Devolved AI

Subject : Basic Learning of NLP and LLMs including various technique and models.

Date of Submission : 13th March, 2024

Submitted To,

Nathan Peterson,
Founder and CEO,
Devolved AI

Md. Nazmul Hossain,
Co-Founder & COO,
Devolved AI

Submitted By,

Md Al Amin Tokder

Machine Learning Engineer ,
Devolved AI

Introduction to NLP & Text Processing Techniques

Natural Language Processing (NLP) is a field at the intersection of computer science, artificial intelligence, and linguistics. It focuses on the interaction between computers and human language, with the goal of enabling computers to understand, interpret, and generate human languages.

Real-life example: Consider a simple email filtering system. It uses basic NLP techniques to classify emails into 'spam' or 'non-spam'. The system processes the text of the emails, identifies certain keywords (like 'sale', 'free', 'offer') that are frequently found in spam, and uses this information to filter emails.

Key Techniques:

Tokenization: Breaking down text into words or sentences.

Stop Words Removal: Removing common words that may not contribute to the meaning of the text.

Stemming and Lemmatization: Reducing words to their base or root form.

Part-of-Speech Tagging: Identifying parts of speech (nouns, verbs, adjectives, etc.) in a sentence.

Named Entity Recognition (NER): Identifying important named entities in the text (like people, places, organizations).

```
!pip install nltk==3.6.5
import nltk
```

```
Requirement already satisfied: nltk==3.6.5 in
/opt/conda/lib/python3.10/site-packages (3.6.5)
Requirement already satisfied: click in
/opt/conda/lib/python3.10/site-packages (from nltk==3.6.5) (8.1.7)
Requirement already satisfied: joblib in
/opt/conda/lib/python3.10/site-packages (from nltk==3.6.5) (1.3.2)
Requirement already satisfied: regex>=2021.8.3 in
/opt/conda/lib/python3.10/site-packages (from nltk==3.6.5)
(2023.12.25)
Requirement already satisfied: tqdm in /opt/conda/lib/python3.10/site-
packages (from nltk==3.6.5) (4.66.1)
```

Word Tokenization

Tokenization is the process of breaking down text into smaller units called tokens. Tokens can be words, characters, or subwords. In NLP, tokenization is often the first step in text preprocessing, preparing the text for more complex operations like parsing or entity recognition.

