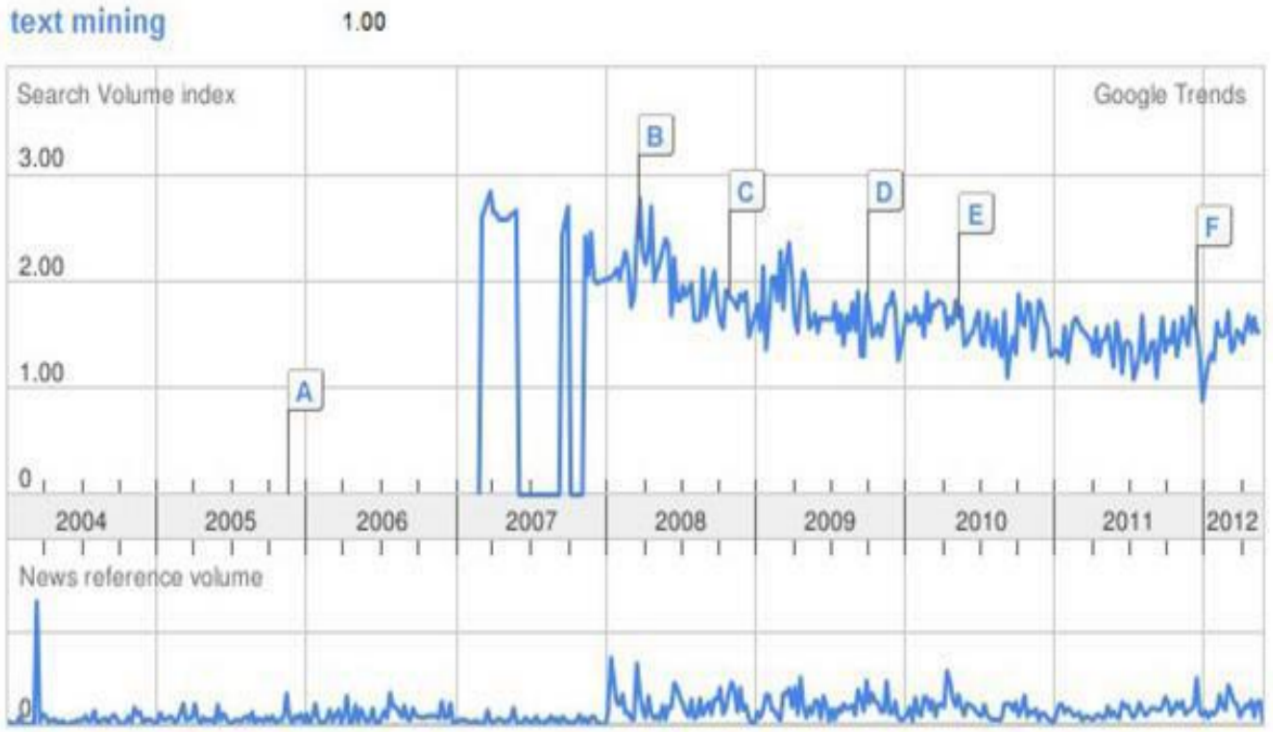
Text Mining Applications –Unsupervised

- Text clustering



Cluster No.	Comment	Key Words
1	1, 3, 4	doctor, staff, friendly, helpful
2	5, 6, 8	treatment, results, time, schedule
3	2, 7	service, clinic, fast

- Trend analysis



Trend for the Term “text mining” from Google Trends

Sentiment Analysis

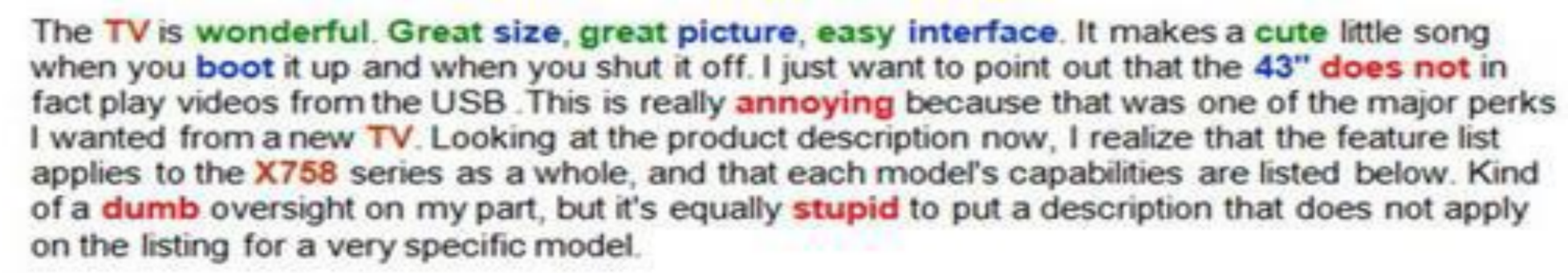
- Sentiments can be either positive or negative.
- Machine learning algorithms can be used to evaluate if a series of words reflect a positive or negative sentiment
- unsupervised or supervised?!

