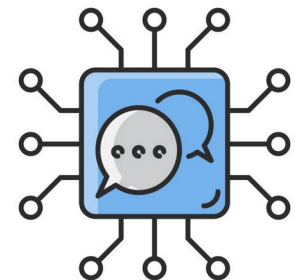


Shallow Parsing and Tools for NLP

Tushar B. Kute,
<http://tusharkute.com>



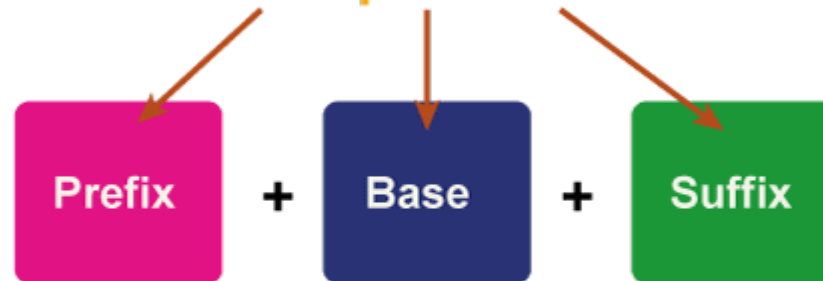
Morphology

- In linguistics, morphology is the study of the internal structure of words, focusing on how smaller units of meaning, known as morphemes, combine to form words.
- Think of it as dissecting words to understand their building blocks and how they connect.

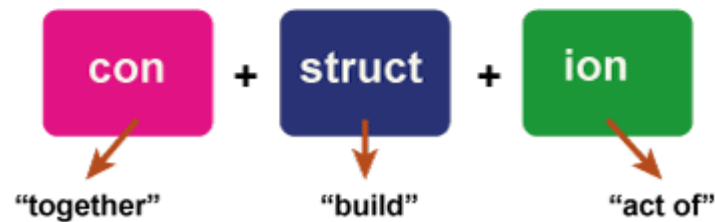
Morphology

Morphology

Words are made up of
morphemes

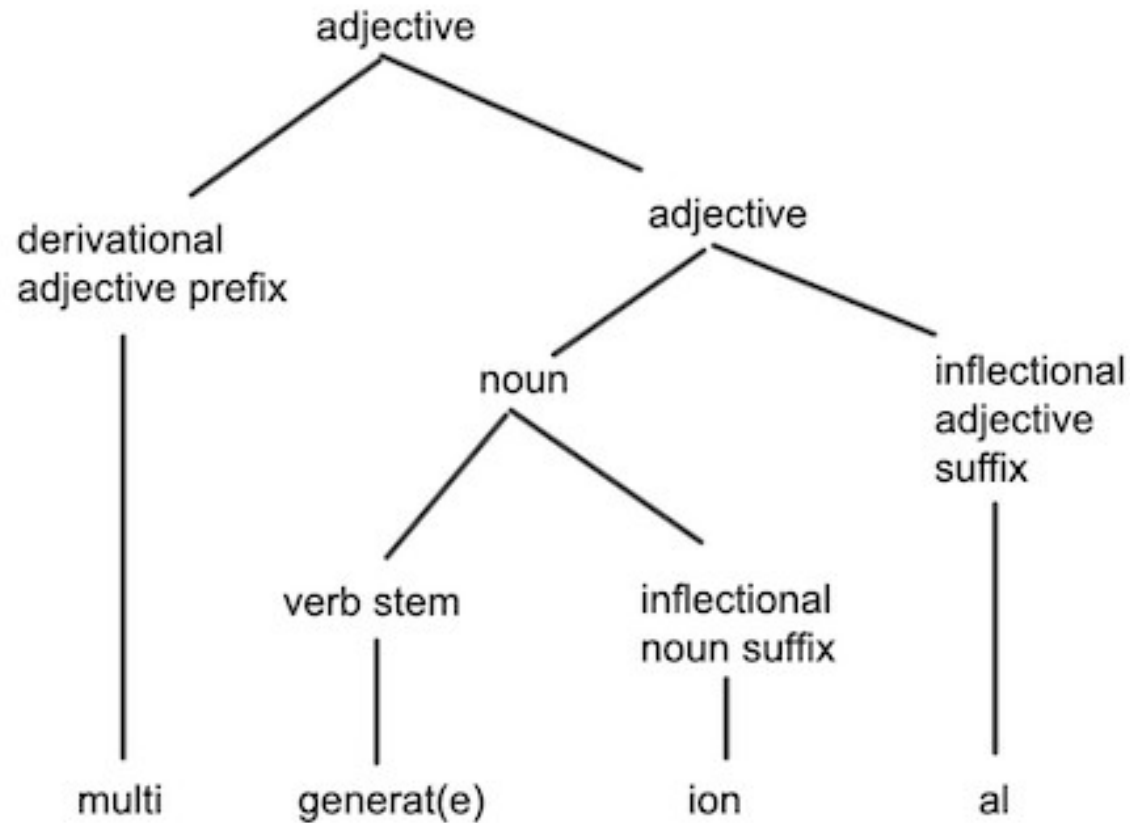


Each morpheme carries meaning.



Construction means the act of building things together.

Morphology



Morphemes

- The smallest units of meaning within a word.
Example: In "unbreakable," "un-" and "breakable" are morphemes.
 - Free vs. Bound:
 - Free morphemes: Can stand alone as words (e.g., "book," "run").
 - Bound morphemes: Must attach to another morpheme to form a word (e.g., "un-," "-able").

Morphemes : Types

- Prefixes:
 - Added to the beginning of a word (e.g., "un-," "re-").
- Suffixes:
 - Added to the end of a word (e.g., "-able," "-ly").
- Roots:
 - The core meaning-carrying morpheme of a word (e.g., "break" in "unbreakable").