- **Word Tokenization:** Splits the text into words. It's useful for tasks where the semantics of individual words are important, such as in sentiment analysis or topic modeling.
- **Character Tokenization:** Splits the text into characters. This is often used in character-level models or for certain languages where the concept of a word is not straightforward.
- **Subword Tokenization:** Breaks text into subword units. This can help in handling rare words or morphologically rich languages. Models like BERT use subword tokenization.

```
#Word Tokenization Example
import nltk
from nltk.tokenize import word_tokenize

nltk.download('punkt')

text = "Natural language processing is an exciting field."
tokens = word_tokenize(text)
print(tokens)
```

```
[nltk_data] Downloading package punkt to /usr/share/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
['Natural', 'language', 'processing', 'is', 'an', 'exciting', 'field',
'.']
```

```
p="""I have three visions for India. In 3000 years of our history,
people from all over
               the world have come and invaded us, captured our lands,
conquered our minds.
               From Alexander onwards, the Greeks, the Turks, the
Moguls, the Portuguese, the British,
               the French, the Dutch, all of them came and looted us,
took over what was ours.
               Yet we have not done this to any other nation. We have
not conquered anyone.
               We have not grabbed their land, their culture,
               their history and tried to enforce our way of life on
them.
               Why? Because we respect the freedom of others.That is
why my
               first vision is that of freedom. I believe that India
got its first vision of
               this in 1857, when we started the War of Independence.
It is this freedom that
               we must protect and nurture and build on. If we are not
free, no one will respect us.
               My second vision for India’s development. For fifty
```

```
years we have been a developing nation.
               It is time we see ourselves as a developed nation. We
are among the top 5 nations of the world
               in terms of GDP. We have a 10 percent growth rate in
most areas. Our poverty levels are falling.
               Our achievements are being globally recognised today.
Yet we lack the self-confidence to
               see ourselves as a developed nation, self-reliant and
self-assured. Isn’t this incorrect?
               I have a third vision. India must stand up to the
world. Because I believe that unless India
               stands up to the world, no one will respect us. Only
strength respects strength. We must be
               strong not only as a military power but also as an
economic power. Both must go hand-in-hand.
               My good fortune was to have worked with three great
minds. Dr. Vikram Sarabhai of the Dept. of
               space, Professor Satish Dhawan, who succeeded him and
Dr. Brahm Prakash, father of nuclear material.
               I was lucky to have worked with all three of them
closely and consider this the great opportunity of my life.
               I see four milestones in my career"""

# Tokenizing sentences
sentence=nltk.sent_tokenize(p)
print(sentence)

# Tokenizing word
words=nltk.word_tokenize(p)
# print(words)
words
```

```
['I have three visions for India.', 'In 3000 years of our history,
people from all over \n               the world have come and invaded
us, captured our lands, conquered our minds.', 'From Alexander
onwards, the Greeks, the Turks, the Moguls, the Portuguese, the
British,\n               the French, the Dutch, all of them came and
looted us, took over what was ours.', 'Yet we have not done this to
any other nation.', 'We have not conquered anyone.', 'We have not
grabbed their land, their culture, \n               their history and
tried to enforce our way of life on them.', 'Why?', 'Because we
respect the freedom of others.That is why my \n               first
vision is that of freedom.', 'I believe that India got its first
vision of \n               this in 1857, when we started the War of
Independence.', 'It is this freedom that\n               we must
protect and nurture and build on.', 'If we are not free, no one will
respect us.', 'My second vision for India’s development.', 'For fifty
years we have been a developing nation.', 'It is time we see ourselves
as a developed nation.', 'We are among the top 5 nations of the world\
n               in terms of GDP.', 'We have a 10 percent growth rate
