Course: Artificial Intelligence and Machine Learning Code: 20CS51I

WEEK- 11

NATURAL LANGUAGE PROCESSING

Session 1

11.1 UNDERSTANDING NATURAL LANGUAGE PROCESSING

Language is the primary means of communication used by humans. It is the tool we use to express the greater part of our ideas and emotions. It shapes thought, has a structure, and carries meaning. Learning new concepts and expressing ideas through them is so natural that we hardly realize how we process natural language.

"A natural language or ordinary language is any language that has evolved naturally in humans through use and repetition without conscious planning or premeditation." In contrast to artificial languages like Python, C, Java, etc. natural languages like English, Kannada, Hindi, French, etc. have evolved over time and use and it is difficult to express them in strict formal rules.

"Natural Language Processing (NLP) is a branch of Artificial Intelligence that helps computers to understand, interpret and manipulate human languages to analyze and derive its meaning."

NLP incorporates **machine learning models, statistics, and deep learning models** into **computational linguistics** i.e., rule-based modeling of human language to allow computers to understand text, spoken words and understands human language, intent, and sentiment. It helps developers to organize and structure knowledge to perform tasks like translation, summarization, named entity recognition, relationship extraction, speech recognition, topic segmentation, etc.

11.1.1 Why we need NLP?

Natural language is full of **ambiguities.** Ambiguity can be referred to as the ability of having more than one meaning or being understood in more than one way.

There are different types of ambiguities present in natural language:

1. Lexical Ambiguity: It is defined as the ambiguity associated with the meaning of a single word. A single word can have different meanings. Also, a single word can be a noun, adjective, or verb. For example: The word "bank" can have different meanings. It can be a financial bank or a riverbank. Similarly, the word "clean" can be a noun, adverb, adjective, or verb.

2. Syntactic Ambiguity: It is defined as the ambiguity associated with the way the words are parsed. For example: The sentence "Visiting relatives can be boring." This sentence can have two different meanings. One is that visiting a relative's house can be boring. The second is that visiting relatives at your place can be boring.

3. Semantic Ambiguity: It is defined as ambiguity when the meaning of the words themselves can be ambiguous. For example: The sentence "Mary knows a little French." In this sentence the word "little French" is ambiguous. As we don't know whether it is about the language French or a person.

Let us now try to understand why NLP is considered hard using a few examples.

1. "There was not a single man at the party"

- Does it mean that there were no men at the party? or
- Does it mean that there was no one at the party?
- Here does man refer to the gender "man" or "mankind"?

2. "The chicken is ready to eat"

- Does this mean that the bird (chicken) is ready to feed on some grains? or
- Does it mean that the meat is cooked well and is ready to be eaten by a human?

3. "Google is a great company." and "Google this word and find its meaning."

- Google is being used as a noun in the first statement and as a verb in the second.

4. The man saw a girl with a telescope.

- Did the man use a telescope to see the girl? or
- Did the man see a girl who was holding a telescope?

This is a primary reason why NLP is considered hard. Another reason why NLP is hard is because it deals with the extraction of knowledge from unstructured data.

11.2 NLP APPROACHES

11.2.1 Rule Based NLP

Rule-based approaches are the oldest approaches to NLP. The rule-based or grammar-based approach implies that a human is involved in the process of stepwise system development and improvement. A rule-based NLP system simply follows these rules to categorize the language it's analyzing. If the rule doesn't exist, the system will be unable to 'understand' the human language and thus will fail to categorize it. Regular expressions and context free grammars are some examples of rule-based approaches to NLP.

Advantages of Rule based approach:

- A rule-based system is good at capturing a specific language phenomenon: it will decode the linguistic relationships between words to interpret the sentence.
- It tends to focus on pattern-matching or parsing.

- Rule-based systems are low precision, high recall, meaning they can have high performance in specific use cases, but often suffer performance degradation when generalized.

Disadvantage of Rule-based approach:

- It requires skilled experts: it takes a linguist or a knowledge engineer to manually encode each rule in NLP.
- Rules need to be manually crafted and enhanced all the time.
- Moreover, the system can become so complex, that some rules can start contradicting each other.
- Accuracy of the NLP system is dependent on the rules provided
- They cannot easily scale to accommodate a seemingly endless stream of exceptions or the increasing volumes of text and voice data.