Morphological Processes

- Inflection:
 - Modifying a word to express grammatical information like tense, number, or case (e.g., "sing," "sings," "sung").
- Derivation:
 - Creating new words from existing ones by adding affixes (e.g., "happy" -> "unhappy").
- Compounding:
 - Combining two or more words to form a new word (e.g., "blackboard," "sunflower").

Morphology Types: Process

- Inflection:
 - Modifying a word to express grammatical information like tense, number, case, or mood. Examples: "sing," "sings," "sung," "book," "books."
- Derivation:
 - Creating new words from existing ones by adding affixes (prefixes, suffixes, infixes). Examples: "happy" -> "unhappy," "teach" -> "teacher," "run" -> "running."

Morphology Types: Process

- Compounding:
 - Combining two or more words to form a new word. Examples: "blackboard," "sunflower," "bookstore."
- Conversion:
 - Changing the part of speech of a word without adding affixes. Examples: "run" (verb) -> "run" (noun), "fast" (adjective) -> "fast" (adverb).

Morphology Types: Affix

- Prefixation:
 - Adding an affix at the beginning of a word.
Examples: "un-," "re-," "non-."
- Suffixation:
 - Adding an affix at the end of a word.
Examples: "-able," "-ly," "-ness."

Morphology Types: Affix

- Infixation:
 - Adding an affix within a word stem.
Examples: "s" in "sing-s," "umlaut" in German.
- Circumfixation:
 - Adding affixes both at the beginning and end of a word. Examples: "ge-" and "-t" in German "gearbeitet" (worked).

Morphology Types: Derivation

- English:
 - Primarily uses prefixes and suffixes to change the meaning or part of speech of a word.
 - Examples: "unhappy," "playable," "conversion."

Morphology Types: Derivation

- Indian Languages:
 - Reduplication: Many languages like Telugu and Oriya repeat parts of words for emphasis or to denote grammatical changes. Example: "chala-chala" (going repeatedly).
 - Internal changes: Some languages like Punjabi alter vowel sounds or consonants within words for derivation. Example: "padhna" (to read) vs. "parh" (reading).

Morphology Types: Compounding

- English: Combines two or more words to form new ones. Examples: "blackboard," "sunflower," "bookstore."

Morphology Types: Compounding

- Indian Languages:
 - Tātkriya: Certain Indian languages like Sanskrit and Hindi form compound verbs by linking nouns or adjectives with verbs. Example: "jal-pīna" (to drink water) from "jal" (water) and "pīna" (to drink).
 - Bahuvrīhi: Combining nouns creates new nouns with descriptive meanings. Example: "kamal-phool" (lotus) from "kamal" (lotus) and "phool" (flower).

Morphological Analysis

- As a school of thought morphology is the creation of astrophysicist Fritz Zwicky. Zwicky contrived the methodology to address non quantified problems that have many apparent solutions.
- For problems to be suited to morphological analysis they are generally inexpressible in numbers.
- Other problems are better addressed with the more traditional decomposition method where complexity is broken down in parts and trivial elements are ignored to produce a simplified problem and solution.

Morphological Analysis

- Practical

Tokenization

- The first thing you need to do in any NLP project is text preprocessing.
- Preprocessing input text simply means putting the data into a predictable and analyzable form. It's a crucial step for building an amazing NLP application.
- There are different ways to preprocess text:
 - stop word removal,
 - tokenization,
 - stemming.

Tokenization

- Tokenization is the first step in any NLP pipeline. It has an important effect on the rest of your pipeline.
- A tokenizer breaks unstructured data and natural language text into chunks of information that can be considered as discrete elements.
- The token occurrences in a document can be used directly as a vector representing that document.
- This immediately turns an unstructured string (text document) into a numerical data structure suitable for machine learning.

Tokenization



Types of Tokenizers

- Word Tokenizer
- Sentence Tokenizer
- White Space Tokenizer
- Word Tokenizer
- Space Tokenizer
- Tab Tokenizer
- Line Tokenizer
- Tree Bank Word Tokenizer
- Tweet Tokenizer
- MWET tokenizer

Types of Tokenizers

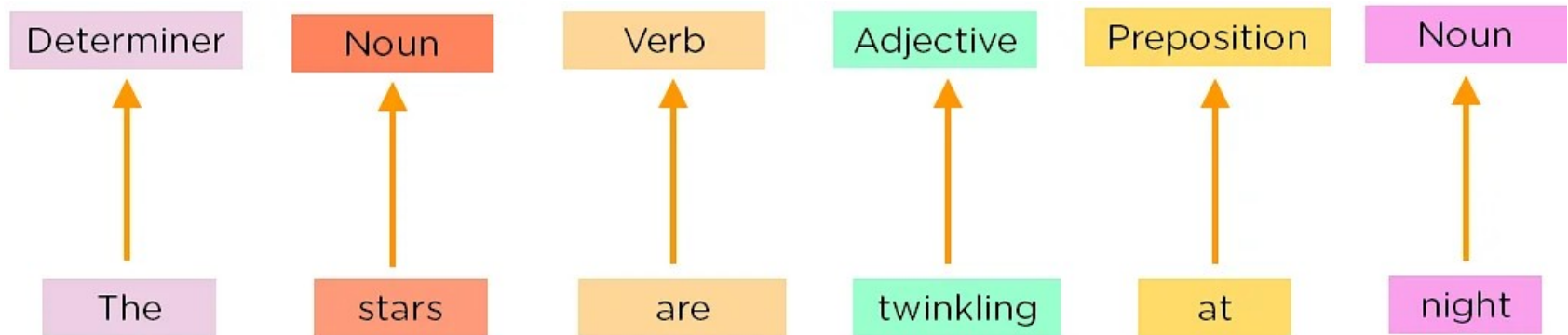
- Practical

POS Tagging

- Part-of-speech (POS) tagging is the process of assigning a word to its grammatical category, in order to understand its role within the sentence. Traditional parts of speech are nouns, verbs, adverbs, conjunctions, etc.
- Part-of-speech taggers typically take a sequence of words (i.e. a sentence) as input, and provide a list of tuples as output, where each word is associated with the related tag.
- Part-of-speech tagging is what provides the contextual information that a lemmatiser needs to choose the appropriate lemma.

