Faculty Orientation Program
on
Natural Language Processing

[Elective V : 410252 A]

BE Computer Engineering 2019 Course

Organized By
S. T. E. S.'s Sinhgad Institute of Technology
in association with
BOS Computer Engineering, SPPU, Pune

(20th January 2023)

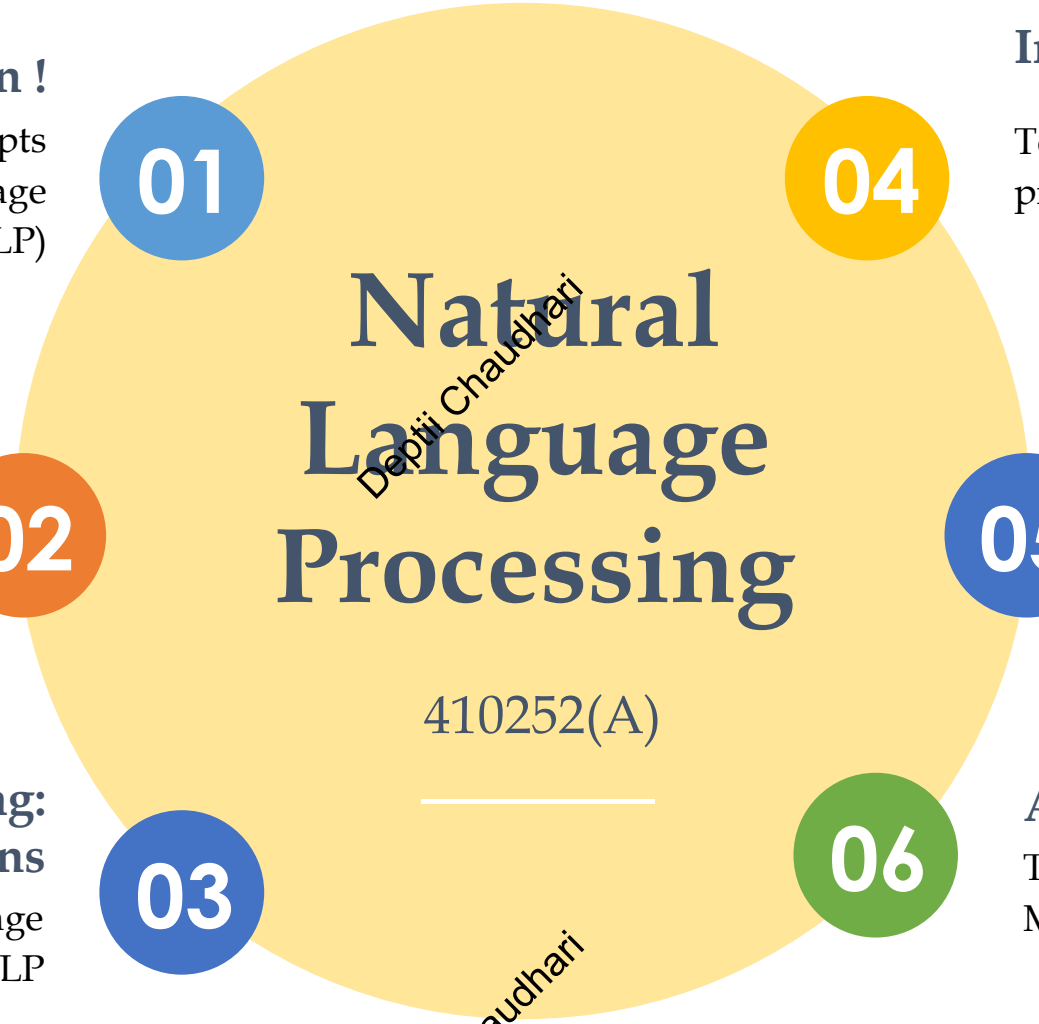
Prof. Deptii Chaudhari

Assistant Professor, Department of Computer Engineering

Hope Foundation's International Institute of Information Technology, Hinjawadi, Pune

deptiic@isquareit.edu.in, www.isquareit.edu.in

Course Objectives



01 Introduction !

To be familiar with fundamental concepts and techniques of natural language processing (NLP)

02 Language Syntax and Semantics: Core Knowledge

To acquire the knowledge of various morphological, syntactic, and semantic NLP tasks

03 Language Modelling: Illustrations

To develop the various language modeling techniques for NLP

04 Integrate

To use appropriate tools and techniques for processing natural languages

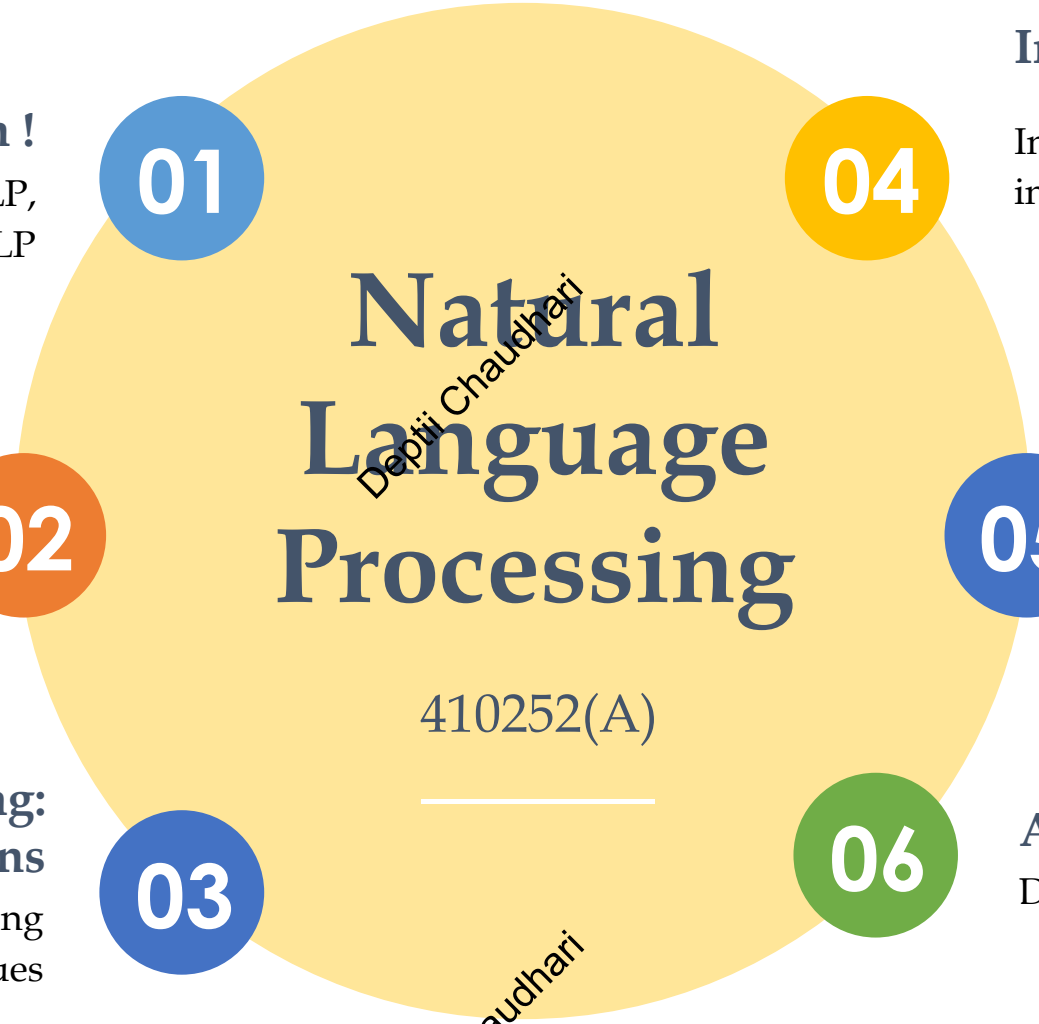
05 Recent Advances, Tools and Techniques

