IEEE
STUDENT CHAPTER

THAPAR
UNIVERSITY

NATURAL LANGUAGE PROCESSING

ANSHUL
AGGARWAL

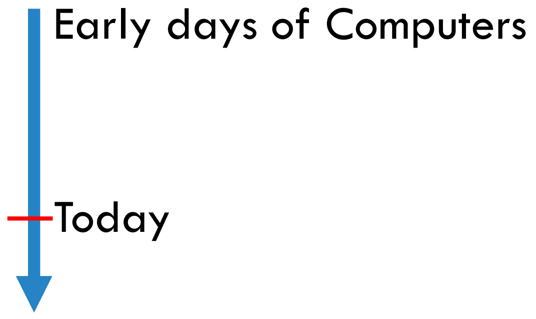
What is AI?

INTRODUCTION | 1

COMPUTER LANGUAGE VS. NATURAL LANGUAGE

- Computer Language – 0s and 1s.
- Natural Language – The one we use in daily life.

- Timeline of communication with a computer:
  - Manual binary inputs
  - Programming Languages
  - Natural Languages



NATURAL LANGUAGE COMMANDS

Two types of Assistive AIs:

- Passive
- Active

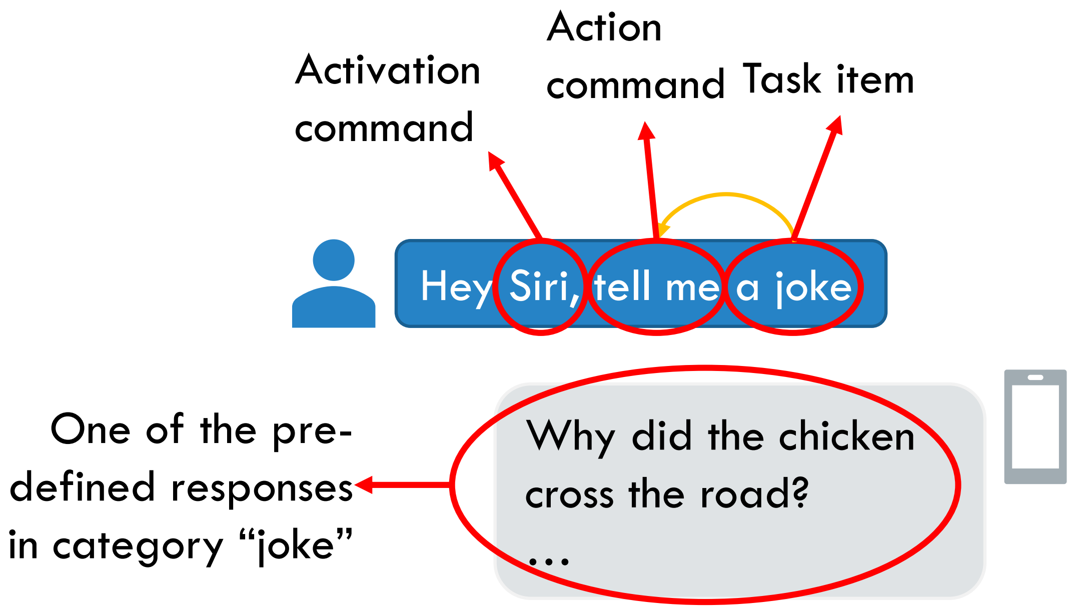
Passive AI: Lower level of interaction (“commanding”), somewhat hardcoded set of commands. Eg. Siri, Cortana, Google Assistant, chatbots.

Active AI: Conversational level of interaction (“talking”), not command based. Unachieved goal.

NLP IN PASSIVE AI

Objectives:

1. Convert speech to text, and text to speech.
2. Find keywords, and some contextual information to recognize commands.


Activation command
Action command
Task item
Hey Siri, tell me a joke
Why did the chicken cross the road?
…
One of the pre-defined responses in category "joke"

ACTIVE AI - CONVERSATIONAL

PASSIVE AI

Hey Siri, tell me a joke

Why did the chicken cross the road?
…

Hey Siri, what's funny?

Searching for "what's funny" on the internet…

ACTIVE AI

Hey Siri, what's funny?

Tell me, why did the Roman chicken cross the road?

I don't know

She was afraid someone would Caesar!

LANGUAGE PROCESSING

Human Language Task	Equivalent NLP Task
Reading	Natural Language Understanding
Writing	Natural Language Generation
Listening	Speech-to-text
Speaking	Text-to-speech

KEY TERMS | 2

MORPHEME, STEM AND LEMMA

The smallest meaningful grammatical unit of a language is called **Morpheme.**

E.g.

un · inspir · ing

un, *inspire* and *ing* are all morphemes

Stem is the basic meaningful unit of a word.

In the above example, *inspire* is the stem

Lemma is the dictionary form of the stem.

E.g.: The dictionary form of *better* is *good.*

CORPUS

Plural - *corpora*

A collection of text documents, generally related in genre, is a **corpus**.

E.g.: The Reuters News Corpus, Shakespeare Corpus, etc.

A corpus may contain additional information about the text contained, from metadata of the documents to part-of-speech tags for words.

These corpora can also be used as the training dataset for creating different Machine Learning models for various uses.

PLATFORM USED

Natural Language Toolkit (NLTK) & Python3

http://www.nltk.org/

(Alternatives: Stanford CoreNLP - Java, Apache OpenNLP – Java)

SYNTAX ANALYSIS | 3

TOKENIZATION AND SENTENCE SEGMENTATION

Tokenization: The process of segmenting text into independent words.

