DATA ANALYSIS AND VISUALIZATION

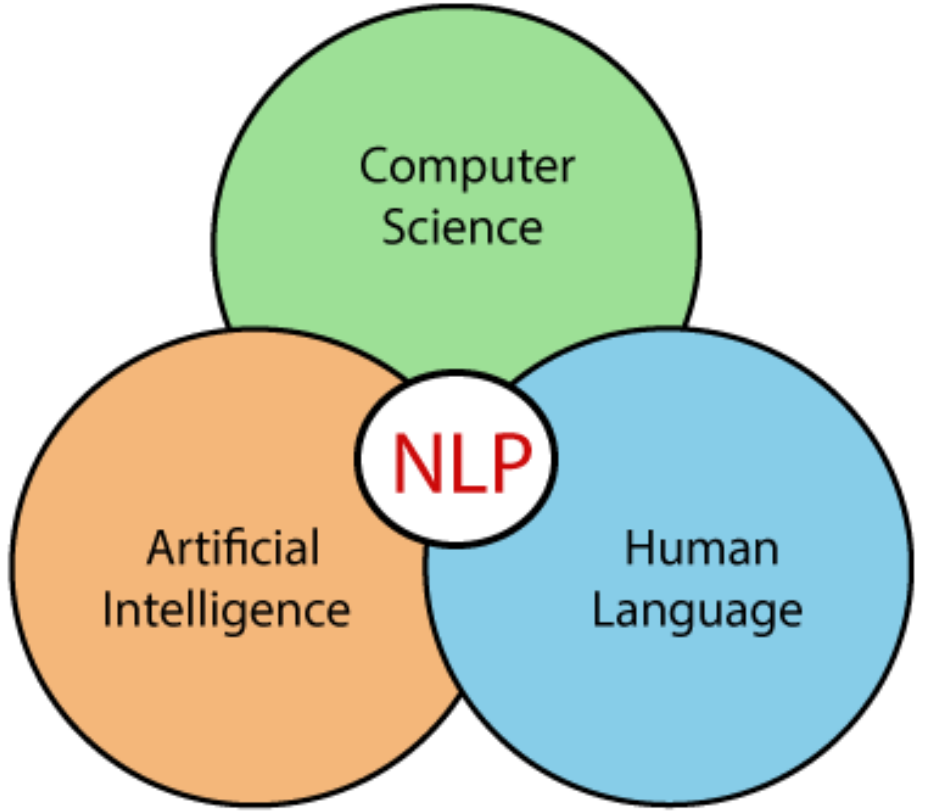
INSTRUCTOR: UMME AMMARAH

NATURAL LANGUAGE PROCESSING (NLP)

INTRODUCTION

INTRODUCTION

- How to program computers to process and analyze large amounts of natural language data.
- NLP gives machine ability to understand human language better and to assist in language related tasks.
- NLP uses statistical models, machine learning, deep learning to understand text.



APPLICATIONS

- Contextual advertisements
- Email clients – spam filtering, smart reply
- Social Media – controlling content for display
- Search engines
- Chatbots
- Caption generations
- Text summarization
- Auto correct
- Market intelligence

NLP LEXICAL TASKS

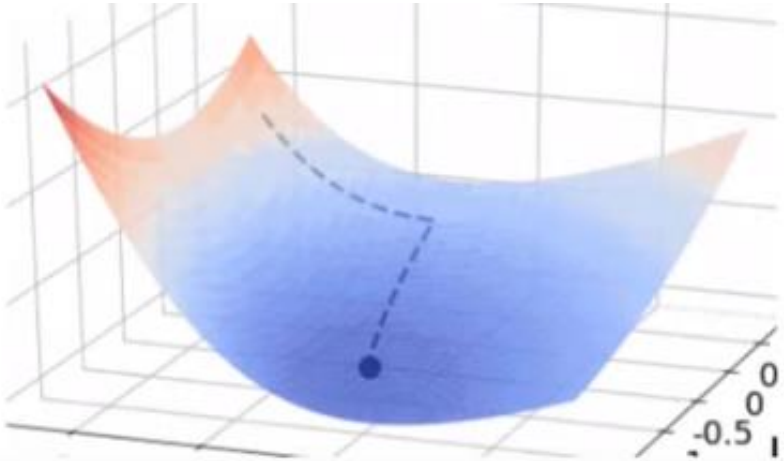
- Part of speech tagging
- Identifying relationships
- Word sense disambiguation
- Named entity recognition
- Co-reference resolution

HIGH LEVEL TASK

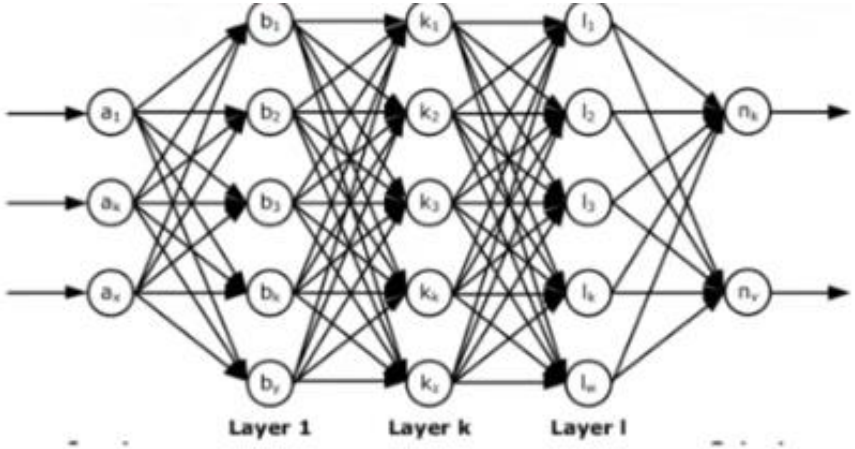
- Sentiment analysis
- Text/Document Classification
- Speech recognition
- Recognizing Textual Entailment
- Natural language generation
- Language Modelling

