We at The Data Monk hold the vision to make sure everyone in the IT industry has an equal stand to work in an open domain such as analytics. Analytics is one domain where there is no formal under-graduation degree and which is achievable to anyone and everyone in the World.

We are a team of 30+ mentors who have worked in various product-based companies in India and abroad, and we have come up with this idea to provide study materials directed to help you crack any analytics interview.

Every one of us has been interviewing for at least the last 6 to 8 years for different positions like Data Scientist, Data Analysts, Business Analysts, Product Analysts, Data Engineers, and other senior roles. We understand the gap between having good knowledge and converting an interview to a top product-based company.

Rest assured that if you follow our different mediums like our blog cum questions-answer portal www.TheDataMonk.com , our youtube channel - The Data Monk, and our e-books, then you will have a very strong candidature in whichever interview you participate in.

There are many blogs that provide free study materials or questions on different analytical tools and technologies, but we concentrate mostly on the questions which are asked in an interview. We have a set of 100+ books which are available both on Amazon and on The Data Monk e-shop page

We would recommend you to explore our website, youtube channel, and e-books to understand the type of questions covered in our articles. We went for the question-answer approach both on our website as well as our e-books just because we feel that the best way to go from beginner to advance level is by practicing a lot of questions on the topic.

We have launched a series of 50 e-books on our website on all the popular as well as niche topics. Our range of material ranges from SQL, Python, and Machine Learning algorithms to ANN, CNN, PCA, etc.

We are constantly working on our product and will keep on updating it. It is very necessary to go through all the questions present in this book.

Give a rating to the book on Amazon, do provide your feedback and if you want to help us grow then please subscribe to our Youtube channel.

NATURAL LANGUAGE PROCESSING

- As NLP is one of the major topics nowadays and better understood with examples and code. So, I have included codes after every topic. Hope you will like it.

Q1.What is the need of NLP?

A1. We have many machine learning models for predicting numerical values. Almost all the algorithms work on numerical values as they cannot work on characters. So, NLP is introduced to make algorithms work with text also.

Q2. How is NLP used in ml?

A2. Suppose we have a sentence "I am a data scientist". We cannot pass it directly to the machine learning algorithms. So, we convert the word in sentence in numerical value (vector). Now, we can pass it to the machine learning algorithms and they can work n it flawlessly.

Q3. What is the path to use NLP?

A3. These are the steps for using NLP:

- Text Preprocessing (Level-1) – Tokenization, Lemmatization, Stopwords
- Text preprocessing (Level-2) – Bag of words(BOW) , TFIDF , Unigrams, Bigrams, n-grams
- Text Preprocessing – Genism, Word2Vec, AvgWord2Vec
- Solve Machine Learning Use Cases – Like sentiment analysis, Fake news classifier
- Understand the usage of ANN
- Understand the working of RNN, LSTM, GRU
- Text Preprocessing (Level-3) – Word embedding, Word2Vec

- Bidirectional LSTM, RNN, Encoders and Decoders, Attention Models
- Transformers
- BERT

Q4. What are the applications of NLP?

A4. NLP is used in various industries nowadays. It is becoming one of the most popular algorithms in data science field. Some of the applications of NLP are:

- Amazon’s Alexa
- Google Voice Command
- G-Mail spam classifier
- Chat bots
- Language Translator
- Google search’s Autocorrect and Auto complete

Q5. What are the libraries used for performing NLP tasks?

A5. Libraries used for performing NLP tasks are:

- Tensorflow
- Keras
- NLTK
- Spacepy
- Sklearn

Q6. What is Tokenization?

A6. It is one of the initial steps of text preprocessing. It is a technique which is used to convert big texts like paragraph into small groups (tokens) like sentences or word. If, we will use that big text it will take very much computational power as well as much time to get computed. According to Wikipedia, “Tokens are building blocks of NLP”. We can break that big text into smaller packets. We can convert the text into either word, Character, Sub words(n-grams characters).

Q7. What are different types of tokenization techniques?

A7. There are many ways of tokenizing techniques. You can break down the text in the following ways:

- Sentences
- Words
- Sub words (n-gram character)

Q8. What is the need of tokenization?

A8. Suppose we have an input of words which is of length 10,000 and we will pass it directly to the model then the process we will be slow, more time consumption, more computational power will be used. So, we will apply tokenization to break the whole input into small pieces due to which accuracy will increase, less computational power will be used and less time will be required.

Q9. How sentence tokenization works?

A9. In sentence tokenization, big texts such as paragraphs are converted into sentences. For example, “Hey! My name is Dev Chauhan. I am from Delhi. My age is 21. I am pursuing data science as my career.”. So, this will be converted to separate sentences after applying sentence tokenization as: “Hey! My name is Dev Chauhan”, “I am from Delhi”, “My age is 21”, “I am pursuing data science as my career”.

Q10. How word tokenization works?

A10. In word tokenization, a big text can be break down into words. For example, “My name is Dev Chauhan”. Now when we apply word tokenization to it, we will get the result as : “My-name-is-Dev-Chauhan”.

Q11. How character tokenization works?

A11. In character tokenization, a big text can be break down into single characters. For example, “Bestest”, after applying character tokenization on it, we will get the result as: “B-e-s-t-e-s-t”.