Part of Speech Tagging

- Now, you must explain the concept of nouns, verbs, articles, and other parts of speech to the machine by adding these tags to our words. This is called 'part of'.



POS Tags

- CC coordinating conjunction
- CD cardinal digit
- DT determiner
- EX existential there (like: “there is” ... think of it like “there exists”)
- FW foreign word
- IN preposition/subordinating conjunction
- JJ adjective ‘big’
- JJR adjective, comparative ‘bigger’
- JJS adjective, superlative ‘biggest’
- LS list marker 1)
- MD modal could, will
- NN noun, singular ‘desk’
- NNS noun plural ‘desks’
- NNP proper noun, singular ‘Harrison’
- NNPS proper noun, plural ‘Americans’
- PDT predeterminer ‘all the kids’
- POS possessive ending parent’s
- PRP personal pronoun I, he, she
- PRP\$ possessive pronoun my, his, hers
- RB adverb very, silently,
- RBR adverb, comparative better
- RBS adverb, superlative best
- RP particle give up
- TO, to go ‘to’ the store.
- UH interjection, errrrrrrrm
- VB verb, base form take
- VBD verb, past tense took
- VBG verb, gerund/present participle taking
- VBN verb, past participle taken
- VBP verb, sing. present, non-3d take
- VBZ verb, 3rd person sing. present takes
- WDT wh-determiner which
- WP wh-pronoun who, what
- WP\$ possessive wh-pronoun whose
- WRB wh-adverb where, when

POS Tags

- Practical

Stopwords

- The words which are generally filtered out before processing a natural language are called stop words.
- These are actually the most common words in any language (like articles, prepositions, pronouns, conjunctions, etc) and does not add much information to the text.
- Examples of a few stop words in English are “the”, “a”, “an”, “so”, “what”.

Stopwords



Why to remove stopwords?

- Stop words are available in abundance in any human language.
- By removing these words, we remove the low-level information from our text in order to give more focus to the important information.
- In other words, we can say that the removal of such words does not show any negative consequences on the model we train for our task.
- Removal of stop words definitely reduces the dataset size and thus reduces the training time due to the fewer number of tokens involved in the training.

Do we remove stopwords always?

- The answer is no!
- We do not always remove the stop words. The removal of stop words is highly dependent on the task we are performing and the goal we want to achieve.
- For example, if we are training a model that can perform the sentiment analysis task, we might not remove the stop words.
- Movie review: “The movie was not good at all.”
- Text after removal of stop words: “movie good”

Stopwords Removal

- Practical

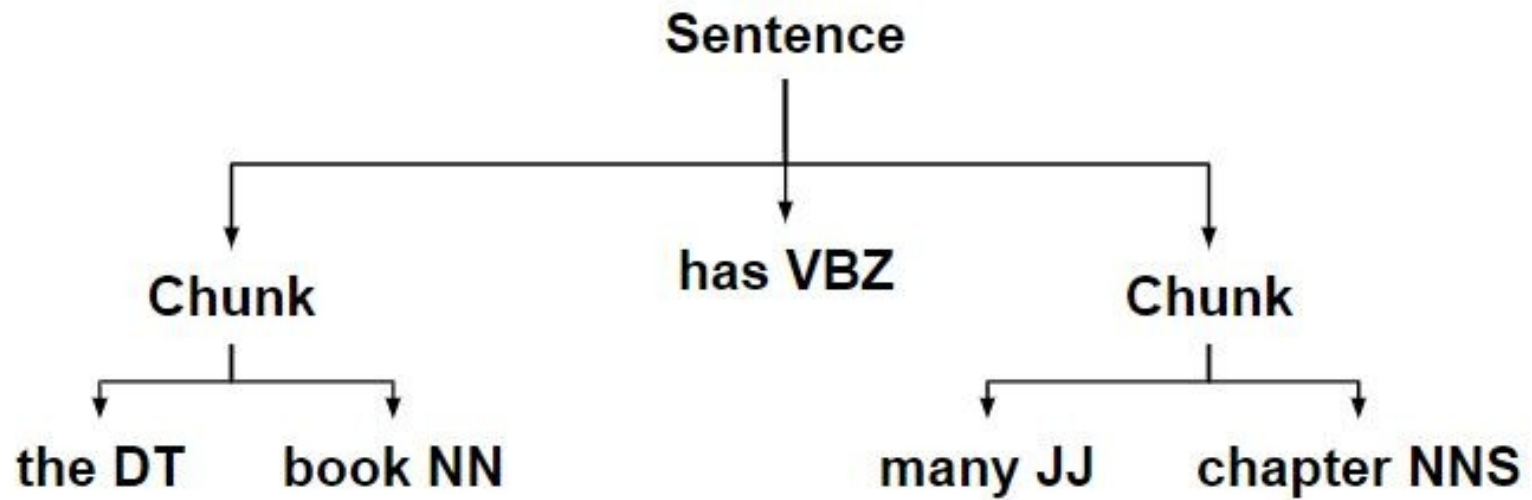
Chunking

- Chunking is defined as the process of natural language processing used to identify parts of speech and short phrases present in a given sentence.
- Recalling our good old English grammar classes back in school, note that there are eight parts of speech namely the noun, verb, adjective, adverb, preposition, conjunction, pronoun, and interjection.
- Also, in the above definition of chunking, short phrases refer to the phrases formed by including any of these parts of speech.