Sentiment Analysis (supervised vs unsupervised)

- We can have actual humans to determine and label the sentiment of our data and treat it like a text classification problem.
- You may find some human-labeled tweets data on the (data.world). Data contains over 8000 tweets that have been labeled to be positive, negative, neutral, or unknown (“I can’t tell’).
- How can we approach the same problem, if we didn’t have labels? (unsupervised learning)

Sentiment Analysis

- The field of sentiment analysis deals with categorization (or classification) of opinions expressed in textual documents.



- Green color represents positive tone, red color represents negative tone, and product features and model names are highlighted in blue and brown, respectively.

Text Data in Predictive Models

- Use of both types of data in building predictive models.
- the process involves running algorithms on the data set in which the prediction is going to take place.
  - Training the model,
  - multiple models being used on the same data set and finally arriving on the model which is the best fit based on the business data understanding.
- The predictive models' category includes predictive, descriptive, and decision models.

NLP Tasks

- NLP applications require several NLP analyses:

–Word tokenization

–Sentence boundary detection

–Part-of-speech (POS) tagging

  - to identify the part-of-speech (e.g. noun, verb) of each word

–Named Entity (NE) recognition

  - to identify proper nouns (e.g. names of person, location, organization; domain terminologies)

–Parsing

  - to identify the syntactic structure of a sentence

–Semantic analysis

  - to derive the meaning of a sentence

Tokenization

- Tokenization is essentially splitting a phrase, sentence, paragraph, or an entire text document into smaller units, such as individual words or terms. Each of these smaller units are called tokens.

Natural Language Processing

['Natural', 'Language', 'Processing']

- The tokens could be words, numbers or punctuation marks. In tokenization, smaller units are created by locating word boundaries.

Tokenization using Python's split() function

```
1  text = """Founded in 2002, SpaceX's mission is to enable humans to become a spacefaring civilization and a multi-planet
2  species by building a self-sustaining city on Mars. In 2008, SpaceX's Falcon 1 became the first privately developed
3  liquid-fuel launch vehicle to orbit the Earth."""
4  # Splits at space
5  text.split()
```

```
Output : ['Founded', 'in', '2002,', 'SpaceX's', 'mission', 'is', 'to', 'enable', 'humans',
          'to', 'become', 'a', 'spacefaring', 'civilization', 'and', 'a', 'multi-planet',
          'species', 'by', 'building', 'a', 'self-sustaining', 'city', 'on', 'Mars.', 'In',
          '2008,', 'SpaceX's', 'Falcon', '1', 'became', 'the', 'first', 'privately',
          'developed', 'liquid-fuel', 'launch', 'vehicle', 'to', 'orbit', 'the', 'Earth.']
```

```
text = """Founded in 2002, SpaceX’s mission is to enable humans to become a spacefaring civilization and a multi-planet
species by building a self-sustaining city on Mars. In 2008, SpaceX’s Falcon 1 became the first privately developed
liquid-fuel launch vehicle to orbit the Earth."""
# Splits at '.'
text.split('. ')
```

```
Output : ['Founded in 2002, SpaceX’s mission is to enable humans to become a spacefaring
          civilization and a multi-planet \nspecies by building a self-sustaining city on
          Mars',
         'In 2008, SpaceX’s Falcon 1 became the first privately developed \nliquid-fuel
          launch vehicle to orbit the Earth.']
```