Q12. How sub words (n-gram) tokenization works?

A12. In sub words tokenization, a text can be break down into sub words. For example, “Bestest”, after applying n-gram tokenization on it, we will get the result as: “Best-est”.

Code for Better Understanding of Tokenization

#Import Libraries

```
import nltk
nltk.download()
```

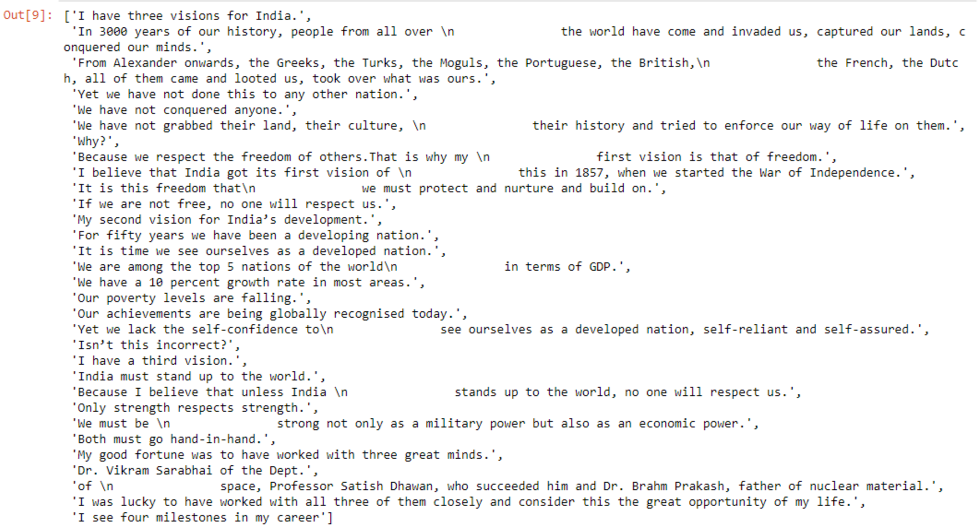
#Paragraph

```
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."""
```

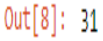
#Converting paragraph into sentences

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



```
Out[9]: ['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, c
        onquered our minds.',
         'From Alexander onwards, the Greeks, the Turks, the Moguls, the Portuguese, the British,\n              the French, the Dutc
        h, 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']
```

len(sentences)

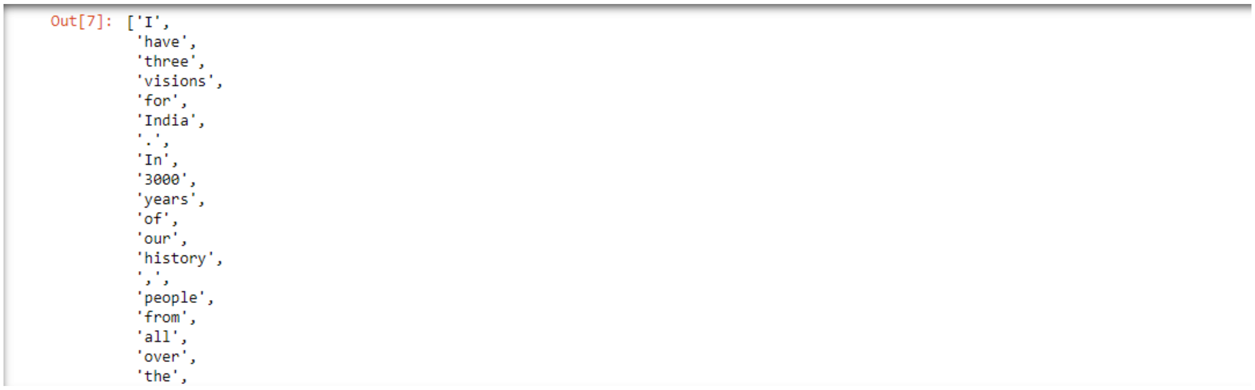


```
Out[8]: 31
```

#Converting paragraph into words, all the words, punctuation marks and all will be counted in it.

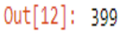
words=nltk.word_tokenize(paragraph)

words



```
Out[7]: ['I',
         'have',
         'three',
         'visions',
         'for',
         'India',
         '.',
         'In',
         '3000',
         'years',
         'of',
         'our',
         'history',
         ',',
         'people',
         'from',
         'all',
         'over',
         'the',
```

len(words)



```
Out[12]: 399
```

Q13. What is Stemming?

A13. After tokenization, we reach at the stemming process where we try to bring word to its original root/base. According to Wikipedia "It is the process of reducing the infected words to their word stem". As there can be many words in the text which are same, just used written differently.

Q14. How does it work?

A14.Stemming brings down word to base/root word which may be having some meaning or no meaning at all. For example, history , historical → histori

finally , final , finalized → fina

Here, we can see history and historical are converted into histori which does not make any sense.

Q15. What is the limitation of stemming?

A15. The word converted into root/base word after stemming are most of the times are not meaningful. For example, histori, fina, etc. So, to overcome this problem we use lemmatizer.

