COMP 6751 Natural Language Analysis

Project Report 1

Student name: Md Sakib Ullah Sourav

ID: 40264066

Expectations of originality:

I, Md Sakib Ullah Sourav (student id 40264066), certify that this submission is my original work and meets the Faculty's Expectations of Originality.

Date: September 22, 2023

Contents

1. Project Goal

This project aims to investigate several modules and features of the Natural Language Toolkit (NLTK) in order to gain a deeper understanding of the field of natural language processing. The reference data that we are using is the NLTK version of the Reuter's text training/267 dataset. The voyage that we shall undertake will consist of the following essential elements:

The tokenizer module facilitates the segmentation of text into discrete units, such as words or tokens, which serves as the foundational component for a majority of natural language processing (NLP) endeavours.

Sentence splitting is a fundamental process that allows for the segmentation of text into individual phrases. This phase is crucial in several natural language processing (NLP) applications, such as text summarization and sentiment analysis.

Part-of-Speech (POS) tagging is a linguistic process that entails assigning a grammatical category, such as noun, verb, or adjective, to each word inside a given phrase. The acquisition of this knowledge is of utmost importance in comprehending the organisation and significance of written discourse.

The process of gazetteer annotation entails the identification and annotation of distinct things, such as nation names, currencies, and units, by referencing predetermined lists known as gazetteers.

Named Entity Recognition (NER) is a sophisticated technology that surpasses the rudimentary process of gazetteer annotation. The task at hand is the identification and categorization of named entities inside a given text, including individuals, organisations, geographical places, dates, and other relevant entities.

The goal of Measured Entity Detection (MED) involves the identification and extraction of numeric or quantitative information from unstructured text data, focusing on locating and classifying entities. Unlike Named Entity Recognition (NER), which is concerned with

identifying and categorising things, MED specifically aims to extract numerical or quantitative data from textual sources.

In the process of this project, we will explore the NLTK modules in order to execute diverse tasks, therefore enhancing our comprehension of their practical applications in addressing natural language processing difficulties.

2. Modules of the Project

The given passage in the of Reuter's text training/267 is as below-

INDONESIA UNLIKELY TO IMPORT PHILIPPINES COPRA

Indonesia is unlikely to import copra from the Philippines in 1987 after importing 30,000 tonnes in 1986, the U.S. Embassy's annual agriculture report said. The report said the 31 pct devaluation of the Indonesian rupiah, an increase in import duties on copra and increases in the price of Philippines copra have reduced the margin between prices in the two countries. Indonesia's copra production is forecast at 1.32 mln tonnes in calendar 1987, up from 1.30 mln tonnes in 1986.

2.1 Tokenizer

Tokenizers are a useful tool in the field of text analysis since they facilitate the decomposition of raw text into smaller units, such as individual words or phrases. This process aids in the management and organisation of textual data, enabling more effective analysis.

Initially, the tokenizer from the Natural Language Toolkit (NLTK) was integrated into the integrated development environment (IDE) I used in order to extract smaller segments from the provided text.

```
tokens = nltk.word_tokenize(document_content)
```

However, it could not tokenize some words properly. One of the primary challenges is handling languages with complex grammatical rules and word formations. Tokenizers must also address punctuation marks, contractions, and hyphenated words correctly

To encounter the challenges, I then implemented below regular expression tokenizer-

Operator	Explanation
(?:)	Non-capturing group
(?:[A-Z]\.)+	abbreviations, e.g. U.S.A.
|\d+(?:,\d{3})*(?:\.\d+)?	Numbers with optional commas and decimals
\w+(?:-\w+)*	words with optional internal hyphens and apostrophes
|[.,;!?()]	Common punctuation
|\$?\d+(?:\.\d+)?%?	currency and percentages
|(?:[...])	ellipsis …
|[“”‘’"']	Quotation marks
|[\[\]{}:<>]	Brackets and colons
|(?:\S)	Any other character (non-space)
|[——-]	Dashes and hyphens
|[/]	Slashes

2.2 Sentence Splitting (SS)

A significant obstacle in sentence segmentation (SS) is to the handling of punctuation marks that occur at the conclusion of sentences and include ambiguity, such as periods found within abbreviations or ellipses, as they might potentially be misinterpreted as sentence borders. Sentences may also extend across numerous lines or paragraphs, posing challenges in identifying sentence boundaries.

```
sentences = nltk.sent_tokenize(document_content)
```

The script nltk.sent_tokenize returns a sentence-tokenized copy of the given passage. Here in this script, I used a function called “enumerate” which is a built-in Python function that allows you to iterate over an iteral (e.g., a list or string) while keeping track of the current index or position of each item.

2.3 POS Tagging

First of all, I imported the needed functions along with the below one which uses the Penn Treebank POS tags.

pos_tags = pos_tag(words)

Eventually, this function typically returns a list of tuples those represent the grammatical roles of words, such as nouns (NN), verbs (VB), adjectives (JJ), and others.

2.4 Gazetteer

The process of gazetteer annotation pertains to the annotation of certain named things or terms inside a given text, such as nation names, currencies, and units of measurement.

```
# Define gazetteers
countries = ["INDONESIA", "PHILIPPINES"]
currencies = ["rupiah", "pct"]
units = ["tonnes", "mln"]
# Create regular expression patterns for each category
country_pattern = r'\b(?:' + '|'.join(re.escape(country) for country in countries) + r')\b'
currency_pattern = r'\b(?:' + '|'.join(re.escape(currency) for currency in currencies) + r')\b'
unit_pattern = r'\b(?:' + '|'.join(re.escape(unit) for unit in units) + r')\b'

# Define chunking rules using regular expressions
chunk_grammar = r"""
        COUNTRY: {<NNP>{1,}}
CURRENCY: {<NN><NN><NN><NN>?}
UNIT: {<NN><NN><NN><NN>?}
"""
# Create a chunk parser with the defined grammar
```

```
chunk_parser = nltk.RegexpParser(chunk_grammar)

# Tag the words with part-of-speech tags
pos_tags = nltk.pos_tag(words)

# Parse the tagged words using the chunk parser
chunks = chunk_parser.parse(pos_tags)
```

I had to create regular expressions in code to specify country, currency and unit. Then parsing the words using chunk parser. Interruption in data, contextual identification and domain adaptation is important to get the desired output in terms of gazetteer annotation using NLTK.

2.5 Named Entity Recognition (NER)

Identifying named entities can be challenging when a word can have multiple interpretations. For example, "Washington" could refer to a city or a person's name. Here I adopted the below command to perform the NER on our given passage.

```
ner_tags = ne_chunk(nltk.pos_tag(words))
```

The main four categories in NER are person (PER), organization (ORG), Geo-political entity (GPE), and GSP which refers a specific place or entity.

2.6 Measured Entity Detection (MET)

By using regular expression and domain specific tagging, such as list of words and symbols, MET can be effectively extracted. Here I used the below expressions to get the best results:

```
(\d{1,3}(,\d{3})*(\.\d+)?\s*(mln|pct|tonnes)
```

I printed the value and corresponding units in the end to show the METs.

3. Limitations

i) During sentence splitting in module 2, initially the output takes the header phrase as the first sentence. Then I made two approaches. In the first approach it separates he header phrase from the sentences. But the second approach excludes the first sentence along with the header phrase.

ii) In the NER, the output (shown below) of the word “U. S. Embassy” should be Organization but our model splits it into two different entity

Entity: U.S., Label: GPE
Entity: Embassy, Label: ORGANIZATION