UNIT-5

NLP Applications: Introduction, Information Extraction, Automatic Text Summarization, Question-Answering System, **Lexical Resources:** Introduction, Word Net, Frame Net, Stemmers, Part-of-Speech Tagger, Research Corpora, Journals and Conferences in the Area.

INTRODUCTION

- Natural language processing (NLP) is the ability of a computer program to understand human language as it is spoken and written -- referred to as natural language. It is a component of artificial intelligence (AI).
- Natural language processing (NLP) is an area of computer science and artificial intelligence concerned with the interaction between computers and humans in natural language.
- **The ultimate goal of NLP is to help computers understand language as well as we do.**
- It is the driving force behind things like virtual assistants, speech recognition, sentiment analysis, automatic text summarization, machine translation and much more.
- NLP has existed for more than 50 years and has roots in the field of linguistics.
- It has a variety of real-world applications in a number of fields, including medical research, search engines and business intelligence.
- Businesses use massive quantities of unstructured, text-heavy data and need a way to efficiently process it. A lot of the information created online and stored in databases is natural human language, and until recently, businesses could not effectively analyze this data. This is where natural language processing is useful.
- The advantage of natural language processing can be seen when considering the following two statements:
- "Cloud computing insurance should be part of every service-level agreement," and, "A good SLA ensures an easier night's sleep -- even in the cloud." If a user relies on natural language processing for search, the program will recognize that cloud computing is an entity, that cloud is an abbreviated form of cloud computing and that SLA is an industry acronym for service-level agreement.
- Applications of NLP techniques include voice assistants like Amazon’s Alexa and Apple’s Siri, but also things like machine translation and text-filtering.
- **customer feedback analysis** -- where AI analyzes social media reviews;
- **customer service automation** -- where voice assistants on the other end of a customer service phone line are able to use speech recognition to understand what the customer is saying, so that it can direct the call correctly;
- **automatic translation** -- using tools such as Google Translate, Bing Translator and Translate Me;
- **academic research and analysis** -- where AI is able to analyze huge amounts of academic material and research papers not just based on the metadata of the text, but the text itself;

- **analysis and categorization of medical records** -- where AI uses insights to predict, and ideally prevent, disease;
- **word processors used for plagiarism and proofreading** -- using tools such as Grammarly and Microsoft Word;
- Natural language processing has heavily benefited from recent advances in machine learning, especially from deep learning techniques. The field is divided into the three parts:
- **Speech recognition—**the translation of spoken language into text.
- **Natural language understanding—**a computer's ability to understand language.
- **Natural language generation—**the generation of natural language by a computer.

What is Information Extraction?

- Information Extraction is the process of parsing through unstructured data and extracting essential information into more editable and structured data formats.
- For example, consider we're going through a company's financial information from a few documents. Usually, we search for some required information when the data is digital or manually checks the same. But with information extraction NLP algorithms, we can automate the data extraction of all required information such as tables, company growth metrics, and other financial details from various kinds of documents (PDFs, Docs, Images etc.).
- Information Extraction from text data can be achieved by leveraging Deep Learning and NLP techniques like Named Entity Recognition.

How Does Information Extraction Work?

- To understand the mechanics of Information Extraction NLP algorithms, we should understand the kind of data we are working on. This will help us to sort out the information we want to extract from the unstructured data.
- For example, for invoice related information, the algorithm should understand the invoice items, company name, billing address etc. While working on medical reports, it should identify and extract patient names, drug information, and other general reports.
- After curating the data, we'll then start applying the information extraction NLP techniques, to process and build models around the data. Below are some of the most common techniques that are frequently used.

Tokenization:

- Computers usually won't understand the language we speak or communicate with. Hence, we break the language, basically the words and sentences, into tokens and then load it into a program. The process of breaking down language into tokens is called tokenization.
- For example, consider a simple sentence: "NLP information extraction is fun''. This could be tokenized into:
- **One-word (sometimes called unigram token):** NLP, information, extraction, is, fun

- **Two-word phrase (bigram tokens):** NLP information, information extraction, extraction is, is fun, fun NLP
- **Three-word sentence (trigram tokens):** NLP information extraction, information extraction is, extraction is fun

Parts of Speech Tagging:

- Tagging parts of speech is very crucial for information extraction from text.
- It'll help us understand the context of the text data. We usually refer to text from documents as "unstructured data" – data with no defined structure or pattern. Hence, with POS tagging we can use techniques that will provide the context of words or
- tokens used to categorize them in specific ways.

Dependency Graphs:

- Dependency graphs help us find relationships between neighboring words using directed graphs.
- This relation will provide details about the dependency type (e.g. Subject, Object etc.).
- An Example of Information Extraction
- Several industries deal with lots of documents every day and rely on manual work. Those include finance, medical chains, transportation, and construction. Using NLP information extraction techniques on documents will allow everyone on the teams to search, edit, and analyze important transactions and details across business processes.

1. Information Collection:

Firstly, we'll need to collect the data from different sources to build an information extraction model. Usually, we see documents on emails, cloud drives, scanned copies, computer software, and many other sources for business. Hence, we'll have to write different scripts to collect and store information in one place. This is usually done by either using APIs
on the web or building RPA (Robotic Process Automation) pipelines.