Code for performing Stemming

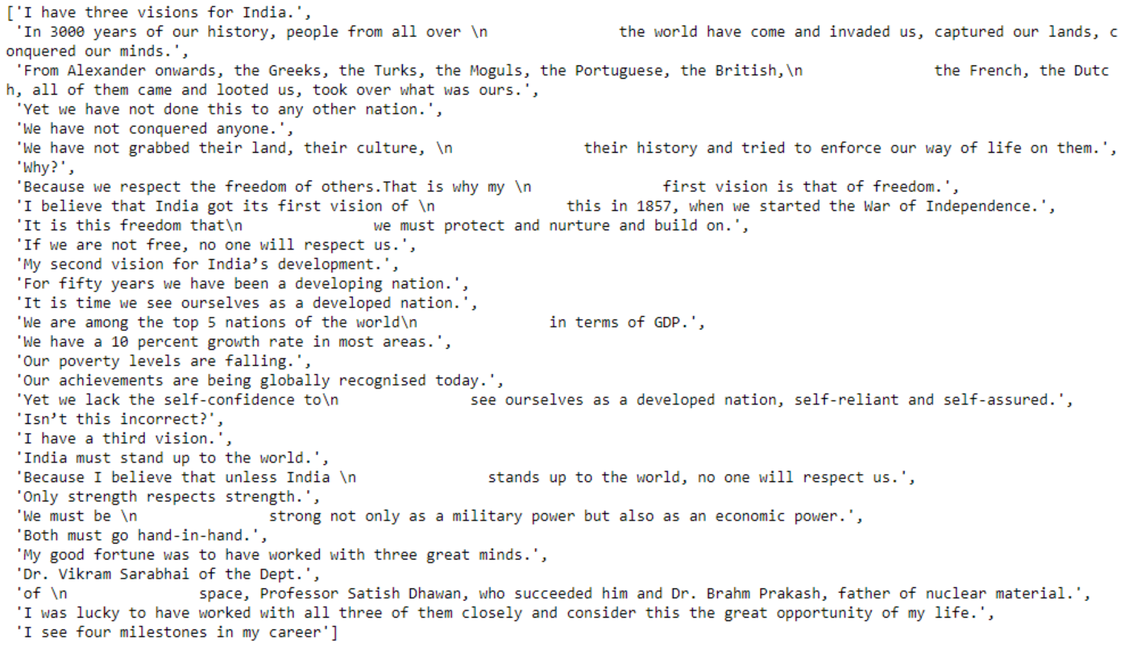
#Import Libraries

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

#Paragraph

```
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
```

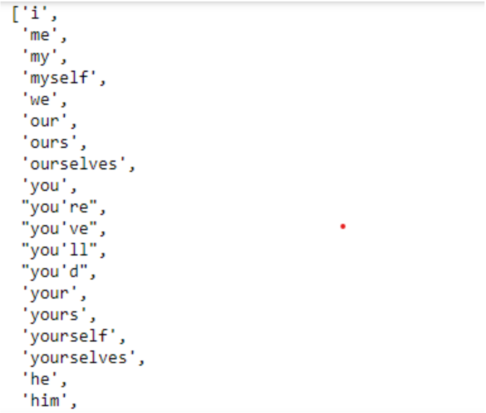
```
        friends, I love you dearly; you know who you are. And lastly,
        I just want to say this: Making The Revenant was about
        man's relationship to the natural world. A world that we
        collectively felt in 2015 as the hottest year in recorded
        history. Our production needed 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."""


#Converting paragraph into sentences
sentences=nltk.sent_tokenize(paragraph)
sentences
```



```
['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, c
onquered our minds.',
 'From Alexander onwards, the Greeks, the Turks, the Moguls, the Portuguese, the British,\n               the French, the Dutc
h, 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']
```

```
#Stemming
stemmer=PorterStemmer()
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"))]    #stopwords.words("english") it means i want
all the stopwords from english language
   sentences[i]=" ".join(words)
```

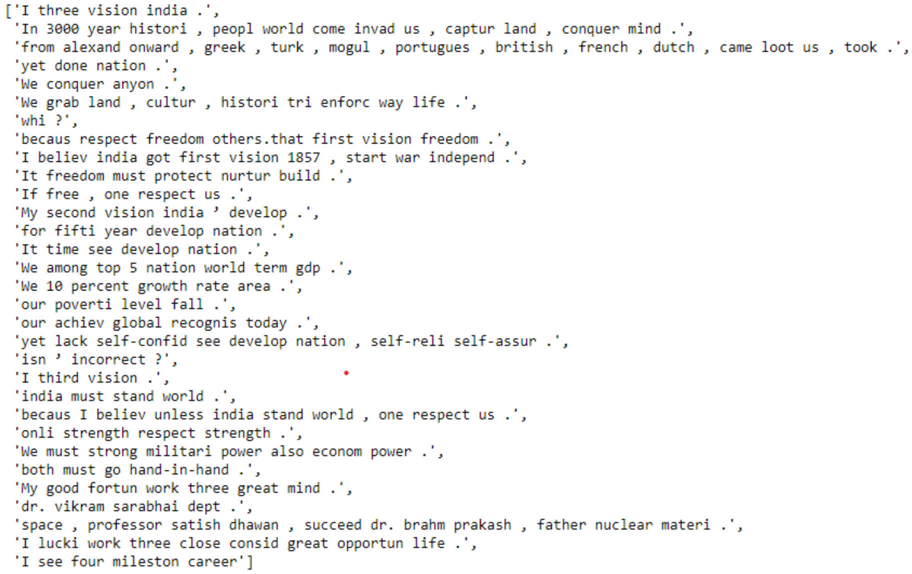
```
#They do not provide any help in the model to predict. They may rather cause
confusions.
stopwords.words("english")
```



```
['i',
 'me',
 'my',
 'myself',
 'we',
 'our',
 'ours',
 'ourselves',
 'you',
 "you're",
 "you've",
 "you'll",
 "you'd",
 'your',
 'yours',
 'yourself',
 'yourselves',
 'he',
 'him',
```