```

```
in most areas.', 'Our poverty levels are falling.', 'Our achievements
are being globally recognised today.', 'Yet we lack the self-
confidence to\n               see ourselves as a developed nation,
self-reliant and self-assured.', 'Isn’t this incorrect?', 'I have a
third vision.', 'India must stand up to the world.', 'Because I
believe that unless India \n               stands up to the world, no
one will respect us.', 'Only strength respects strength.', 'We must be
\n                strong not only as a military power but also as an
economic power.', 'Both must go hand-in-hand.', 'My good fortune was
to have worked with three great minds.', 'Dr. Vikram Sarabhai of the
Dept.', 'of \n               space, Professor Satish Dhawan, who
succeeded him and Dr. Brahm Prakash, father of nuclear material.', 'I
was lucky to have worked with all three of them closely and consider
this the great opportunity of my life.', 'I see four milestones in my
career']

['I',
 'have',
 'three',
 'visions',
 'for',
 'India',
 '.',
 'In',
 '3000',
 'years',
 'of',
 'our',
 'history',
 ',',
 'people',
 'from',
 'all',
 'over',
 'the',
 'world',
 'have',
 'come',
 'and',
 'invaded',
 'us',
 ',',
 'captured',
 'our',
 'lands',
 ',',
 'conquered',
 'our',
 'minds',
 '.',
 'From',
 'Alexander',
 'onwards',
 ',',
 'the',
 'Greeks',
 ',',
 'the',
 'Turks',
 ',',
 'the',
 'Moguls',
 ',',
 'the',
 'Portuguese',
 ',',
 'the',
 'British',
 ',',
 'the',
 'French',
 ',',
 'the',
 'Dutch',
 ',',
 'all',
 'of',
 'them',
 'came',
 'and',
 'looted',
 'us',
 ',',
 'took',
 'over',
 'what',
 'was',
 'ours',
 '.',
 'Yet',
 'we',
 'have',
 'not',
 'done',
 'this',
 'to',
 'any',
 'other',
 'nation',
 '.',
 'We',
 'have',
 'not',
 'conquered',
 'anyone',
 '.',
 'We',
 'have',
 'not',
 'grabbed',
 'their',
 'land',
 ',',
 'their',
 'culture',
 ',',
 'their',
 'history',
 'and',
 'tried',
 'to',
 'enforce',
 'our',
 'way',
 'of',
 'life',
 'on',
 'them',
 '.',
 'Why',
 '?',
 'Because',
 'we',
 'respect',
 'the',
 'freedom',
 'of',
 'others.That',
 'is',
 'why',
 'my',
 'first',
 'vision',
 'is',
 'that',
 'of',
 'freedom',
 '.',
 'I',
 'believe',
 'that',
 'India',
 'got',
 'its',
 'first',
 'vision',
 'of',
 'this',
 'in',
 '1857',
 ',',
 'when',
 'we',
 'started',
 'the',
 'War',
 'of',
 'Independence',
 '.',
 'It',
 'is',
 'this',
 'freedom',
 'that',
 'we',
 'must',
 'protect',
 'and',
 'nurture',
 'and',
 'build',
 'on',
 '.',
 'If',
 'we',
 'are',
 'not',
 'free',
 ',',
 'no',
 'one',
 'will',
 'respect',
 'us',
 '.',
 'My',
 'second',
 'vision',
 'for',
 'India',
 '’',
 's',
 'development',
 '.',
 'For',
 'fifty',
 'years',
 'we',
 'have',
 'been',
 'a',
 'developing',
 'nation',
 '.',
 'It',
 'is',
 'time',
 'we',
 'see',
 'ourselves',
 'as',
 'a',
 'developed',
 'nation',
 '.',
 'We',
 'are',
 'among',
 'the',
 'top',
 '5',
 'nations',
 'of',
 'the',
 'world',
 'in',
 'terms',
 'of',
 'GDP',
 '.',
 'We',
 'have',
 'a',
 '10',
 'percent',
 'growth',
 'rate',
 'in',
 'most',
 'areas',
 '.',
 'Our',
 'poverty',
 'levels',
 'are',
 'falling',
 '.',
 'Our',
 'achievements',
 'are',
 'being',
 'globally',
 'recognised',
 'today',
 '.',
 'Yet',
 'we',
 'lack',
 'the',
 'self-confidence',
 'to',
 'see',
 'ourselves',
 'as',
 'a',
 'developed',
 'nation',
 ',',
 'self-reliant',
 'and',
 'self-assured',
 '.',
 'Isn',
 '’',
 't',
 'this',
 'incorrect',
 '?',
 'I',
 'have',
 'a',
 'third',
 'vision',
 '.',
 'India',
 'must',
 'stand',
 'up',
 'to',
 'the',
 'world',
 '.',
 'Because',
 'I',
 'believe',
 'that',
 'unless',
 'India',
 'stands',
 'up',
 'to',
 'the',
 'world',
 ',',
 'no',
 'one',
 'will',
 'respect',
 'us',
 '.',
 'Only',
 'strength',
 'respects',
 'strength',
 '.',
 'We',
 'must',
 'be',
 'strong',
 'not',
 'only',
 'as',
 'a',
 'military',
 'power',
 'but',
 'also',
 'as',
 'an',
 'economic',
 'power',
 '.',
 'Both',
 'must',
 'go',
 'hand-in-hand',
 '.',
 'My',
 'good',
 'fortune',
 'was',
 'to',
 'have',
 'worked',
 'with',
 'three',
 'great',
 'minds',
 '.',
 'Dr.',
 'Vikram',
 'Sarabhai',
 'of',
 'the',
 'Dept',
 '.',
 'of',
 'space',
 ',',
 'Professor',
 'Satish',
 'Dhawan',
 ',',
 'who',
 'succeeded',
 'him',
 'and',
 'Dr.',
 'Brahm',
 'Prakash',
 ',',
 'father',
 'of',
 'nuclear',
 'material',
 '.',
 'I',
 'was',
 'lucky',
 'to',
 'have',
 'worked',
 'with',
 'all',
 'three',
 'of',
 'them',
 'closely',
 'and',
 'consider',
 'this',
 'the',
 'great',
 'opportunity',
 'of',
 'my',
 'life',
 '.',
 'I',
 'see',
 'four',
 'milestones',
 'in',
 'my',
 'career']
```

