Natural Language Processing

# Introduction to NLP

|  |
| --- |
| **NLP: Areas of Application:** |
| **Understanding Text:**     * There are 3 modules * **Lexical Processing** * Identify the language and **Syntactical processing** * Derive the meaning from the language, **semantic processing**   Once all the 3 modules are done, we can apply the processed language to some of the application shown in above picture.   * In the **lexical processing** it needs to find the underlying term for a given sentence/word/random text.      * In the **Syntactical processing** there are 2 kind of things done. * Identify the language * Try to understand the word meaning from the sentence. In below sentence “can” is having different meaning in different sentence. How to analyse the sentence.        * In **semantic processing**:       Discourse context is basically whether the meaning of the sentence coming out well or not. Synonym and antonyms of the word.  Deep Learning Architecture for NLP is quite different which is not covered in NLP section. Traditional NLP models are covered in the course. |
|  |
| **Text Encoding:**        [**https://docs.python.org/3/howto/unicode.html**](https://docs.python.org/3/howto/unicode.html)    The default encoding for strings in python is Unicode UTF-8. You can also look at [this](https://mothereff.in/utf-8) UTF-8 encoder-decoder to look how a string is stored. Note that, the online tool gives you the hexadecimal codes of a given string. ([**https://mothereff.in/utf-8**](https://mothereff.in/utf-8)**)** |

## Regular Expression

|  |
| --- |
| Refer to “Regular+Expressions” Jupyter notebook |

# Basic Lexical Processing (Module-1)

|  |  |  |
| --- | --- | --- |
| **Word Frequencies and Stop Words:** | | |
|  | |  |
|  | |  |
| In below Spam and Ham message the frequently occurred word should be identified which are shown below. Which are nothing but the stop words. | | In the Spam text, few words are shown above can be categorized as other section to take separate action. |
|  | | |
|  | | |
| **Tokenisation:**        **Bag-of-Words Representation:** | | |
|  |  | |
| **Stemming and Lemmatization:**                **TF-IDF Representation:** | | |

# Advanced Lexical Processing (Module-1)

|  |
| --- |
| **Canonicalization:**       * Converting the word into its base form is one of the major steps in the “Lexical Processing” * In the basic “Lexical Processing” we have studied “Stemming” and “Lemmatization” where we are converting the words to its original form. Such process is called canonicalization. * “Stemming” and “Lemmatization” are the basic steps of canonicalization where singular to plural or present tense to past or something similar kind of words are converted. * In this advance form we will learn 3 major technique. * Deal with the **Miss Spelling** using “Edit Distance” technique Dynamic Programming to find/suggest the words by editing minimum number of letters from original words, if original word not present in the dictionary and minimum edit distance word present in dictionary. * Some of the words are having different pronunciation in British and American English. To avoid the redundant, need to convert both types to single type by checking their “**Phonetic Hash**” value. * Some of the words meaning is by combining multiple words together which can be called as Term. E.g: IIIT-B (International Institute of Information Technology Bangalore) should be consider as single term in place of individual word taken as separate word. “**Pointwise Mutual Information**” technique can be used consider such term as single word. |
| **Phonetic Hashing:** |
|  |
| **Spell-Corrector:**  Using “Edit Distance” dynamic programming, Spell-Corrector code is being written. |
| **Pointwise Mutual Information:** |

# Introduction to Syntactic Processing (Module-2)

|  |  |
| --- | --- |
| The What and Why of Syntactical Processing:               * **Word Orders:** In above example bag of word in lexical processing is same but order of the words gives a completely different meaning. 1st sentence in the example is meaningful where the 2nd sentence is not meaningful (Man bites dog) * **Role of Stop words:** During our lexical processing, we used to drop the stop words. But in above example, stop words are playing major role to give the meaning of the sentence. * **Role of Morphological Forms:** In above example, workers, working and work are having different meaning. In the stemming and lemmatization, we used to consider all these 3 words are same in lexical processing. To understand the meaning of the sentence morphological form is important. * **Roles of Parts of Speech:** In above example, driving is having different meaning based on its position in the sentence. Hence position of the word in the sentence matters to understand the meaning of the word. * **Dependencies:** In above example, 1st sentence capital and India are interrelated where in the 2nd sentence name of the country is the dependant word. | |
| Parsing: | |
|  |  |
|  |  |
| <https://explosion.ai/demos/displacy?text=The%20little%20boy%20went%20to%20the%20park&model=en_core_web_sm&cpu=0&cph=0> | |