To comprehend the advance real world applications in NLP domain.

06 Applications

To Describe Applications of NLP and Machine Translations.

Course Outcomes



01 Introduction !

Describe the fundamental concepts of NLP, challenges and issues in NLP

02 Language Syntax and Semantics: Core Knowledge

Analyze Natural languages morphologically, syntactical and semantically

03 Language Modelling: Illustrations

Illustrate various language modelling techniques

04 Integrate

Integrate the NLP techniques for the information retrieval task

05 Recent Advances: Tools and Techniques

Demonstrate the use of NLP tools and techniques for text-based processing of natural languages

06 Applications

Develop real world NLP applications

Unit V: NLP Tools and Techniques

Prominent NLP Libraries:

- ✓ Natural Language Tool Kit (NLTK)
- ✓ spaCy
- ✓ TextBlob
- ✓ Gensim

Linguistic Resources:

- ✓ Lexical Knowledge Networks
- ✓ WordNets
- ✓ Indian Language WordNet (IndoWordnet)
- ✓ VerbNets
- ✓ PropBank
- ✓ Treebanks
- ✓ Universal Dependency Treebanks
- ✓ Word Sense Disambiguation: Lesk Algorithm, Walker's algorithm
- ✓ WordNets for Word Sense Disambiguation

Case Study: Hindi Wordnet, Sanskrit WordNet, Indic Library

Mapping to CO: CO5

Various NLP Libraries

NLP Library	Description
NLTK	This is one of the most usable and mother of all NLP libraries.
spaCy	This is a completely optimized and highly accurate library widely used in deep learning
Stanford CoreNLP Python	For client-server-based architecture, this is a good library in NLTK. This is written in JAVA, but it provides modularity to use it in Python.
TextBlob	This is an NLP library which works in Pyhton2 and python3. This is used for processing textual data and provide mainly all type of operation in the form of API.
Gensim	Genism is a robust open source NLP library support in Python. This library is highly efficient and scalable.
Pattern	It is a light-weighted NLP module. This is generally used in Web-mining, crawling or such type of spidering task. p
Polyglot	For massive multilingual applications, Polyglot is best suitable NLP library. Feature extraction in the way on Identity and Entity.
PyNLPl	PyNLPI also was known as 'Pineapple' and supports Python. It provides a parser for many data formats like FoLiA/GiZa/Moses/ARPA/Timbl/CQL.
Vocabulary	This library is best to get Semantic type information from the given text.

Natural Language Toolkit (NLTK)

- Leading platform for building Python programs to work with human language data
- Provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet
- Provides suite of text processing libraries for
  - classification
  - tokenization
  - stemming
  - tagging
  - parsing, and semantic reasoning,
  - wrappers for industrial-strength NLP libraries
  - an active discussion forum
- Suitable for linguists, engineers, students, educators, researchers, and industry users alike.

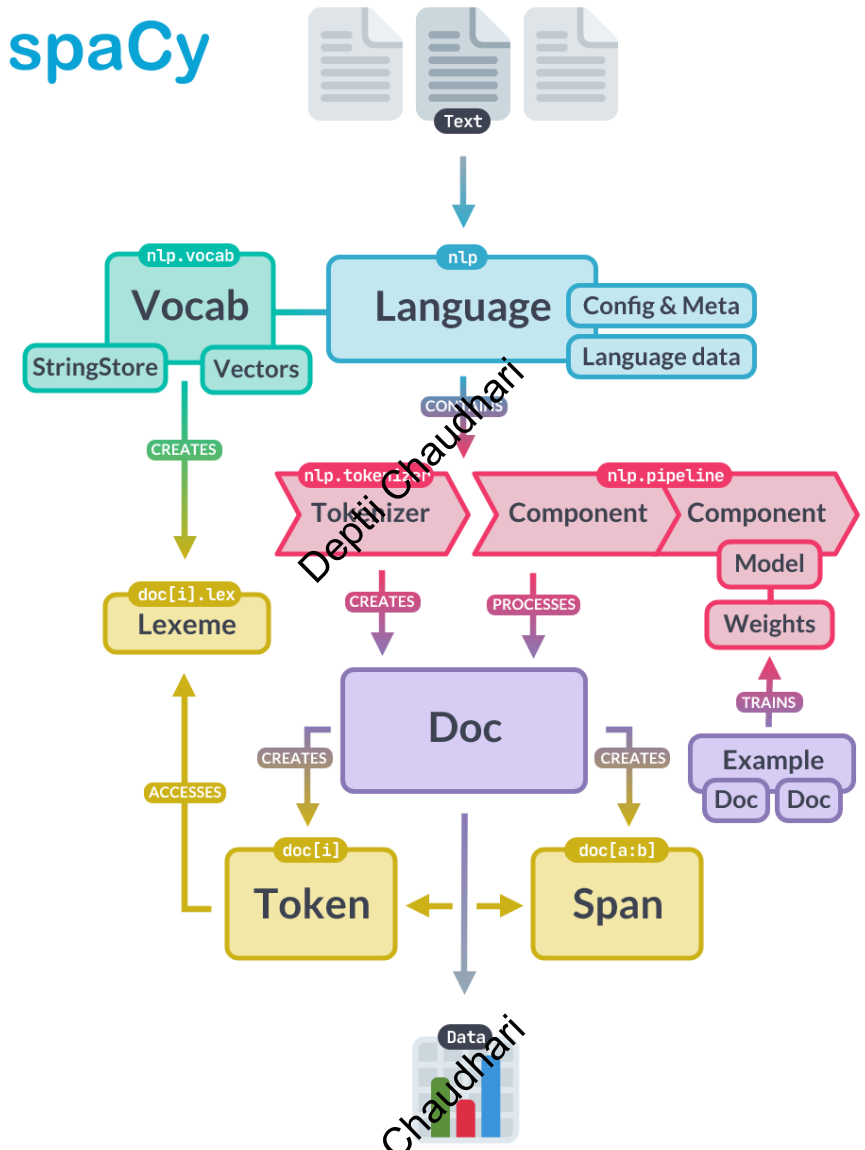
Natural Language Toolkit (NLTK)