#Here you can see that all the stop words are removed and words are stemmed to their base.

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']
```

Q16. What is lemmatization?

A16. It is used after we have done tokenization, we use lemmatization to bring the word to its root/base value. We stem them to its root value but here we see that the root/base word will always be a meaningful word. And also it can also give another form of a word having same meaning.

Q17. How does lemmatization work?

A17. Lemmatization brings the word to its root/base value. But, in lemmatization, the root/base word is a meaningful word. For example,

history, historical → history

finally, final, finalized → final

As we can see, the root/base words are having some value unlike stemming.

Q18. How lemmatization and stemming different?

A18.

- Lemmatization brings down the word to its stem such that it is a meaningful word and readable and understood by human beings. But in case of stemming, it brings back words to its root but, it may not be a meaningful word.
- Lemmatization takes a lot of time to compute as it brings down the value to a meaningful word. As, it has to understand each and everything inside that particular word. Whereas, Stemming, it does not take that much time, as it just the base word
- Stemming is used in → Sentiment analysis, g-mail spam detection
  Lemmatization is used in → Chatbots, question/answer application (as they require meaningful words)

Q19. What is the limitation of lemmatization?

A19. Lemmatization takes a lot of time to compute as it brings down the value to a meaningful word. As, it has to understand each and everything inside that particular word.

Q20.What are stop words?

A20. Stop words are the words in the sentence that are not meaningful and does not help us to understand anything about that sentence. So, we remove the stop words from the sentence as they do not provide much information about that sentence. For example, in a sentence “My name is Dev” after removing stop words we will “name Dev”, as “my is” does not provide any information about that sentence.

Q21. Why is it important to remove stop words?

A21. It is important to remove stop words because if do not remove stop words then they will be cause miscalculations and confusions.

Q22. What are the places where stemming is used?

A22. Stemming can be used in Sentiment analysis, g-mail spam detection.

Q23. What are the places where lemmatization is used?

A23. Lemmatization can be used in Chat bots, question-answer applications (as they require meaningful words).

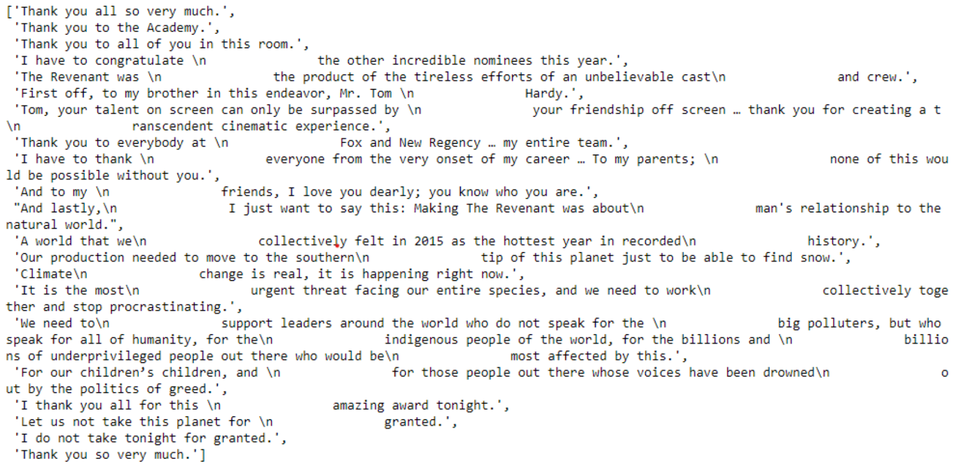
Code for performing Lemmatization

```
#Import libraries

import nltk
from nltk.stem import WordNetLemmatizer
from nltk.corpus import stopwords

#Paragraph
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."""