11.2.2 Statistical NLP

Statistical NLP aims to perform statistical inference for the field of NLP. It combines computer algorithms with machine learning and deep learning models to automatically extract, classify, and label elements of text and voice data and then assign a statistical likelihood to each possible meaning of those elements. Today, deep learning models and learning techniques based on convolutional neural networks (CNNs) and recurrent neural networks (RNNs) enable NLP systems that 'learn' as they work and extract ever more accurate meaning from huge volumes of raw, unstructured, and unlabeled text and voice data sets.

Advantages of Statistical NLP:

- The main advantage of statistical NLP is machine learning algorithm's "learnability", which is why no manual rule/grammar coding is needed, requiring high skills.
- The corpus can be annotated using the low-skilled workforce.
- The data fed to such system will be huge and there are a lot of data points (e.g. keywords etc.), which makes it easy for the machine to learn statistical clues of the words for a given task.
- Machine learning approaches can significantly speed up the development of a capability of certain NLP systems, when good training data sets are available

Disadvantages of Statistical NLP:

- Lack of training data
- Poorly labelled, insufficient data
- New "preparation" of data is required each time as, once created and labelled, the corpus often can't be reused on new data schemas.

11.3 NLP USE CASES

Natural language processing is the driving force behind machine intelligence in many modern real-world applications. Here are a few examples:

- **Spam detection:** NLP's text classification capabilities can be used to scan emails for language that often indicates spam or phishing. These indicators can include overuse of financial terms, characteristic bad grammar, threatening language, inappropriate urgency, misspelled company names, and more.

- **Machine translation:** Machine translation involves translating words/sentences in one language to another. Effective translation has to capture accurately the meaning and tone of the input language and translate it to text with the same meaning and desired impact in the output language. Example: Google translate

- **Speech Recognition:** This is the process of mapping acoustic speech signals to a set of words. Difficulty arises due to wide variations in the pronunciations of words, homonym (e.g., dear and deer) and acoustic ambiguities (e.g., in the rest and interest)

- **Speech Synthesis:** It refers to automatic production of speech (uttering sentences in natural language). Such systems can read out your mail or messages for you.

- **Virtual agents and chatbots:** Virtual agents such as Apple's Siri and Amazon's Alexa use speech recognition to recognize patterns in voice commands and natural language generation to respond with appropriate action or helpful comments. Chatbots perform the same magic in response to typed text queries.

- **Social media sentiment analysis:** Sentiment analysis can analyze language used in social media posts, responses, reviews, and more to extract attitudes and emotions in response to products, promotions, and events. companies can use this information in product designs, advertising campaigns, and more.

- **Text summarization:** Text summarization uses NLP techniques to digest huge volumes of digital text and create summaries and synopses for indexes, research databases, or busy readers who don't have time to read full text. The best text summarization applications use semantic reasoning and natural language generation (NLG) to add useful context and conclusions to summaries.

11.4 NLP TOOLS & LIBRARIES

Some commonly used NLP tools and libraries are as follows:

1. NLTK - entry-level open-source NLP Tool

Natural Language Toolkit (AKA NLTK) is an open-source software powered with Python NLP. NLTK provides users with a basic set of tools for text-related operations. It is a good starting point for beginners in Natural Language Processing.

Natural Language Toolkit features include:

- Text classification
- Part-of-speech tagging
- Entity extraction
- Tokenization
- Parsing
- Stemming
- Semantic reasoning

Natural Language Toolkit is useful for simple text analysis. But, if you need to work on a massive amount of data, it requires significant resources.

2. Stanford Core NLP - Data Analysis, Sentiment Analysis, Conversational UI

Stanford NLP library is a multi-purpose tool for text analysis. Like NLTK, Stanford CoreNLP provides many different natural language processing software. But if you need more, you can use custom modules. The main advantage of Stanford NLP tools is scalability. Unlike NLTK, Stanford Core NLP is a perfect choice for processing large amounts of data and performing complex operations.

3. Apache OpenNLP - Data Analysis and Sentiment Analysis

Apache OpenNLP is an open-source library for those who prefer practicality and accessibility. Like Stanford CoreNLP, it uses Java NLP libraries with Python decorators. While NLTK and Stanford CoreNLP are state-of-the-art libraries with tons of additions, OpenNLP is a simple yet useful tool. Besides, you can configure OpenNLP in the way you need and get rid of unnecessary features.