Tokenization using NLTK

```
1  from nltk.tokenize import word_tokenize
2  text = """Founded in 2002, SpaceX’s mission is to enable humans to become a spacefaring civilization and a multi-planet
3  species by building a self-sustaining city on Mars. In 2008, SpaceX’s Falcon 1 became the first privately developed
4  liquid-fuel launch vehicle to orbit the Earth."""
5  word_tokenize(text)
```

```
Output: ['Founded', 'in', '2002', ',', 'SpaceX', '’', 's', 'mission', 'is', 'to', 'enable',
         'humans', 'to', 'become', 'a', 'spacefaring', 'civilization', 'and', 'a',
         'multi-planet', 'species', 'by', 'building', 'a', 'self-sustaining', 'city', 'on',
         'Mars', '.', 'In', '2008', ',', 'SpaceX', '’', 's', 'Falcon', '1', 'became',
         'the', 'first', 'privately', 'developed', 'liquid-fuel', 'launch', 'vehicle',
         'to', 'orbit', 'the', 'Earth', '.']
```

```
1  from nltk.tokenize import sent_tokenize
2  text = """Founded in 2002, SpaceX’s mission is to enable humans to become a spacefaring civilization and a multi-planet
3  species by building a self-sustaining city on Mars. In 2008, SpaceX’s Falcon 1 became the first privately developed
4  liquid-fuel launch vehicle to orbit the Earth."""
5  sent_tokenize(text)
```

```
Output: ['Founded in 2002, SpaceX’s mission is to enable humans to become a spacefaring
         civilization and a multi-planet \nspecies by building a self-sustaining city on
         Mars.',
        'In 2008, SpaceX’s Falcon 1 became the first privately developed \nliquid-fuel
         launch vehicle to orbit the Earth.']
```

Part-Of-Speech (POS) Tagging

- POS tagging is a process of assigning a POS or lexical class marker to each word in a sentence (and all sentences in a corpus).

Input: **the lead paint is unsafe**

Output: **the/Det lead/N paint/N is/V unsafe/Adj**

Named Entity Recognition (NER)

- NER is to process a text and identify named entities in a sentence
- also called entity identification or entity extraction – is a natural language processing (NLP) technique that automatically identifies named entities in a text and classifies them into predefined categories.
- Entities can be names of people, organizations, locations, times, quantities, monetary values, percentages, and more.

[ORG U.N.] official [PER Ekeus] heads for [LOC Baghdad] .

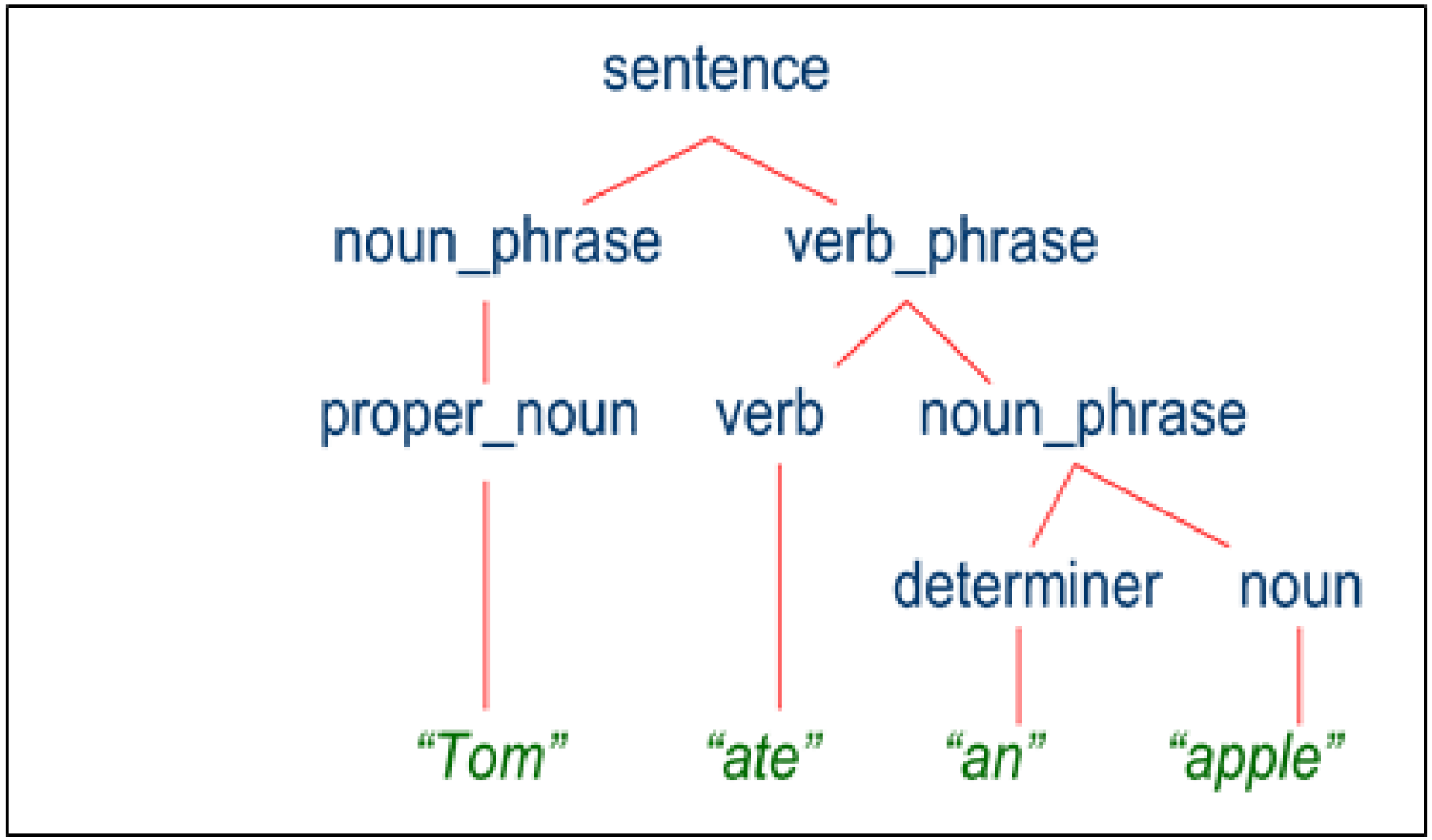
–e.g. “U.N. official Ekeus heads for Baghdad.”

