natural-language-processing-2

March 11, 2024

[27]:
```
import pandas as pd
# Import necessary libraries
import nltk
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer, WordNetLemmatizer
import spacy
import gensim

from sklearn.feature_extraction.text import CountVectorizer
import pandas as pd

def generate_features_multiple(unwanted_sms_list):
    # Initialize lists to store feature values
    binary_features_list = []
    categorical_features_list = []

    # Binary features
    keywords = ["CONGRATULATIONS", "loan", "CRB", "Dial", "register",␣
 ↪"STOP","KES","KSH"]

    # Process each unwanted SMS
    for sms in unwanted_sms_list:
        # Initialize dictionaries to store binary and categorical features for␣
 ↪each SMS
        binary_features_dict = {}
        categorical_features_dict = {}

        # Check for presence of keywords
        for keyword in keywords:
            binary_features_dict[keyword] = 1 if keyword in sms else 0

        # Categorical features
        sender_name = "Unknown"  # Sender's name not provided in the example
        type_of_offer = "Unknown"  # Type of offer not provided in the example
        call_to_action = "Unknown"  # Call-to-action not provided in the example
        # Append binary features dictionary to the list
        binary_features_list.append(binary_features_dict)

        # Append categorical features to dictionary
        categorical_features_dict['sender_name'] = sender_name
        categorical_features_dict['type_of_offer'] = type_of_offer
        categorical_features_dict['call_to_action'] = call_to_action

        # Append categorical features dictionary to the list
        categorical_features_list.append(categorical_features_dict)

    # Convert lists to DataFrames
    binary_features_df = pd.DataFrame(binary_features_list)
    categorical_features_df = pd.DataFrame(categorical_features_list)

    return binary_features_df, categorical_features_df

# Unwanted Smses
unwanted_sms_list = [
    "CONGRATULATIONS,Samson You qualify for a loan of 10,500 KSH In CRB?,NO␣
  ↪PROBLEM We have the INFO Dial *336*16# to register & borrow STOP *456*9*5#",
    "KCB M_PESA LOANS is now qualify to your M-PESA users 21% Savings from␣
  ↪5,000 ,10,000, 20,00 ,50,000,100,000 To apply send 555555 To 0755120267␣
  ↪>Call",
    "PERSONAL LOAN You qualify for KES 1,000-15,000 to MPESA 791848007.No CRB␣
  ↪check.Use a simple step here.Dial *336*16# to register & borrow. STOP␣
  ↪*456*9*5#",
    "You qualify for KES 7,600 to YOUR  MPESA NO. DIAL *453*210# for direct␣
  ↪application OR visit https://jijengenaloan.com to REGISTER and BORROW. STOP␣
  ↪*456*9*5#",
    "BUSINESS LOAN Samson! You qualify for KES 1,500-15,000 to MPESA 791848007.
  ↪No CRB check.Use a simple step here. Dial *453*200# to register & borrow.",
    "Hey , tried Branch yet? Click here to apply at https://branch.co/my-loan␣
  ↪and get your cash in under 10 minutes if you qualify."
]
binary_features_df, categorical_features_df =␣
  ↪generate_features_multiple(unwanted_sms_list)

print("Binary Features Table")
binary_features_df.head()
```

```
Binary Features Table
```

[27]:

	CONGRATULATIONS	loan	CRB	Dial	register	STOP	KES	KSH
0	1	1	1	1	1	1	0	1
1	0	0	0	0	0	0	0	0
2	0	0	1	1	1	1	1	0

```
3                 0     1    0     0         0     1    1    0
4                 0     0    1     1         1     0    1    0
```