- Text Tokenization
- Frequency Distribution of Words
- Stemming
- Lemmatization
- Filtering Stop words
- Parts of Speech(POS) Tagging
- Named Entity Recognition
- Getting Synonyms from Wordnet
- Getting Antonyms from Wordnet

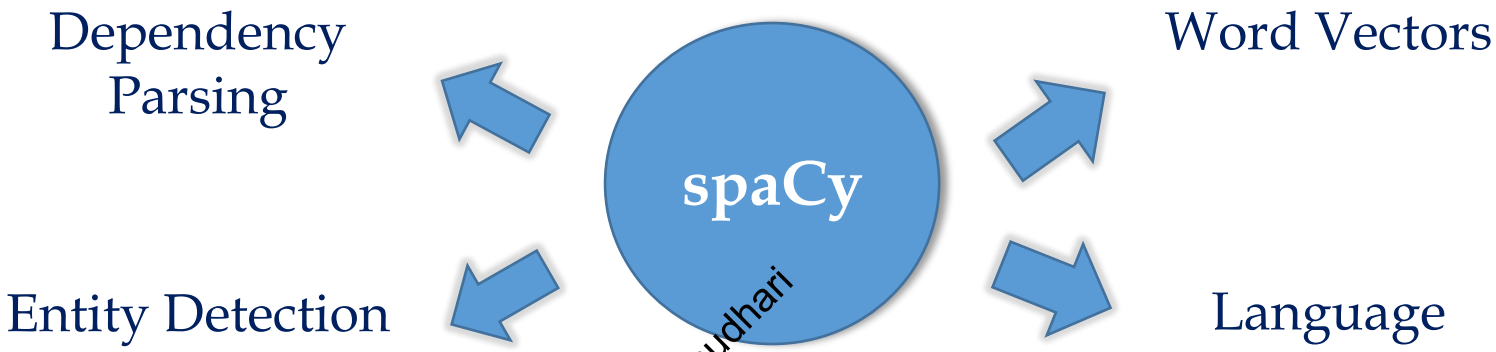
spaCy Features

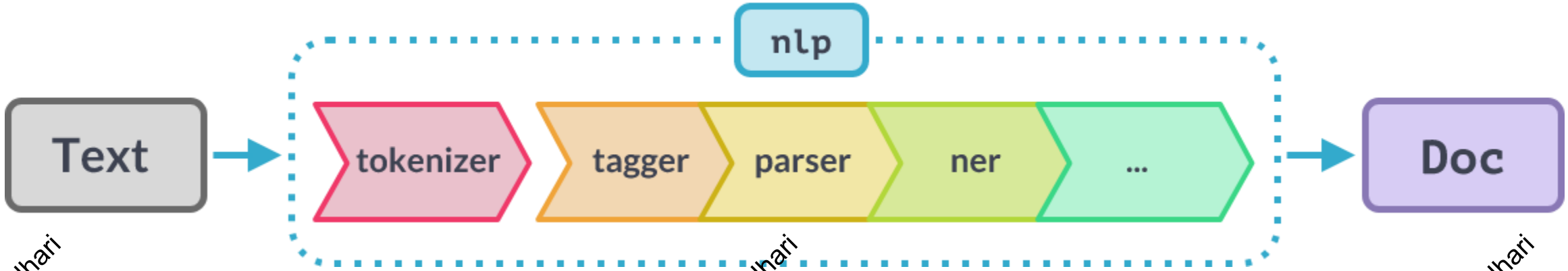
- ✓ Support for **72+ languages**
- ✓ **80 trained pipelines** for 24 languages
- ✓ Multi-task learning with pretrained **transformers** like BERT
- ✓ Pretrained **word vectors**
- ✓ State-of-the-art speed
- ✓ Production-ready **training system**
- ✓ Linguistically-motivated **tokenization**
- ✓ Components for named entity recognition, part-of-speech tagging, dependency parsing, sentence segmentation, text classification, lemmatization, morphological analysis, entity linking and more
- ✓ Easily extensible with **custom components** and attributes
- ✓ Support for custom models in **PyTorch**, **TensorFlow** and other frameworks
- ✓ Built in **visualizers** for syntax and NER
- ✓ Easy **model packaging**, deployment and workflow management
- ✓ Robust, rigorously evaluated accuracy

spaCy



spaCy - Language Processing Pipelines





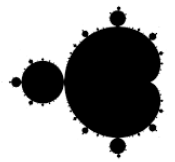

```
pipeline = ["tok2vec", "tagger", "parser", "ner"]
```

SpaCy - Language Processing Pipelines

```
import spacy
texts = ["Net income was $9.4 million compared to the prior year of $2.7 million.",   "Revenue exceeded twelve billion dollars, with a loss of $1b.",]
nlp = spacy.load("en_core_web_sm")
for doc in nlp.pipe(texts, disable=["tok2vec", "tagger", "parser", "attribute_ruler", "lemmatizer"]):
    # Do something with the doc here
    print([(ent.text, ent.label_) for ent in doc.ents])
```

```
[('$9.4 million', 'MONEY'), ('the prior year', 'DATE'), ('$2.7 million', 'MONEY')]
[('twelve billion dollars', 'MONEY'), ('1b', 'MONEY')]
```