2. Process Data:

After we collect the data, the next step is to process them. Usually, documents are two types: electronically generated (editable) and the other non-electronically generated (scanned documents). For the electronically generated documents, we can directly send them into the preprocessing pipelines. Still, we'll need OCR to first read all the data from images and then send them into preprocessing pipelines for the scanned copies. We can either use open-source tools like Tesseract or any online services like Nanonets or Textract. After all the data is in editable or electronic format, we can then apply to pre-process steps like

Tokenization and POS tagging and then use data loaders to load the data into the NLP information extraction models.

3. Choosing the Right Model:

As discussed in the above sections, choosing a suitable model mostly depends on the type of data we're working with. Today, there are several state-of-the-art models we could rely on. Below are some of the frequently use open-source models:

Named Entity Recognition on CoNLL 2003 (English)

Key Information Extraction From Documents: Evaluation And Generator.

Deep Reader: Information extraction from Document images via relation extraction and Natural Language

These are some of the information extraction models.

4. Evaluation of the Model:

We evaluate the training process is crucial before we use the models in production. This is usually done by creating a testing dataset and finding some key metrics:

Accuracy: the ratio of correct predictions made against the size of the test data.

Precision: the ratio of true positives and total predicted positives.

Recall: the ratio of true positives and total actual positives.

F1-Score: harmonic mean of precision and recall.

5. Deploying Model in Production:

The full potential of the NLP models only knows when they are deployed in production. Today, as the world is entirely digital, these models are stored on cloud servers with a suitable background. In most cases, Python is utilized as its more handy programming language when it comes to Text data and machine learning. The model is either exported as API or an SDK (software development kit) for integrating with business tools.

applications of Information Extraction:

Invoice Automation: Automate the process of invoice information extraction.

Healthcare Systems: Manage medical records by identifying patient information and their prescriptions.

KYC Automation: Automate the process of KYC by extracting ethical information from customer's identity documents.

Financial Investigation: Extract import information from financial documents. (Tax, Growth, Quarterly Revenue, Profit/Losses)

***AUTOMATIC TEXT SUMMARIZATION

Text summarization is the process of generating short, fluent, and most importantly accurate summary of a respectively longer text document.

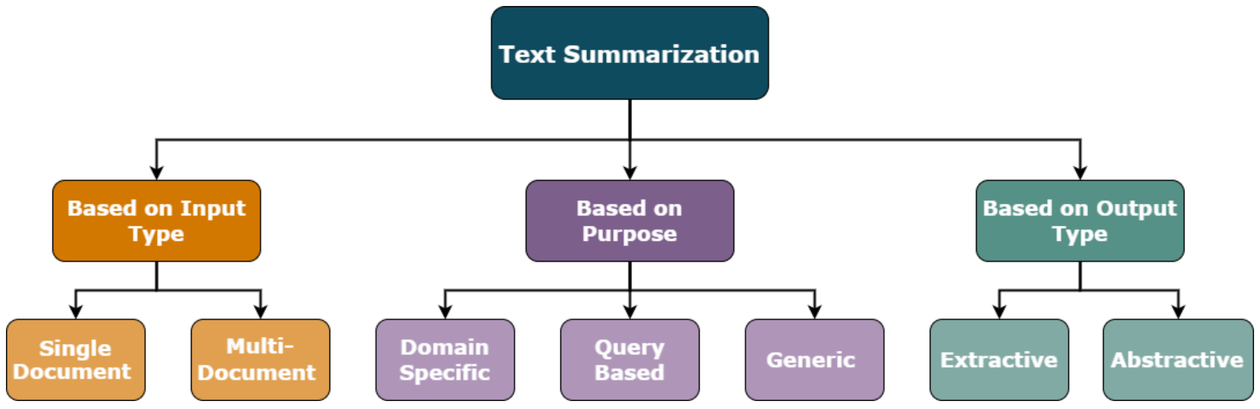
The main idea behind automatic text summarization is to be able to find a short subset of the most essential information from the entire set and present it in a human-readable format.

As online textual data grows, automatic text summarization methods have the potential to be very helpful because more useful information can be read in a short time.

Why automatic text summarization?

Summaries reduce reading time.
When researching documents, summaries make the selection process easier.
Automatic summarization improves the effectiveness of indexing.
Automatic summarization algorithms are less biased than human summarization.
Personalized summaries are useful in question-answering systems as they provide personalized information.
Using automatic or semi-automatic summarization systems enables commercial abstract services to increase the number of text documents they are able to process.

Type of summarization:



Based on input type:

Single Document, where the input length is short. Many of the early summarization systems dealt with single-document summarization.

Multi-Document, where the input can be arbitrarily long.

Based on the purpose:
—-----------------------------

Generic, where the model makes no assumptions about the domain or content of the text to be summarized and treats all inputs as homogeneous. The majority of the work that has been done revolves around generic summarization.

Domain-specific, where the model uses domain-specific knowledge to form a more accurate summary. For example, summarizing research papers of a specific domain, biomedical documents, etc.

Query-based, where the summary only contains information that answers natural language questions about the input text.

Based on output type:
—-----------------------------

Extractive, where important sentences are selected from the input text to form a summary. Most summarization approaches today are extractive in nature.

Abstractive, where the model forms its own phrases and sentences to offer a more coherent summary, like what a human would generate. This approach is definitely more appealing, but much more difficult than extractive summarization.