[9]:
```python
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.preprocessing import OneHotEncoder

def extract_categorical_features(sms_text):
    # Placeholder values
    type_of_offer = "Unknown"
    call_to_action = "Unknown"



    # Extract type of offer
    if "LOAN" in sms_text:
        type_of_offer = "Loan"

    # Extract call to action
    if "Dial" in sms_text:
        call_to_action = "Dial"
    elif "apply" in sms_text:
        call_to_action = "Apply"

    return type_of_offer, call_to_action

def generate_features_multiple(unwanted_sms_list):
    categorical_features_list = []

    for sms in unwanted_sms_list:
      type_of_offer, call_to_action = extract_categorical_features(sms)

        # Create dictionary to store categorical features
      categorical_features_dict = {
            'type_of_offer': type_of_offer,
            'call_to_action': call_to_action
        }

        # Append categorical features to list
      categorical_features_list.append(categorical_features_dict)

    # Convert list of dictionaries to DataFrame
    categorical_features_df = pd.DataFrame(categorical_features_list)

    return categorical_features_df


unwanted_sms_list = [
    "CONGRATULATIONS,Samson You qualify for a loan of 10,500 KSH In CRB?,NO␣
↪PROBLEM We have the INFO Dial *336*16# to register & borrow STOP *456*9*5#",
    "KCB M_PESA LOANS is now qualify to your M-PESA users 21% Savings from␣
↪5,000 ,10,000, 20,00 ,50,000,100,000 To apply send 555555 To 0755120267␣
↪>Call",
    "PERSONAL LOAN You qualify for KES 1,000-15,000 to MPESA 791848007.No CRB␣
↪check.Use a simple step here.Dial *336*16# to register & borrow. STOP␣
↪*456*9*5#",
    "You qualify for KES 7,600 to YOUR  MPESA NO. DIAL *453*210# for direct␣
↪application OR visit https://jijengenaloan.com to REGISTER and BORROW. STOP␣
↪*456*9*5#",
    "BUSINESS LOAN Samson! You qualify for KES 1,500-15,000 to MPESA 791848007.
↪No CRB check.Use a simple step here. Dial *453*200# to register & borrow.",
    "Hey , tried Branch yet? Click here to apply at https://branch.co/my-loan␣
↪and get your cash in under 10 minutes if you qualify."
]

categorical_features_df = generate_features_multiple(unwanted_sms_list)
print("Categorical Features Table")
categorical_features_df
```

```
Categorical Features Table
```

[9]:
```
  type_of_offer call_to_action
0       Unknown           Dial
1          Loan          Apply
2          Loan           Dial
3       Unknown        Unknown
4          Loan           Dial
5       Unknown          Apply
```

[10]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
```

[11]:
```
# Download NLTK resources if not already downloaded
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data…
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package stopwords to /root/nltk_data…
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data…
```

[11]: True

[12]:
```
# Load spaCy model
nlp = spacy.load("en_core_web_sm")
```

[13]:
```
# Text from the online Document
document = """
Introduction
Natural Language Processing (NLP) is one of the hottest areas of artificial
intelligence (AI) thanks to applications like text generators that compose␣
 ↪coherent essays,
chatbots that fool people into thinking they’re sentient,
and text-to-image programs that produce photorealistic images of anything you␣
 ↪can describe.
Recent years have brought a revolution in the ability of computers to␣
 ↪understand human languages,
programming languages, and even biological and chemical sequences, such as DNA␣
 ↪and protein structures,
that resemble language.
The latest AI models are unlocking these areas to analyze the meanings of input␣
 ↪text and generate meaningful,
expressive output.
What is Natural Language Processing (NLP)Natural language processing (NLP)
is the discipline of building machines that can manipulate human language - or␣
 ↪data that resembles human language - in the way that it is written,
spoken, and organized. It evolved from computational linguistics,
which uses computer science to understand the principles of language, but␣
 ↪rather than developing theoretical frameworks,
NLP is an engineering discipline that seeks to build technology to accomplish␣
 ↪useful tasks.
NLP can be divided into two overlapping subfields: natural language␣
 ↪understanding (NLU),
which focuses on semantic analysis or determining the intended meaning of text,␣
 ↪and natural language generation (NLG),
which focuses on text generation by a machine. NLP is separate from - but often␣
 ↪used in conjunction with - speech recognition,
which seeks to parse spoken language into words, turning sound into text and␣
 ↪vice versa.
"""
```

[14]:
```
len(document)
```

[14]: 1611

Tokenization

breaking of texts into phrases ,words or other meaningful elements

[15]:
```python
# Tokenization in nltk
tokens=word_tokenize(document)
print(tokens)
```

