DESIGN AND IMPLEMENT INDEXER FOR ODIA USING NLP

GROUP NAME

RAHUL SAHOO   BINATA RANJAN BHARTI

What is Indexer:

The purpose of storing an index is to optimize speed and performance in finding relevant documents for a search query.

DESIGN AND IMPLEMENT INDEXER FOR ODIA USING NLP:

NLTK is a powerful Python package that provides a set of diverse natural languages algorithms. It is free, opensource, easy to use, large community, and well documented. NLTK consists of the most common algorithms such as tokenizing, part-of-speech tagging.

Tokenization:

Tokenization is the first step in text analytics. The process of breaking down a text paragraph into smaller chunks such as words or sentence is called Tokenization. Token is a single entity that is building blocks for sentence or paragraph.

Syntax:

```
 Import nltk                     #in python library
nltk.download('popular')         #download and install all packages
```

Sentence Tokenization:

```
>>> from nltk.tokenize import sent_tokenize

>>>  odia_text="""ସବୁ ମନୁଷ୍ୟ ଜନ୍ମକାଳରୁ ସ୍ୱାଧୀନ. ସେମାନଙ୍କର ମର୍ଯ୍ୟାଦା ଓ ଅଧିକାର ସମାନ."""

>>>  from nltk.tokenize import sent_tokenize

>>>  tokenized_text=sent_tokenize(odia_text)

>>>  print(tokenized_text)

['ସବୁ ମନୁଷ୍ୟ ଜନ୍ମକାଳରୁ ସ୍ୱାଧୀନ.', 'ସେମାନଙ୍କର ମର୍ଯ୍ୟାଦା ଓ ଅଧିକାର ସମାନ.']
```

Word Tokenization:

Word tokenizer breaks text paragraph into words.

```
>>> odia_text="""ସବୁ ମନୁଷ୍ୟ ଜନ୍ମକାଳରୁ ସ୍ୱାଧୀନ. ସେମାନଙ୍କର ମର୍ଯ୍ୟାଦା ଓ ଅଧିକାର ସମାନ."""

>>> tokens=nltk.word_tokenize(odia_text)

>>> print(tokens)
```

#OUTPUT

```
['ସବୁ', 'ମନୁଷ୍ୟ', 'ଜନ୍ମକାଳରୁ', 'ସ୍ୱାଧୀନ', '.', 'ସେମାନଙ୍କର', 'ମର୍ଯ୍ୟାଦା', 'ଓ', 'ଅଧିକାର', 'ସମାନ', '.']
```

Frequency Distribution:

```
>>> from nltk.probability import FreqDist

>>> fdist = FreqDist(odia_text)

>>>print(fdist)
```

#OUTPUT

```
<FreqDist with 10 samples and 11 outcomes>

>>> fdist.most_common(2)
```

#OUTPUT

```
[('.', 2), ('ସବୁ', 1)]
```

POS Tagging:

POS Tagging is the process of assigning a part of speech, like noun, verb, pronoun, adverb, adverb or other lexical class marker to each word in a sentence. POS Tagging looks for relationships within the sentence and assigns a corresponding tag to the word.

```
>>> nltk.pos_tag(tokens)
[('ସବୁ', 'JJ'), ('ମନୁଷ୍ୟ', 'NNP'), ('ଜନ୍ମକାଳରୁ', 'NNP'), ('ସ୍ୱାଧୀନ', 'NNP'),
('.', '.'), ('ସେମାନଙ୍କର', 'VB'), ('ମର୍ଯ୍ୟାଦା', 'JJ'), ('ଓ', 'NNP'),
('ଅଧିକାର', 'NNP'), ('ସମାନ', 'NNP'), ('.', '.')]
```