How to do text summarization
—--------------------------------------

Text cleaning
Sentence tokenization
Word tokenization
Word-frequency table
Summarization

Text cleaning:
—----------------

```
# !pip instlla -U spacy
# !python -m spacy download en_core_web_sm
import spacy
from spacy.lang.en.stop_words import STOP_WORDS
from string import punctuation
stopwords = list(STOP_WORDS)
nlp = spacy.load(‘en_core_web_sm’)
doc = nlp(text)
```

Word tokenization:
—-----------------------

```
tokens = [token.text for token in doc]
print(tokens)
punctuation = punctuation + ‘\n’
```

```
punctuation
word_frequencies = {}
for word in doc:
if word.text.lower() not in stopwords:
if word.text.lower() not in punctuation:
if word.text not in word_frequencies.keys():
word_frequencies[word.text] = 1
else:
word_frequencies[word.text] += 1
print(word_frequencies)
```

Sentence tokenization:

—-----------------------------

```
max_frequency = max(word_frequencies.values())
max_frequency
for word in word_frequencies.keys():
word_frequencies[word] = word_frequencies[word]/max_frequency
print(word_frequencies)
sentence_tokens = [sent for sent in doc.sents]
print(sentence_tokens)
```

Word frequency table:

—----------------------------

```
sentence_scores = {}
for sent in sentence_tokens:
for word in sent:
if word.text.lower() in word_frequencies.keys():
if sent not in sentence_scores.keys():
sentence_scores[sent] = word_frequencies[word.text.lower()]
else:
sentence_scores[sent] += word_frequencies[word.text.lower()]
sentence_scores
```

Summarization:

—------------------

```
from heapq import nlargest
select_length = int(len(sentence_tokens)*0.3)
select_length
summary = nlargest(select_length, sentence_scores, key = sentence_scores.get)
summary
final_summary = [word.text for word in summary]
```

```
summary = ‘ ‘.join(final_summary)
```

Input:

```
text = “””
```

Maria Sharapova has basically no friends as tennis players on the WTA Tour. The Russian player has no problems in openly speaking about it and in a recent interview she said: ‘I don’t really hide any feelings too much.

I think everyone knows this is my job here. When I’m on the courts or when I’m on the court playing, I’m a competitor and I want to beat every single person whether they’re in the locker room or across the net.

So I’m not the one to strike up a conversation about the weather and know that in the next few minutes I have to go and try to win a tennis match.

I’m a pretty competitive girl. I say my hellos, but I’m not sending any players flowers as well. Uhm, I’m not really friendly or close to many players.

I have not a lot of friends away from the courts.’ When she said she is not really close to a lot of players, is that something strategic that she is doing? Is it different on the men’s tour than the women’s tour? ‘No, not at all.

I think just because you’re in the same sport doesn’t mean that you have to be friends with everyone just because you’re categorized, you’re a tennis player, so you’re going to get along with tennis players.

I think every person has different interests. I have friends that have completely different jobs and interests, and I’ve met them in very different parts of my life.

I think everyone just thinks because we’re tennis players we should be the greatest of friends. But ultimately tennis is just a very small part of what we do.

There are so many other things that we’re interested in, that we do.’

“””

Output(final summary): summary

I think just because you’re in the same sport doesn’t mean that you have to be friends with everyone just because you’re categorized, you’re a tennis player, so you’re going to get along with tennis players.

Maria Sharapova has basically no friends as tennis players on the WTA Tour. I have friends that have completely different jobs and interests, and I’ve met them in very different parts of my life.

I think everyone just thinks because we’re tennis players So I’m not the one to strike up a conversation about the weather and know that in the next few minutes I have to go and try to win a tennis match.

When she said she is not really close to a lot of players, is that something strategic that she is doing?

Link: https://medium.com/analytics-vidhya/text-summarization-using-nlp-3e85ad0c6349

Question-Answering System

What is a question-answering System?

Question answering (QA) is a field of natural language processing (NLP) and artificial intelligence (AI) that aims to develop systems that can understand and answer questions posed in natural language.

The point of a QA system is to understand the question and give an answer that is correct and helpful.

QA systems can be based on various techniques, including information retrieval, knowledge-based, generative, and rule-based approaches. Each method has its strengths and weaknesses, and the choice of method depends on the project's specific needs.

QA systems can be used in many places, like customer service, search engines, healthcare, education, finance, e-commerce, voice assistants, chatbots, and virtual assistants.

How does a natural language question-answering system work?

A natural language question-answering (QA) system is a computer program that automatically answers questions using NLP. The basic process of a natural language QA system includes the following steps:

1. **Text pre-processing**: The question is pre-processed to remove irrelevant information and standardize the text's format. This step includes tokenisation, lemmatisation, and stop-word removal, among others.
2. **Question understanding**: The pre-processed question is analyzed to extract the relevant entities and concepts and to identify the type of question being asked. This step can be done using natural language processing (NLP) techniques such as named entity recognition, dependency parsing, and part-of-speech tagging.
3. **Information retrieval**: The question is used to search a database or corpus of text to retrieve the most relevant information. This can be done using information retrieval techniques such as keyword search or semantic search.
4. **Answer generation:** The retrieved information is analyzed to extract the specific answer to the question. This can be done using various techniques, such as machine learning algorithms, rule-based systems, or a combination.
5. **Ranking**: The extracted answers are ranked based on relevance and confidence score.

For example, some systems are based on a knowledge base, others on information retrieval, and others on generative models. Hybrid systems can also be designed to combine several approaches to improve overall performance.

It's also worth noting that the quality of the input data, pre-processing, tokenization, and the model's architecture are essential to achieve an excellent question-answering system.