Chunking

- For example, chunking can be done to identify and thus group noun phrases or nouns alone, adjectives or adjective phrases, and so on. Consider the sentence below:
 - “I had burgers and pastries for breakfast.”
- In this case, if we wish to group or chunk noun phrases, we will get “burgers”, “pastries” and “lunch” which are the nouns or noun groups of the sentence.

Chunking



Chunking: Where it is used?

- Why would we want to learn something without knowing where it is widely used?!
- Chunking is used to get the required phrases from a given sentence.
- However, POS tagging can be used only to spot the parts of speech that every word of the sentence belongs to.
- When we have loads of descriptions or modifications around a particular word or the phrase of our interest, we use chunking to grab the required phrase alone, ignoring the rest around it.

Chunking: Types

- Chunking up:
 - Here, we don't dive deep; instead, we are happy with just an overview of the information. It just helps us get a brief idea of the given data.
- Chunking down:
 - Unlike the previous type of chunking, chunking down helps us get detailed information.
- So, if you just want an insight, consider “chunking up” otherwise prefer “chunking down”.

Chunking

- Practical

Named Entity Recognition

- Named entity recognition (NER) — sometimes referred to as entity chunking, extraction, or identification — is the task of identifying and categorizing key information (entities) in text.
- An entity can be any word or series of words that consistently refers to the same thing. Every detected entity is classified into a predetermined category.
- For example, an NER machine learning (ML) model might detect the word “MITU Skillologies” in a text and classify it as a “Company”.

Named Entity Recognition

- NER is a form of natural language processing (NLP), a subfield of artificial intelligence.
- NLP is concerned with computers processing and analyzing natural language, i.e., any language that has developed naturally, rather than artificially, such as with computer coding languages.

Named Entity Recognition

contentSkip to site indexPoliticsSubscribeLog InSubscribeLog InToday's PaperAdvertisementSupported **ORG** byF.B.I. Agent **Peter Strzok PERSON** ,
Who Criticized Trump PERSON in Texts, Is FiredImagePeter Strzok, a top **F.B.I. GPE** counterintelligence agent who was taken off the special counsel
 investigation after his disparaging texts about President **Trump PERSON** were uncovered, was fired. **CreditT.J. Kirkpatrick PERSON** for **The New York**
TimesBy Adam Goldman ORG and **Michael S. SchmidtAug PERSON** . **13 CARDINAL** , **2018WASHINGTON CARDINAL** — **Peter Strzok**
PERSON , the **F.B.I. GPE** senior counterintelligence agent who disparaged President **Trump PERSON** in inflammatory text messages and helped
 oversee the **Hillary Clinton PERSON** email and **Russia GPE** investigations, has been fired for violating bureau policies, Mr. **Strzok PERSON** 's lawyer
 said **Monday DATE** .Mr. Trump and his allies seized on the texts — exchanged during the **2016 DATE** campaign with a former **F.B.I. GPE** lawyer,
Lisa Page — in PERSON assailing the **Russia GPE** investigation as an illegitimate “witch hunt.” Mr. **Strzok PERSON** , who rose over **20 years**
DATE at the **F.B.I. GPE** to become one of its most experienced counterintelligence agents, was a key figure in **the early months DATE** of the
 inquiry.Along with writing the texts, Mr. **Strzok PERSON** was accused of sending a highly sensitive search warrant to his personal email account.The
F.B.I. GPE had been under immense political pressure by Mr. **Trump PERSON** to dismiss Mr. **Strzok PERSON** , who was removed **last summer**
DATE from the staff of the special counsel, **Robert S. Mueller III PERSON** . The president has repeatedly denounced Mr. **Strzok PERSON** in posts on

Named Entity Recognition

- Person
 - E.g., Elvis Presley, Audrey Hepburn, David Beckham
- Organization
 - E.g., Google, Mastercard, University of Oxford
- Time
 - E.g., 2006, 16:34, 2am
- Location
 - E.g., Trafalgar Square, MoMA, Machu Picchu
- Work of art
 - E.g., Hamlet, Guernica, Exile on Main St.

How NER used?

- NER is suited to any situation in which a high-level overview of a large quantity of text is helpful.
- With NER, you can, at a glance, understand the subject or theme of a body of text and quickly group texts based on their relevancy or similarity.

How NER used?

- Human resources
 - Speed up the hiring process by summarizing applicants' CVs; improve internal workflows by categorizing employee complaints and questions
- Customer support
 - Improve response times by categorizing user requests, complaints and questions and filtering by priority keywords

How NER used?

- Search and recommendation engines
 - Improve the speed and relevance of search results and recommendations by summarizing descriptive text, reviews, and discussions
 - Booking.com is a notable success story here
- Content classification
 - Surface content more easily and gain insights into trends by identifying the subjects and themes of blog posts and news articles

How NER used?