Tokenization:

Splitting text into sentences or words.

Stemming and Lemmatization:

Reducing words to their base or root form.

Stop-word Removal:

Eliminating common words that add little value in text analysis.

Regular Expressions:

For text pattern recognition and manipulation.

Stemming and Lemmatization

Description: Both stemming and lemmatization are used to reduce words to their base form, but they differ in their approach and complexity.

Stemming: (jesob word decision make korte (for example: of ,this,my , me etc...) temon kono effect fele na segula remove kore fela uttom)

Example:"Histri"

A simpler method that chops off the ends of words, often leading to incorrect generalizations and suboptimal results. For example, "fishing", "fished", and "fisher" might all be reduced to "fish". It's faster but less accurate.

Lemmatization:

Example: "History"

More sophisticated, it considers the morphological analysis of words. Lemmas are the root forms of words. For example, "am", "are", and "is" are all lemmatized into "be". It uses vocabulary and grammatical analysis, leading to better results but at a higher computational cost.

```
#Steaming : jesob word decision make korte (for example: of ,this,my ,
me etc...) temon kono effect fele na segula remove kore fela uttom
#Decision make mane (classification(YES/NO) jemon spam detection)


#Steaming


import nltk
from nltk.stem import PorterStemmer
from nltk.corpus import stopwords

paragraph = """I have three visions for India. In 3000 years of our
history, people from all over
               the world have come and invaded us, captured our lands,
conquered our minds.
               From Alexander onwards, the Greeks, the Turks, the
Moguls, the Portuguese, the British,
               the French, the Dutch, all of them came and looted us,
took over what was ours.
               Yet we have not done this to any other nation. We have
not conquered anyone.
               We have not grabbed their land, their culture,
               their history and tried to enforce our way of life on
them.
               Why? Because we respect the freedom of others.That is
why my
               first vision is that of freedom. I believe that India
```

```
got its first vision of
               this in 1857, when we started the War of Independence.
It is this freedom that
               we must protect and nurture and build on. If we are not
free, no one will respect us.
               My second vision for India’s development. For fifty
years we have been a developing nation.
               It is time we see ourselves as a developed nation. We
are among the top 5 nations of the world
               in terms of GDP. We have a 10 percent growth rate in
most areas. Our poverty levels are falling.
               Our achievements are being globally recognised today.
Yet we lack the self-confidence to
               see ourselves as a developed nation, self-reliant and
self-assured. Isn’t this incorrect?
               I have a third vision. India must stand up to the
world. Because I believe that unless India
               stands up to the world, no one will respect us. Only
strength respects strength. We must be
               strong not only as a military power but also as an
economic power. Both must go hand-in-hand.
               My good fortune was to have worked with three great
minds. Dr. Vikram Sarabhai of the Dept. of
               space, Professor Satish Dhawan, who succeeded him and
Dr. Brahm Prakash, father of nuclear material.
               I was lucky to have worked with all three of them
closely and consider this the great opportunity of my life.
               I see four milestones in my career."""


sentences = nltk.sent_tokenize(paragraph)
stemmer = PorterStemmer()

# Stemming
for i in range(len(sentences)):
    words = nltk.word_tokenize(sentences[i])
    words = [stemmer.stem(word) for word in words if word not in
set(stopwords.words('english'))]
    sentences[i] = ' '.join(words)


sentences
```

```
['I three vision india .',
 'In 3000 year histori , peopl world come invad us , captur land ,
conquer mind .',
 'from alexand onward , greek , turk , mogul , portugues , british ,
french , dutch , came loot us , took .',
 'yet done nation .',
 'We conquer anyon .',
 'We grab land , cultur , histori tri enforc way life .',
 'whi ?',
 'becaus respect freedom others.that first vision freedom .',
 'I believ india got first vision 1857 , start war independ .',
 'It freedom must protect nurtur build .',
 'If free , one respect us .',
 'My second vision india ’ develop .',
 'for fifti year develop nation .',
 'It time see develop nation .',
 'We among top 5 nation world term gdp .',
 'We 10 percent growth rate area .',
 'our poverti level fall .',
 'our achiev global recognis today .',
 'yet lack self-confid see develop nation , self-reli self-assur .',
 'isn ’ incorrect ?',
 'I third vision .',
 'india must stand world .',
 'becaus I believ unless india stand world , one respect us .',
 'onli strength respect strength .',
 'We must strong militari power also econom power .',
 'both must go hand-in-hand .',
 'My good fortun work three great mind .',
 'dr. vikram sarabhai dept .',
 'space , professor satish dhawan , succeed dr. brahm prakash , father
nuclear materi .',
 'I lucki work three close consid great opportun life .',
 'I see four mileston career .']
```

```
# kon kon word useless for classification model segula stopwords.words
ei library call dilei peye jabo jemon:

import nltk
from nltk.corpus import stopwords
stopwords.words('english')
```

```
['i',
 'me',
 'my',
 'myself',
 'we',
 'our',
 'ours',
 'ourselves',
 'you',
 "you're",
 "you've",
 "you'll",
 "you'd",
 'your',
 'yours',
 'yourself',
 'yourselves',
 'he',
 'him',
 'his',
 'himself',
 'she',
 "she's",
 'her',
 'hers',
 'herself',
 'it',
 "it's",
 'its',
 'itself',
 'they',
 'them',
 'their',
 'theirs',
 'themselves',
 'what',
 'which',
 'who',
 'whom',
 'this',
 'that',
 "that'll",
 'these',
 'those',
 'am',
 'is',
 'are',
 'was',
 'were',
 'be',
 'been',
 'being',
 'have',
 'has',
 'had',
 'having',
 'do',
 'does',
 'did',
 'doing',
 'a',
 'an',