#Converting paragraph into sentences
sentences=nltk.sent_tokenize(paragraph)
lemmatizer=WordNetLemmatizer()
```

#Before applying lematization

Sentences



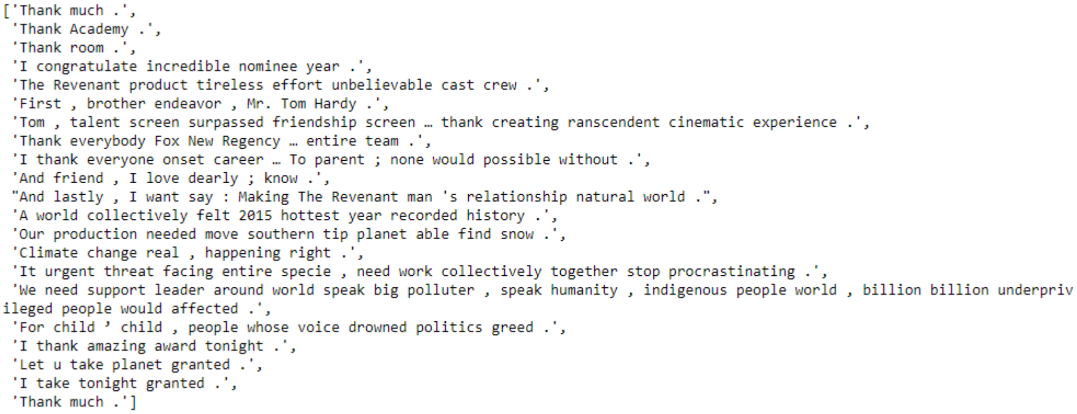
```
['Thank you all so very much.',
 'Thank you to the Academy.',
 'Thank you to all of you in this room.',
 'I have to congratulate \n               the other incredible nominees this year.',
 'The Revenant was \n               the product of the tireless efforts of an unbelievable cast\n          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

#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)
```

#After applying lemmatization

Sentences



```
['Thank much .',
 'Thank Academy .',
 'Thank room .',
 'I congratulate incredible nominee year .',
 'The Revenant product tireless effort unbelievable cast crew .',
 'First , brother endeavor , Mr. Tom Hardy .',
 'Tom , talent screen surpassed friendship screen … thank creating ranscendent cinematic experience .',
 'Thank everybody Fox New Regency … entire team .',
 'I thank everyone onset career … To parent ; none would possible without .',
 'And friend , I love dearly ; know .',
 "And lastly , I want say : Making The Revenant man 's relationship natural world .",
 'A world collectively felt 2015 hottest year recorded history .',
 'Our production needed move southern tip planet able find snow .',
 'Climate change real , happening right .',
 'It urgent threat facing entire specie , need work collectively together stop procrastinating .',
 'We need support leader around world speak big polluter , speak humanity , indigenous people world , billion billion underpriv
ileged people would affected .',
 'For child ’ child , people whose voice drowned politics greed .',
 'I thank amazing award tonight .',
 'Let u take planet granted .',
 'I take tonight granted .',
 'Thank much .']
```

Q24. What is Bag of Words?

A24. In machine learning, we cannot provide the string directly to the model. As they only work with numbers. So we convert the string into numerical value with the help of bag of words. Bag of words, preprocesses the text by converting it into a *bag of words*, where we keep the track of frequency of each word in a sentence and whole.

Q25. What are the steps for creating BOW?

A25. The steps to create BOW are as follows:

- Firstly, we will remove the stop words from the raw text.
- Then, we will count the frequency of each word in all the sentences and sort them I descending order.
- Then, we will create a table representing the BOW, where if in a sentence, a word is present in that sentence we will place"1", if not present we will place "0".

Q26. Explain BOW with an example?

A26. Let me explain BOW with an example to make you understand it nicely.

Suppose we have 3 sentences,

Sent-1 → He is a good boy.
Sent-2 → He is a good girl.
Sent-3 → Boy and girl are good.

Now we have to remove stop words from the sentence,

Sent-1 → good boy
Sent-2 → good girl
Sent-3 → boy girl good

Now we will calculate the frequency of a particular word in all the sentences,

Word	Frequency
Good	3
Boy	2
Girl	2

Always remember that the words are kept in sorted order.
Now, the main work comes into play we will convert the words into vectors with the help of the table given above.

Sentences	Good	Boy	Girl
Sent-1	1	1	0
Sent-2	1	0	1
Sent-3	1	1	1

Now let us understand what table represents, we first take sent-1, in it there are only 2 words, so we marked "1" for words which are in the sentence and "0" if not in the sentence. If the frequency of a word is "n" in a sentence then we will place "n" in the table.

Q27. What are different types of BOW?

A27. There are 2 types of BOW:

- Binary Bow – In this only "0" is used if the word is not present in that sentence and "1" is used if the word is present in that sentence. We cannot determine the frequency that how many times the word appeared I a given sentence.
- Simple Bow – In this "0" is used if the word is not present in that sentence and "n" where n is the frequency of word present in the sentence.

Q28. What is the disadvantage of using BOW?

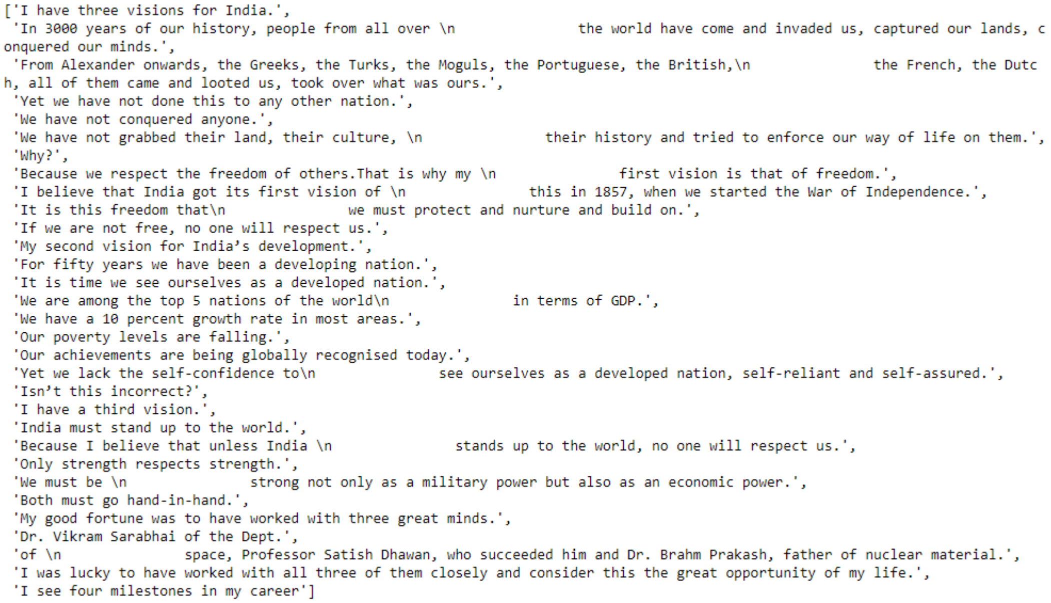
A28. In the above table, we are having "1" for both "good" and "boy" in sent-1. They both are having equal representation/semantics. They are almost same, with same semantics. So, model will not be able to know which of them is more important. For example, if we are using sentiment analysis, we have to find which word is more important so that model can focus on it and give result accordingly and accurately. To overcome this we have TFIDF (Term Frequency and Inverse Document Frequency)

Code for implementing Bag Of Words(BOW)