Training a QA model requires a large dataset of questions and corresponding answers.

Types of question answering system

Question answering (QA) implementation in natural language processing (NLP) involves using various NLP techniques to answer questions in natural language automatically. There are several different approaches to QA implementation in NLP.

1.Information retrieval-based QA

Information retrieval-based question answering (QA) is a method of automatically answering questions by searching for relevant documents or passages that contain the answer. This approach uses information retrieval techniques, such as keyword or semantic search, to identify the documents or passages most likely to hold the answer to a given question.

Information retrieval-based QA systems are generally easy to implement and can be used to answer a wide range of questions. However, their performance can be limited by the quality and relevance of the indexed text and the effectiveness of the retrieval and extraction methods used.

It’s also important to note that IR-based QA systems are often used with other types of QA, like knowledge-based or generative QA, to improve the system’s overall performance.

2.Knowledge-based QA

Knowledge-based question answering (QA) automatically answers questions using a knowledge base, such as a database or ontology, to retrieve the relevant information. This strategy’s foundation is that searching for a structured knowledge base for a question can yield the answer.

Knowledge-based QA systems are generally more accurate and reliable than other QA approaches based on structured and well-curated knowledge. But their performance can be limited by how well the knowledge base is covered and how well the methods used to make queries and get information from their work.

It’s also important to note that knowledge-based QA systems are often used with other QA methods, like information retrieval-based or generative QA, to improve the overall performance of the QA system.

3.Generative QA

Generative question answering (QA) automatically answers questions using a generative model, such as a neural network, to generate a natural language answer to a given question.

This method is based on the idea that a machine can be taught to understand and create text in natural language to provide a correct answer in terms of grammar and meaning.

Generative QA systems are powerful as they can answer a wide range of questions and generate more human-like answers.

However, their performance can be limited by the training data's quality and diversity and the model's complexity.

It's also worth noting that Generative QA systems are often used with other QA approaches, such as information retrieval-based or knowledge-based QA, to improve the overall performance of the QA system.

These combinations are known as Hybrid QA systems.

4.Hybrid QA

Hybrid question answering (QA) automatically answers questions by combining multiple QA approaches, such as information retrieval-based, knowledge-based, and generative QA. This approach is based on the idea that different QA approaches have their strengths and weaknesses, and by combining them, the overall performance of the QA system can be improved.

Hybrid QA systems are considered more robust and accurate than a single QA approach, as they can leverage the strengths of multiple QA methods. Hybrid QA systems can also be more flexible, as they can adapt to different types of questions and different levels of complexity. But designing and putting together a hybrid QA system can be more complex and take more resources than a single QA method.

Hybrid QA systems can be built to be used in a specific domain or a general-purpose QA system. In both cases, the system's performance will depend on the data quality, pre-processing, tokenization, and the model's architecture.

5. Rule-based QA

Rule-based question answering (QA) automatically answers questions using a predefined set of rules based on keywords or patterns in the question. This approach is based on the idea that many questions can be answered by matching the question to a set of predefined rules or templates.

Rule-based QA systems are generally simple and easy to implement. Still, their performance can be limited by the coverage and completeness of the rules and the effectiveness of the pattern matching and extraction methods used. In addition, rule-based QA systems are more prone to errors and can only handle questions covered by predefined rules.

It's also worth noting that rule-based QA systems are often combined with other QA approaches, such as information retrieval-based, knowledge-based, or generative QA, to improve the overall performance of the QA system. In these cases, the rule-based QA can filter out irrelevant answers and improve the efficiency of the comprehensive system.

All of these approaches require significant training data, including questions and their corresponding answers, to improve the accuracy of the QA system.

Additionally, the quality of the input data, pre-processing, tokenization, and the model's architecture is essential to achieve a good question-answering system.

Applications of question and answering systems

Question-answering (QA) systems have various applications in various industries and domains. Some of the most common applications of QA systems include:

1. **Customer service:** QA systems can be used to answer customers' questions quickly and correctly, reducing the need for human customer service reps.
2. **Search engines:** QA systems can make search results more accurate and valuable by answering specific questions instead of just giving a list of relevant documents.
3. **Healthcare:** QA systems can give patients accurate and reliable information about their health conditions and treatment options.
4. **Education:** QA systems can be used in education to give students immediate feedback and explanations for their answers, which helps them learn better.

5. **Finance:** QA systems can tell financial advisors about the latest market trends and investment strategies.
6. **E-Commerce:** QA systems can be used to recommend products to customers and answer their questions about the features and availability of those products.
7. **Voice assistants:** QA systems can be connected to voice assistants so that users can conversationally get answers to their questions.
8. **Chatbots:** QA systems can be linked to chatbots so that users can naturally get answers to their questions.
9. **Virtual assistants:** QA systems can be connected to virtual assistants so that users can conversationally get answers to their questions.
10. **Business intelligence:** QA systems can extract relevant information from large datasets and provide decision-making insights.