E.g.: The above sentence is tokenized into "The", "process", "of", "segmenting", "text", "into", "independent", "words"

This is done with the help of whitespaces. All punctuation is removed in this process.

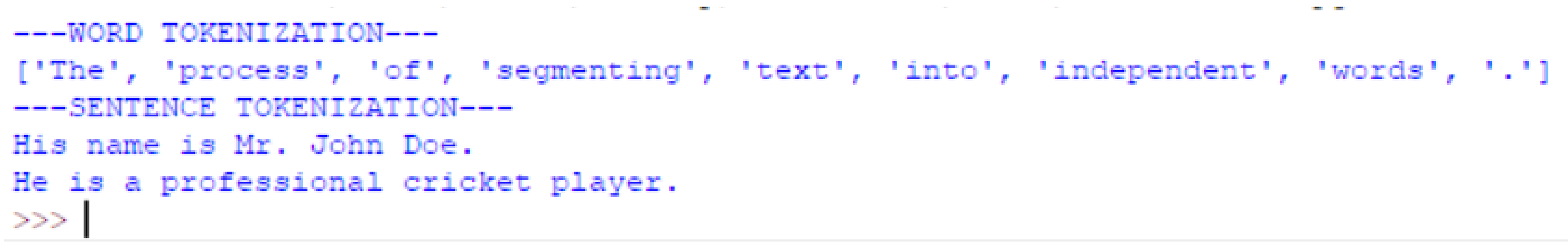
Sentence Segmentation: The process of separating individual sentences in the document. This is done with the help of punctuation marks like '?', '!', '.' which are followed by whitespaces.

E.g.: "His name is Mr. John Doe. He is a professional cricket player." is segmented into

1. His name is Mr. John Doe.
2. He is a professional cricket player.

SAMPLE CODE - 1

```
from nltk.tokenize import word_tokenize, sent_tokenize
print("---WORD TOKENIZATION---")
sentence = "The process of segmenting text into independent words."
words = word_tokenize(sentence)
print(words)
paragraph = "His name is Mr. John Doe. He is a professional cricket player."
sentences = sent_tokenize(paragraph)
print("---SENTENCE TOKENIZATION---")
for sent in sentences:
    print(sent)
```



```
---WORD TOKENIZATION---
['The', 'process', 'of', 'segmenting', 'text', 'into', 'independent', 'words', '.']
---SENTENCE TOKENIZATION---
His name is Mr. John Doe.
He is a professional cricket player.
>>> 
```

STEMMING

- Words can be viewed as consisting of:
  - A **Stem**
  - One or more Affixes – **Prefixes, Suffixes** or both
- Stemming extracts the stem from any word.
- Popular stemming algorithms: Porter Stemmer, Snowball Stemmer
- This process involves a set of rules, that are applied if they satisfy some condition. One such rule, concerning ‘s’ at the end, is:

SSES $\longrightarrow$ SS

caresses $\longrightarrow$ *caress*

IES $\longrightarrow$ I

ponies $\longrightarrow$ *poni*

SS $\longrightarrow$ SS

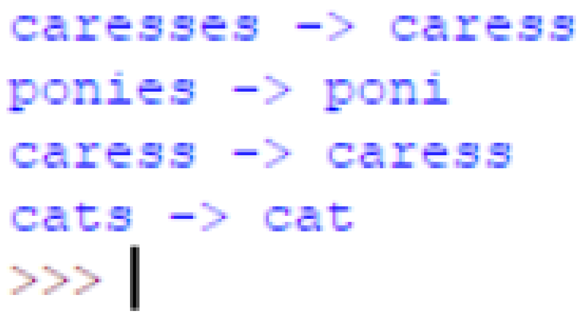
caress $\longrightarrow$ *caress*

S $\longrightarrow$ ϵ

cats $\longrightarrow$ *cat*

SAMPLE CODE - 2

```python
import nltk
from nltk.stem.snowball import SnowballStemmer
stemmer = SnowballStemmer("english")
words = ['caresses', 'ponies', 'caress','cats']
for word in words:
    print(word, "->", stemmer.stem(word))
```



```
caresses -> caress
ponies -> poni
caress -> caress
cats -> cat
>>> |
```

Fill in the blank:

Microsoft issued a new security _____

N-GRAM

- A useful part of the knowledge needed to allow Word Prediction can be captured using simple statistical techniques
- We'll rely on the notion of the probability of a sequence (a phrase, a sentence)
- N-Grams are sequences of tokens.
- The N stands for how many terms are used

  Unigram: 1 term Bigram: 2 terms Trigrams: 3 terms
- N-Grams give us some idea of the context around the token we are looking at.

PART-OF-SPEECH TAGGING

- Annotate each word in a sentence with a part-of-speech marker.
- Lowest level of syntactic analysis. E.g.:

Amit	saw	the	saw	and	decided	to	take	it	to	the	table.
NNP	VBD	DT	NN	CC	VBD	TO	VB	PRP	IN	DT	NN

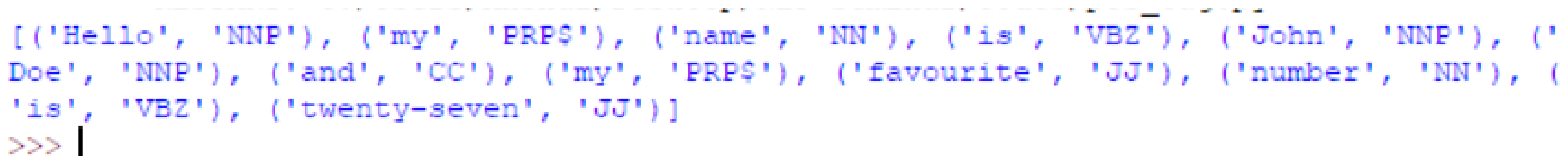
- Useful for subsequent syntactic parsing and word sense disambiguation.