- Health care
 - Improve patient care standards and reduce workloads by extracting essential information from lab reports
 - Roche is doing this with pathology and radiology reports
- Academia
 - Enable students and researchers to find relevant material faster by summarizing papers and archive material and highlighting key terms, topics, and themes
 - The EU's digital platform for cultural heritage, Europeana, is using NER to make historical newspapers searchable

- Practical

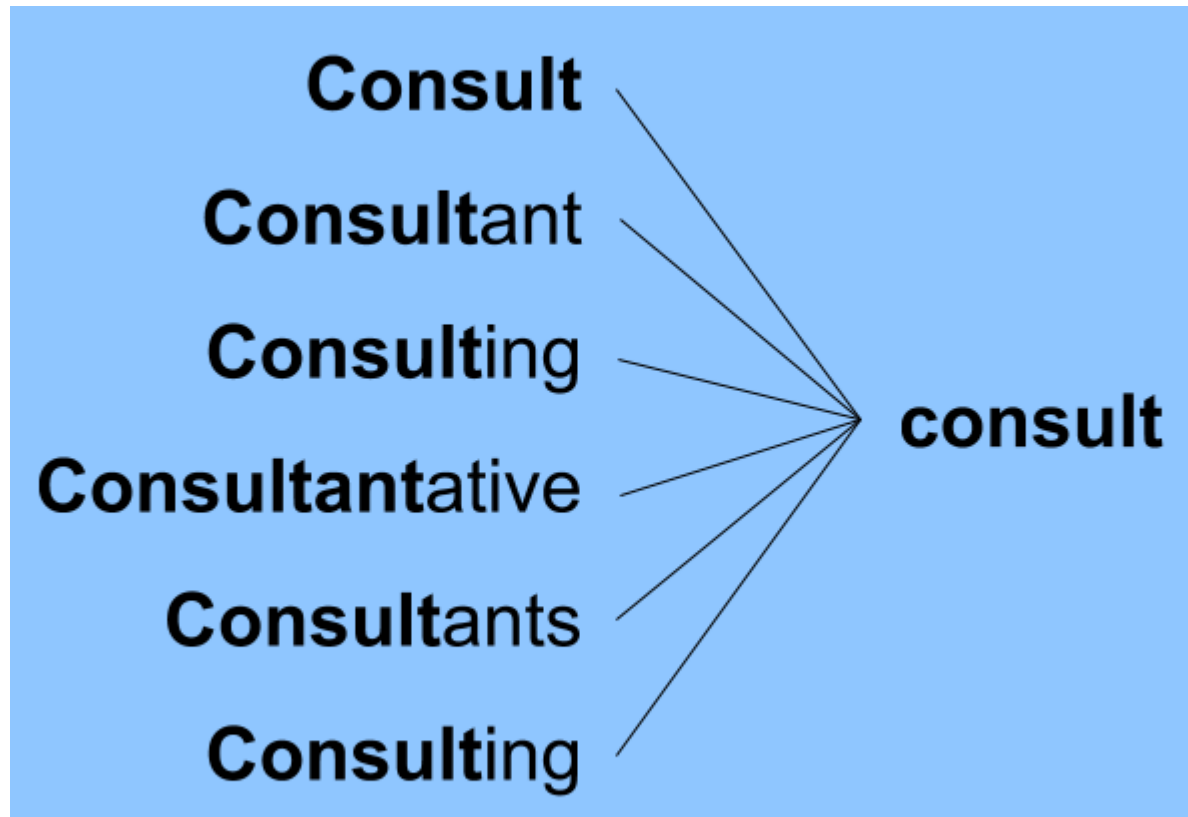
Stemming

- Stemming is the process of reducing a word to its stem that affixes to suffixes and prefixes or to the roots of words known as "lemmas".
- Stemming is important in natural language understanding (NLU) and natural language processing (NLP).
- Stemming is a part of linguistic studies in morphology as well as artificial intelligence (AI) information retrieval and extraction.

Stemming

- Stemming and AI knowledge extract meaningful information from vast sources like big data or the internet since additional forms of a word related to a subject may need to be searched to get the best results.
- Stemming is also a part of queries and internet search engines.
- Recognizing, searching and retrieving more forms of words returns more results.
- When a form of a word is recognized, it's possible to return search results that otherwise might have been missed.

Stemming



Stemming

- Practical

Stemming

Word	Porter	Lancaster	Lemmatiser
wrote	wrote	wrot	write
thinking	think	think	think
remembered	rememb	rememb	remember
relies	reli	rely	rely
ate	ate	at	eat
gone	gone	gon	go
won	won	won	win
ran	ran	ran	run
swimming	swim	swim	swim
mistreated	mistreat	mist	mistreat

Lemmatization

- Lemmatization is a text normalization technique used in Natural Language Processing (NLP), that switches any kind of a word to its base root mode.
- Lemmatization is responsible for grouping different inflected forms of words into the root form, having the same meaning.

Lemmatization

likes



like

better



good

worse



bad

Lemmatization

- Lemmatization is among the best ways to help chatbots understand your customers' queries to a better extent.
- Since this involves a morphological analysis of the words, the chatbot can understand the contextual form of the words in the text and can gain a better understanding of the overall meaning of the sentence that is being lemmatized.
- Lemmatization is also used to enable robots to speak and converse. This makes lemmatization a rather important part of natural language understanding (NLU) and natural language processing (NLP) in artificial intelligence.

Lemmatization

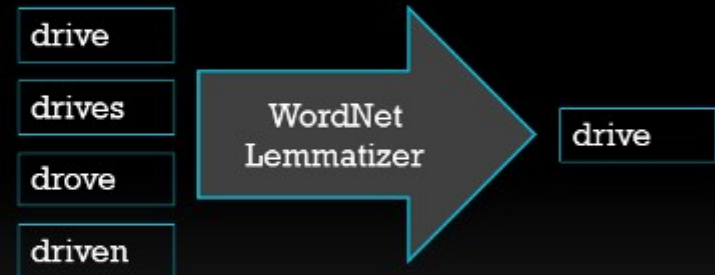
- Lemmatization is a vital part of Natural Language Understanding (NLU) and Natural Language Processing (NLP). It plays critical roles both in Artificial Intelligence (AI) and big data analytics.
- Lemmatization is extremely important because it is far more accurate than stemming.
- This brings great value when working with a chatbot where it is crucial to understand the meaning of a user's messages.
- The major disadvantage to lemmatization algorithms, however, is that they are much slower than stemming algorithms.