Tools

Several NLP tools and frameworks are available for implementing a question-answering (QA) system. Some of the most popular include:

1. TensorFlow: An open-source machine learning framework that can train and deploy QA models. TensorFlow provides a wide range of tools for natural language processing (NLP) tasks, such as sentiment analysis, language translation, and text generation, which can be used to implement QA systems.
2. BERT: A pre-trained transformer-based model for natural language processing tasks, including question answering. BERT has been trained on a large corpus of text and achieved state-of-the-art performance on several NLP benchmarks. BERT can be fine-tuned on specific datasets to perform QA tasks and easily integrated into other models.
3. GPT-3: A pre-trained transformer-based model for natural language processing tasks, including question answering. GPT-3 has been trained on a massive amount of text and has achieved state-of-the-art performance on several NLP benchmarks, including QA tasks. GPT-3 can be fine-tuned on specific datasets to perform QA tasks and easily integrated into other models.
4. Hugging Face: An open-source platform that provides a wide range of pre-trained models for NLP tasks, including question answering. Hugging Face models can be tuned for specific datasets and integrated into other models, making it simple to implement QA systems.

5. SpaCy: A popular open-source library for natural language processing in Python. SpaCy provides a wide range of tools for text processing, including tokenization, lemmatization, and named entity recognition, which can be used to implement QA systems.
6. NLTK: The Natural Language Toolkit (NLTK) is a Python library for working with human language data. It provides several tools for text pre-processing, tokenization, stemming, tagging, parsing, semantic reasoning and wrappers for industrial-strength NLP libraries.
7. OpenNLP: OpenNLP is an open-source library for natural language processing that provides tools for tokenization, stemming, tagging, parsing, and named entity recognition, among others. It can be used with other tools and libraries to make NLP-based applications, such as quality assurance (QA) systems.

Lexical Resources:

A lexical resource is a language resource consisting of data regarding the lexemes of the lexicon of one or more languages e.g., in the form of a database.

Depending on the type of languages that are addressed, a lexical resource may be qualified as monolingual, bilingual or multilingual.

For bilingual and multilingual lexical resources, the words may be connected or not connected from one language to another.

Lexical resources in digital lexicography are often referred to as machine-readable dictionaries (MRD).

WORDNET

WordNet is a large lexical database of English words. Nouns, verbs, adjectives, and adverbs are grouped into sets of cognitive synonyms called ‘synsets’, each expressing a distinct concept.

WordNet is a lexical database of semantic relations between words that links words into semantic relations including synonyms, hyponyms, and meronyms.

WordNet was first created in 1985, in English only, in the Cognitive Science Laboratory of Princeton University under the direction of psychology professor George Armitage Miller. It was later directed by Christiane Fellbaum.

APPLICATIONS

WordNet has been used for a number of purposes in information systems, including word-sense disambiguation, information retrieval, automatic text classification, automatic text summarization, machine translation and even automatic crossword puzzle generation.

A common use of WordNet is to determine the similarity between words.

LIMITATIONS

The most widely discussed limitation of WordNet (and related resources like ImageNet) is that some of the semantic relations are more suited to concrete concepts than to abstract concepts.[11] For example, it is easy to create hyponyms/hypernym relationships to capture that a "conifer" is a type of "tree", a "tree" is a type of "plant", and a "plant" is a type of "organism", but it is difficult to classify emotions like "fear" or "happiness" into equally deep and well-defined hyponyms/hypernym relationships.

WordNet does not include information about the etymology or the pronunciation of words and it contains only limited information about usage.

WordNet does not include much domain-specific terminology.

FRAMENET

FrameNet is a group of online lexical databases based upon the theory of meaning known as Frame semantics, developed by linguist Charles J. Fillmore.

A frame is a schematic representation of a situation involving various participants, props, and other conceptual roles. Examples of frame names are Being_born and Locative_relation.

A frame in FrameNet contains a textual description of what it represents (a frame definition), associated frame elements, lexical units, example sentences, and frame-to-frame relations.

APPLICATIONS

FrameNet has been used in applications like question answering, paraphrasing, recognizing textual entailment, and information extraction, either directly or by means of Semantic Role Labeling tools.

LIMITATIONS

FrameNet does not focus on the peculiarities of a single verb but on the common semantic features of a frame.

Introduction to Stemming:-

Stemming is a method in **text processing** that eliminates prefixes and suffixes from words, transforming them into their fundamental or root form, The main objective of stemming is to streamline and standardize words, enhancing the effectiveness of the **natural language processing** tasks. The article explores more on the stemming technique and how to perform stemming in Python.

What is Stemming in NLP?

Simplifying words to their most basic form is called stemming, and it is made easier by stemmers or stemming algorithms. For example, "chocolates" becomes "chocolate" and "retrieval" becomes "retrieve." This is crucial for pipelines for natural language processing, which use tokenized words that are acquired from the first stage of dissecting a document into its constituent words.

Stemming in natural language processing reduces words to their base or root form, aiding in text normalization for easier processing. This technique is crucial in tasks like text classification, information retrieval, and text summarization. While beneficial, stemming has drawbacks, including potential impacts on text readability and occasional inaccuracies in determining the correct root form of a word.

Why is Stemming important?

It is important to note that stemming is different from Lemmatization. Lemmatization is the process of reducing a word to its base form, but unlike stemming, it takes into account the context of the word, and it produces a valid word, unlike stemming which may produce a non-word as the root form.

```
Some more example of stemming for root word "like" include:

->"likes"

->"liked"

->"likely"

->"liking"
```

Types of Stemmer in NLTK :

1. Porter's Stemmer:- It is one of the most popular stemming methods proposed in 1980. It is based on the idea that the suffixes in the English language are made up of a combination of smaller and simpler suffixes. This stemmer is known for its speed and simplicity. The main applications of Porter Stemmer include data mining and Information retrieval. However, its applications are only limited to English words. Also, the group of stems is mapped on to the same stem and the output stem is not necessarily a meaningful word. The algorithms are fairly lengthy in nature and are known to be the oldest stemmer.