```
['Introduction', 'Natural', 'Language', 'Processing', '(', 'NLP', ')', 'is',
'one', 'of', 'the', 'hottest', 'areas', 'of', 'artificial', 'intelligence', '(',
'AI', ')', 'thanks', 'to', 'applications', 'like', 'text', 'generators', 'that',
'compose', 'coherent', 'essays', ',', 'chatbots', 'that', 'fool', 'people',
'into', 'thinking', 'they', '’', 're', 'sentient', ',', 'and', 'text-to-image',
'programs', 'that', 'produce', 'photorealistic', 'images', 'of', 'anything',
'you', 'can', 'describe', '.', 'Recent', 'years', 'have', 'brought', 'a',
'revolution', 'in', 'the', 'ability', 'of', 'computers', 'to', 'understand',
'human', 'languages', ',', 'programming', 'languages', ',', 'and', 'even',
'biological', 'and', 'chemical', 'sequences', ',', 'such', 'as', 'DNA', 'and',
'protein', 'structures', ',', 'that', 'resemble', 'language', '.', 'The',
'latest', 'AI', 'models', 'are', 'unlocking', 'these', 'areas', 'to', 'analyze',
'the', 'meanings', 'of', 'input', 'text', 'and', 'generate', 'meaningful', ',',
'expressive', 'output', '.', 'What', 'is', 'Natural', 'Language', 'Processing',
'(', 'NLP', ')', 'Natural', 'language', 'processing', '(', 'NLP', ')', 'is',
'the', 'discipline', 'of', 'building', 'machines', 'that', 'can', 'manipulate',
'human', 'language', '-', 'or', 'data', 'that', 'resembles', 'human',
'language', '-', 'in', 'the', 'way', 'that', 'it', 'is', 'written', ',',
'spoken', ',', 'and', 'organized', '.', 'It', 'evolved', 'from',
'computational', 'linguistics', ',', 'which', 'uses', 'computer', 'science',
'to', 'understand', 'the', 'principles', 'of', 'language', ',', 'but', 'rather',
'than', 'developing', 'theoretical', 'frameworks', ',', 'NLP', 'is', 'an',
'engineering', 'discipline', 'that', 'seeks', 'to', 'build', 'technology', 'to',
'accomplish', 'useful', 'tasks', '.', 'NLP', 'can', 'be', 'divided', 'into',
'two', 'overlapping', 'subfields', ':', 'natural', 'language', 'understanding',
'(', 'NLU', ')', ',', 'which', 'focuses', 'on', 'semantic', 'analysis', 'or',
'determining', 'the', 'intended', 'meaning', 'of', 'text', ',', 'and',
'natural', 'language', 'generation', '(', 'NLG', ')', ',', 'which', 'focuses',
'on', 'text', 'generation', 'by', 'a', 'machine', '.', 'NLP', 'is', 'separate',
'from', '-', 'but', 'often', 'used', 'in', 'conjunction', 'with', '-', 'speech',
'recognition', ',', 'which', 'seeks', 'to', 'parse', 'spoken', 'language',
'into', 'words', ',', 'turning', 'sound', 'into', 'text', 'and', 'vice',
'versa', '.']
```

[16]:
```python
# Tokenization using spacy
tokens_spacy = [token.text for token in spacy.load("en_core_web_sm")(document)]
print(tokens_spacy)
```