TextBlob



- ✓ A Python (2 and 3) library for processing textual data.
- ✓ Provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more
- ✓ Features
  - Noun phrase extraction
  - Part-of-speech tagging
  - Sentiment analysis
  - Classification (Naive Bayes, Decision Tree)
  - Tokenization (splitting text into words and sentences)
  - Word and phrase frequencies
  - Parsing
  - n-grams
  - Word inflection (pluralization and singularization) and lemmatization
  - Spelling correction
  - Add new models or languages through extensions
  - WordNet integration

https://textblob.readthedocs.io/en/dev/

Gensim

- ✓ Gensim = “Generate Similar”
- ✓ Created by Radim Řehůřek
- ✓ A free open-source Python library for representing documents as semantic vectors, as efficiently (computer-wise) and painlessly (human-wise) as possible.
- ✓ Designed to process raw, unstructured digital texts (“plain text”) using unsupervised machine learning algorithms.
- ✓ The algorithms in Gensim, such as Word2Vec, FastText, Latent Semantic Indexing (LSI, LSA, LsiModel), Latent Dirichlet Allocation (LDA, LdaModel) etc, automatically discover the semantic structure of documents by examining statistical co-occurrence patterns within a corpus of training documents.
- ✓ These algorithms are unsupervised, which means no human input is necessary – you only need a corpus of plain text documents.
- ✓ Features
  - ✓ Super fast
  - ✓ Data Streaming - no "dataset must fit in RAM" limitations
  - ✓ Platform independent
  - ✓ Open source
  - ✓ Ready-to-use models and corpora

Linguistic Resources -Lexical Knowledge Networks

- Automatic text understanding requires knowledge and, so far, machines know only what we give or teach them.
- Therefore, most natural language processing (NLP) tasks crucially rely on the existence of linguistic resources that encode information about language, be it of morphological, syntactic, or semantic nature.
- Such resources are typically acquired via two main approaches
  - The knowledge-based approach, or top–down, where information is manually curated by humans
  - The corpus-based approach, or bottom–up, where information is automatically learned from corpora.
- Lexical knowledge Networks (LKNs), also known as lexico-semantic resources, provide information about words and potentially entities, and are at the core of knowledge-based approaches.

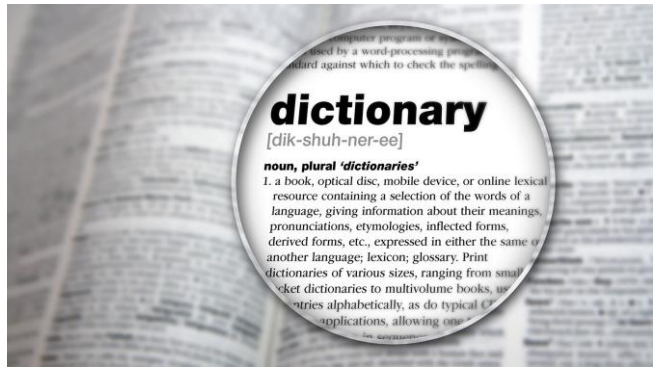
Lexical Knowledge Networks - Motivation

- How do you disambiguate 'web' in *"the spider spun a web"* from *"go surf the web"*?
- How do you summarize a long paragraph?
- How do you automatically construct language phrase books for tourists?
- Can a search query such as *"a game played with bat and ball"* be answered as *"cricket"*?
- Can the emotional state of a person who blogs *"I didn't expect to win the prize!"* be determined?
- Many of these issues can be (partially) resolved just by knowing more about the meaning of words - lexical semantics theory
- Need a lexicon that provides:
  - dictionary or thesaurus-like information
  - more rich associations among words
- Key elements:
  - collection of words
  - useful relations among them
  - ability to query the network

Lexical Knowledge Networks - Motivation

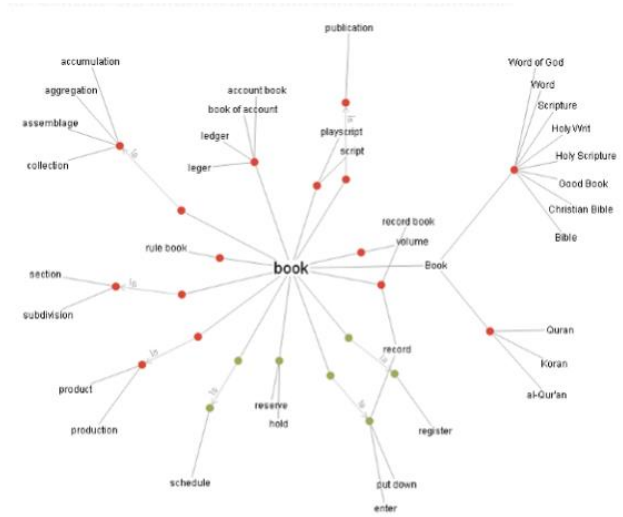
- LKNs embed a Conceptualization of the world
- Share universal properties across different languages
- The properties are combinatorial and graph theoretic in nature and pertain to the path length, degree, density etc.
- Examples of Lexical Networks
  - WordNets
  - ConceptNet
  - IndoWordnet
  - VerbNets
  - EuroWordNet
  - BabelNet

WordNets



=

Words + Meaning



=

Words + Meaning + Semantic and Lexical Relations

WordNets

- Constituent elements are senses
- Relational Semantics as opposed to componential semantics
- **Principles**: Differentiation, Minimality, Coverage, Replaceability
- **Basic entities**: "Synsets", i.e., Sets of Synonymous words
- **Relations**:
  - **Lexical**: Synonymy, Antonymy
  - **Semantic**: Hypernymy/Hyponymy, Meronymy/Holonymy and more

Principles used for Synset Creation

- Synsets are sets of synonymous words: basic elements of WNs
- **Minimality**: Only the minimal set that uniquely identifies the concept is used to create the sysnet, e.g.,{ghar, kamaraa} (room)
- **Coverage**: The synset should contain all the words denoting a concept. The words are listed in order of (decreasing) frequency of their occurrence in the corpus {ghar, kamaraa, kaksh} (room)
- **Replaceability**: The words forming the synset should be mutually replaceable in a specific context. Two synonyms may mutually replace each other in a context C, if the substitution of the one for the other in C does not alter the meaning of the sentence:
  - {svadesh, ghar} (motherland)– {apanaa desh} (the country where one is born)
  - amerikaa meN do saal bitaane ke baad shyaam svadesh/ ghar lauTaa
  - (America in two years stay after Shyam motherland returned)
  - ('Shyam returned to his motherland after spending two years in America')

WordNet Lexico-Semantic Relations

- **Synonymy**: relationship between words in a synset
- **Antonymy** : relationship between words having an opposite meaning
- **Gradation** : 'morning', 'afternoon', 'evening' are related through gradation relation
- **Hypernymy/Hyponymy :** is-a-kind-of relation – 'fruit' is a hypernym of 'mango' and 'mango' is a hyponym of 'fruit'.
- **Meronymy/Holonymy** : part-whole relation – 'hand' is a meronym of 'body' and 'body' is a holonym of 'hand'
- **Entailment**: 'snore' entails 'sleep'
- **Attribute**: relationship between noun and adjective synsets – 'hot' is a value of or attribute of 'temperature'
- **Nominalization**: relationship between noun and verb synsets – 'service' nominalizes the verb 'serve'
- **Ability Link**: specifies the inherited features of a nominal concept – 'animal' and 'walk', 'fish' and 'swim'
- **Capability Link**: relationship specifies the acquired features of a nominal concept – 'person' and 'swim'
- **Function Link**: relationship specifies the function of a nominal concept – 'vehicle' and 'move' and 'teacher' and 'teach'