Apache OpenLP is the right choice for:

- Named Entity Recognition
- Sentence Detection
- POS tagging
- Tokenization

4. SpaCy - Data Extraction, Data Analysis, Sentiment Analysis, Text Summarization

SpaCy is the next step of the NLTK evolution. NLTK is clumsy and slow when it comes to more complex business applications. At the same time, SpaCy provides users with a smoother, faster, and efficient experience.

- SpaCy, an open-source NLP library, is a perfect match for comparing customer profiles, product profiles, or text documents.
- SpaCy is good at syntactic analysis, which is handy for aspect-based sentiment analysis and conversational user interface optimization.

- SpaCy is also an excellent choice for named-entity recognition. You can use SpaCy for business insights and market research.
- Another SpaCy advantage is word vector usage. Unlike OpenNLP and CoreNLP, SpaCy works with word2vec and doc2vec.

5. GenSim - Document Analysis, Semantic Search, Data Exploration

GenSim is the perfect tool to extract particular information to discover business insights. It is an open-source NLP library designed for document exploration and topic modeling. It would help you to navigate the various databases and documents.

- The key GenSim feature is word vectors. It sees the content of the documents as sequences of vectors and clusters, and then, classifies them.
- GenSim is also resource-saving when it comes to dealing with a large amount of data.

The main GenSim use cases are:

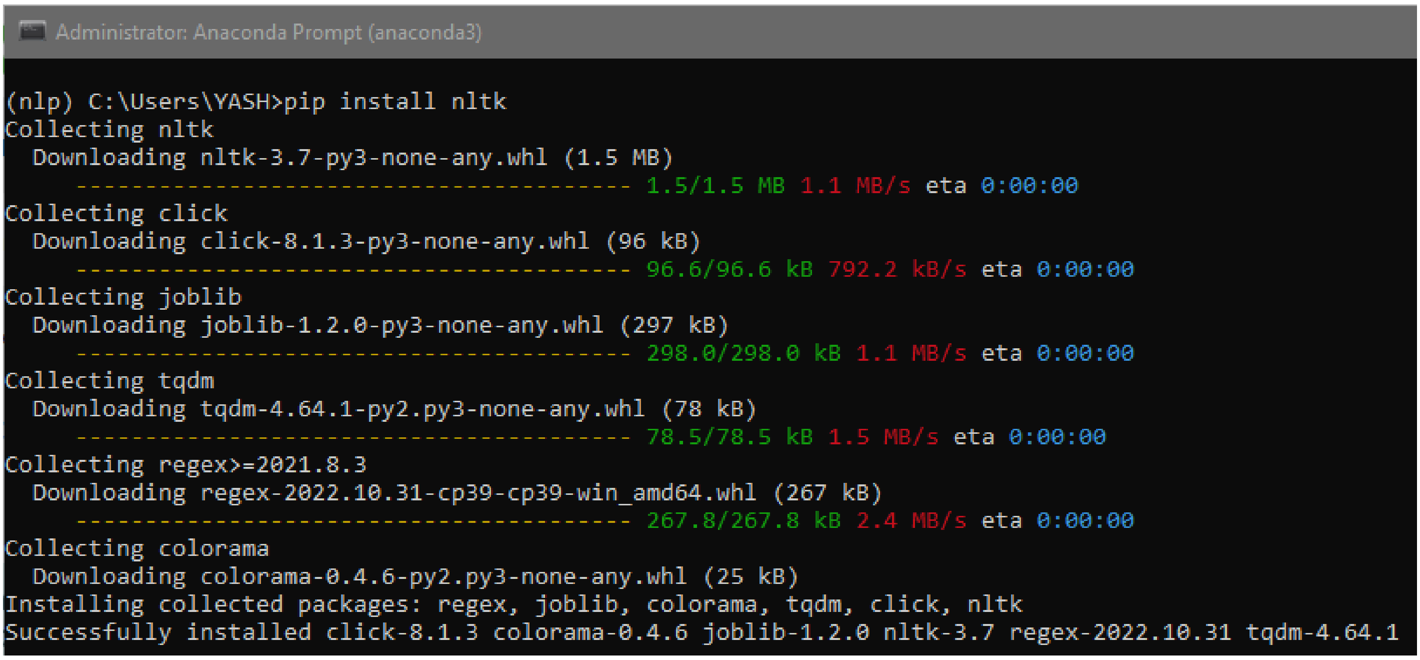
- Data analysis
- Semantic search applications
- Text generation applications (chatbot, service customization, text summarization, etc.)