```
['\n', 'Introduction', '\n', 'Natural', 'Language', 'Processing', '(', 'NLP',
')', 'is', 'one', 'of', 'the', 'hottest', 'areas', 'of', 'artificial', '\n',
'intelligence', '(', 'AI', ')', 'thanks', 'to', 'applications', 'like', 'text',
'generators', 'that', 'compose', 'coherent', 'essays', ',', '\n', 'chatbots',
'that', 'fool', 'people', 'into', 'thinking', 'they', '’re', 'sentient', ',',
'\n', 'and', 'text', '-', 'to', '-', 'image', 'programs', 'that', 'produce',
'photorealistic', 'images', 'of', 'anything', 'you', 'can', 'describe', '.',
'\n', 'Recent', 'years', 'have', 'brought', 'a', 'revolution', 'in', 'the',
'ability', 'of', 'computers', 'to', 'understand', 'human', 'languages', ',',
'\n', 'programming', 'languages', ',', 'and', 'even', 'biological', 'and',
'chemical', 'sequences', ',', 'such', 'as', 'DNA', 'and', 'protein',
'structures', ',', '\n', 'that', 'resemble', 'language', '.', '\n', 'The',
'latest', 'AI', 'models', 'are', 'unlocking', 'these', 'areas', 'to', 'analyze',
'the', 'meanings', 'of', 'input', 'text', 'and', 'generate', 'meaningful', ',',
'\n', 'expressive', 'output', '.', '\n', 'What', 'is', 'Natural', 'Language',
'Processing', '(', 'NLP)Natural', 'language', 'processing', '(', 'NLP', ')',
'\n', 'is', 'the', 'discipline', 'of', 'building', 'machines', 'that', 'can',
'manipulate', 'human', 'language', '-', 'or', 'data', 'that', 'resembles',
'human', 'language', '-', 'in', 'the', 'way', 'that', 'it', 'is', 'written',
',', '\n', 'spoken', ',', 'and', 'organized', '.', 'It', 'evolved', 'from',
'computational', 'linguistics', ',', '\n', 'which', 'uses', 'computer',
'science', 'to', 'understand', 'the', 'principles', 'of', 'language', ',',
'but', 'rather', 'than', 'developing', 'theoretical', 'frameworks', ',', '\n',
'NLP', 'is', 'an', 'engineering', 'discipline', 'that', 'seeks', 'to', 'build',
'technology', 'to', 'accomplish', 'useful', 'tasks', '.', '\n', 'NLP', 'can',
'be', 'divided', 'into', 'two', 'overlapping', 'subfields', ':', 'natural',
'language', 'understanding', '(', 'NLU', ')', ',', '\n', 'which', 'focuses',
'on', 'semantic', 'analysis', 'or', 'determining', 'the', 'intended', 'meaning',
'of', 'text', ',', 'and', 'natural', 'language', 'generation', '(', 'NLG', ')',
',', '\n', 'which', 'focuses', 'on', 'text', 'generation', 'by', 'a', 'machine',
'.', 'NLP', 'is', 'separate', 'from', '-', 'but', 'often', 'used', 'in',
'conjunction', 'with', '-', 'speech', 'recognition', ',', '\n', 'which',
'seeks', 'to', 'parse', 'spoken', 'language', 'into', 'words', ',', 'turning',
'sound', 'into', 'text', 'and', 'vice', 'versa', '.', '\n']
```

[17]:
```python
#Tokenization using Gensim
tokens_gensim = list(gensim.utils.tokenize(document))
print(tokens_gensim)
```

```
['Introduction', 'Natural', 'Language', 'Processing', 'NLP', 'is', 'one', 'of',
'the', 'hottest', 'areas', 'of', 'artificial', 'intelligence', 'AI', 'thanks',
'to', 'applications', 'like', 'text', 'generators', 'that', 'compose',
'coherent', 'essays', 'chatbots', 'that', 'fool', 'people', 'into', 'thinking',
'they', 're', 'sentient', 'and', 'text', 'to', 'image', 'programs', 'that',
'produce', 'photorealistic', 'images', 'of', 'anything', 'you', 'can',
'describe', 'Recent', 'years', 'have', 'brought', 'a', 'revolution', 'in',
'the', 'ability', 'of', 'computers', 'to', 'understand', 'human', 'languages',
'programming', 'languages', 'and', 'even', 'biological', 'and', 'chemical',
'sequences', 'such', 'as', 'DNA', 'and', 'protein', 'structures', 'that',
'resemble', 'language', 'The', 'latest', 'AI', 'models', 'are', 'unlocking',
'these', 'areas', 'to', 'analyze', 'the', 'meanings', 'of', 'input', 'text',
'and', 'generate', 'meaningful', 'expressive', 'output', 'What', 'is',
'Natural', 'Language', 'Processing', 'NLP', 'Natural', 'language', 'processing',
'NLP', 'is', 'the', 'discipline', 'of', 'building', 'machines', 'that', 'can',
'manipulate', 'human', 'language', 'or', 'data', 'that', 'resembles', 'human',
'language', 'in', 'the', 'way', 'that', 'it', 'is', 'written', 'spoken', 'and',
'organized', 'It', 'evolved', 'from', 'computational', 'linguistics', 'which',
'uses', 'computer', 'science', 'to', 'understand', 'the', 'principles', 'of',
'language', 'but', 'rather', 'than', 'developing', 'theoretical', 'frameworks',
'NLP', 'is', 'an', 'engineering', 'discipline', 'that', 'seeks', 'to', 'build',
'technology', 'to', 'accomplish', 'useful', 'tasks', 'NLP', 'can', 'be',
'divided', 'into', 'two', 'overlapping', 'subfields', 'natural', 'language',
'understanding', 'NLU', 'which', 'focuses', 'on', 'semantic', 'analysis', 'or',
'determining', 'the', 'intended', 'meaning', 'of', 'text', 'and', 'natural',
'language', 'generation', 'NLG', 'which', 'focuses', 'on', 'text', 'generation',
'by', 'a', 'machine', 'NLP', 'is', 'separate', 'from', 'but', 'often', 'used',
'in', 'conjunction', 'with', 'speech', 'recognition', 'which', 'seeks', 'to',
'parse', 'spoken', 'language', 'into', 'words', 'turning', 'sound', 'into',
'text', 'and', 'vice', 'versa']
```

Stemming

process of reducing words into their roots/base

[18]:
```python
# Stemming using NLTK
from nltk.stem import PorterStemmer
ps=PorterStemmer()
stemmed_tokens = [ps.stem(token) for token in tokens]
# Join stemmed words back into a document
stemmed_document = ' '.join(stemmed_tokens)
print(stemmed_document)
```