```
#Import Libraries
import nltk
import re
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from nltk.stem import WordNetLemmatizer


#Paragraph
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."""
```

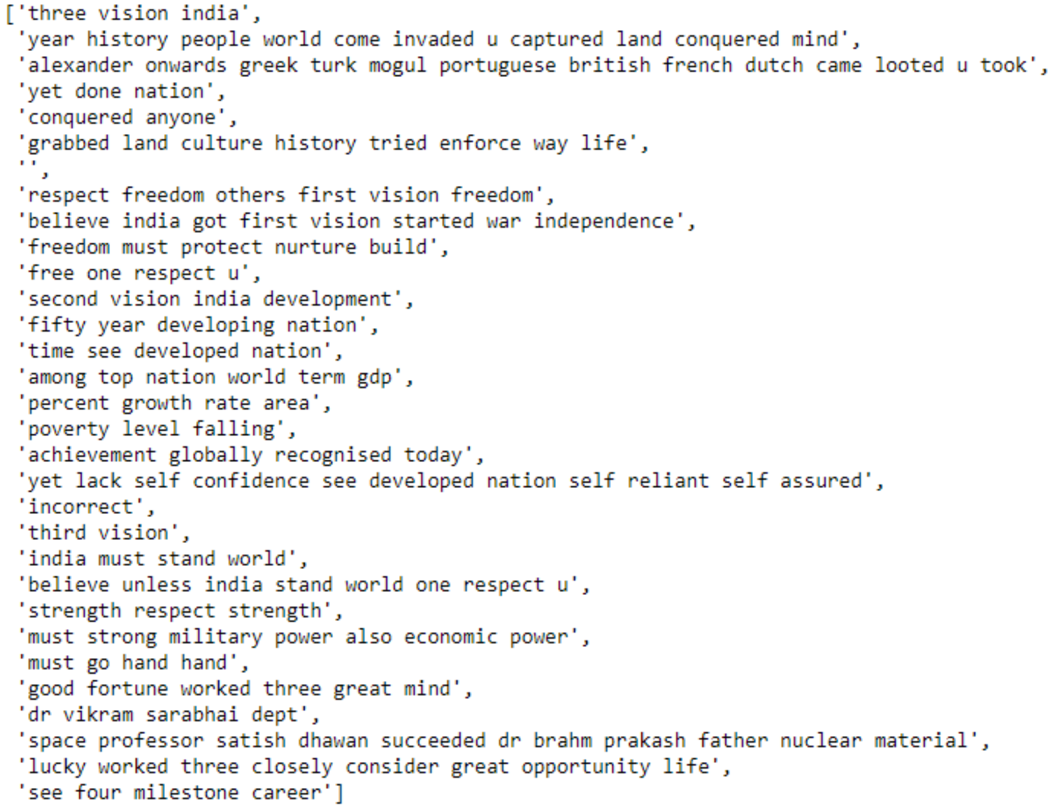
#Cleaning Text

```
stemmer=PorterStemmer()
lemmatizer=WordNetLemmatizer()
sentences=nltk.sent_tokenize(paragraph)
corpus=[]
sentences
```