POS TAGS – SOME EXAMPLES

NN	noun	chair, bandwidth, pacing
VB	verb	study, debate, munch
JJ	adjective	purple, tall, ridiculous
RB	adverb	unfortunately, slowly
IN	preposition	of, by, to
PRP	pronoun	I, me, mine
DT	determiner	the, a, that, those

SAMPLE CODE - 3

```
import nltk
from nltk import word_tokenize
txt = Hello my name is John Doe and my favourite number is 
twenty-seven"
tokens = word_tokenize(txt)
pos = nltk.pos_tag(tokens)
print(pos)
```



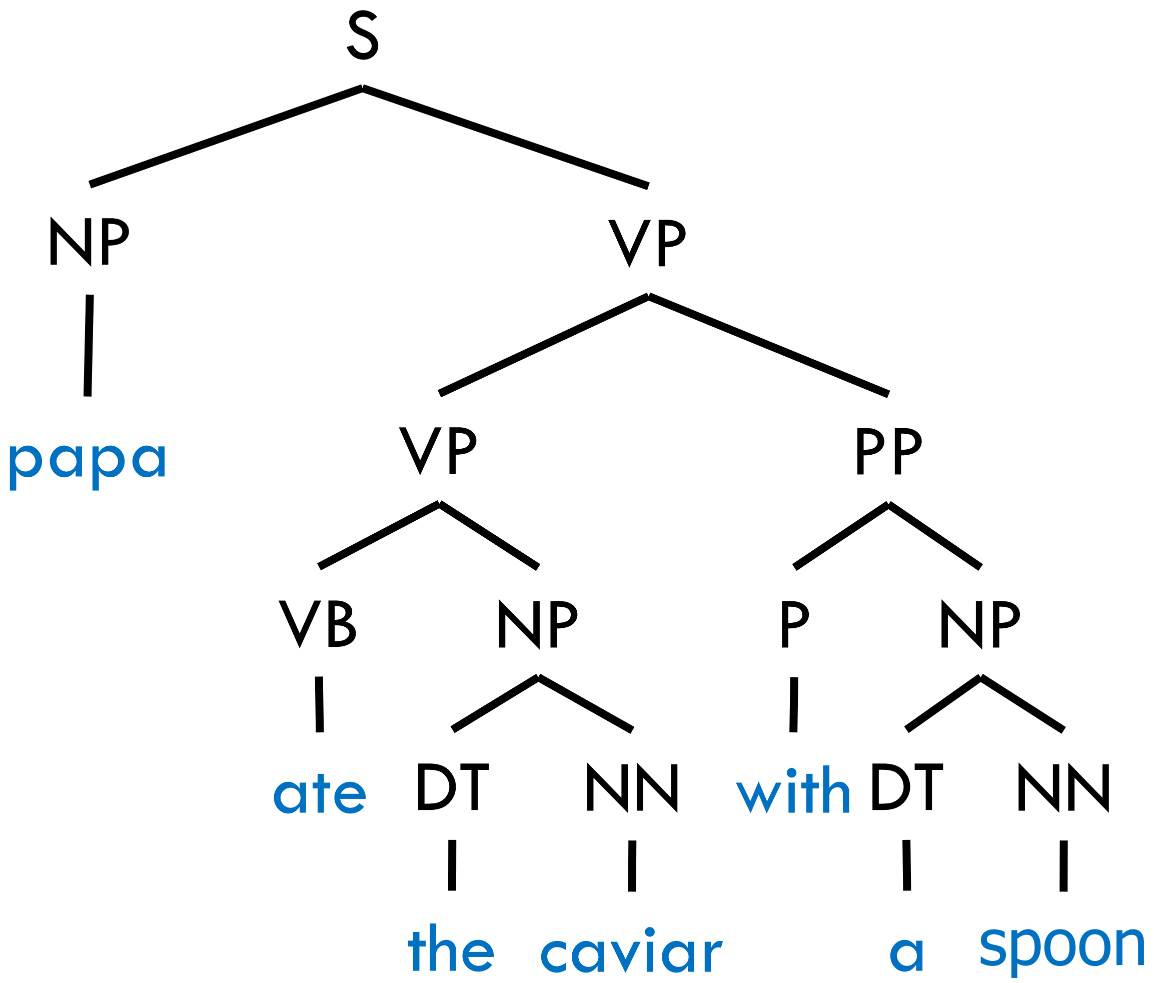
```
[('Hello', 'NNP'), ('my', 'PRP$'), ('name', 'NN'), ('is', 'VBZ'), ('John', 'NNP'), ('
Doe', 'NNP'), ('and', 'CC'), ('my', 'PRP$'), ('favourite', 'JJ'), ('number', 'NN'), (
'is', 'VBZ'), ('twenty-seven', 'JJ')]
>>> 
```

GRAMMAR RULES

- Every language needs a set of rules to form valid sentences, called grammar.
- The sentence should conform to the grammar, to be a part of a language.

- As a prerequisite of understanding the meaning of a sentence, we need to parse the sentence, using some defined grammar rules.