WordNet Example

WordNet Search - 3.1

- WordNet home page - Glossary - Help

Word to search for: school [Search WordNet]

Display Options: (Select option to change) [Change]

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations

Display options for sense: (gloss) "an example sentence"

Noun

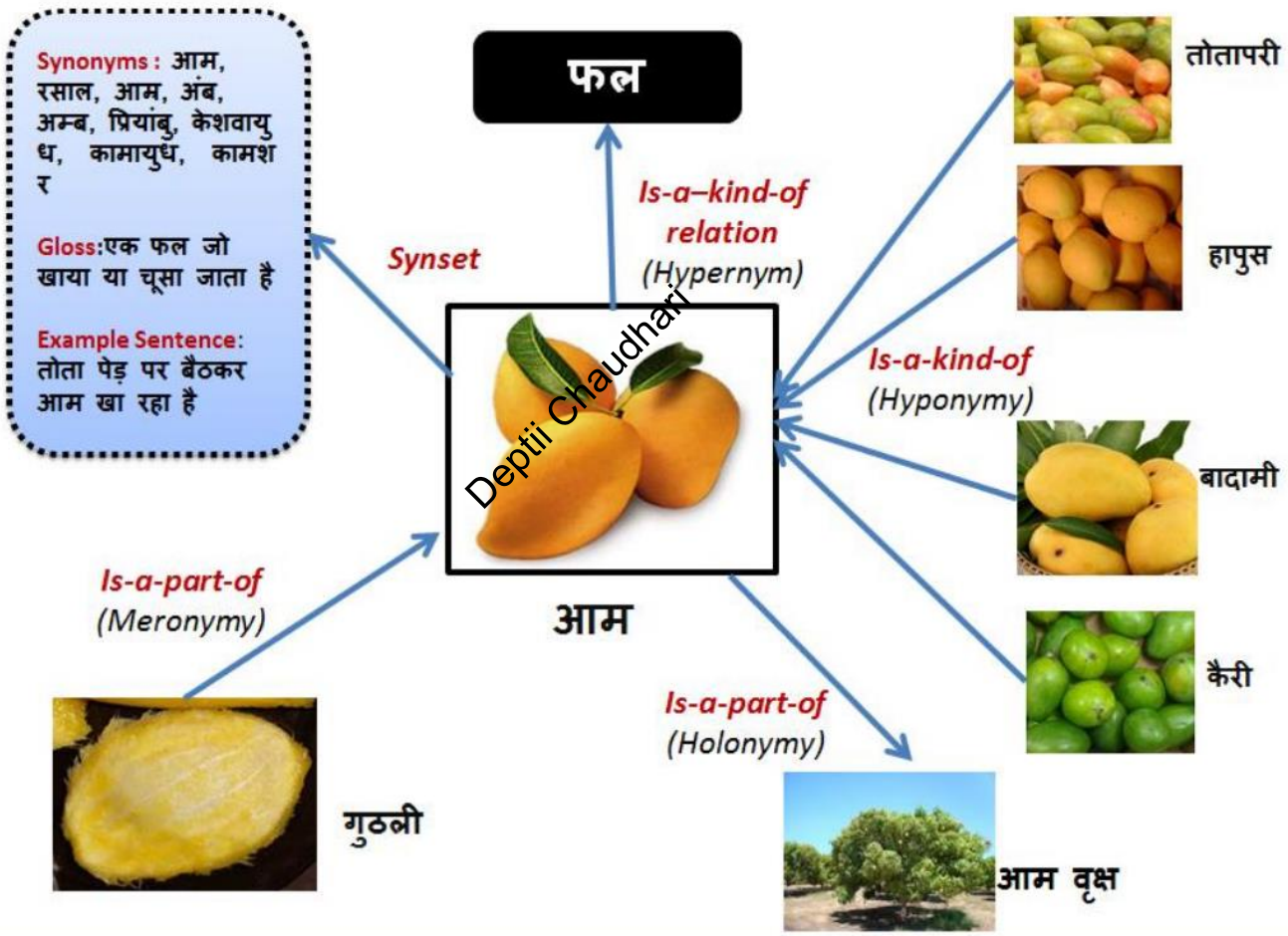
- S: (n) **school** (an educational institution) *"the school was founded in 1900"*
- S: (n) **school**, schoolhouse (a building where young people receive education) *"the school was built in 1932"; "he walked to school every morning"*
- S: (n) **school**, schooling (the process of being formally educated at a school) *"what will you do when you finish school?"*
- S: (n) **school** (a body of creative artists or writers or thinkers linked by a similar style or by similar teachers) *"the Venetian school of painting"*
- S: (n) **school**, schooltime, school day (the period of instruction in a school; the time period when school is in session) *"stay after school"; "he didn't miss a single day of school"; "when the school day was done we would walk home together"*
- S: (n) **school** (an educational institution's faculty and students) *"the school keeps parents informed"; "the whole school turned out for the game"*
- S: (n) **school**, shoal (a large group of fish) *"a school of small glittering fish swam by"*

Verb

- S: (v) **school** (educate in or as if in a school) *"The children are schooled at great cost to their parents in private institutions"*
- S: (v) educate, **school**, train, cultivate, civilize, civilise (teach or refine to be discriminative in taste or judgment) *"Cultivate your musical taste"; "Train your tastebuds"; "She is well schooled in poetry"*
- S: (v) **school** (swim in or form a large group of fish) *"A cluster of schooling fish was attracted to the bait"*

http://wordnetweb.princeton.edu/perl/webwn

WordNet Example



https://www.cfilt.iitb.ac.in/wordnet/webhwn/

Institutes involved in creating IndoWordNet

Indian Institute of Technology, Bombay	Hindi, Marathi, Sanskrit
Goa University, Goa	Konkani
Guwahati University, Guwahati	Assamese, Bodo
University of Hyderabad, Hyderabad	Odia
Jawaharlal Nehru University, New Delhi	Urdu
Dharmsinh Desai University, Nadiad	Gujarati
University of Kashmir, Srinagar	Kashmiri
Punjabi University, Patiala	Punjabi
Thapar University, Patiala	Punjabi
Manipur University, Imphal	Manipuri
Assam University, Silchar	Nepali
Amrita Vishwa Vidyapeetham, Coimbatore	Malayalam
University of Mysore, Mysore	Kannada
Tamil University , Tanjavur	Tamil
Dravidian University, Kuppam	Telugu