 'the',
 'and',
 'but',
 'if',
 'or',
 'because',
 'as',
 'until',
 'while',
 'of',
 'at',
 'by',
 'for',
 'with',
 'about',
 'against',
 'between',
 'into',
 'through',
 'during',
 'before',
 'after',
 'above',
 'below',
 'to',
 'from',
 'up',
 'down',
 'in',
 'out',
 'on',
 'off',
 'over',
 'under',
 'again',
 'further',
 'then',
 'once',
 'here',
 'there',
 'when',
 'where',
 'why',
 'how',
 'all',
 'any',
 'both',
 'each',
 'few',
 'more',
 'most',
 'other',
 'some',
 'such',
 'no',
 'nor',
 'not',
 'only',
 'own',
 'same',
 'so',
 'than',
 'too',
 'very',
 's',
 't',
 'can',
 'will',
 'just',
 'don',
 "don't",
 'should',
 "should've",
 'now',
 'd',
 'll',
 'm',
 'o',
 're',
 've',
 'y',
 'ain',
 'aren',
 "aren't",
 'couldn',
 "couldn't",
 'didn',
 "didn't",
 'doesn',
 "doesn't",
 'hadn',
 "hadn't",
 'hasn',
 "hasn't",
 'haven',
 "haven't",
 'isn',
 "isn't",
 'ma',
 'mightn',
 "mightn't",
 'mustn',
 "mustn't",
 'needn',
 "needn't",
 'shan',
 "shan't",
 'shouldn',
 "shouldn't",
 'wasn',
 "wasn't",
 'weren',
 "weren't",
 'won',
 "won't",
 'wouldn',
 "wouldn't"]
```

```
#Steaming practice myself:

import nltk
from nltk.stem import PorterStemmer
from nltk.corpus import stopwords

p="Hello I am Al Amin. My favorite hobby is to researching in new
technology and invent something new."

sentences=nltk.sent_tokenize(p)


sentences = nltk.sent_tokenize(p)
stemmer = PorterStemmer()

# Stemming
for i in range(len(sentences)):
    words = nltk.word_tokenize(sentences[i])
    words = [stemmer.stem(word) for word in words if word not in
set(stopwords.words('english'))]
    sentences[i] = ' '.join(words)

sentences
```

```
['hello I Al amin .',
 'My favorit hobbi research new technolog invent someth new .']
```

```
import nltk
import subprocess
```

```
# Download and unzip wordnet
try:
    nltk.data.find('wordnet.zip')
except:
    nltk.download('wordnet', download_dir='/kaggle/working/')
    command = "unzip /kaggle/working/corpora/wordnet.zip -d
/kaggle/working/corpora"
    subprocess.run(command.split())
    nltk.data.path.append('/kaggle/working/')

# Now you can import the NLTK resources as usual
from nltk.corpus import wordnet
```

```
[nltk_data] Downloading package wordnet to /kaggle/working/...
[nltk_data]   Package wordnet is already up-to-date!
Archive:  /kaggle/working/corpora/wordnet.zip
```

```
replace /kaggle/working/corpora/wordnet/lexnames? [y]es, [n]o, [A]ll,
[N]one, [r]ename:  NULL
(EOF or read error, treating as "[N]one" ...)
```

```
#lemmatization myself
import nltk
nltk.download('/root/nltk_data')
nltk.download('stopwords')

from nltk.stem import WordNetLemmatizer
from nltk.corpus import stopwords
paragraph = """Thank you all so very much. Thank you to the Academy.
               Thank you to all of you in this room. I have to
congratulate
               the other incredible nominees this year. The Revenant
was
               the product of the tireless efforts of an unbelievable
cast
               and crew. First off, to my brother in this endeavor,
Mr. Tom
               Hardy. Tom, your talent on screen can only be surpassed
by
               your friendship off screen … thank you for creating a t
               ranscendent cinematic experience. Thank you to
everybody at
               Fox and New Regency … my entire team. I have to thank
               everyone from the very onset of my career … To my
parents;
               none of this would be possible without you. And to my
               friends, I love you dearly; you know who you are. And
lastly,
               I just want to say this: Making The Revenant was about
               man's relationship to the natural world. A world that
we
               collectively felt in 2015 as the hottest year in
recorded
               history. Our production needed to move to the southern
               tip of this planet just to be able to find snow.
Climate
               change is real, it is happening right now. It is the
most
               urgent threat facing our entire species, and we need to
work
               collectively together and stop procrastinating. We need
to
               support leaders around the world who do not speak for
the
               big polluters, but who speak for all of humanity, for
the
               indigenous people of the world, for the billions and
               billions of underprivileged people out there who would
be
               most affected by this. For our children’s children, and

               for those people out there whose voices have been
drowned
               out by the politics of greed. I thank you all for this
               amazing award tonight. Let us not take this planet for
               granted. I do not take tonight for granted. Thank you
so very much."""


sentences = nltk.sent_tokenize(paragraph)
lemmatizer = WordNetLemmatizer()
# Lemmatization
for i in range(len(sentences)):
    words = nltk.word_tokenize(sentences[i])
    words = [lemmatizer.lemmatize(word) for word in words if word not
in set(stopwords.words('english'))]
    sentences[i] = ' '.join(words)
```

```
[nltk_data] Error loading /root/nltk_data: Package '/root/nltk_data'
[nltk_data]     not found in index
[nltk_data] Downloading package stopwords to /usr/share/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Bag of Words (BoW)

Explanation The Bag of Words model represents text data as a 'bag' (multiset) of words, disregarding grammar and word order but keeping multiplicity. In this model, a text (such as a sentence or a document) is represented as the bag (multiset) of its words.

Mathematical Representation:

- Let's consider a vocabulary of D words from a corpus.
- Each document is represented as a vector in a D-dimensional space.
- Each dimension corresponds to a unique word in the vocabulary.
- If a word occurs in the document, its value in the vector is the count of times it appears; if not, the value is zero.

Example: Suppose we have two documents:

- Doc1: "blue house"
- Doc2: "red house" Vocabulary: ["blue", "house", "red"]

The BoW vectors would be:

- Doc1: [1, 1, 0] (1 "blue", 1 "house", 0 "red")
- Doc2: [0, 1, 1] (0 "blue", 1 "house", 1 "red")