PARSING USING GRAMMAR RULES



GRAMMAR RULES

S → NP VP

NP → DT NN | NP PP

VP → VB NP | VP PP

PP → P NP

NP → papa

NN → caviar | spoon

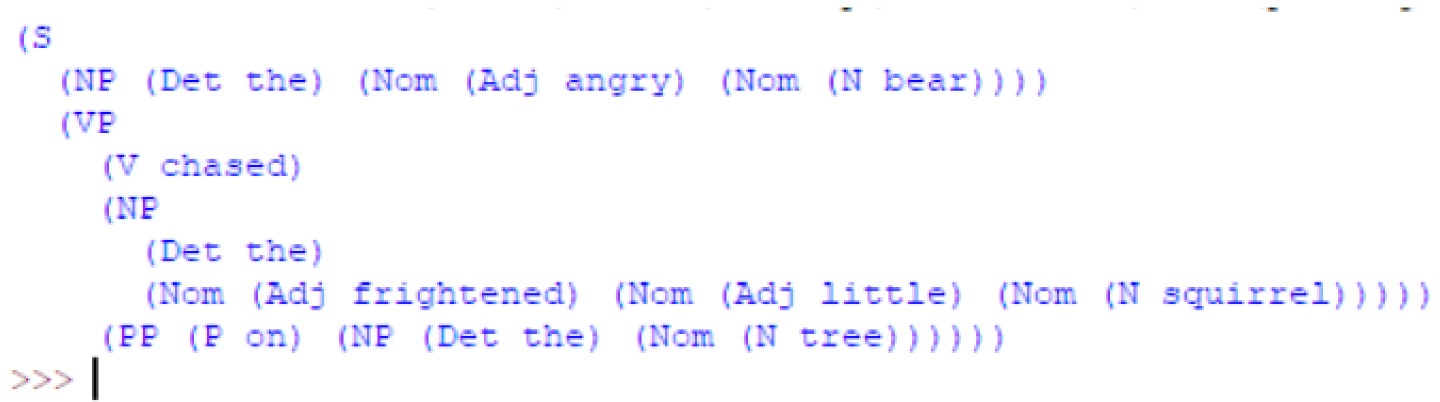
VB → spoon | ate

P → with

DT → the | a

SAMPLE CODE - 4

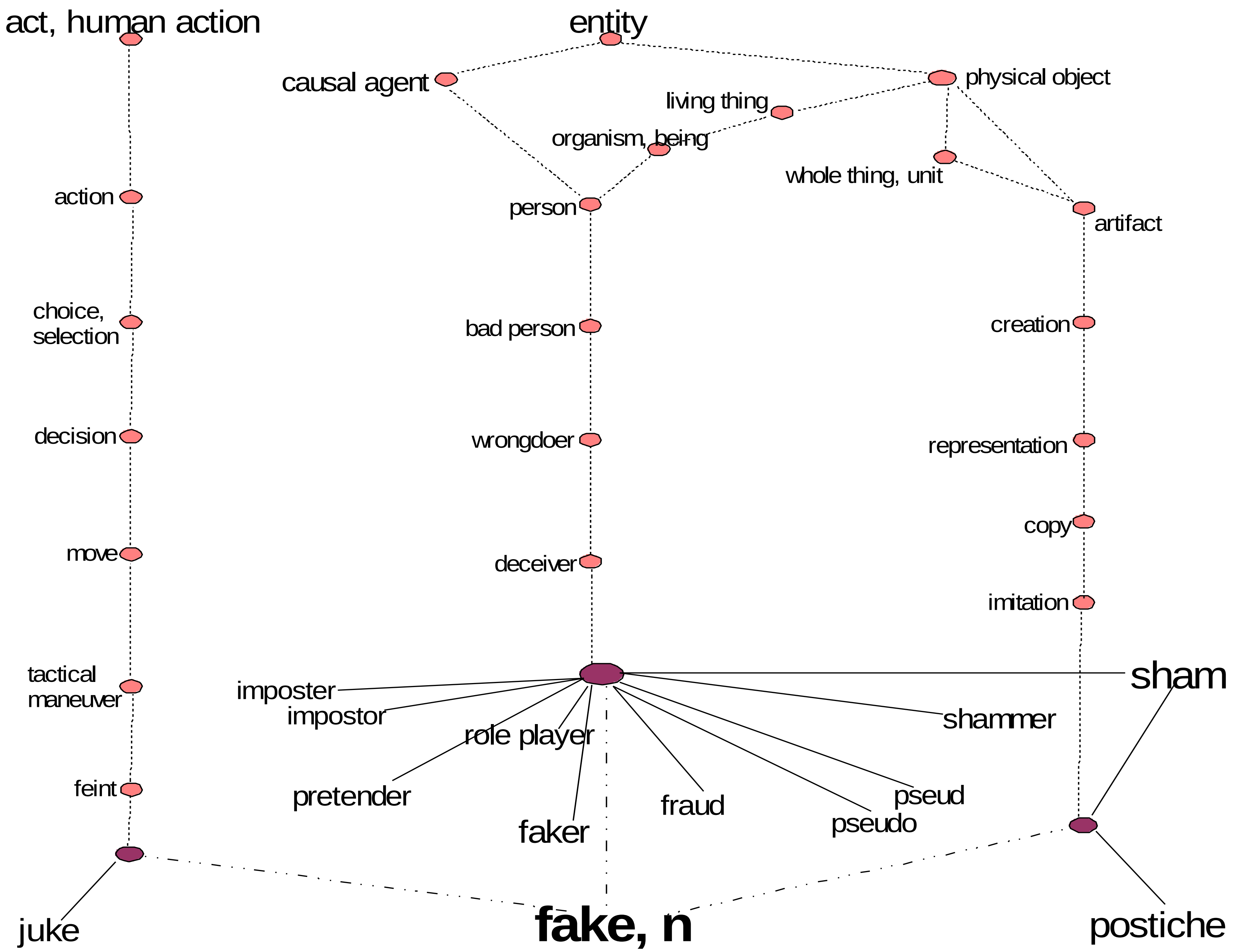
```
import nltk
from nltk import CFG
grammar1 = CFG.fromstring("""
S -> NP VP
NP -> Det Nom | PropN
Nom -> Adj Nom | N
VP -> V Adj | V NP | V S | V NP PP
PP -> P NP
Det -> 'the' | 'an' | 'a'
N -> 'bear' | 'squirrel' | 'tree'
Adj -> 'angry' | 'frightened' | 'little' | 'short'
V -> 'chased'
P -> 'on' | 'at'
""")
parser = nltk.RecursiveDescentParser(grammar1)
sentence1 = "the angry bear chased the frightened little squirrel on the tree".split()
for t in parser.parse(sentence1):
    print(t)
```

