Name:- Deepak Prasad Div:- D16AD Roll no:- 44

NLP Assignment-01

DATE: 11/09/2023

Q.1 Explain with example different types of ambiguity in natural language?

Ans

1) Lexical ambiguity:-
This occurs when a word has multiple meanings, for instance, the word 'bank' can refer to a financial institute or the side of a river.

2) Syntatic ambiguity:-
This arises when a sentence can be parse in multiple ways due to its structure for example 'I saw the man with the telescope" can mean you saw a man with the telescope or you used a telescope to see a man

3) Semantic ambiguity:-
This occurs when a sentence can be interpreted in more than one way because of the meaning of the words used for instance 'She found a bat in the park' could be refer to a sport equipment bat or a flying mammal bat

4) Pragmatic ambiguity:-
This arises when the meaning of the sentence is unclear due to the context or tone. For example 'Can you pass the salt? could be request or a question about someone ability?

DATE:

Q.2 Discuss challenges of the NLP and how to overcome them

Ans Natural language processing faces several challenges due to the complexity and nuances of human language. Here are some major challenges and potential ways to overcome them.

a) Ambiguity :-
Ambiguity in a language can lead to confusion one way to tackle this is by using context based approaches. Machine learning models that consider surrounding word or sentence can often disambiguate the intended meaning

b) Variability and Informality :-
Language is incredibly diverse with slang colloquialisms and variation across cultures. Incorporating more extension and diverse training data can be help models better understand and generate different language style.

c) Lack of context :-
Understanding context is crucial for accurate comprehension. Models like transformer based architecture leverage attention mechanism to capture context over longer sequence of the aiding in overcoming challenge.

DATE:

Q.3

Ans Tokenization is a fundamental preprocessing step in natural language processing (NLP) that involves breaking down a text into individual words or units called tokens. These tokens sever as the building blocks for the various NLP tasks, such a language modeling text classification and machine translation.

Tokenization is important for many reasons.

a) Text understanding :-
Tokenization divides text into managable units allowing machine to understand the structure of sentence paragraphs and documents.

b) Feature extraction :-
In this tasks like text classification each is taken can be traked as a feature, enabling models to learns patterns and relationships within the text.

c) Language modelling :-
Tokenization aids in the building language models by breaking text into smaller units for predicting the next word or generating coherent text.

Code :

```
import re
text = "Tokenization is good"
tokens = re.findall(r'\b\w+\b', text)
print(tokens).
```

DATE:

Q.4

Stemming	Lemmatization
Stemming is the faster because it chops word without knowing the context of the word in given sentence.	Lemmatization is slower as compared to stemming but it knows the context of the word before processing.
It is a rule-based approach	It is a dictionary-based approach.
Accuracy is less	Accuracy is more
For example studies = 'studi'	For example studies = 'study'.

Porter stemmer algorithm :-

The porter stemmer algorithm is widely used stemming algorithm developed by Martin Porter in 1980. It follows a set of heuristic rules to transform words into their stems. The algorithm consist of several phases, each together targeting specific suffixes. The phases aim to progressively remove common suffixes simple words to their root forms.

DATE:

Q. 5 Explain Edit distance algorithm

Ans The edit distance algorithm measures the similarity between two strings by calculating the minimum number of single character edits required to transform one string into the other form.
The edits can be insertion, deletion or substitutions.
This widely in various applications such as spell checking, DNA sequence analysis and plagiarism detection.

Algorithm

```
function editDistance (str1, str2):
    m = length of str1
    n = length of str2
    Create a dp array of the (m+1) x (n+1)
    for i from 0 to m:
        dp[i][0] = i

    for j from 0 to n:
        dp[0][j] = j

    for i from 1 to m:
        for j from 1 to n:
            if str1[i-1] = str2[j-1]
            dp[i][j] = dp[i-1][j-1].
            else
            dp[i][j] = min(dp[i-1] + 1,
                           dp[i][j-1] + 1, dp[i-1][j-1] + 1).

    return dp[m][n].
```

DATE:

Forexample:-

cat and hat

∴ edit distance is 1
Subsitute the 'c' with 'h'.

Q.6 Use of finite automata and finite state transducers.
Ans

1) Tokenization:
FST can be used for segmenting text into tokens
For example in the english you can used an FST to split a setence into individual words or a sub word.

2) Part of speech tagging:-
FST can help assign parts of speech to words in a sentence
By constructing and fst that recognizes word sequences and assigns Pos tags, you can develop Pos tagging system.

3) Morophological analysis
FSTs are extensively used in morphological analysis to break down words into their constituent morphemes.

4) Spell checking and correction:
It can be used to do spell checking and correction system.

DATE:

Q.7 Importance of Morphological analysis.

1) Language Understanding
Understanding the structure of words is fundamental to understand language.

2) NLP tasks:
Its integral to various NLP tasks, including machine translation, text summarization, information retrival and sentiment analysis.

3) Resource-Scarce Language
In languages with rich morphological complexity morphological analysis vital for accurate processing

4) Search engines
For each search engines morphological analysis enables searching for variations of words.

5) Sentiment analysis:-
In sentiment analysis, morphological analysis helps identifying negations or words forms that change the polarity of a sentence.

Applications
1. Information Retrival
2. Machine translation
3. Sentiment analysis
4. Speech recognition.

DATE:

Q.8 N-grams.

Ans N grams are contiguous sequences of N-1 items words, letters or symbol from a given sample of text or speech. They are a fundamental concept in the NLP. N-grams are used to analyze and model the relationships between elements in a sequence. They are classified as:

1) Unigrams (1-grams):
These are single items in a sequence often individual words.

2) Bigrams (2-grams):
These consist of pairs of adjacent items in a sequence of words.

3) Trigrams (3-grams):
Trigrams are composed of the consective items.

4) N-grams (N>3):
N-grams can be any sequence of the N items.

Example:-

This is an group of words.

Unigrams :- 'This', 'is', 'an', 'group', 'of', 'words'.
Bigrams :- 'This is', 'an group', 'of words'.
Trigrams :- 'This is an', 'group of words'.

DATE:

Q.9 Penn Treebank

Ans Penn Treebank Pos tagset is a widely used tagging scheme for English that assign tags to words based on their grammatical roles in a sentence.
Following are the some of the tags:

1) CC (coordinating conjuction):
connects words prahses or clauses.

2) CD (cardinal number)
Represents numbers like 'one', 'two',

3) DT (Determiner)
modifies the noun and specifices their references such as the 'this', 'some'.

4) (JJ) adjective
Describes noun like, 'happy', 'blue', 'tall'.

5) NN (Noun, singular)
represent singular or mass noun like 'cat'

6) RB (adverb)
modifies verb, adjective, or other adverb.

7) To (to)
Part of the infinitive form of the verbs like 'to' 'in' 'to go'.

DATE:

Q.10 Rule based part of speech (Pos) tagging

1) Tokenization
The first step is to tokenize the input text into individual tokens

2) Linguistic Rules
Develop a set of linguistic rules that describes the characteristic and patterns.

3) Ambiguity Handling
Implementing rule based tagging may lead to ambiguity rules should be designed to handle such cases.

4) Expection Handling :-
Rules should account for exception and irgulon's in the language.

5) Iterative improvment
This involve analyzing.