Parsing

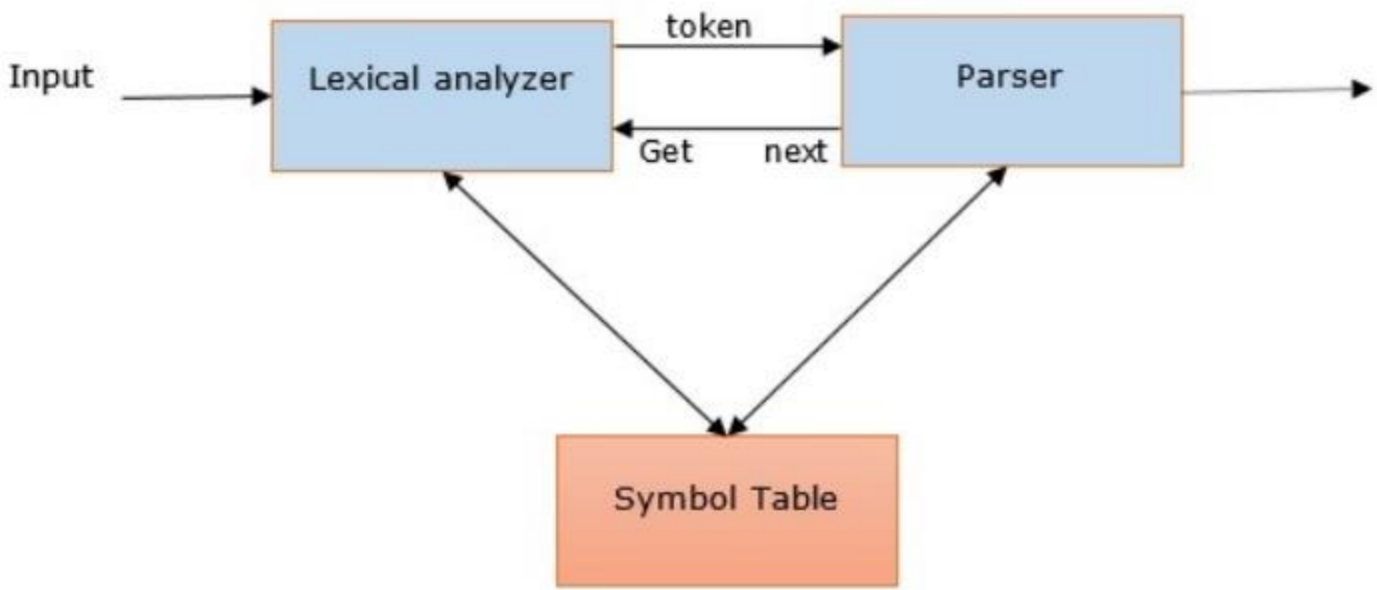
- is the process of determining the syntactic structure of a text by analyzing its constituent words based on an underlying grammar (of the language).