Applications of WordNets

- Machine Translation
- Word Sense Disambiguation
- Sentiment Analysis
- Information Retrieval
- Multi Word Expression Detection
- Document structuring and categorization
- Cognitive NLP

VerbNet

- VerbNet (VN) is the **largest on-line verb lexicon** currently available for English.
- VerbNet is part of the SemLink project in development at the University of Colorado.
- It is a hierarchical domain-independent, broad-coverage verb lexicon with mappings to other lexical resources such as WordNet, Xtag, and FrameNet.
- VerbNet is used in a variety of natural language processing applications and research projects.
- VerbNet can be used to distinguish between multiple possible senses for a given verb.
- VerbNet can be used for metaphor and figurative language detection by looking for violations to selectional restrictions imposed on thematic roles.
  - Literal :
    - The bomb destroyed the building [+concrete]
    - The rock damaged the fence [+conrete]
  - Figurative :
    - The speaker destroyed his argument [-concrete]
    - The politician damaged her reputation [-concrete]
- Caused-Motion Constructions (CMC)

Treebanks

- Context-free grammar rules can be used, in principle, to assign a parse tree to any sentence.
- This means that it is possible to build a corpus in which every sentence is syntactically annotated with a parse tree. Such a syntactically annotated corpus is called a **treebank**.
- A wide variety of treebanks have been created, generally by using parsers to automatically parse each sentence, and then using humans (linguists) to hand-correct the parses.
- The Penn Treebank project has produced treebanks from the Brown, Switchboard, ATIS, and Wall Street Journal corpora of English, as well as treebanks in Arabic and Chinese.
- Other treebanks include the Prague Dependency Treebank for Czech, the Negra treebank for German, and the Susanne treebank for English.

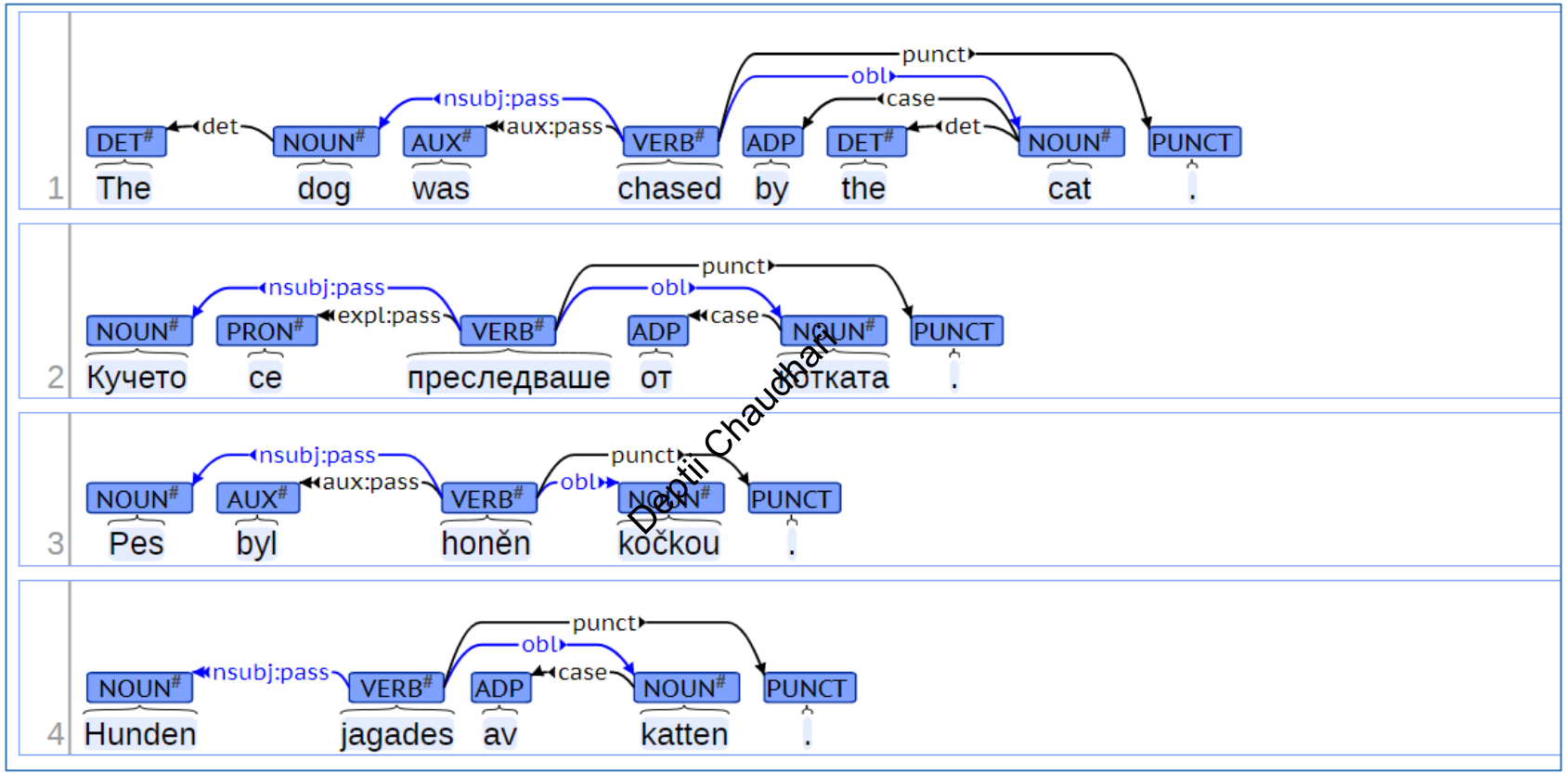
Treebanks – Interesting Reads

- Bhojpuri Treebank: https://aclanthology.org/2020.wildre-1.7.pdf
- Marathi Treebank: https://aclanthology.org/W17-7623.pdf
- AnnCorra: https://aclanthology.org/W02-1202.pdf
- Indian Languages Treebanking Project: https://kcis.iiit.ac.in/LT/

Universal Dependency Treebanks

- Universal Dependencies (UD) is a framework for consistent annotation of grammar (parts of speech, morphological features, and syntactic dependencies) across different human languages.
- UD is an open community effort with over 300 contributors producing nearly **200 treebanks** in over **100 languages**.
- The annotation scheme is based on an evolution of (universal) Stanford dependencies, Google universal part-of-speech tags, and the Interset interlingua for morphosyntactic tagsets.
- The general philosophy is to provide **a universal inventory of categories and guidelines to facilitate consistent annotation of similar constructions across languages**, while allowing language-specific extensions when necessary.

