Natural Language Processing Questions

Complete & Continue

Enable fullscreen

Q1. What is Lemmatization?

Ans: Lemmatization is the process of arriving at a lemma of a word. What is a lemma, then?

Lemma is the root from which a word is formed.

For example: If we lemmatize ‘studies’ and ‘studying’, we will end up with ‘study’ as its lemma. We got to this conclusion after the morphological analysis of both words. These were mapped in a dictionary which helped in arriving at the lemma

Q2 What is Stemming?

Ans- stemming is the process of reducing inflected (or sometimes derived) words to their word stem, base or root form—generally a written word form. ... A computer program or subroutine that stems word may be called a stemming program, stemming algorithm, or stemmer

Q3 What is the difference between Lemmatizing and Stemming?

Ans- **stemming** technique only looks at the form of the word whereas the lemmatization technique looks at the meaning of the word. It means after applying lemmatization, we will always get a valid word.

Most people think that lemmatizing is almost the same as stemming. This notion is incorrect. Lemmatization is a more detailed approach to reduce a given string and extract the core lemma from it. It is more complex than stemming. It should be noted that if ‘studies’ undergoes stemming, we will end up with ‘studi’ as it just removes the suffix without considering the linguistic morphology.

Q4. How NER works?

Ans- NER Stands for Named - Entity Recognition. It is also known as entity chunking/extraction, which is a popular technique used in information extraction to identify and segment the named entities and classify or categorize them under various predefined classes.

Basic NER processes structured and unstructured texts by identifying and locating entities. For example, instead of identifying “Steve” and “Jobs” as different entities, NER understands that “Steve Jobs” is a single entity. More developed NER processes can classify identified entities as well. In this case, NER not only identifies but classifies “Steve Jobs” as a person.

For more details visit the link below

1. https://expertsystem.com/entity-extraction-work/

2. https://medium.com/@Innoplexus/what-is-named-entity-recognition-7ed05beb7171#:~:text=NER%20plays%20a%20major%20role,by%20identifying%20and%20locating%20entities.&text=In%20this%20case%2C%20NER%20not,Steve%20Jobs%E2%80%9D%20as%20a%20person.

Q5. Why do we use Feature Scaling?

Ans- Feature Scaling or Standardization: It is a step of Data Pre Processing which is applied to independent variables or features of data. It basically helps to normalize the data within a particular range. Sometimes, it also helps in speeding up the calculations in an algorithm.

Q6 What is the significance of TF-IDF?

Ans - TF*IDF is an information retrieval technique that weighs a term's frequency (TF) and its inverse document frequency (IDF). Each word or term has its respective TF and IDF score. The product of the TF and IDF scores of a term is called the TF*IDF weight of that term.

Put simply, the higher the TF*IDF score (weight), the rarer the term and vice versa..

Words with high TF-IDF numbers imply a strong relationship with the document they appear in, suggesting that if that word to appear in a query, the document could be of interest to the person.

For a term t in document d, the weight Wt,d of term t in document d is given by:

Wt,d = TFt,d log (N/DFt)

Where:

TFt,d is the number of occurrences of t in document d.

DFt is the number of documents containing the term t.

N is the total number of documents in the corpus

For More details, Visit the below link

https://www.onely.com/blog/what-is-tf-idf/

Q7: What is ngram in NLP. ?

Ans - An N-gram means a sequence of N words. So for example, "Medium blog" is a 2-gram (a bigram), "A Medium blog post" is a 4-gram, and "Write on Medium" is a 3-gram (trigram). Basically, an N-gram model predicts the occurrence of a word based on the occurrence of its N – 1 previous words.

Let us see a way to assign a probability to a word occurring next in a sequence of words. First of all, we need a very large sample of English sentences (called a corpus).

For the purpose of our example, we'll consider a very small sample of sentences, but in reality, a corpus will be extremely large. Say our corpus contains the following sentences:

He said thank you.