```
introduct natur languag process ( nlp ) is one of the hottest area of artifici
intellig ( ai ) thank to applic like text gener that compos coher essay ,
chatbot that fool peopl into think they ’ re sentient , and text-to-imag program
that produc photorealist imag of anyth you can describ . recent year have
brought a revolut in the abil of comput to understand human languag , program
languag , and even biolog and chemic sequenc , such as dna and protein structur
, that resembl languag . the latest ai model are unlock these area to analyz the
mean of input text and gener meaning , express output . what is natur languag
process ( nlp ) natur languag process ( nlp ) is the disciplin of build machin
that can manipul human languag - or data that resembl human languag - in the way
that it is written , spoken , and organ . it evolv from comput linguist , which
use comput scienc to understand the principl of languag , but rather than
develop theoret framework , nlp is an engin disciplin that seek to build
technolog to accomplish use task . nlp can be divid into two overlap subfield :
natur languag understand ( nlu ) , which focus on semant analysi or determin the
intend mean of text , and natur languag gener ( nlg ) , which focus on text
gener by a machin . nlp is separ from - but often use in conjunct with - speech
recognit , which seek to pars spoken languag into word , turn sound into text
and vice versa .
```

Lemmatization

transforming words to ther base form or dictionry form

[19]:
```
#lemmatization using nltk
from nltk.stem import WordNetLemmatizer
lemmatizer=WordNetLemmatizer()
tokenized_document=[lemmatizer.lemmatize(token) for token in tokens]
lemmatized_words=','.join(tokenized_document)
print(lemmatized_words)
```

```
Introduction,Natural,Language,Processing,(,NLP,),is,one,of,the,hottest,area,of,a
rtificial,intelligence,(,AI,),thanks,to,application,like,text,generator,that,com
pose,coherent,essay,,,chatbots,that,fool,people,into,thinking,they,’,re,sentient
,,,and,text-to-image,program,that,produce,photorealistic,image,of,anything,you,c
an,describe,.,Recent,year,have,brought,a,revolution,in,the,ability,of,computer,t
o,understand,human,language,,,programming,language,,,and,even,biological,and,che
mical,sequence,,,such,a,DNA,and,protein,structure,,,that,resemble,language,.,The
,latest,AI,model,are,unlocking,these,area,to,analyze,the,meaning,of,input,text,a
nd,generate,meaningful,,,expressive,output,.,What,is,Natural,Language,Processing
,(,NLP,),Natural,language,processing,(,NLP,),is,the,discipline,of,building,machi
ne,that,can,manipulate,human,language,–,or,data,that,resembles,human,language,–,
in,the,way,that,it,is,written,,,spoken,,,and,organized,.,It,evolved,from,computa
tional,linguistics,,,which,us,computer,science,to,understand,the,principle,of,la
nguage,,,but,rather,than,developing,theoretical,framework,,,NLP,is,an,engineerin
g,discipline,that,seek,to,build,technology,to,accomplish,useful,task,.,NLP,can,b
e,divided,into,two,overlapping,subfields,:,natural,language,understanding,(,NLU,
),,,which,focus,on,semantic,analysis,or,determining,the,intended,meaning,of,text
,,,and,natural,language,generation,(,NLG,),,,which,focus,on,text,generation,by,a
,machine,.,NLP,is,separate,from,–,but,often,used,in,conjunction,with,–,speech,re
cognition,,,which,seek,to,parse,spoken,language,into,word,,,turning,sound,into,t
ext,and,vice,versa,.
```

Stopwords

these are common words like the,at during text processing they don’t typically carry much meaning

[21]:
```
# Remove stop words
stop_words = set(stopwords.words("english"))
filtered_words = [word for word in tokens if word.lower() not in stop_words]
print(filtered_words)
```