```
['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, c
onquered our minds.',
 'From Alexander onwards, the Greeks, the Turks, the Moguls, the Portuguese, the British,\n               the French, the Dutc
h, 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']
```

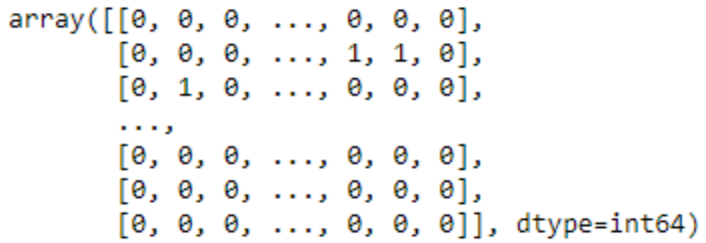
```
for i in range(len(sentences)):
    #For replacing everything other then alphabets with space
    review=re.sub("[^a-zA-Z]"," ",sentences[i])
    #Converting text to lowercase
    review=review.lower()
    #After splitting this review will become list of words
    review=review.split()
    #Stemming the words other than stopwords
    review=[lemmatizer.lemmatize(word) for word in review if word not in set(stopwords.words("english"))]
    #After stemming all the words we will join the list as review
    review=" ".join(review)
    #Appending the review in the list
    corpus.append(review)
corpus
```



```
['three vision india',
 'year history people world come invaded u captured land conquered mind',
 'alexander onwards greek turk mogul portuguese british french dutch came looted u took',
 'yet done nation',
 'conquered anyone',
 'grabbed land culture history tried enforce way life',
 '',
 'respect freedom others first vision freedom',
 'believe india got first vision started war independence',
 'freedom must protect nurture build',
 'free one respect u',
 'second vision india development',
 'fifty year developing nation',
 'time see developed nation',
 'among top nation world term gdp',
 'percent growth rate area',
 'poverty level falling',
 'achievement globally recognised today',
 'yet lack self confidence see developed nation self reliant self assured',
 'incorrect',
 'third vision',
 'india must stand world',
 'believe unless india stand world one respect u',
 'strength respect strength',
 'must strong military power also economic power',
 'must go hand hand',
 'good fortune worked three great mind',
 'dr vikram sarabhai dept',
 'space professor satish dhawan succeeded dr brahm prakash father nuclear material',
 'lucky worked three closely consider great opportunity life',
 'see four milestone career']
```

#Creating the Bag of Words model

```
from sklearn.feature_extraction.text import CountVectorizer        #In CountVectorizer almost all the work is done of creating BOW
cv=CountVectorizer(max_features=1500)
X=cv.fit_transform(corpus).toarray()