Example: EED -> EE means "if the word has at least one vowel and consonant plus EED ending, change the ending to EE" as 'agreed' becomes 'agree'.

- **Advantage:** It produces the best output as compared to other stemmers and it has less error rate.
- **Limitation:** Morphological variants produced are not always real words.

2. Lovins Stemmer:-It is proposed by Lovins in 1968, that removes the longest suffix from a word then the word is recorded to convert this stem into valid words.

Example: sitting -> sitt -> sit

Advantage: It is fast and handles irregular plurals like 'teeth' and 'tooth' etc

Limitation: It is time consuming and frequently fails to form words from stem.

3. Dawson Stemmer :-It is an extension of Lovins stemmer in which suffixes are stored in the reversed order indexed by their length and last letter.

Advantage: It is fast in execution and covers more suffices.

Limitation: It is very complex to implement.

4. Krovetz Stemmer:-

1) Convert the plural form of a word to its singular form.

2) Convert the past tense of a word to its present tense and remove the suffix 'ing'.

Example: 'children' -> 'child'

- **Advantage:** It is light in nature and can be used as pre-stemmer for other stemmers.
- **Limitation:** It is inefficient in case of large documents.

5. Xerox Stemmer:-Capable of processing extensive datasets and generating valid words, it has a tendency to over-stem, primarily due to its reliance on lexicons, making it language-dependent. This constraint implies that its effectiveness is limited to specific languages.

Example:

'children' -> 'child'

'understood' -> 'understand'

‘whom’ -> ‘who’

‘best’ -> ‘good’

6. N-Gram Stemmer :-The algorithm, aptly named n-grams (typically n=2 or 3), involves breaking words into segments of length n and then applying statistical analysis to identify patterns. An n-gram is a set of n consecutive characters extracted from a word in which similar words will have a high proportion of n-grams in common.

Example: ‘INTRODUCTIONS’ for n=2 becomes : *I, IN, NT, TR, RO, OD, DU, UC, CT, TI, IO, ON, NS, S*

- **Advantage:** It is based on string comparisons and it is language dependent.
- **Limitation:** It requires space to create and index the n-grams and it is not time efficient.

7. Snowball Stemmer:-The Snowball Stemmer, compared to the Porter Stemmer, is multi-lingual as it can handle non-English words. It supports various languages and is based on the ‘Snowball’ programming language, known for efficient processing of small strings.

The Snowball stemmer is way more aggressive than Porter Stemmer and is also referred to as Porter2 Stemmer. Because of the improvements added when compared to the Porter Stemmer, the Snowball stemmer is having greater computational speed.

8. Lancaster Stemmer:-The Lancaster stemmers are more aggressive and dynamic compared to the other two stemmers. The stemmer is really faster, but the algorithm is really confusing when dealing with small words. But they are not as efficient as Snowball Stemmers. The Lancaster stemmers save the rules externally and basically uses an iterative algorithm.

9. Regexp Stemmer:-The Regexp Stemmer, or Regular Expression Stemmer, is a stemming algorithm that utilizes regular expressions to identify and remove suffixes from words. It allows users to define custom rules for stemming by specifying patterns to match and remove.
This method provides flexibility and control over the stemming process, making it suitable for specific applications where custom rule-based stemming is desired.

Applications of Stemming

1. **Stemming is used in information retrieval systems like search engines.**
2. **It is used to determine domain vocabularies in domain analysis.**
3. **To display search results by indexing while documents are evolving into numbers and to map documents to common subjects by stemming**

Advantages of Stemming:-Stemming in natural language processing offers advantages such as text normalization, simplifying word variations to a common base form. It aids in information retrieval, text mining, and reduces feature dimensionality in machine learning. Stemming enhances

computational efficiency, making it a valuable step in text pre-processing for various NLP applications.

Disadvantages in Stemming

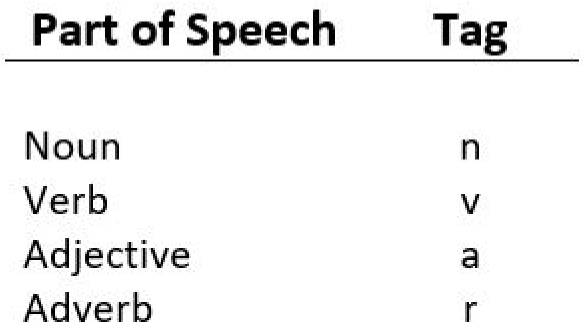
- **Over-stemming: Over-stemming in natural language processing occurs when a stemmer produces incorrect root forms or non-valid words. This can result in a loss of meaning and readability. For instance, "arguing" may be stemmed to "argu," losing meaning. To address this, choosing an appropriate stemmer, testing on sample text, or using lemmatization can mitigate over-stemming issues. Techniques like semantic role labeling and sentiment analysis can enhance context awareness in stemming.**

Part-of-speech Tagger:-

One of the core tasks in **Natural Language Processing (NLP)** is **Parts of Speech (PoS) tagging**, which is giving each word in a text a grammatical category, such as nouns, verbs, adjectives, and adverbs. Through improved comprehension of phrase structure and semantics, this technique makes it possible for machines to study and comprehend human language more accurately.In many NLP applications, including machine translation, sentiment analysis, and information retrieval, PoS tagging is essential. PoS tagging serves as a link between language and machine understanding, enabling the creation of complex language processing systems and serving as the foundation for advanced linguistic analysis.