```
#Bag of Words model

import nltk

paragraph =  """I have three visions for India. In 3000 years of our history, people from all over
               the world have come and invaded us, captured our lands, conquered our minds.
               From Alexander onwards, the Greeks, the Turks, the Moguls, the Portuguese, the British,
               the French, the Dutch, all of them came and looted us, took over what was ours.
               Yet we have not done this to any other nation. We have not conquered anyone.
               We have not grabbed their land, their culture,
               their history and tried to enforce our way of life on them.
               Why? Because we respect the freedom of others.That is why my
               first vision is that of freedom. I believe that India got its first vision of
               this in 1857, when we started the War of Independence. It is this freedom that
               we must protect and nurture and build on. If we are not free, no one will respect us.
               My second vision for India’s development. For fifty
```

```
years we have been a developing nation.
               It is time we see ourselves as a developed nation. We
are among the top 5 nations of the world
               in terms of GDP. We have a 10 percent growth rate in
most areas. Our poverty levels are falling.
               Our achievements are being globally recognised today.
Yet we lack the self-confidence to
               see ourselves as a developed nation, self-reliant and
self-assured. Isn’t this incorrect?
               I have a third vision. India must stand up to the
world. Because I believe that unless India
               stands up to the world, no one will respect us. Only
strength respects strength. We must be
               strong not only as a military power but also as an
economic power. Both must go hand-in-hand.
               My good fortune was to have worked with three great
minds. Dr. Vikram Sarabhai of the Dept. of
               space, Professor Satish Dhawan, who succeeded him and
Dr. Brahm Prakash, father of nuclear material.
               I was lucky to have worked with all three of them
closely and consider this the great opportunity of my life.
               I see four milestones in my career"""


# Cleaning the texts
import re
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from nltk.stem import WordNetLemmatizer

ps = PorterStemmer()
wordnet=WordNetLemmatizer()
sentences = nltk.sent_tokenize(paragraph)
corpus = []
for i in range(len(sentences)):
    review = re.sub('[^a-zA-Z]', ' ', sentences[i])
    review = review.lower()
    review = review.split()
    review = [ps.stem(word) for word in review if not word in
set(stopwords.words('english'))]
    review = ' '.join(review)
    corpus.append(review)

# Creating the Bag of Words model
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features = 1500)
X = cv.fit_transform(corpus).toarray()
X
# print(X)
```

```
array([[0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 1, 1, 0],
       [0, 1, 0, ..., 0, 0, 0],
       ...,
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0]])
```

```python
import matplotlib.pyplot as plt
from wordcloud import WordCloud

# Join all the processed sentences back into a single string
all_words = ' '.join(corpus)

# Create a WordCloud object
wordcloud = WordCloud(width = 800, height = 800,
                background_color ='white',
                stopwords = stopwords.words('english'),
                min_font_size = 10).generate(all_words)

# Plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

Term Frequency-Inverse Document Frequency (TF-IDF)

Explanation TF-IDF is a numerical statistic intended to reflect how important a word is to a document in a collection or corpus. It's a weight often used in information retrieval and text mining.

