**FAQs and Resources**

**Topics: Introduction to Natural Language Processing**

1. **What is preprocessing for language data and why is it necessary?**Natural language data in its raw form cannot be fed to a machine-learning algorithm, therefore, it is paramount to preprocess the dataset. The process of transforming language data into something that an algorithm can digest is called preprocessing. Cleaning the text is the first step in preprocessing. This step involves converting data to lowercase, removing stop words, unnecessary characters or tags, redundant spaces or newlines from the data. Next step is normalizing the dataset. This step involves transforming the data using standard reductions such as stemming and lemmatization.
2. **What are the different frameworks that can be used for working with natural language data?**The most frequently used framework for natural language processing in python is NLTK (Natural Language Toolkit). NLTK comes with off the shelf tokenizers, additional methods for regular expression based tokenizer, pre-trained models for stemming and lemmatization, part of speech tagging and entity recognition. NLTK also offers a wide range of natural language in multiple languages. Spacy is another most commonly used python package for natural language processing. While NLTK mostly offers traditional algorithms as a pre-trained model, Spacy also has a suite of pre-trained neural networks for a variety of tasks. This includes pre-trained neural network-based models for POS tagging, named entity recognition, language models, etc.
3. **What are the frequentist methods that are used in NLP and why are they called frequentist?**Methods such as a binary bag-of-words, bag-of-words, Tf-IDF are frequentist methods. All of these methods do not take into consideration the order in which words in a sentence occur. Only the frequency of each word i.e.number of times each word occurs in a sentence is taken into consideration, therefore, they are called as frequentist methods. One of the core problems with such methods in dealing with ambiguous words i.e. words that are both noun and verb. For e.g. the word book can be a noun e.g. reading a book and a verb e.g. book a ticket. Since the order of the words in the sentence is not taken into consideration frequentist methods fail in classifying samples containing ambiguous words.
4. **Is stemming and lemmatization always useful in the training the model?**Stemming and lemmatization are techniques used for information retrieval. Motivation is that different forms of words refer to the same concept. For e.g., if you search for the word “running” in the corpus, you may want to extract the articles containing the word “run”, “runner” and “runs” as well. Stemming does that by reducing the word to its root form. In many cases, this leads to the ambiguity between words. For e.g., “busy” and “business” are mapped to the same root. Therefore, depending on the corpus and vocabulary it's not always useful to use stemming or lemmatization.
5. **What are stop words and should we always remove them from the data?**Words that occur frequently in the dataset as connecting words within sentences and convey almost zero information towards solving the given problem are called stop words. For e.g. words like, is, the, and, are, they, there, etc. occur in almost all sentences and cannot be used as features for classification of a sentence.   
   Another aspect of looking at stop words is that they provide context and coherent meaning to the sentence. Therefore, tasks, where context is more important like in the case of occurrence of ambiguous words considering stopwords, is a good idea. Whereas task such as basic chatbot commands like what is time, etc. stop words can be removed.
6. **What is the advantage of Tf-idf over the traditional bag of words model?**A word that occurs in all the samples does not provide any information. While bag of words model considers the contribution of that all words in a sentence same, Tf-idf penalizes the contribution of words by inverse document frequency count. This normalizes the contribution of each word. If a word occurs in a few sentences, it most probably represents the context of the sentence. On the other hand, if a word occurs in every sentence it is probably a stop word or an irrelevant word and its contribution should be penalized. This objective is achieved by TF-Idf.
7. **What do we mean by the word context in NLP? Why is it necessary to encode context in the model?**The word context in NLP generally refers to the complete story of a sentence or a document. There are many words that have different meanings in different contexts. Therefore, tasks like Machine Translation, Question-Answering that require the precise meaning of each word in the given sentence require the context of the sentence also to be known. This allows models to differentiate between meanings of ambiguous words.
8. **What are n-grams and do they encode context information?**One of the traditional methods to encode context information in the model is n-grams. Frequentist methods count the occurrences of two or more words occurring together in the corpus. Two words model is the bi-gram model, similarly trigram and n-gram model. By looking at the occurrence at two or more words together we indirectly encode the context information in the model.
9. **What is the Markovian assumption? Does it always hold true for natural language processing?**As per the markovian assumption, the current state of the model is dependent only on the previous state of the model. In NLP for a language model, this can be understood as current word prediction depending on the previous word only. Though this assumption makes solving the problem statement much easier, It’s not true for most of the cases in NLP. The current state of the model will most of the time depend on more than one previous state. For e.g. consider the question, “London is related to Britain in the same as Paris is to \_\_\_\_\_\_ “. Here blank is the current we are trying to find and rest of the word in the sentence are the previous state. Answering this question will require information right from the beginning of the sentence.
10. **Where to find datasets to work with?**

Kaggle, NLTK, UCI dataset repository.