What is POS(Parts-Of-Speech) Tagging?

Parts of Speech tagging is a linguistic activity in Natural Language Processing(NLP) wherein each word in a document is given a particular part of speech (adverb, adjective, verb, etc.) or grammatical category. Through the addition of a layer of syntactic and semantic information to the words, this procedure makes it easier to comprehend the sentence's structure and meaning



Part of Speech	Tag
Noun	n
Verb	v
Adjective	a
Adverb	r

Default tagging is a basic step for the part-of-speech tagging. It is performed using the DefaultTagger class. The DefaultTagger class takes 'tag' as a single argument. **NN** is the tag for a singular noun. DefaultTagger is most useful when it gets to work with most common part-of-speech tag. that's why a noun tag is recommended

Example of POS Tagging

Consider the sentence: "The quick brown fox jumps over the lazy dog."

After performing POS Tagging:

- "The" is tagged as determiner (DT)
- "quick" is tagged as adjective (JJ)
- "brown" is tagged as adjective (JJ)
- "fox" is tagged as noun (NN)
- "jumps" is tagged as verb (VBZ)
- "over" is tagged as preposition (IN)
- "the" is tagged as determiner (DT)

- “lazy” is tagged as adjective (JJ)
- “dog” is tagged as noun (NN)

  By offering insights into the grammatical structure, this tagging aids machines in comprehending not just individual words but also the connections between them inside a phrase. For many NLP applications, like text summarization, sentiment analysis, and machine translation, this kind of data is essentia

Workflow of POS Tagging in NLP:-The following are the processes in a typical natural language processing (NLP) example of part-of-speech (POS) tagging:

- Tokenization: Divide the input text into discrete tokens, which are usually units of words or subwords. The first stage in NLP tasks is tokenization.
  - **Loading Language Models:** To utilize a library such as NLTK or SpaCy, be sure to load the relevant language model. These models offer a foundation for comprehending a language’s grammatical structure since they have been trained on a vast amount of linguistic data.
  - **Text Processing**: If required, preprocess the text to handle special characters, convert it to lowercase, or eliminate superfluous information. Correct PoS labeling is aided by clear text.
  - **Linguistic Analysis**: To determine the text’s grammatical structure, use linguistic analysis. This entails understanding each word’s purpose

inside the sentence, including whether it is an adjective, verb, noun, or other.

- **Part-of-Speech Tagging:** To determine the text's grammatical structure, use linguistic analysis. This entails understanding each word's purpose inside the sentence, including whether it is an adjective, verb, noun, or other.
- **Results Analysis:** Verify the accuracy and consistency of the PoS tagging findings with the source text. Determine and correct any possible problems or mistagging.

Types of POS Tagging in NLP

1. Rule-Based Tagging:

Rule-based part-of-speech (POS) tagging involves assigning words their respective parts of speech using predetermined rules, contrasting with machine learning-based POS tagging that requires training on annotated text corpora. In a rule-based system, POS tags are assigned based on specific word characteristics and contextual cues.

Let's consider an example of how a rule-based part-of-speech (POS) tagger might operate:
Rule: Assign the POS tag "noun" to words ending in "-tion" or "-ment."

Text: "The presentation highlighted the key achievements of the project's development."

Rule based Tags:

- "The" – Determiner (DET)

- “presentation” – Noun (N)
- “highlighted” – Verb (V)
- “the” – Determiner (DET)
- “key” – Adjective (ADJ)
- “achievements” – Noun (N)
- “of” – Preposition (PREP)
- “the” – Determiner (DET)
- “project’s” – Noun (N)
- “development” – Noun (N)

2. Transformation Based tagging

Transformation-based tagging (TBT) is a part-of-speech (POS) tagging method that uses a set of rules to change the tags that are applied to words inside a text. In contrast, statistical POS tagging uses trained algorithms to predict tags probabilistically, while rule-based POS tagging assigns tags directly based on predefined rules.

Consider the transformation rule: Change the tag of a verb to a noun if it follows a determiner like “the.”

Text: “The cat chased the mouse”.

Initial Tags:

- “The” – Determiner (DET)
- “cat” – Noun (N)

- “chased” – Verb (V)
- “the” – Determiner (DET)
- “mouse” – Noun (N)

Transformation rule applied: Change the tag of “chased” from Verb (V) to Noun (N) because it follows the determiner “the.”

Updated tags:

- “The” – Determiner (DET)
- “cat” – Noun (N)
- “chased” – Noun (N)
- “the” – Determiner (DET)
- “mouse” – Noun (N)

3. Statistical POS Tagging

Utilizing probabilistic models, statistical part-of-speech (POS) tagging is a computer linguistics technique that places grammatical categories on words inside a text. If rule-based tagging uses massive annotated corpora to train its algorithms, statistical tagging uses machine learning.

Advantages of POS Tagging:-There are several advantages of Parts-Of-Speech (POS) Tagging including:

- **Text Simplification:** Breaking complex sentences down into their constituent parts makes the material easier to understand and easier to simplify.
- **Information Retrieval:** Information retrieval systems are enhanced by point-of-sale (POS) tagging, which allows for more precise indexing and search based on grammatical categories.