Stemming vs. Lemmatization

STEMMING

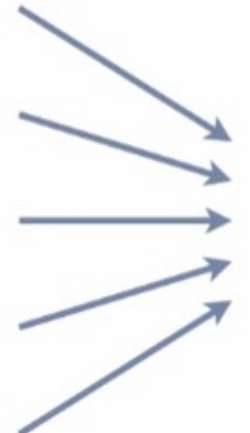


LEMMATIZATION



Stemming vs. Lemmatization

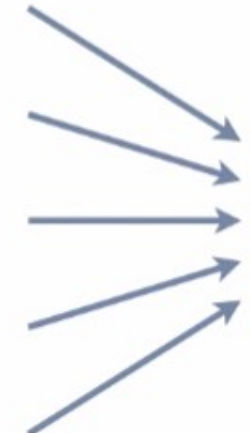
change
changing
changes
changed
changer



chang

The diagram illustrates the process of stemming. On the left, five words are listed: 'change', 'changing', 'changes', 'changed', and 'changer'. Five blue arrows point from each of these words to a single word on the right, 'chang', which is colored blue. This represents the process of reducing different inflected forms of a word to their common root.

change
changing
changes
changed
changer



change

The diagram illustrates the process of lemmatization. On the left, the same five words are listed: 'change', 'changing', 'changes', 'changed', and 'changer'. Five blue arrows point from each of these words to a single word on the right, 'change', which is colored green. This represents the process of reducing different inflected forms of a word to their base or dictionary form.

Semantic Parsing

- Semantic parsing, a crucial task in Natural Language Processing (NLP), delves deeper than syntax parsing by aiming **to extract the meaning** behind a sentence.
- It goes beyond understanding the grammatical structure (syntax) and focuses on converting natural language sentences into a formal representation that **captures their semantics**.

Semantic Parsing

- Core Objective:
 - Bridging the Gap: Semantic parsing bridges the gap between human language and machine understanding by translating natural language into a machine-readable format that expresses the intended meaning.

Types of Meaning Representations

- Logical Forms: These represent the meaning of a sentence in a logical format, often using predicate logic or lambda calculus. They capture the relationships between entities and actions described in the sentence.
- Abstract Meaning Representations (AMRs): This graphical format represents the meaning of a sentence as a network of interconnected concepts and relations.
- Execution Trees: These represent the meaning of a sentence as a sequence of instructions or actions to be performed by a machine.

Word Sense

- In Natural Language Processing (NLP), word sense refers to the specific meaning a word conveys within a particular context.
- Many words have multiple meanings, and identifying the intended sense in a given situation is crucial for accurate language understanding.

Word Sense

- Understanding Word Ambiguity:
 - Polysemy: This refers to words with multiple, related meanings.
 - For example, the word "bat" can refer to a flying mammal or a wooden club used in sports.
 - Homonymy: This refers to words with the same spelling and pronunciation but unrelated meanings.
 - For example, "bat" can also refer to a piece of fabric used to blindfold someone.

Wordnet

- WordNet is a large lexical database that groups words in the English language into sets of synonyms, called synsets.
- These synsets capture the different senses or meanings that a word can have.
- It's essentially a vast electronic thesaurus combined with elements of a dictionary, designed to provide a deeper understanding of word relationships.

Wordnet : Structure

- Synsets:
 - The core of WordNet is the synset, a collection of synonymous words representing a single concept.
 - For example, the synset {king, monarch, sovereign} represents the concept of a ruler.
- Semantic Relations:
 - WordNet connects synsets through various semantic relations,

Wordnet : Structure

- Semantic Relations: WordNet connects synsets through various semantic relations, including:
- Hypernymy/Hyponymy: A hierarchical relationship where a general term (hypernym) is linked to more specific terms (hyponyms) that fall under it. For example, "animal" is a hypernym of "dog" and "cat" (hyponyms).
- Meronymy/Holonymy: A part-whole relationship where a part (meronym) is linked to the whole it belongs to (holonym). For example, "wheel" is a meronym of "car" (holonym).

Wordnet : Structure

- **Antonymy:** A relationship between words with opposite meanings. For example, "happy" is an antonym of "sad".
- **Parts of Speech:** WordNet categorizes words into different parts of speech (nouns, verbs, adjectives, adverbs).

Wordnet : Why?

- Word Sense Disambiguation (WSD):
 - By providing information about different senses of a word and their relationships, WordNet helps identify the intended meaning of a word in context.
- Information Retrieval:
 - WordNet can be used to improve the accuracy of search engines by understanding the semantic relationships between words in a query and documents.

Wordnet : Limitations

- Coverage: While extensive, WordNet primarily focuses on everyday words and might not include domain-specific terminology.
- Granularity: Some argue that WordNet's synsets might be too fine-grained, leading to challenges in accurately WSD tasks.
- Language Focus: The current primary focus is on English, with limited availability for other languages.

Word sense disambiguation (WSD)

- Word sense disambiguation (WSD) is a crucial task in Natural Language Processing (NLP) that deals with identifying the specific meaning of a word within a particular context.
- Many words in a language have multiple meanings, and WSD aims to pinpoint the intended sense based on the surrounding words and the overall sentence structure.

Word sense disambiguation (WSD)

- Imagine encountering the sentence "The bank is on the river." Here, "bank" could refer to the financial institution or the edge of a body of water.
- Without WSD, a machine might struggle to understand the true meaning of the sentence.
- However, with WSD, the machine can identify the intended sense based on the context (river suggesting the edge meaning).

WSD: Applications

- Machine Translation: Choosing the correct translation equivalent based on the intended meaning in the source language.
- Text Summarization: Identifying the key points of a text and ensuring the summary accurately reflects the meaning of the original content.
- Sentiment Analysis: Understanding the sentiment expressed in a sentence requires considering the specific sense of the words used.
- Question Answering: Answering a question accurately might rely on identifying the intended meaning of words in the question itself and the relevant passages.