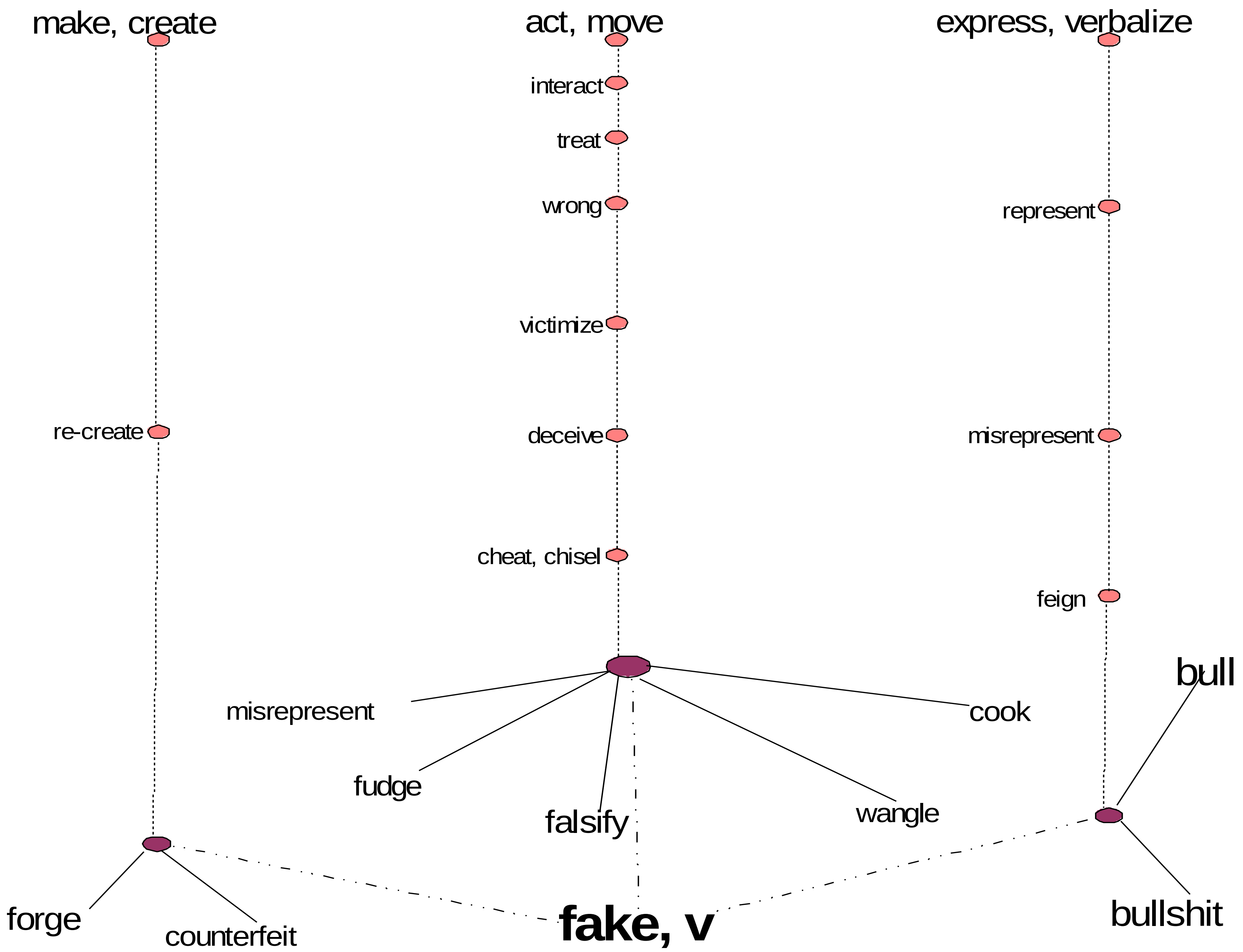


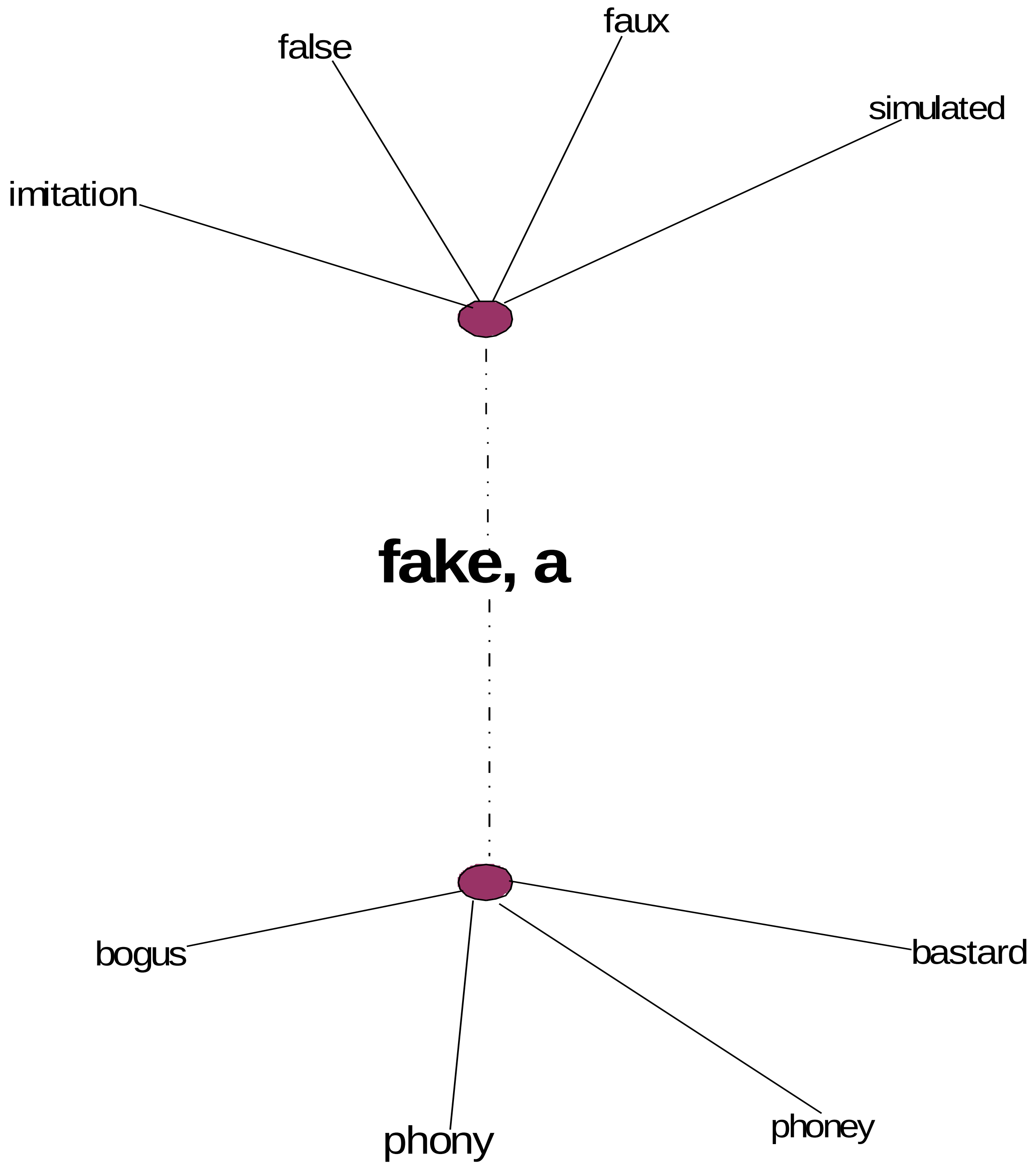
```
(S
  (NP (Det the) (Nom (Adj angry) (Nom (N bear))))
  (VP
    (V chased)
    (NP
      (Det the)
      (Nom (Adj frightened) (Nom (Adj little) (Nom (N squirrel)))))
    (PP (P on) (NP (Det the) (Nom (N tree))))))
>>>
```

WORDNET

- A lexical semantic network relating word forms and lexicalized concepts.
- Four main categories of relations:
  1. **Hyponyms:** A more specific meaning for a generalized term. E.g.: *Spoon* and *Cutlery*
  2. **Meronyms:** These words refer to the whole while mentioning a part. E.g.: "*I see several familiar faces present*." Here, *faces* is a meronym for *people*.
  3. **Antonyms:** Words with opposite meaning. E.g.: *lose* and *win*
  4. **Synonyms:** Words with similar meaning. E.g.: *win* and *victory*


act, human action
entity
causal agent
physical object
living thing
organism, being
whole thing, unit
action
person
artifact
choice, selection
bad person
creation
decision
wrongdoer
representation
copy
move
deceiver
imitation
tactical maneuver
imposter
impostor
role player
sham
shammer
feint
pretender
faker
fraud
pseud
pseudo
juke
fake, n
postiche


make, create
act, move
express, verbalize
interact
treat
wrong
represent
victimize
re-create
deceive
misrepresent
cheat, chisel
feign
bull
misrepresent
cook
fudge
falsify
wangle
forge
counterfeit
fake, v
bullshit


faux
false
simulated
imitation
fake, a
bogus
bastard
phony
phoney

SEMANTIC ANALYSIS | 4

NAMED ENTITY RECOGNITION (NER)

- Identifying named entities in text.
- Different from POS tagging, which identifies nouns at the word (token) level.
- NER identifies only entities, with meaning into consideration.