Disadvantages of POS Tagging:-Some common disadvantages in part-of-speech (POS) tagging include:

- **Ambiguity:** The inherent ambiguity of language makes POS tagging difficult since words can signify different things depending on the context, which can result in misunderstandings.
- **Idiomatic Expressions:** Slang, colloquialisms, and idiomatic phrases can be problematic for POS tagging systems since they don't always follow formal grammar standards.

research Corpora

Research corpora in Natural Language Processing (NLP) are large collections of texts used to train, test, and evaluate NLP models. These corpora can include a variety of text types such as books, news articles, scientific papers, social media posts, and transcribed speech. They are essential for developing and refining algorithms in areas such as machine translation, sentiment analysis, text classification, and language generation.

What is corpora in NLP

In Natural Language Processing (NLP), a "corpus" (plural: "corpora") is a large and structured set of texts used for linguistic research and in various computational models. Corpora are foundational in NLP as they provide the raw material upon which algorithms and models are trained and tested.

Types of Corpora in NLP

Monolingual Corpora

- General Corpora: These contain text from a wide range of sources and topics, often representing a particular language's everyday use. Examples include the Brown Corpus, COCA, and BNC for English.

Multilingual Corpora

- Parallel Corpora: These contain texts in two or more languages that are translations of each other. Examples include the Europarl Corpus, United Nations Parallel Corpus, and OpenSubtitles, which contain texts in multiple languages.

Time-Annotated Corpora

- Historical Corpora: Texts collected from historical periods, helping researchers study language evolution, changes in grammar, and vocabulary over time.

Spoken Corpora

- Transcribed Speech Corpora: Collections of spoken language transcribed into text format, such as the Switchboard Corpus or Fisher Corpus, used for research in speech recognition, speaker identification, and dialogue systems.

Social Media Corpora

- Twitter Corpus: Consists of tweets and associated metadata, often used for sentiment analysis, trend detection, and studying language use in social media.

Why are research corpora important in NLP?

Research corpora provide the raw material necessary for training, testing, and evaluating NLP models. They allow researchers to analyze language patterns, develop algorithms, and study linguistic phenomena across different domains and languages.

What are the characteristics of a good research corpus in NLP?

A good research corpus should be large enough to represent diverse language use, be well-annotated with linguistic information (such as part-of-speech tags, syntactic structure, named entities), and ideally cover various domains or genres to ensure comprehensive coverage.

Applications of research corpora (corpus)

Machine Learning Model Training:

Research corpora are used to train supervised and unsupervised machine learning models for various NLP tasks such as text classification, sentiment analysis, named entity recognition, machine translation, and more. These corpora provide the necessary data for algorithms to learn patterns, relationships, and structures within language.

Language Modeling and Generation:

Corpora are used to build language models that predict the likelihood of a sequence of words, enabling tasks like autocomplete, spell checking, and text generation. Models like GPT (Generative Pre-trained Transformer) use vast corpora to generate coherent and contextually relevant text.

Dialog Systems and Chatbots:

Dialogue corpora capture conversational interactions and are used to train chatbots and dialogue systems, improving their ability to understand and generate human-like responses.

Sentiment Analysis and Opinion Mining:

Corpora containing labeled data help in training sentiment analysis models that determine sentiment polarity (positive, negative, neutral) in texts. These models are used for analyzing social media sentiment, customer feedback, and reviews.

Named Entity Recognition (NER) and Information Extraction:

Corpora annotated with named entities (such as names of people, organizations, locations) assist in training NER models, used in information extraction tasks like entity recognition from text and event extraction.

Speech Recognition and Speech-to-Text Systems:

Spoken language corpora transcribed into text format serve as training data for speech recognition models, enabling accurate conversion of spoken language into text, powering voice assistants and dictation software.

Journals and Conferences in the area(in NLP)

Journals:

Computational Linguistics (CL): Official journal of the Association for Computational Linguistics (ACL), covering a wide range of NLP topics.

Natural Language Engineering (NLE): Focuses on practical applications of language processing.

Journal of Artificial Intelligence Research (JAIR):Publishes articles covering AI-related topics, including NLP.

Transactions of the Association for Computational Linguistics (TACL): ACL's journal, publishing papers on computational linguistics and NLP.

Language Resources and Evaluation Journal: Covers the creation, annotation, and use of language resources.

Conferences:

Association for Computational Linguistics (ACL): One of the most prominent conferences in NLP, covering a wide range of topics in computational linguistics.

Conference on Empirical Methods in Natural Language Processing (EMNLP):Focuses on empirical and data-driven approaches in NLP.

International Conference on Learning Representations (ICLR): Covers representation learning, which is crucial in NLP.

International Joint Conference on Artificial Intelligence (IJCAI): Includes NLP as part of its broader AI topics.

Conference on Neural Information Processing Systems (NeurIPS): Covers machine learning and includes NLP-related papers.

Conference on Computational Natural Language Learning (CoNLL): Focuses on NLP-related machine learning and learning from natural language data.

European Chapter of the Association for Computational Linguistics (EACL): A regional chapter of ACL, organizing conferences focused on NLP in Europe.

International Conference on Web and Social Media (ICWSM): Focuses on research at the intersection of social media and web technology, including NLP-related topics.

International Conference on Language Resources and Evaluation (LREC): Focuses on language resources and evaluation methods used in NLP.

These journals and conferences serve as important venues for researchers, scientists, and practitioners to present their work, exchange ideas, and stay updated on the latest developments in the field of NLP.