WSD: challenges

- **Context Dependency:** The meaning of a word can be highly dependent on the specific context in which it appears. For instance, "bat" can mean a flying mammal or a sports equipment depending on the context.
- **Limited Resources:** Training accurate WSD models often requires vast amounts of annotated data, where each word instance is labeled with its specific sense in that context. Creating such data can be expensive and time-consuming.
- **Word Ambiguity:** Some words have very subtle differences in meaning that can be challenging for WSD models to distinguish. For example, the difference between "light" as in weight and "light" as in illumination.

WSD: Approaches

- **Dictionary-based Methods:** These methods rely on pre-defined dictionaries that provide different senses for a word. The system analyzes the context of the word and chooses the sense that best fits the surrounding words and the overall sentence meaning.
- **Machine Learning Methods:** Supervised learning algorithms can be trained on large datasets of text annotated with word senses. These models learn to identify patterns in the context that indicate the intended meaning of a word.
- **Unsupervised Learning Methods:** These methods group words with similar contexts together, assuming that words appearing in similar contexts likely share the same sense.

WSD: Algorithms

- Lesk Algorithm: This classic approach leverages pre-defined dictionaries with information about word senses and their definitions. It compares the overlap between the definitions of the target word's senses and the definitions of surrounding words in the sentence. The sense with the most overlap is chosen as the most likely meaning.
- Gloss Overlap: Similar to Lesk, this method compares the words used in the definitions of the target word's senses with the words in the surrounding context. The sense with the highest number of matching words is considered the most likely meaning.

WSD: Algorithms

- Machine Learning Methods:
 - Naive Bayes:
 - Support Vector Machines
 - Neural Networks (RNN)

WSD: Algorithms

- Unsupervised Learning Methods:
 - Clustering
 - EM Algorithms

Lesk Algorithm

- 1. Input:
 - The algorithm takes two main inputs:
 - A sentence containing a word with multiple senses (ambiguous word).
 - A dictionary resource that provides definitions for the different senses of words.
- 2. Identify Ambiguous Word:
 - The algorithm first identifies the word in the sentence that has multiple possible meanings (the ambiguous word).

Lesk Algorithm

- 3. Extract Dictionary Senses:
 - The algorithm retrieves all the different senses (definitions) listed for the ambiguous word in the dictionary.
- 4. Analyze Context:
 - The algorithm defines a "context window" around the ambiguous word in the sentence. This context window typically includes a few words before and after the ambiguous word.

Lesk Algorithm

- 5. Signature Creation:
 - For each sense of the ambiguous word retrieved from the dictionary:
 - The algorithm creates a "signature" which is essentially a set of words.
 - This signature can be formed by including all the words present in the definition of that particular sense.

Lesk Algorithm

- 6. Overlap Calculation:
 - The algorithm calculates the overlap between the signature of each sense and the words in the context window of the sentence.
 - This overlap is typically measured as the number of words that appear in both the signature and the context window (excluding stop words like "the", "a", etc.).

Lesk Algorithm

- 7. Disambiguation:
 - The algorithm identifies the sense of the ambiguous word with the highest overlap between its signature and the context window. This sense is considered the most likely meaning based on the surrounding words.
- 8. Output:
 - The algorithm outputs the disambiguated word, which is the ambiguous word along with its most likely sense according to the analysis.

Lesk Algorithm: Example

- Sentence:
 - "I went to the bank to deposit my money."
- Ambiguous Word: "bank"
- Dictionary Senses:
 - bank (financial institution)
 - bank (edge of a river)
- Context Window: "the bank" (assuming a small window)

Lesk Algorithm: Example

- Signature Creation:
 - bank (financial institution) signature: {financial institution, money, deposit}
 - bank (edge of a river) signature: {river, edge}
- Overlap Calculation:
 - Overlap (financial institution): 2 (money, deposit)
 - Overlap (edge of a river): 0
- Disambiguation:
 - Based on the highest overlap (2), the algorithm suggests "bank (financial institution)" as the most likely meaning in this context.