```
['Introduction', 'Natural', 'Language', 'Processing', '(', 'NLP', ')', 'one',
'hottest', 'areas', 'artificial', 'intelligence', '(', 'AI', ')', 'thanks',
'applications', 'like', 'text', 'generators', 'compose', 'coherent', 'essays',
',', 'chatbots', 'fool', 'people', 'thinking', '’', 'sentient', ',', 'text-to-
image', 'programs', 'produce', 'photorealistic', 'images', 'anything',
'describe', '.', 'Recent', 'years', 'brought', 'revolution', 'ability',
'computers', 'understand', 'human', 'languages', ',', 'programming',
'languages', ',', 'even', 'biological', 'chemical', 'sequences', ',', 'DNA',
'protein', 'structures', ',', 'resemble', 'language', '.', 'latest', 'AI',
'models', 'unlocking', 'areas', 'analyze', 'meanings', 'input', 'text',
'generate', 'meaningful', ',', 'expressive', 'output', '.', 'Natural',
'Language', 'Processing', '(', 'NLP', ')', 'Natural', 'language', 'processing',
'(', 'NLP', ')', 'discipline', 'building', 'machines', 'manipulate', 'human',
'language', '-', 'data', 'resembles', 'human', 'language', '-', 'way',
'written', ',', 'spoken', ',', 'organized', '.', 'evolved', 'computational',
'linguistics', ',', 'uses', 'computer', 'science', 'understand', 'principles',
'language', ',', 'rather', 'developing', 'theoretical', 'frameworks', ',',
'NLP', 'engineering', 'discipline', 'seeks', 'build', 'technology',
'accomplish', 'useful', 'tasks', '.', 'NLP', 'divided', 'two', 'overlapping',
'subfields', ':', 'natural', 'language', 'understanding', '(', 'NLU', ')', ',',
'focuses', 'semantic', 'analysis', 'determining', 'intended', 'meaning', 'text',
',', 'natural', 'language', 'generation', '(', 'NLG', ')', ',', 'focuses',
'text', 'generation', 'machine', '.', 'NLP', 'separate', '-', 'often', 'used',
'conjunction', '-', 'speech', 'recognition', ',', 'seeks', 'parse', 'spoken',
'language', 'words', ',', 'turning', 'sound', 'text', 'vice', 'versa', '.']
```

Role and comparison

Gensim

Is primarily focused on topic modeling, similarity retrieval, and other natural language processing tasks. It provides efficient algorithms for tokenization and text processing.

spaCy

Is known for its high performance and provides tokenization, lemmatization, and stop word removal along with advanced linguistic features.

NLTK

Is a comprehensive library for natural language processing tasks and provides implementations for tokenization, stemming, lemmatization, and stop word removal.

[22]:
```python
# Bag of words analysis using CountVectorizer
vectorizer = CountVectorizer()
X = vectorizer.fit_transform([document])
features = vectorizer.get_feature_names_out()
```

[23]:
```python
# Convert to Pandas DataFrame for presentation
df = pd.DataFrame(X.toarray(), columns=features)
df
```

[23]:
```
   ability  accomplish  ai  an  analysis  analyze  and  anything  \
0        1           1   2   1         1        1    8         1

   applications  are  …  versa  vice  way  what  which  with  words  \
0             1    1  …      1     1    1     1      4     1      1

   written  years  you
0        1      1    1
```

```
[1 rows x 143 columns]
```

```
[ ]:
```