Mathematical Representation

TF (Term Frequency) The frequency of a term t in a document d, calculated as:

$$TF(t,d)=\frac{\text{Number of times term } t \text{ appears in document } d}{\text{Total number of terms in document } d}$$

IDF (Inverse Document Frequency) The log of the number of documents � D divided by the number of documents that contain term � t:

$$IDF(t,D)=\log\left(\frac{\text{Total number of documents } D}{\text{Number of documents containing term } t}\right)$$

TF-IDF The TF-IDF value is obtained by multiplying TF and IDF:

$$\text{TF-IDF}(t,d,D)=TF(t,d)\times IDF(t,D)$$

TF-IDF(t,d,D)=TF(t,d)×IDF(t,D) **Example** Consider two documents in a corpus:

Doc1: "blue house blue" Doc2: "red house" Calculating TF-IDF for "blue" in Doc1:

$$ TF("\text{blue}", \text{Doc1}) = \frac{2}{3} $$

(Here, "blue" appears 2 times out of 3 total terms.)

$$ IDF("\text{blue}", 2) = \log\left(\frac{2}{1}\right) $$

(Here, "blue" appears in 1 document out of 2.)

$$ \text{TF-IDF}("\text{blue}", \text{Doc1}, 2) = \left(\frac{2}{3}\right) \times \log\left(\frac{2}{1}\right) $$

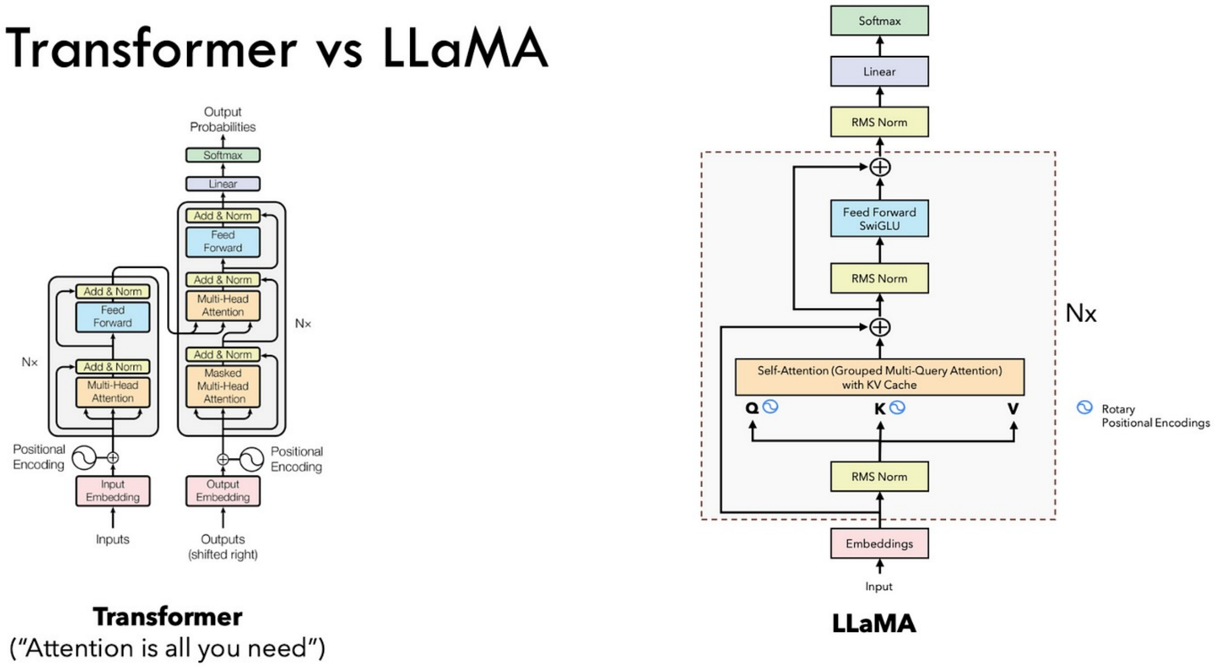
(This is the product of the TF and IDF values calculated above.)

```
#TF-IDF
# -*- coding: utf-8 -*-
"""

@author: Md Al Amin Tokder
"""

import nltk

paragraph =  """I have three visions for India. In 3000 years of our
history, people from all over
               the world have come and invaded us, captured our lands,
conquered our minds.
               From Alexander onwards, the Greeks, the Turks, the
Moguls, the Portuguese, the British,
               the French, the Dutch, all of them came and looted us,
took over what was ours.
               Yet we have not done this to any other nation. We have
not conquered anyone.
               We have not grabbed their land, their culture,
               their history and tried to enforce our way of life on
them.
               Why? Because we respect the freedom of others.That is
why my
               first vision is that of freedom. I believe that India
got its first vision of
               this in 1857, when we started the War of Independence.
It is this freedom that
               we must protect and nurture and build on. If we are not
free, no one will respect us.
               My second vision for India’s development. For fifty
```

```
years we have been a developing nation.
               It is time we see ourselves as a developed nation. We
are among the top 5 nations of the world
               in terms of GDP. We have a 10 percent growth rate in
most areas. Our poverty levels are falling.
               Our achievements are being globally recognised today.
Yet we lack the self-confidence to
               see ourselves as a developed nation, self-reliant and
self-assured. Isn’t this incorrect?
               I have a third vision. India must stand up to the
world. Because I believe that unless India
               stands up to the world, no one will respect us. Only
strength respects strength. We must be
               strong not only as a military power but also as an
economic power. Both must go hand-in-hand.
               My good fortune was to have worked with three great
minds. Dr. Vikram Sarabhai of the Dept. of
               space, Professor Satish Dhawan, who succeeded him and
Dr. Brahm Prakash, father of nuclear material.
               I was lucky to have worked with all three of them
closely and consider this the great opportunity of my life.
               I see four milestones in my career"""


# Cleaning the texts
import re
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from nltk.stem import WordNetLemmatizer

ps = PorterStemmer()
wordnet=WordNetLemmatizer()
sentences = nltk.sent_tokenize(paragraph)
corpus = []
for i in range(len(sentences)):
    review = re.sub('[^a-zA-Z]', ' ', sentences[i])
    review = review.lower()
    review = review.split()
    review = [wordnet.lemmatize(word) for word in review if not word
in set(stopwords.words('english'))]
    review = ' '.join(review)
    corpus.append(review)

# Creating the TF-IDF model
from sklearn.feature_extraction.text import TfidfVectorizer
cv = TfidfVectorizer()
X = cv.fit_transform(corpus).toarray()
X
```

```
array([[0.        , 0.        , 0.        , ..., 0.        , 0.
,
        0.        ],
       [0.        , 0.        , 0.        , ..., 0.25883507,
0.30512561,
        0.        ],
       [0.        , 0.28867513, 0.        , ..., 0.        , 0.
,
        0.        ],
       ...,
       [0.        , 0.        , 0.        , ..., 0.        , 0.
,
        0.        ],
       [0.        , 0.        , 0.        , ..., 0.        , 0.
,
        0.        ],
       [0.        , 0.        , 0.        , ..., 0.        , 0.
,
        0.        ]])
```