APPROACHES TO NLP

Heuristic Methods



Machine Learning Based Methods



Deep Learning Based Methods

USEFUL PYTHON LIBRARIES FOR NLP

- NLTK
- SpaCy
- TextBlob
- Gensim
- PolyGlot

TERMINOLOGIES

- **Document**: collection of many words.
- **Vocabulary**: set of unique words in a document.
- **Token:** basic unit of discrete data. It often refers to a single word or a punctuation mark.
- **Corpus:** collection of documents.
- **Context** : Context of a word/token is the words/tokens that surround it on left and right in the document.
- **Vector Embedding**: Vector based numerical representations of text.

NLP PIPELINE

- Data Acquisition
- Text Preparation
- Feature Engineering
- Modelling

TEXT PREPARATION

- Cleaning
  - Html tags cleaning
  - Removing emojis
  - Spelling checker
- Basic Preprocessing
  - Tokenization (sentence or word)
  - Stop word cleaning
  - Stemming/Lemmatization
  - Lower case conversion
  - Language detection
- Advance Preprocessing
  - POS tagging
  - Parsing
  - coreference resolution

FEATURE ENGINEERING

- Converting words and sentences into numeric vectors for use with computational algorithms.

- The process of representing a token/a sentence as a numerical vector is called ‘Embedding’,

PROPERTIES FOR EMBEDDING

- It should uniquely identify a word.
- Arithmetic operations should be possible on these representations.
- Tasks like computing word similarities and relationships should be easy.
- Should capture the morphological, syntactic and semantic similarity among words.
- It should be easy to map from word to its embedding and vice versa.

PROMINENT TECHNIQUES OF EMBEDDING

- One Hot Encoding
- Bag of Words (BoW)
- Term Frequency-Inverse Document Frequency (TF-IDF)
- Word2Vec
- GloVe
- Fast Text
- Transformers

ONE HOT ENCODING

- Every word is represented as a unique 'One-Hot' binary vector of 0s and 1s.

- For every unique word in the vocabulary, the vector contains a single 1 and rest all values are 0s, the position of 1 in the vector uniquely identifies a word.

X	MacOS	Linux	Windows
MacOS	1	0	0
Windows	0	0	1
MacOS	1	0	0
Linux	0	1	0
Windows	0	0	1

BAG OF WORDS (BOW)

- A text (such as a sentence or a document) is represented as the bag (multiset) of its words, disregarding grammar and even word order but keeping multiplicity.

- It involves two things:
  - Vocabulary of known words
  - Measure of the presence of known words

EXAMPLE

- If we count the number of unique words in all the four reviews we will be getting a total of 12 unique words.

Example: If we are given 4 reviews for an Italian pasta dish.

Review 1 : This pasta is very tasty and affordable.

Review 2: This pasta is not tasty and is affordable.

Review 3 : This pasta is delicious and cheap.

Review 4: Pasta is tasty and pasta tastes good.

- 'This' , 'pasta', 'is', 'very', 'tasty', 'and', 'affordable', 'not', 'delicious', 'cheap', 'tastes', 'good'
- Review 4:

1	2	3	4	5	6	7	8	9	10	11	12
0	2	1	0	1	1	0	0	0	0	1	1

BOW

- After converting the documents into such vectors we can compare different sentences and calculate the Euclidean distance between them so as to check if two sentences are similar or not. If there would be no common words distance would be much larger and vice-versa.

- BOW doesn't work very well when there are small changes in the terminology we are using as here we have sentences with similar meaning but with just different words.

TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY (TF-IDF)

- TF-IDF is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.
- The tf–idf value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word.

- This concept includes:
  - Count the number of times each word appears in a document.
  - Calculate the frequency that each word appears in a document out of all the words in the document.

TERM FREQUENCY

- Term frequency (TF) shows how frequently an expression (term, word) occurs in a document.

- Probability of finding a word in a document.

$$TF(w_i, r_j) = \frac{\text{No. of times } w_i \text{ occurs in } r_j}{\text{Total no. of words in } r_j}$$

INVERSE DOCUMENT FREQUENCY (IDF)

- IDF is a measure of how much information the word provides, i.e., if it's common or rare across all documents.
- It is used to calculate the weight of rare words across all documents in the corpus.
- It is the logarithmically scaled inverse fraction of the documents that contain the word.

$$\mathrm{idf}(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

TF-IDF

$$\mathrm{tfidf}(t, d, D) = \mathrm{tf}(t, d) \cdot \mathrm{idf}(t, D)$$

- TF-IDF gives larger values for less frequent words in the document corpus. TF-IDF value is high when both IDF and TF values are high i.e the word is rare in the whole documents but frequent in a document.

EXAMPLE

- Sentence 1: The car is driven on the road.
- Sentence 2: The truck is driven on the highway.

Word	TF		IDF	TF*IDF	
	A	B		A	B
The	1/7	1/7	log(2/2) = 0	0	0
Car	1/7	0	log(2/1) = 0.3	0.043	0
Truck	0	1/7	log(2/1) = 0.3	0	0.043
Is	1/7	1/7	log(2/2) = 0	0	0
Driven	1/7	1/7	log(2/2) = 0	0	0
On	1/7	1/7	log(2/2) = 0	0	0
The	1/7	1/7	log(2/2) = 0	0	0
Road	1/7	0	log(2/1) = 0.3	0.043	0
Highway	0	1/7	log(2/1) = 0.3	0	0.043