He said bye as he walked through the door.

He went to San Diego.

San Diego has nice weather.

It is raining in San Francisco.

Let's assume a bigram model. So we are going to find the probability of a word based only on its previous word. Let us see a way to assign a probability to a word occurring next in a sequence of words. First of all, we need a very large sample of English sentences **(called a corpus).**

For the purpose of our example, we'll consider a very small sample of sentences, but in reality, a corpus will be extremely large. Say our corpus contains the following sentences:

He said thank you.

He said bye as he walked through the door.

He went to San Diego.

San Diego has nice weather.

It is raining in San Francisco.

Let's assume a bigram model. So we are going to find the probability of a word based only on its previous word. In general, we can say that this probability is (the number of times the previous word 'wp' occurs before the word 'wn') / (the total number of times the previous word 'wp' occurs in the corpus) =

(Count (wp wn))/(Count (wp))

Let's work this out with an example.

To find the probability of the word "you" following the word "thank", we can write this as P (you | thank) which is a conditional probability.

This becomes equal to:

=(No. of times "Thank You" occurs) / (No. of times "Thank" occurs) = 1/1 = 1

We can say with certainty that whenever "Thank" occurs, it will be followed by "You" (This is because we have trained on a set of only five sentences,and "Thank" occurred only once in the context of "Thank You").

Refernce:- https://blog.xrds.acm.org/2017/10/introduction-n-grams-need/

Q 8 What is Perplexed in NLP ?

Ans- The word "perplexed" means "puzzled" or "confused", thus Perplexity in general means the inability to tackle something complicated and a problem that is not specified. Therefore, Perplexity in NLP is a way to determine the extent of uncertainty in predicting some text.

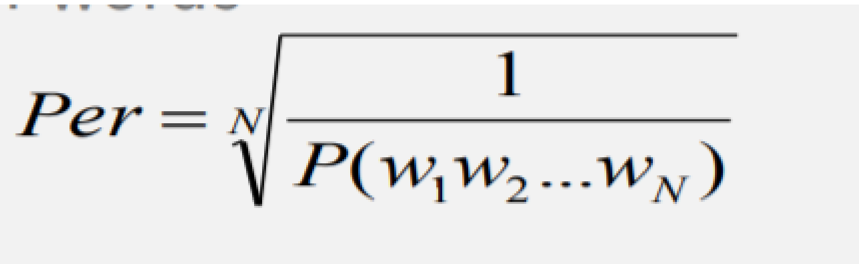
In NLP, perplexity is a way of evaluating language models. Perplexity can be high and low; Low perplexity is ethical because the inability to deal with any complicated problem is less while high perplexity is terrible because the failure to deal with a complicated is high.

Points to take care:

1. Average branching factor in predicting the next word .

2. Lower is better (lower perplexity -> higher probability)

N = number of words NP



$$Per = \sqrt[N]{\frac{1}{P(w_1 w_2 \ldots w_N)}}$$

Reference:- https://towardsdatascience.com/perplexity-intuition-and-derivation-105dd481c8f3

Q 9 What is Pragmatic Ambiguity in NLP?

Ans- Ambiguity, generally used in natural language processing, can be referred to as the ability to be understood in more than one way. In simple terms, we can say that ambiguity is the capability of being understood in more than one way. Natural language is very ambiguous. NLP has the following types of ambiguities.

1. Lexical Ambiguity

The ambiguity of a single word is called lexical ambiguity. For example, treating the word silver as a noun, an adjective, or a verb.

2. Syntactic Ambiguity

This kind of ambiguity occurs when a sentence is parsed in different ways. For example, the sentence “The man saw the girl with the telescope”. It is ambiguous whether the man saw the girl carrying a telescope or he saw her through his telescope.

