Natural Language Processing (NLP): Step-by-Step Explanation

1. Introduction to NLP

Natural Language Processing (NLP) is a field of artificial intelligence that focuses on the interaction between computers and human (natural) languages. The goal is to enable machines to read, understand, and derive meaningful insights from human language.

NLP is primarily used to work with unstructured data—text that does not follow a pre-defined format, such as news articles, social media posts, or interview transcripts.

2. Input Text Example

The notebook begins by defining a block of text about investment behaviors in Kenya. This is a typical example of unstructured data used in NLP.

3. Installing and Downloading NLTK Resources

The Natural Language Toolkit (NLTK) is a Python library used extensively in NLP tasks. The command `nltk.download('punkt')` downloads the Punkt tokenizer models, which are used to split text into words or sentences.

4. Tokenization

Tokenization is the process of breaking down text into smaller units called tokens. The notebook uses `word_tokenize` to split the text into individual words and punctuation marks.

5. Stopwords Removal

Stopwords are commonly used words such as 'the', 'is', and 'in' that usually do not contain significant meaning. Removing these helps to focus on the more meaningful words in the text.

6. Stemming and Lemmatization

Stemming reduces a word to its root form, e.g., 'running' becomes 'run'. Lemmatization also reduces words to their root form, but uses a dictionary to ensure the root word is meaningful.

7. Part-of-Speech (POS) Tagging

POS tagging assigns parts of speech to each word, such as noun, verb, adjective, etc. It helps in understanding the grammatical structure of the sentence.

8. Named Entity Recognition (NER)

NER identifies names of people, organizations, locations, and other proper nouns in the text. This is useful for information extraction tasks.

9. Summary and Next Steps

The notebook demonstrates the basic steps of processing unstructured text data using NLTK. You can extend it with additional tasks such as frequency distribution, visualization using WordCloud, or topic modeling for more advanced insights.

10. Detailed Code Cell Explanations

Code Cell 1

```
# NLP works with unstructured data
text = '''
  Investors are also looking at property and gold as potential hedges against inflation, it added.

Also read: Inside Ruto's radical plan to address high cost of commodities

"People are spending less and looking to invest more to prepare for the higher cost of future financial liabilities. Consequently, investors expect to reduce their allocation to cash in the coming years," said Marc Van de Walle, the global head of wealth management in deposits and mortgages at Standard Chartered.

In Kenya, 95 percent of respondents said they have set up new goals for the future, with 50 percent of them citing rising inflation as the main challenge to achieving their investment goals.

Meanwhile, one-third (33 percent) cited uncertainty in the global economy and a fifth (22 percent) cited fear of poor returns on investments.

Some 53 percent of the respondents said they are saving for their children's education, 51 percent said they are saving to keep up with the rising costs, and 50 percent for retirement.

Further, 73 percent of people surveyed are currently expected to invest more in digital assets in 2023, while only 10 percent are expected to invest less in 2023.

"Despite the recent upheaval in digital assets, Kenyans are still interested in them. This is reinforced by our research which shows that 75 percent of Kenyans surveyed still believe that digital assets are an important part of any investment portfolio and just 9 percent disagree," said the study.

This comes amid economic uncertainty due to rising commodity prices, recession risks, and geopolitical shifts.
```

```
The survey by the lender noted that high inflation is prompting shifts in how investors allocate their funds across different asset classes, from cash and equities to digital assets and sustainable investments.

Nearly a third of Kenyan investors are changing their investment plans in a bid to beat rising inflation, a new poll by Standard Chartered Bank revealed.

'''
```

This code block initializes a variable `text` with a multi-line string. It represents a news article discussing investment trends in Kenya. This unstructured text serves as input for NLP processing.

Code Cell 2

```
import nltk
nltk.download("punkt")
```

This line downloads the 'punkt' tokenizer models from NLTK. These models are used for sentence and word tokenization.

Code Cell 3

```
# Convert text to sentences or works
from nltk.tokenize import sent_tokenize, word_tokenize
tokenized_text = word_tokenize(text)
print(tokenized_text)
```

This code block initializes a variable `text` with a multi-line string. It represents a news article discussing investment trends in Kenya. This unstructured text serves as input for NLP processing.

Code Cell 4

```
nltk.download("stopwords")
from nltk.corpus import stopwords
print(stopwords.fileids())
# Get english stopwords
mystopwords = set(stopwords.words('english'))
print(mystopwords)
```

This line downloads the 'punkt' tokenizer models from NLTK. These models are used for sentence and word tokenization.

Code Cell 5

```
# justpaste.it/bi8nu
filtered_words =[]
for word in tokenized_text:
    if word.lower() not in mystopwords:
        filtered_words.append(word)

print("Old Words ", tokenized_text)
print("New Words ", filtered_words)
```

This cell contains custom code related to NLP processing.

Code Cell 6

```
# justpaste.it/biph5
clean_words = [word for word in filtered_words if word.isalpha()]
print(clean_words)
```

This cell contains custom code related to NLP processing.

Code Cell 7

```
from nltk.probability import FreqDist
frequency = FreqDist(clean_words)
frequency.most_common(30)
```

This cell contains custom code related to NLP processing.

Code Cell 8

```
# justpaste.it/cq844
from wordcloud import WordCloud, STOPWORDS
from PIL import Image
#Function to Create Wordcloud
def create_wordcloud(text):
    stopwords = set(STOPWORDS)
    wc = WordCloud(background_color="white",
    max_words=3000,
    stopwords=stopwords,repeat=True)
    wc.generate(str(text))
```

```
wc.to_file("wc.png")
print("Word Cloud Saved Successfully")
path="wc.png"
display(Image.open(path))
```

This cell contains custom code related to NLP processing.

Code Cell 9

This cell contains custom code related to NLP processing.

Code Cell 10

```
create_wordcloud(frequency.most_common(30))
```

This cell contains custom code related to NLP processing.

Code Cell 11

```
#Function to ngram
from sklearn.feature_extraction.text import CountVectorizer
def get_top_n_gram(corpus,ngram_range,n=None):
 vec = CountVectorizer(ngram_range=ngram_range,stop_words =
'english').fit(corpus)
 bag_of_words = vec.transform(corpus)
 sum_words = bag_of_words.sum(axis=0)
 words_freq = [(word, sum_words[0, idx]) for word, idx in
vec.vocabulary_.items()]
 words_freq =sorted(words_freq, key = lambda x: x[1], reverse=True)
 return words_freq[:n]
```

This cell contains custom code related to NLP processing.