Universal Dependency Treebanks



Parallel examples from English, Bulgarian, Czech and Swedish, where the main grammatical relations involving a passive verb, a nominal subject and an oblique agent are the same, but where the concrete grammatical realization varies.

https://universaldependencies.org/introduction.html

Word Sense Disambiguation

- Word Sense Disambiguation (WSD) is the problem of computationally determining the 'sense' or 'meaning' of a word in a particular context.
- Why do you need it?
- **Ambiguity**
- **Structural ambiguity: due to the sentences structure**
  - *A boy saw a man with a telescope* (English)
  - राम ने दौड़ते हुए शेर को देखा।
- **Lexical ambiguity : due to polysemous words**
  - *She put her glasses on the table* (English)
  - पड़ोसी ने हमारे घर में आग लगायी (Hindi)

Word Sense Disambiguation – Why is it difficult?

- Sometimes human even fails to disambiguate.

'उसका हाथ मशीन के नीचे आ गया'

1. **हाथ**, बाजू, हस्त, बाँह, बाहु, बा… कर - कन्धे से पंजे तक का वह अंग जिससे चीज़ें पकड़ते और … *…धीजी के हाथ बहुत लंबे थे । / भीम की भुजाओं में बहुत बल था*

2. **हाथ**, कर, पंजा, पाणि - …भाग *"उसका हाथ मशीन के नीचे आ गया ।"*

3. **हाथ**, हस्त, कर, पाणि - कोहनी से पंजे के सिरे तक का भाग *"दुर्घटना में उसका दाहिना हाथ टूट गया ।"*

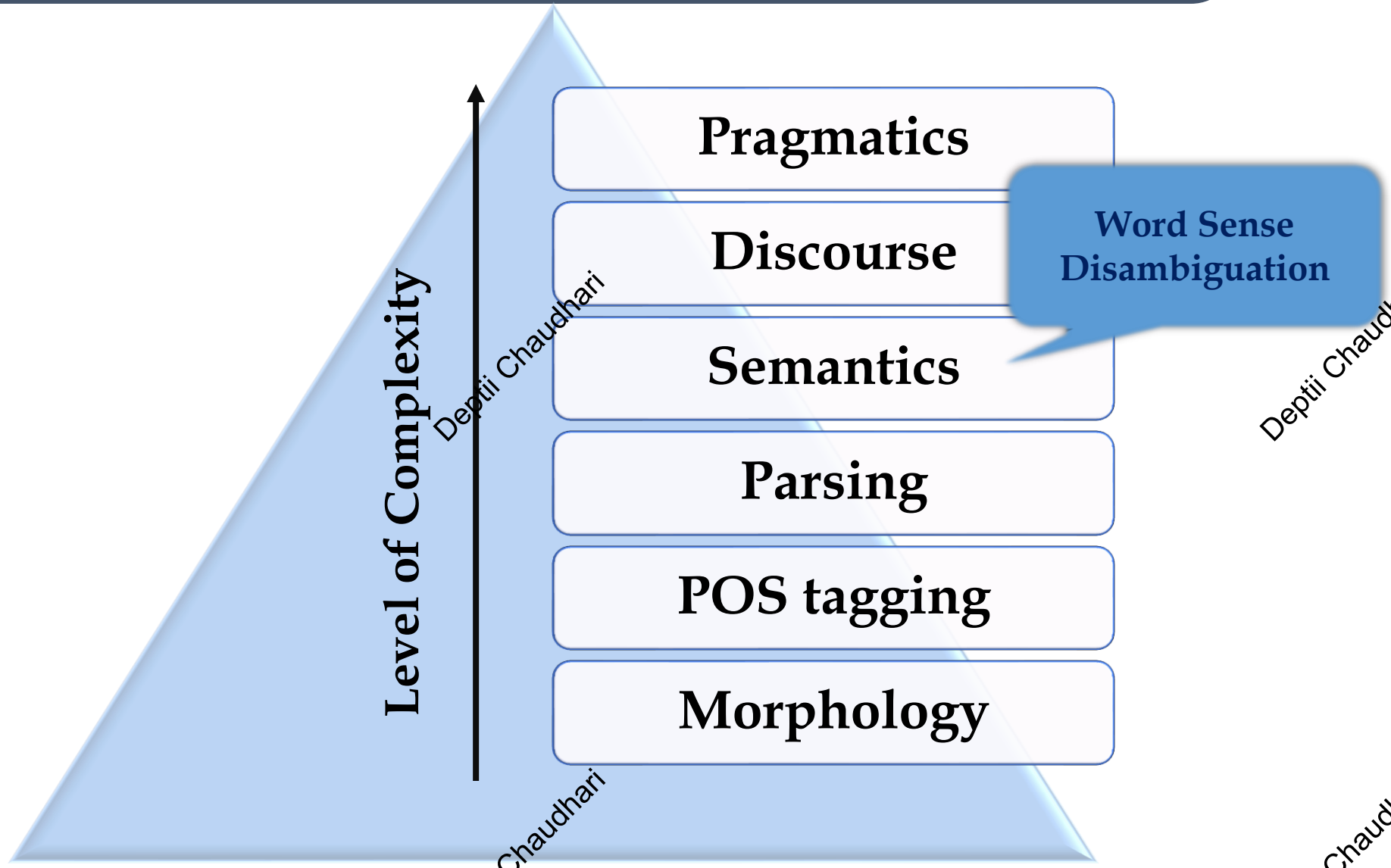
Fine-grained senses

4. **हाथ**, हस्त - चौबीस अंगुल की एक नाप … सिरे तक की लंबाई की नाप , *"इस वस्त्र की लंबाई दो हाथ है"*