Basic Overview Of LLMs Structure



Overview of Large Language Models (LLMs)

Large Language Models (LLMs) are a type of artificial intelligence model designed to understand, generate, and interact with human language at a sophisticated level. They are "large" both in terms of the size of the neural networks used and the amount of data they have been trained on.

Key Features of LLMs:

Size: LLMs are built with a massive number of parameters (weights in the neural network). Models like GPT-3, for example, have hundreds of billions of parameters.

Training Data: These models are trained on extensive corpora of text data, encompassing a wide range of topics, styles, and languages.

Capabilities: LLMs can perform a variety of language tasks without task-specific training. This includes translation, summarization, question-answering, and content generation.

Structure of Large Language Models

The structure of LLMs is based on advanced neural network architectures, predominantly transformers (discussed in detail below). They typically use a variant of the sequence-to-sequence model, which is adept at handling variable-length input and output sequences.

Input Processing: Text inputs are tokenized (converted into a series of tokens) and then embedded into a numerical space that the model can process.

Attention Mechanisms: LLMs heavily rely on attention mechanisms, especially self-attention, to weigh the importance of different parts of the input data.

Layered Architecture: The models consist of multiple layers of processing, with each layer performing complex transformations of the input data.

Output Generation: For generation tasks, LLMs use their understanding of language syntax and semantics to produce coherent and contextually relevant text.

Transformer Architecture

Transformers, introduced in the paper "Attention is All You Need" by Vaswani et al., represent a significant shift in how neural networks are applied to NLP tasks. They are the backbone of most modern LLMs.

Key Features of Transformers:

- **Attention Mechanism:** The core innovation of transformers is the attention mechanism, specifically self-attention. It allows the model to focus on different parts of the input sequence when producing each part of the output sequence, enabling it to capture long-range dependencies in text.
- **No Recurrence or Convolution:** Unlike previous models like RNNs and CNNs, transformers do not rely on recurrence or convolution. This allows for more parallel processing and hence faster training times.
- **Encoder-Decoder Structure:** Many transformers use an encoder-decoder structure. The encoder processes the input data and the decoder generates the output. This structure is particularly effective for translation tasks.

Working of a Transformer:

- **Encoding Phase:** Each word (token) in the input sequence is processed in parallel. The self-attention mechanism allows each token to interact with every other token in the sequence, capturing their contextual relationships.
- **Decoding Phase:** The decoder similarly uses attention mechanisms to focus on different parts of the encoder's output while generating the text sequence.

Importance in NLP: Transformers have drastically improved the performance of NLP models. They are more efficient and scalable compared to previous architectures, leading to breakthroughs in language understanding and generation tasks.