Natural Language Toolkit

Introduction

NLTK provides users opportunities to manipulate text date like classification, tokenization, stemming, tagging, parsing, and sematic reasoning. It's an open-source library called the *Natural Language Toolkit (NLTK)* in Python. Potential users like students, engineers, educators and so on can utilize this toll to process raw text and build linguistic model with technological applications. In this review, we are mainly focused on processing raw text and categorizing and tagging words.

Body

Tokenization transforms a sentence, document, or a collection of documents to a list of words and punctuation. This is the initial step of language processing where *word_tokenize* function breaks up the string words and punctuation. Make sure the unwanted material from website like footer and header are removed before we do any linguistic processing. We might want to normalize text at first place, like converting text to lower case before doing anything with its words. NLTK has built-in stemmers like Porter and Lancaster for stripping affixes. In a word, NLTK in python makes text mining easier, it helps us to extract text from different sources such as websites, electronic books, and text file. Data type "string" in python make the whole process more efficient because it has so many useful functions, like split, join, find, index and so on. This builds a good foundation for the following text modelling.

Tagging is the second step after tokenization which is the process of adding tags or annotation to various components of unstructured data. [1&2] Brining tagging into work, we can use this technique to integrate structured data and unstructured data to build a complete data warehouse. On the one hand, business can obtain useful information from daily news articles and reports in an efficient way by adding tags for each object, so that people are bound to query and search. On the other hand, product-oriented company can learn from the comments from users all over the world. The insight brought from large volumes of warranty claims data makes company better understand the root of the problem and make changes for it. [2] Therefore, it's important for us to implement this strategy.

NLTK provides a direct function to gives each word a tag, using ntlk.pos_tag(text), it also gives documentation for each tag. As we talked before, a text a treated in python as a list of words, and an index is used for each word so that we can use a number to get back a word. With this idea, we can use dictionary data type in Python to achieve word-frequency goal, where we input the word and get back a number as frequency. There are different types of tagging methods, such as default tagger, regular expression tagger, unigram tagger and n-gram tagger [3]. To optimize our accuracy, we use a combined technique known as backoff, when a more specialized model cannot assign a tag in each context, we backoff to a more general model. For example, bigrams, like a more specialized model, since it just gives word pairs from the existing sentence. If the sentence is, "I am a dog person", we are expected to gain 4 pairs, and one of them is "dog person", which is more likely to occur together. But if we can't tag the pair, we will go back to the unigram model which we assume the occurrence of each word is independent of its previous word. Similar idea applies to trigram.. n-gram. This brings the idea of automatic tagging, which is an important step in the NLP pipeline. We start at the bigram, if it cannot find a tag for the token, we would like to try unigram. If the unigram is also unable to find a tag, we will use a default tagger.

Conclusion

This review introduces a useful toolkit in Python to perform text mining. Python's built-in function makes beginning steps of text mining (tokenization and tagging) much easier and more efficient. We can easily grab text from websites, electronic books and text file. The data type in python like string and dictionary also make the whole process clearer and more intuitive. With this strong foundation, we can extract information from text, analyze sentence structure and analyze the meaning of sentences more accurately.

Reference

1. *ch03.rst2*. (n.d.). Retrieved November 5, 2022, from https://www.nltk.org/book/ch03.html
2. *What is Tagging? | Text Analytics - Textrics*. (2022, January 6). Medium. https://textrics.medium.com/what-is-tagging-text-analytics-954f5f9f01ab
3. *5. Categorizing and Tagging Words*. (n.d.). Retrieved November 5, 2022, from https://www.nltk.org/book/ch05.html