## Parts of Speech (POS)

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
| Different Approaches to POS Tagging: |
| * In “Lexicon Based tagger” Most frequently used tag for a word will be replaced in the sentence for that particular word. In above example “driving” comes as VB tag more compare to other tags. Hence, we will use “driving” as VB in the complete sentence. |
| * In “Rule based tagger” we replace some of the tags with some other tags based on some pre-defined rule. |
| * In “Stochastic Tagger” we set the tag in sequential manner based on probability of occurrence of the tags in a given sentence. |
|  |
|  |
| 1. Lexicon and Rule Based POS Tagging:      * Refer the python notebook to understand Lexicon based POS tagging |
| 1. Stochastic Parsing |
|  |
| * Use following formula |
| The Viterbi Heuristic:     1. By considering the probability of each tags shown below, tag of “The” calculated as DT as probability of tag DT for “The” at beginning is higher than other tags.      1. For the next word “High”, last word tag is taken in to consideration. As last word tag was “DT”, for calculation of tag probability for “High” following calculation is followed.          1. The next word “cost” taken previous word tag “JJ” for its own tag calculation.     \*\* The above method of calculation is called “Hidden Markov Model” or “Viterbi Algorithm” |
|  |
| Markov Chain and HMM: |
| Explanation Problem:         * Here “The” starts with DT|Start, NN|Start, JJ|Start with emission probability of “The”. The calculation came out with DT having higher probability. * In the next step “high” being calculated by taking previous tag as DT. |
|  |
|  |
| Learning HMM Model Parameters: |
|  |
|  |
| HMM and Viterbi Algorithm Pseudocode: |
| Deep Learning Based POS Tagging:    Summary: |
| Graded Question: |
|  |

# Parsing (Module-2)

|  |
| --- |
| * In the above 2 sentence POS tagging is same but relation between the words are not proper which makes 1st sentence meaning less and can’t be identified by only doing POS tagging. |

## Constituency Grammars

|  |
| --- |
|  |
|  |
|  |
| * In above context free grammar, we have 2 part. Left part is tag/word which can be used for the right part. E.g: S can be used in NP VP scenario and so on..      * A given tag can be written as tags or set of words.   Top Down Parsing:   * Start from start Symbol and go till the required string.   Bottom Up Parsing:   * From the desired string come till the start symbol. |
|  |

## Top-Down Parsing

|  |
| --- |
|  |
|  |
|  |

## Bottom-Up Parsing

|  |
| --- |
|  |
|  |

## Probabilistic CFG

|  |
| --- |
|  |
| * Man was in the park with a telescope sees the dog.      * Man saw the dog in the park where telescope is there.      * Man saw the dog in the park with the telescope |
|  |
|  |
|  |
|  |

## Chomsky Normal Form

|  |
| --- |
|  |

## Dependency Parsing

|  |
| --- |
|  |

## Summary

|  |
| --- |
|  |
|  |

# Information Extraction

|  |
| --- |
| Information Extraction: |
|  |
|  |
| <https://stackoverflow.com/questions/40879520/nltk-convert-a-chunked-tree-into-a-list-iob-tagging> |
|  |

# Summary (Information Extraction):

|  |
| --- |
|  |

# CRFs (Conditional Random Fields)

|  |
| --- |
|  |
|  |
|  |

## CRF Model Architecture

|  |
| --- |
|  |

# Semantic Processing (Module-3)

## Introduction to Semantic Processing

|  |  |  |
| --- | --- | --- |
|  | | |
|  | | |
|  | | |
|  | | |
|  | | |
|  | | |
| **Entity and Entity Types:** | | |
|  | | * Here one employee reported to another employee. Hence, we have 2 instances of the employee but just 1 entity type. i.e: Employee |
|  | |  |
|  | | |
|  | | |
|  | | |
|  | | |
| * In Reification, we combine multiple entity to give a superset meaning. Which is basically giving the overall meaning for multiple entity. | | |
| Rdf:  <https://www.w3.org/TR/rdf-concepts/>  <https://www.w3.org/TR/PR-rdf-syntax/> | | |
|  | | |
| **Schema:**      <https://schema.org/> | | |
| <https://schema.org/docs/gs.html>  <https://schema.org/docs/financial.html> | | |
| **Semantic Association:** | | |
| e.g: Cat **is an** Animal. Here Animal is the superset. If Animal property is breath, the same for cat as well. | e.g: (Scenario: 1) Orissa **is in** India. If someone is not having permission to enter India, then so for orissa.  (Scenario: 2) Engine is in Car. | |
|  | | |
| Aboutness: | | |
|  | | |
|  | | |
|  | | |
|  | | |
| DATABASES – WordNet and Concept Net | | |
|  | | |
| <https://wordnet.princeton.edu/>  <https://stevenloria.com/wordnet-tutorial/>  <http://conceptnet.io/> | | |
| Word Sense Disambiguation | | |

### 9.1.1 Word Sense Disambiguation – Naïve Bayes

|  |
| --- |
|  |