  - sentence -> noun_phrase, verb_phrase
  - noun_phrase -> proper_noun
  - noun_phrase -> determiner, noun
  - verb_phrase -> verb, noun_phrase
  - proper_noun -> [mary]
  - noun -> [apple]
  - verb -> [ate]
  - determiner -> [the]

Tom ate an apple



Parsing



- The purpose of lexical analyzers is to take a stream of input characters and decode them into higher level tokens that a parser can understand.
- Parsers consume the output of the lexical analyzer and operate by analyzing the sequence of tokens returned.

Parsing

- We can understand the relevance of parsing in NLP with the help of the following points:
  - Parser is used to report any syntax error.
  - It helps to recover from commonly occurring errors so that the processing of the remainder of the program can be continued.
  - Parse tree is created with the help of a parser.
  - Parser is used to create a symbol table, which plays an important role in NLP.
  - Parser is also used to produce intermediate representations (IR).

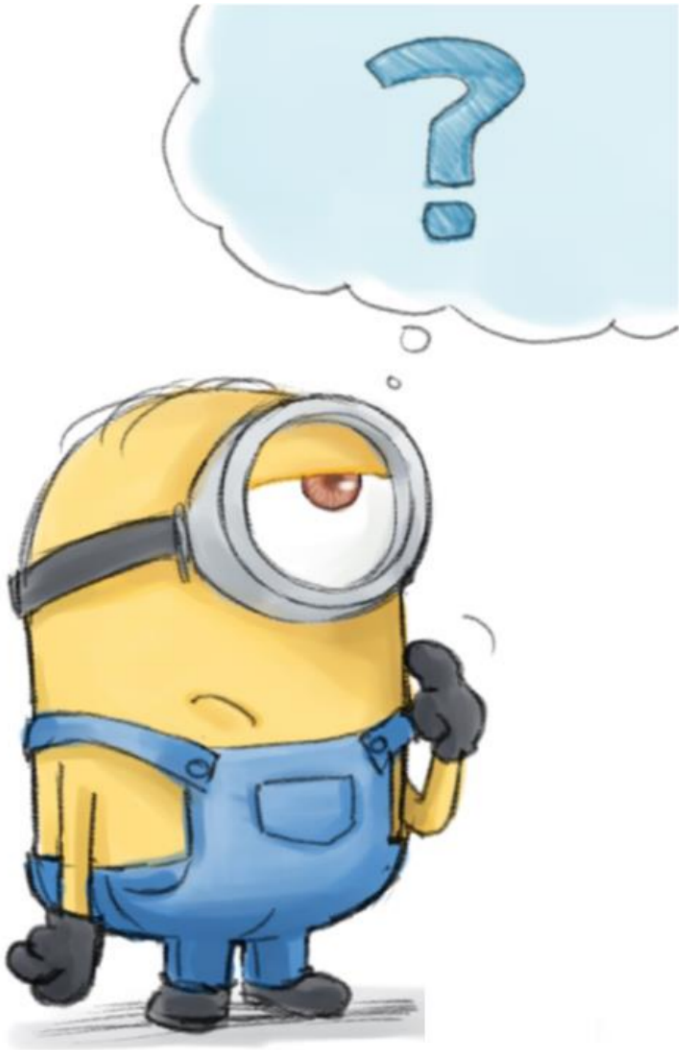
Parsing Methods

- Deep Vs Shallow Parsing

Various types of parsers

- Recursive descent parser
- Shift-reduce parser
- Chart parser
- Regexp parser
- Dependency Parsing

Why NLP is Hard?



Why NLP is Hard?

1. Ambiguity
2. Scale
3. Sparsity
4. Variation
5. Expressivity
6. Unmodeled Variables
7. Unknown representations