E.g.: *The **IEEE Student Chapter** of **Thapar University** organized a seminar on **Natural Language Processing**.*

Named Entities identified:

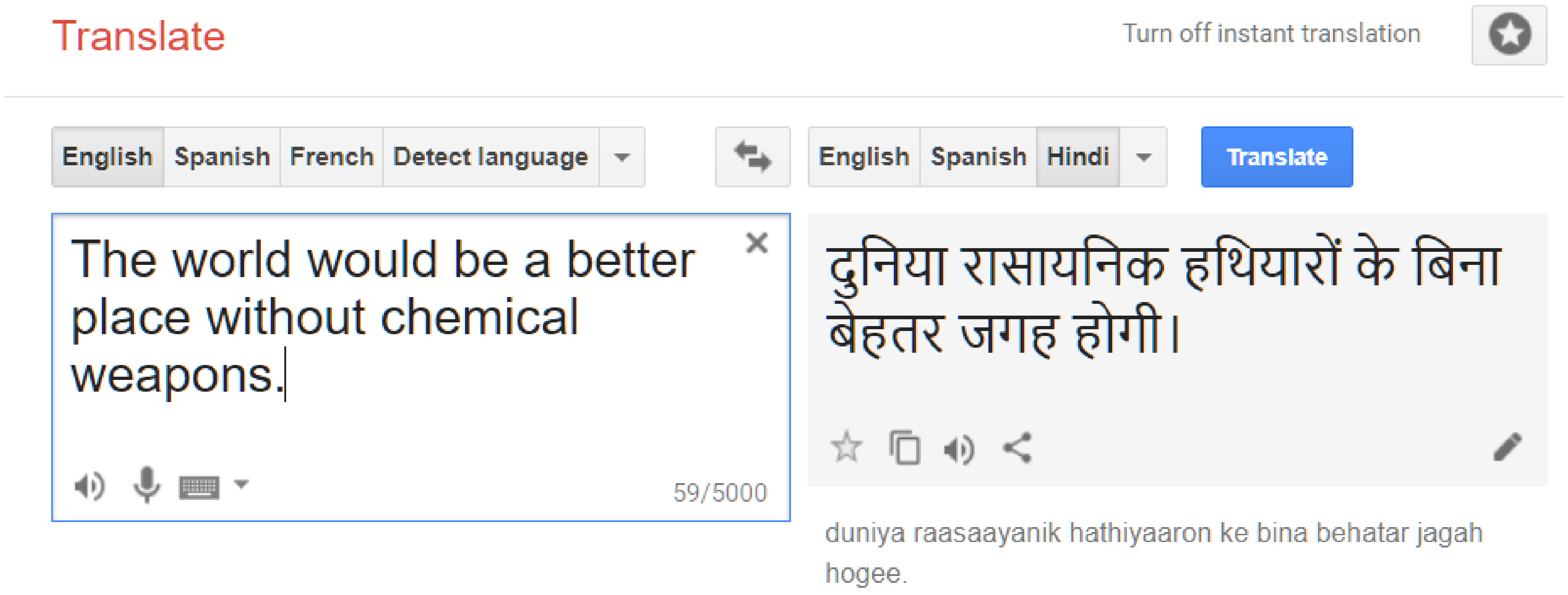
1. IEEE Student Chapter
2. Thapar University
3. Natural Language Processing

MACHINE TRANSLATION

Converting one natural language into another using a machine is Machine Translation (MT).

Complete MT has not yet been achieved.

Online platforms like Google Translate – some level of MT.



Translate

Turn off instant translation

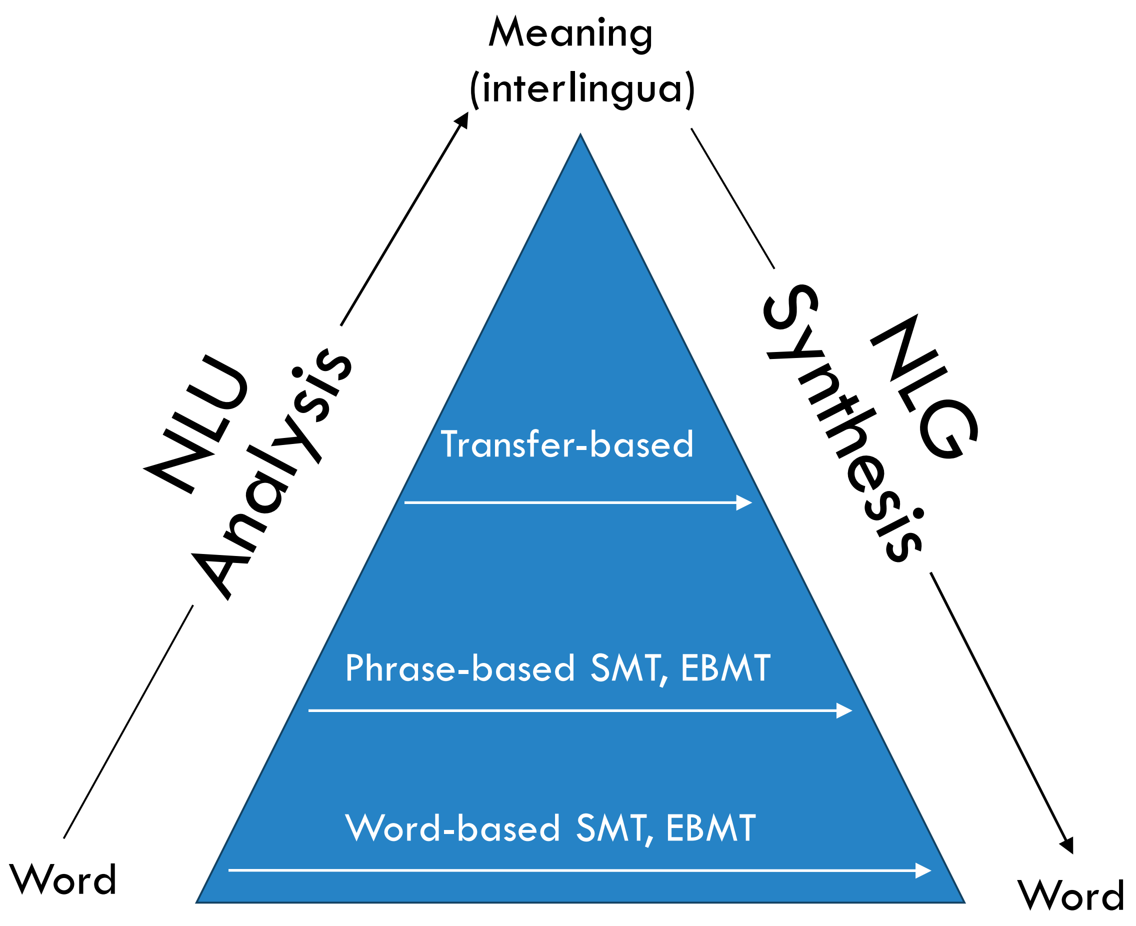
The world would be a better place without chemical weapons.