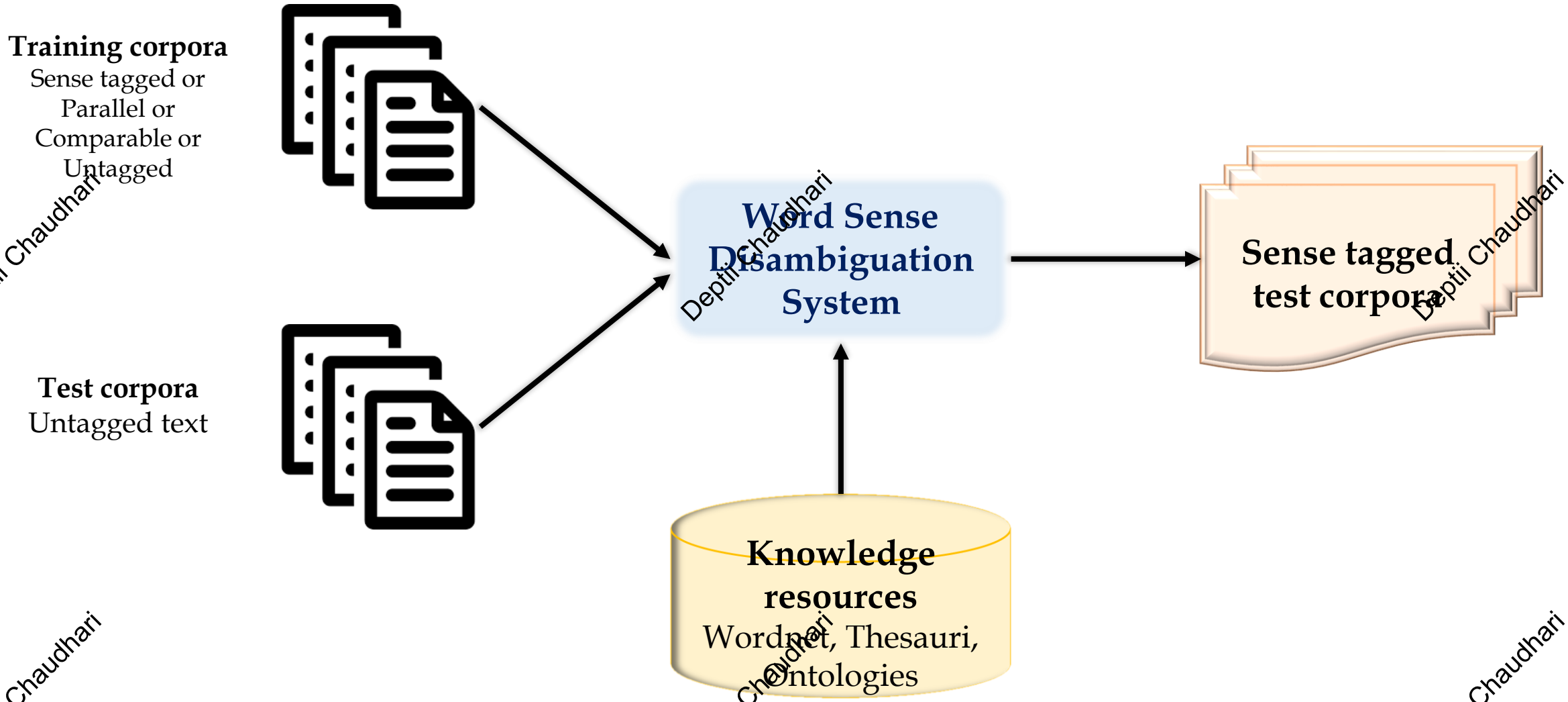
5. **हाथ** - ताश के खेल में एक दौर में गिरने … उसके बाद खेल से बाहर हो जाएँ, *"मेरे सात हाथ बन चुके हैं ।"*

Coarse-grained senses

Word Sense Disambiguation in NLP layers



Working of Word Sense Disambiguation



Word Sense Disambiguation Approaches

uses an explicit lexicon (machine readable dictionary, thesaurus) or ontology (e.g., WordNet).

WSD Approaches

Knowledge-based WSD

- Using Selectional Restrictions
- Lesk's algorithm
- Using conceptual density
- Using Random Walk Algorithms
- Walker'salgorithm

Corpus-based WSD

Supervised

- Naïve Bayes
- Decision Lists
- KNN
- SVM

Unsupervised

Distributional

Type Based

- Hyperlex
- Latent Semantic Indexing (LSA)
- Hyper Space Analogue to Language (HAL)

Token Based

- Context Group Discrimination
- McQuitty'sSimilarity Analysis

Translational Equivalence

- Brown corpus

Hybrid approach

Lesk Algorithm

- The Lesk algorithm assumes that words in a given "neighborhood" (section of text) will tend to share a common topic.
- A simplified version of the Lesk algorithm is to compare the dictionary definition of an ambiguous word with the terms contained in its neighborhood.
- An implementation might look like this:
  - for every sense of the word being disambiguated one should count the number of words that are in both the neighborhood of that word and in the dictionary definition of that sense
  - the sense that is to be chosen is the sense that has the largest number of this count.
- Lesk algorithm works with
  - **Sense Bag:** contains the words in the definition of a candidate sense of the ambiguous word.
  - **Context Bag:** contains the words in the definition of each sense of each context word.

Lesk Algorithm Example

- Let's disambiguate "bank" in this sentence:
  - The **bank** can guarantee deposits will eventually cover future tuition costs because it invests in adjustable-rate mortgage securities.
- Following are the two wordnet senses given for "bank".

bank[1]	**Gloss:**	**a financial institution that accepts deposits and channels the money into lending activities**
	Examples:	**"he cashed a check at the bank", "that bank holds the mortgage on my home"**
bank[2]	Gloss:	sloping land (especially the slope beside a body of water)
	Examples:	"they pulled the canoe up on the bank", "he sat on the bank of the river and watched the currents"

Simplified Lesk Algorithm

- **Choose sense with most word overlap between gloss and context.**

- The **bank** can guarantee deposits will eventually cover future tuition costs because it invests in adjustable-rate mortgage securities.

bank[1]	**Gloss:**	**a financial institution that accepts deposits and channels the money into lending activities**
	Examples:	**"he cashed a check at the bank", "that bank holds the mortgage on my home"**
bank[2]	Gloss:	sloping land (especially the slope beside a body of water)
	Examples:	"they pulled the canoe up on the bank", "he sat on the bank of the river and watched the currents"

The Corpus Lesk algorithm

- Assumes we have some sense-labeled data
- Take all the sentences with the relevant word sense:
  - *These short, "streamlined" meetings usually are sponsored by local banks[1] , Chambers of Commerce, trade associations, or other civic organizations.*
- Now add these to the gloss + examples for each sense, call it the "signature" of a sense.
- Choose sense with most word overlap between context and signature.

Walker's Algorithm

- Walker's algorithm follows a Thesaurus Based approach.
- **Step 1:** For each sense of the target word find a thesaurus category to which that sense belongs.
- **Step 2:** Calculate the score for each sense by using the context words. A context word will add 1 to the score of the sense if the thesaurus category of the word matches that of the sense.
  - ***E.g. The money in this bank fetches an interest of 8% per annum.***
  - **Target word:** bank
  - **Clue words from the context:** money, interest, annum, fetch

	Sense 1: Finance	Sense2: Location
Money	**+1**	**0**
Interest	**+1**	**0**
Fetch	**0**	**0**
Annum	**+1**	**0**
Total	**3**	**0**

Context words add 1 to the sense when the topic of the word matches that of the sense.

Interesting Resources to explore NLP

- https://ruder.io/
- http://nlpprogress.com/
- https://github.com/ivan-bilan/The-NLP-Pandect

Thank You All

Write to Me..

deptiic@isquareit.edu.in

Thank You!