11.5 ENVIRONMENT SETUP

NLTK can be installed by using the pip package installer. Recently NLTK has dropped support for Python 2 so Python 3.5 and above is required to install NLTK.

Pip command to install NLTK is as follows:

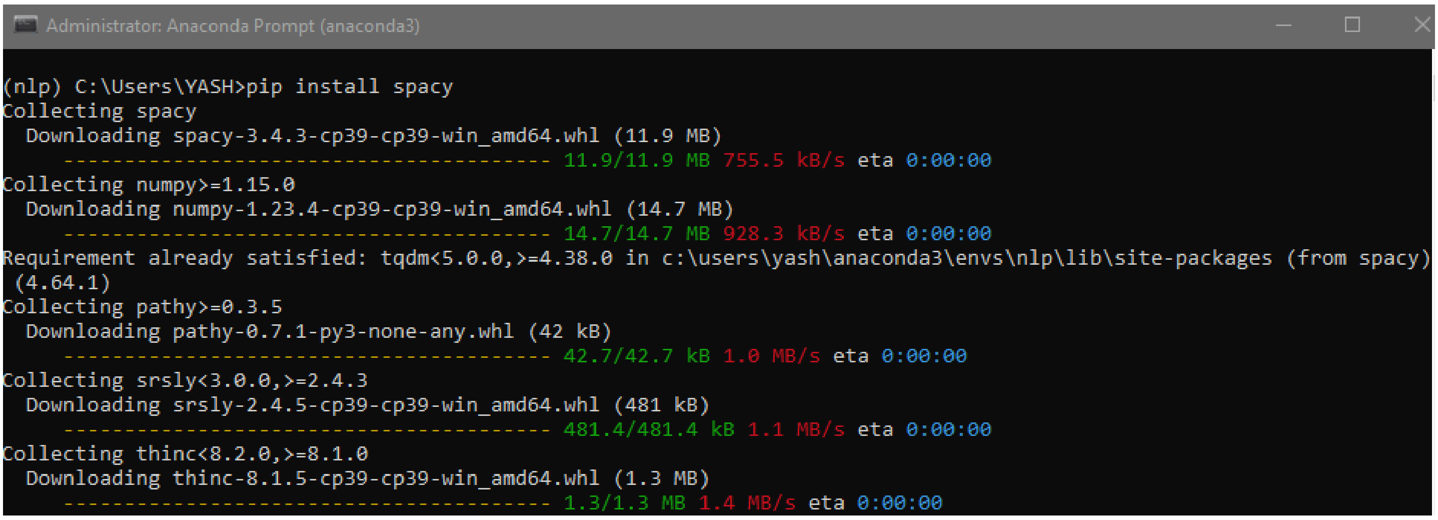
pip install nltk



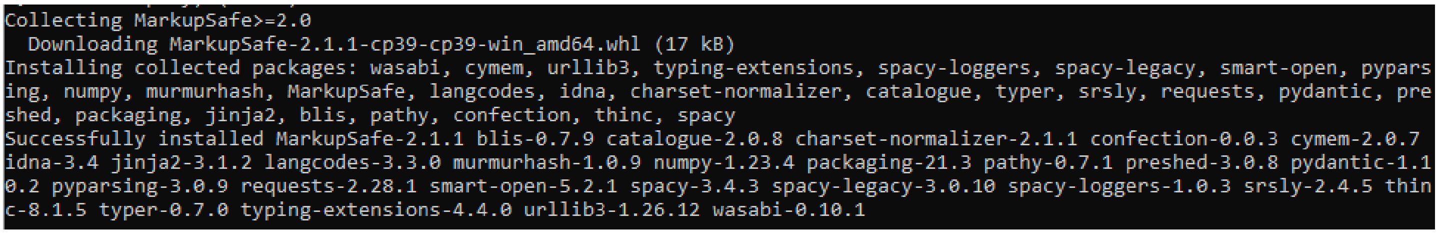
```
(nlp) C:\Users\YASH>pip install nltk
Collecting nltk
  Downloading nltk-3.7-py3-none-any.whl (1.5 MB)
     ---------------------------------------- 1.5/1.5 MB 1.1 MB/s eta 0:00:00
Collecting click
  Downloading click-8.1.3-py3-none-any.whl (96 kB)
     ---------------------------------------- 96.6/96.6 kB 792.2 kB/s eta 0:00:00
Collecting joblib
  Downloading joblib-1.2.0-py3-none-any.whl (297 kB)
     ---------------------------------------- 298.0/298.0 kB 1.1 MB/s eta 0:00:00
Collecting tqdm
  Downloading tqdm-4.64.1-py2.py3-none-any.whl (78 kB)
     ---------------------------------------- 78.5/78.5 kB 1.5 MB/s eta 0:00:00
Collecting regex>=2021.8.3
  Downloading regex-2022.10.31-cp39-cp39-win_amd64.whl (267 kB)
     ---------------------------------------- 267.8/267.8 kB 2.4 MB/s eta 0:00:00
Collecting colorama
  Downloading colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Installing collected packages: regex, joblib, colorama, tqdm, click, nltk
Successfully installed click-8.1.3 colorama-0.4.6 joblib-1.2.0 nltk-3.7 regex-2022.10.31 tqdm-4.64.1
```