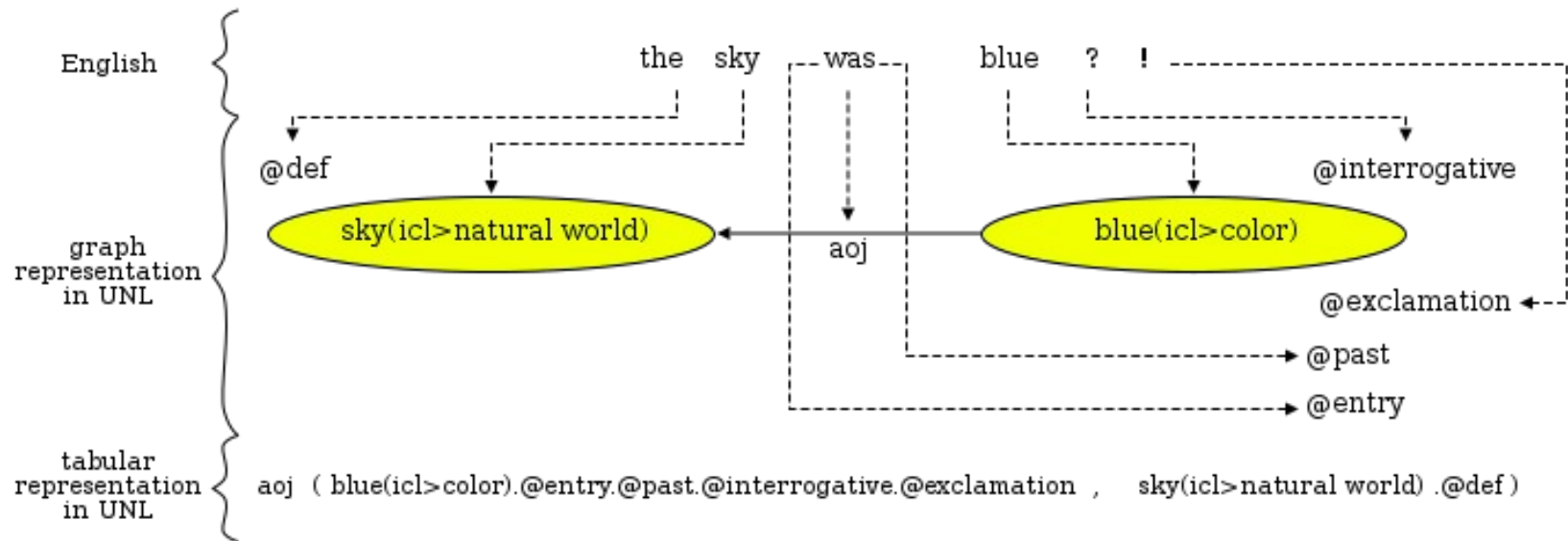
Universal Networking Language

- Universal Networking Language (UNL) is a declarative formal language specifically designed to represent semantic data extracted from natural language texts.
- It can be used as a pivot language in interlingual machine translation systems or as a knowledge representation language in information retrieval applications.

Universal Networking Language

- UNL is designed to establish a simple foundation for representing the most central aspects of information and meaning in a machine- and human-language-independent form.
- As a language-independent formalism, UNL aims to code, store, disseminate and retrieve information independently of the original language in which it was expressed.
- In this sense, UNL seeks to provide tools for overcoming the language barrier in a systematic way.

Universal Networking Language



- <https://www.cfilt.iitb.ac.in>

Synonyms and Antonyms

- NLTK Wordnet can be used to find synonyms and antonyms of words.
- NLTK Corpus package is used to read the corpus to understand the lexical semantics of the words within the document.
- A WordNet involves semantic relations of words and their meanings within a lexical database.
- The semantic relations within the WordNet are hypernyms, synonyms, holonyms, hyponyms, meronyms.
- NLTK WordNet includes the usage of synsets for finding the words within the WordNet with their usages, definitions, and examples.

Synonyms

- To find the synonyms of a word with NLTK WordNet, the instructions below should be followed.
 - Import NLTK.corpus
 - Import WordNet from NLTK.Corpora
 - Create a list for assigning the synonym values of the word.
 - Use the “synsets” method.
 - use the “syn.lemmas” property to assign the synonyms to the list with a for loop.
 - Call the synonyms of the word with NLTK WordNet within a set.

Antonyms

- To find the Antonyms of a Word with NLTK WordNet and Python, the following instructions should be followed.
 - Import NLTK.corpus
 - Import WordNet from NLTK.Corpus
 - Create a list for assigning the synonym values of the word.
 - Use the “synsets” method.
 - use the “syn.lemmas” property to assign the synonyms to the list with a for loop.
 - Use the “antonyms()” method with “name” property for calling the antonym of the phrase.
 - Call the antonyms of the word with NLTK WordNet within a set.

POS Tagging for Synonym and Antonym

- To find the Antonyms of a Word with NLTK WordNet and Python, the following instructions should be followed.
 - Import NLTK.corpus
 - Import WordNet from NLTK.Corpus
 - Create a list for assigning the synonym values of the word.
 - Use the “synsets” method.
 - use the “syn.lemmas” property to assign the synonyms to the list with a for loop.
 - Use the “antonyms()” method with “name” property for calling the antonym of the phrase.
 - Call the antonyms of the word with NLTK WordNet within a set.

POS Tagging for Indian Languages

```
# -*- coding: utf-8 -*-  
from nltk.corpus import indian  
from nltk.tag import tnt  
import nltk  
  
print 'Indian File IDs: ', indian.fileids()  
  
print 'Number of Characters:'  
for ch in indian.fileids():  
    print ch  
    print len(indian.raw(ch))
```

POS Tagging for Indian Languages

```
print 'Number of Words:'  
for wd in indian.fileids():  
    print wd  
    print len(indian.words(wd))
```

```
print 'Number of Sentences:'  
for st in indian.fileids():  
    print st  
    print len(indian.sents(st))
```

```
sents = indian.sents('marathi.pos')  
for sen in sents:  
    print sen[0]
```

POS Tagging in Marathi

```
pos = indian.tagged_sents('marathi.pos')
for sent in pos:
    print sent[0][0], sent[0][1]

train_data = indian.tagged_sents('marathi.pos')
tnt_pos_tagger = tnt.TnT()
tnt_pos_tagger.train(train_data)

word = 'आणि शिक्षण तत्पूर्वी सुरु केले'
tags = tnt_pos_tagger.tag(nltk.word_tokenize
                          (word.decode('utf-8')))
print tags
for tag in tags:
    print 'Word is:', tag[0], 'and POS is:', tag[1]
```

POS Tags output

```
Word is: अणि and POS is: CC  
Word is: शि्षण and POS is: NN  
Word is: तत्पूर्वी and POS is: PRP  
Word is: सुरु and POS is: JJ  
Word is: केल्ले and POS is: VM
```

Thank you

This presentation is created using LibreOffice Impress 7.4.1.2, can be used freely as per GNU General Public License



@mitu_skillologies



@mITuSkillologies



@mitu_group



@mitu-skillologies



@MITUSkillologies

kaggle

@mituskillologies

Web Resources

<https://mitu.co.in>

<http://tusharkute.com>



@mituskillologies

contact@mitu.co.in
tushar@tusharkute.com