3. Semantic Ambiguity

This kind of ambiguity occurs when the meaning of the words themselves can be misinterpreted. In other words, semantic ambiguity happens when a sentence contains an ambiguous word or phrase. For example, the sentence “The car hit the pole while it was moving” is having semantic ambiguity because the interpretations can be “The car, while moving, hit the pole” and “The car hit the pole while the pole was moving”.

.

4. Anaphoric Ambiguity

This kind of ambiguity arises due to the use of anaphora entities in discourse. For example, the horse ran up the hill. It was very steep. It soon got tired. Here, the anaphoric reference of “it” in two situations cause ambiguity.

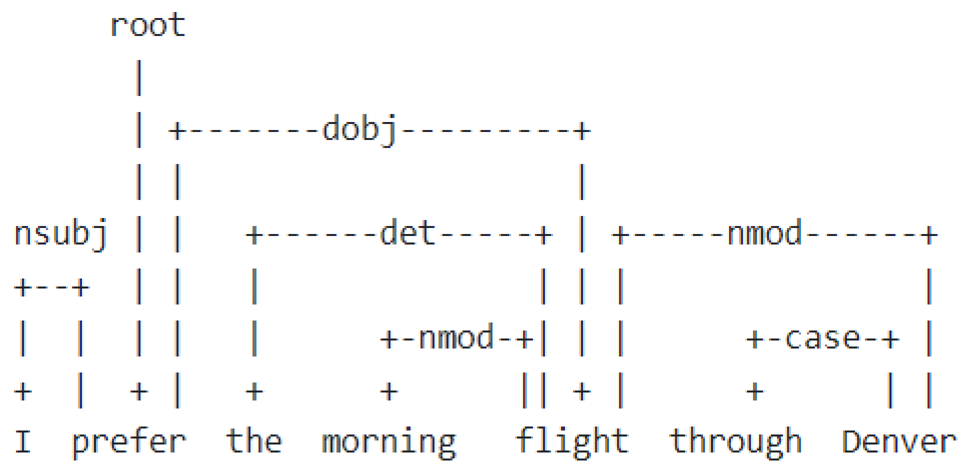
5.Pragmatic ambiguity

Such kind of ambiguity refers to the situation where the context of a phrase gives it multiple interpretations. In simple words, we can say that pragmatic ambiguity arises when the statement is not specific. For example, the sentence “I like you too” can have multiple interpretations as I like you (just like you like me), I like you (just like someone else does).

Q9. Explain Dependency Parsing.

Ans - Dependency parsing is the task of extracting a dependency parse of a sentence that represents its grammatical structure and defines the relationships between "head" words and words, which modify those heads.

Example:



```
     root
      |
      | +-------dobj---------+
      | |                    |
nsubj | |   +------det-----+ | +-----nmod------+
+--+  | |   |              | | |               |
|  |  | |   |      +-nmod-+| | |      +-case-+ |
+  |  + |   +      +      || + |      +      | |
I  prefer  the  morning   flight  through  Denver
```

Relations among the words are illustrated above the sentence with directed, labeled arcs from heads to dependents (+ indicates the dependent).

Refernce:-https://medium.com/@5hirish/dependency-parsing-in-nlp-d7ade014186

Q 10 What is Pragmatic Analysis?

Ans - Pragmatic Analysis is part of the process of extracting information from text. Specifically, it’s the portion that focuses on taking a structured set of text and figuring out what the actual meaning was.

Why is this important? Because a lot of text’s meaning does have to do with the context in which it was said/written. Ambiguity, and limiting ambiguity, are at the core of natural language processing, so needless to say, pragmatic analysis is actually quite crucial with respect to extracting meaning or information.

Q11. Explain Masked language modeling?

Ans- Masked language modeling is a fill-in-the-blank task, where a model uses the context words surrounding a [MASK] token to try to predict what the [MASK] word should be.

The model shown here is BERT, the first large transformer to be trained on this task. Enter text with one or more "[MASK]" tokens and the model will generate the most likely substitution for each.