We can install SpaCy in a similar way using

pip install spacy



```
(nlp) C:\Users\YASH>pip install spacy
Collecting spacy
  Downloading spacy-3.4.3-cp39-cp39-win_amd64.whl (11.9 MB)
     ---------------------------------------- 11.9/11.9 MB 755.5 kB/s eta 0:00:00
Collecting numpy>=1.15.0
  Downloading numpy-1.23.4-cp39-cp39-win_amd64.whl (14.7 MB)
     ---------------------------------------- 14.7/14.7 MB 928.3 kB/s eta 0:00:00
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in c:\users\yash\anaconda3\envs\nlp\lib\site-packages (from spacy)
 (4.64.1)
Collecting pathy>=0.3.5
  Downloading pathy-0.7.1-py3-none-any.whl (42 kB)
     ---------------------------------------- 42.7/42.7 kB 1.0 MB/s eta 0:00:00
Collecting srsly<3.0.0,>=2.4.3
  Downloading srsly-2.4.5-cp39-cp39-win_amd64.whl (481 kB)
     ---------------------------------------- 481.4/481.4 kB 1.1 MB/s eta 0:00:00
Collecting thinc<8.2.0,>=8.1.0
  Downloading thinc-8.1.5-cp39-cp39-win_amd64.whl (1.3 MB)
     ---------------------------------------- 1.3/1.3 MB 1.4 MB/s eta 0:00:00
```



```
Collecting MarkupSafe>=2.0
  Downloading MarkupSafe-2.1.1-cp39-cp39-win_amd64.whl (17 kB)
Installing collected packages: wasabi, cymem, urllib3, typing-extensions, spacy-loggers, spacy-legacy, smart-open, pypars
ing, numpy, murmurhash, MarkupSafe, langcodes, idna, charset-normalizer, catalogue, typer, srsly, requests, pydantic, pre
shed, packaging, jinja2, blis, pathy, confection, thinc, spacy
Successfully installed MarkupSafe-2.1.1 blis-0.7.9 catalogue-2.0.8 charset-normalizer-2.1.1 confection-0.0.3 cymem-2.0.7
idna-3.4 jinja2-3.1.2 langcodes-3.3.0 murmurhash-1.0.9 numpy-1.23.4 packaging-21.3 pathy-0.7.1 preshed-3.0.8 pydantic-1.1
0.2 pyparsing-3.0.9 requests-2.28.1 smart-open-5.2.1 spacy-3.4.3 spacy-legacy-3.0.10 spacy-loggers-1.0.3 srsly-2.4.5 thin
c-8.1.5 typer-0.7.0 typing-extensions-4.4.0 urllib3-1.26.12 wasabi-0.10.1
```

References:

1. Tanveer Siddiqui, U.S. Tiwary, "Natural Language Processing and Information Retrieval", Oxford University Press, 2008.
2. https://medium.com/friendly-data/machine-learning-vs-rule-based-systems-in-nlp-5476de53c3b8
3. https://www.sentisum.com/success-article/machine-learning-nlp
4. https://www.kdnuggets.com/2018/10/main-approaches-natural-language-processing-tasks.html
5. https://www.ibm.com/cloud/learn/natural-language-processing
6. https://theappsolutions.com/blog/development/nlp-tools/
7. Infosys Springboard – Natural Language Processing for developers