59/5000

दुनिया रासायनिक हथियारों के बिना बेहतर जगह होगी।

duniya raasaayanik hathiyaaron ke bina behatar jagah hogee.

THE MACHINE TRANSLATION TRIANGLE



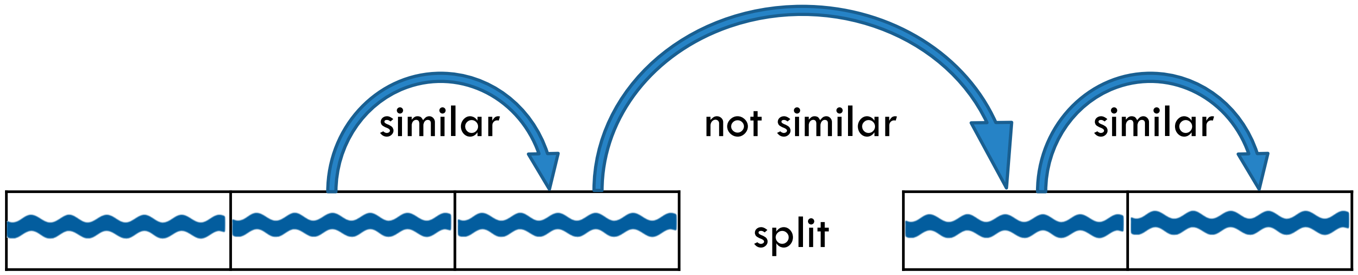
QUESTION ANSWERING

- The vastness of the WWW brings a problem. It is becoming more and more difficult to find answers on the WWW using standard search engines.
- Aim - To present the user with a **short answer** to a question rather than a list of possibly relevant documents.
- Question answering can be approached from one of two existing NLP research areas:
  - **Information Retrieval:** QA can be viewed as short passage retrieval.
  - **Information Extraction:** QA can be viewed as open-domain information extraction.

PARAGRAPH SEGMENTATION -

- Split a piece of text into paragraphs based on independence of content – TextTiling.

1. Block Comparison: Compare consecutive blocks of text, usually with the same word count.
2. Boundary Identification: Mark split point where there is significant difference.



SENTIMENT ANALYSIS

- Classification of pieces of text based on opinion – **positive, negative** or **neutral/no sentiment.**
- Use word-level semantics – whether a given word represents positive or negative emotion. E.g.: *magnificent, wholesome* (positive); *abominable, shambles* (negative).
- Consider negations while detecting sentiment words. E.g.: ***not** good, **isn't** evil.*
- Limitation: cannot detect sarcasm.
- Sentiment words and relations found in SentiWordNet.
- Other methods use machine learning and deep learning.

Uses of Sentiment Analysis?

DISCOURSE ANALYSIS

REFERENCE RESOLUTION

- Resolve references or pronouns to corresponding named-entities.
- Largely unsolved problem, but some progress made by using sentence structures.

E.g.:

Victoria Chen, Chief Financial Officer of Megabucks Banking Corp since 2004, saw her pay jump 20%, to $1.3 million, as the 37-year-old also became the Denver-based financial-services company's president. It has been ten years since she came to Megabucks from rival Lotsabucks.

AUTOMATIC SUMMARIZATION

- Generate a summary of any document, to convey the central idea or concept.
- Two types of Automatic Summarization:
  1. **Extractive** - Pick the best, most informative sentences out of the whole document and use them as-is to form a summary. Use sentence scoring by giving scores to individual words, based on frequency of occurrence.
  2. **Abstractive** - Construct a summary with completely new sentences. Closer to human-generated summaries. Somewhat unsolved problem.

TEXT-TO-SPEECH

- Needs large sample dataset of narrated word-collocations, to build vocabulary. This generates a more “natural” sound.
- The voices are sampled from real recorded speech and split into phonemes, a small unit of human speech. This is called Concatenation synthesis. This is used for **words not in vocabulary**, where the word is broken down into syllables for speech synthesis. This generates a more “artificial” sound.
- These sounds are combined by altering pre-recorded frequencies for a smooth flow of sound.

Try speaking:

It is what it is. ***What is it?***

SPEECH-TO-TEXT

This process involves four steps:

1. **Signal processing:** Convert the audio wave into a sequence of feature vectors
2. **Speech recognition:** Decode the sequence of feature vectors into a sequence of words
3. **Semantic interpretation:** Determine the meaning of the recognized words
4. **Dialog Management:** Correct errors and help get the task done

The process is assisted by word-collocation dictionaries, to correct errors or predict upcoming words.

Anshul Aggarwal

anshulaggarwal987@gmail.com | 98144 99764

linkedin.com/in/anshul-aggarwal

Thanks for your patience!

Content and Code Samples available at

github.com/anshul-aggarwal/NLP-IEEE-TU

Try building NLP applications without significant programming using IBM Watson:

https://www.ibm.com/watson/