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]], dtype=int64)
```

Here we can see that the sentences are converted into array of vectors.

Q29. Why do we use TFIDF?

A29. As we saw in BOW, the words are not given any importance. The common words and uncommon words are given the same importance. But in TFIDF, the uncommon words are given more importance.

Q30. What is TFIDF?

A30. TFIDF stands for Term Frequency and Inverse Document Frequency. It is used to convert text into vectors with the help of Term Frequency and Inverse Document Frequency.

Q31. How do we calculate Term Frequency(TF)?

A31. Term Frequency is calculated by the given formula,

TF = (no. of repeating words in a sentence / no. of words in sentence)

With the help of this formula we will calculate TF for every word and it will help us in making the TF table.

Q32. How do we calculate the Inverse Document frequency (IDF)?

A32. Inverse Document Frequency is calculated by the formula given below:

IDF = log (no. of sentences / no. of sentences containing that word)

With the help of this formula, we will be calculate the IDF for every word and it will help us to make the IDF table.

Q33. Explain TFIDF with an example?

A33. Suppose we have 3 raw sentences as given below:

Sent-1 → He is a good boy.
Sent-2 → He is a good girl.
Sent-3 → Boy and Girl are good.

Now we have to remove stop words from the sentence,

Sent-1 → good boy
Sent-2 → good girl
Sent-3 → boy girl good

Now we will calculate the frequency of a particular word in all the sentences,

Word	Frequency
Good	3
Boy	2
Girl	2

Always remember that the words are kept in sorted order.
Now, we will make a table regarding the Term Frequency,

Term Frequency = (no. of rep words in sentence / no. of words in a sentence)

By using this formula we will create the table for Term Frequency.

words	Sent-1	Sent-2	Sent-3
good	½	½	1/3
Boy	½	0	1/3
Girl	0	½	1/3

Now, we will make a table for Inverse Document Frequency,

IDF = log (no. of sentences / no. of sentences containing that word)

Now, with the help of this formula we will create the table for IDF,

Words	IDF
Good	Log(3/3)=0
Boy	Log(3/2)
Girl	Log(3/2)

Now, we will multiply these two tables and we will get our desired output as TF*IDF,

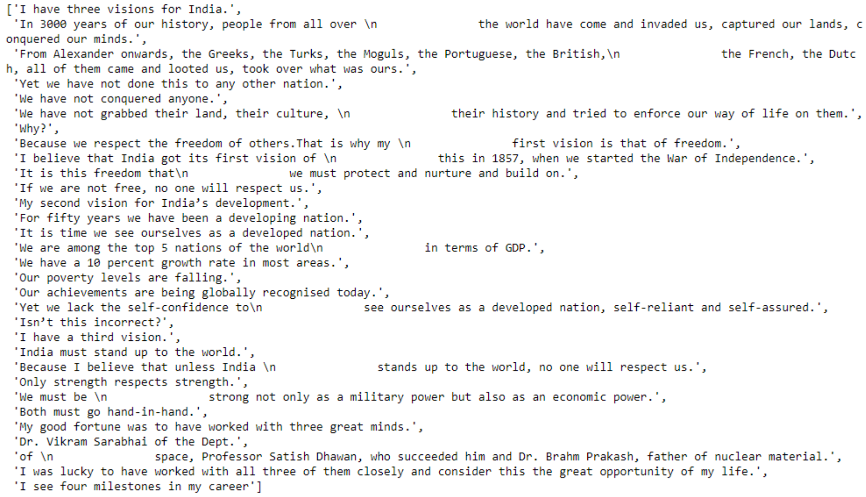
Sentences	Good(f1)	Boy(f2)	Girl(f3)
Sent-1	½ * log(3/3)	½ * log(3/2)	0
Sent-2	0	0	½ * log(3/2)
Sent-3	0	1/3 * log(3/2)	1/3 * log(3/2)

From the above table we can see that words are given importance accordingly.

Code for Implementing TFIDF

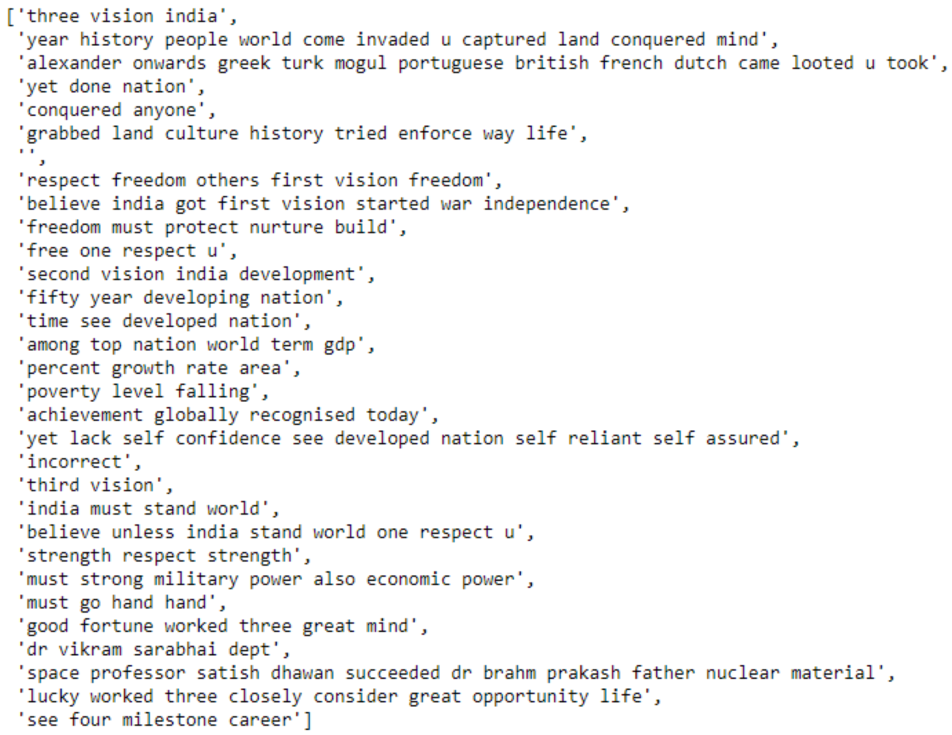
```
#Import Libraries
import nltk
import re
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from nltk.stem import WordNetLemmatizer

#Paragraph
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."""
```

```
#Cleaning the text
stemmer=PorterStemmer()
lemmatizer=WordNetLemmatizer()
sentences=nltk.sent_tokenize(paragraph)
sentences
```



```
['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, c
onquered our minds.',
 'From Alexander onwards, the Greeks, the Turks, the Moguls, the Portuguese, the British,\n              the French, the Dutc
h, 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']
```

```
corpus=[]
for i in range(len(sentences)):
   review=re.sub("[^a-zA-Z]"," ",sentences[i])
   review=review.lower()
   review=review.split()
   review=[lemmatizer.lemmatize(word) for word in review if not word in
set(stopwords.words("english"))]
   review=" ".join(review)
   corpus.append(review)
corpus
```



```
['three vision india',
 'year history people world come invaded u captured land conquered mind',
 'alexander onwards greek turk mogul portuguese british french dutch came looted u took',
 'yet done nation',
 'conquered anyone',
 'grabbed land culture history tried enforce way life',
 '',
 'respect freedom others first vision freedom',
 'believe india got first vision started war independence',
 'freedom must protect nurture build',
 'free one respect u',
 'second vision india development',
 'fifty year developing nation',
 'time see developed nation',
 'among top nation world term gdp',
 'percent growth rate area',
 'poverty level falling',
 'achievement globally recognised today',
 'yet lack self confidence see developed nation self reliant self assured',
 'incorrect',
 'third vision',
 'india must stand world',
 'believe unless india stand world one respect u',
 'strength respect strength',
 'must strong military power also economic power',
 'must go hand hand',
 'good fortune worked three great mind',
 'dr vikram sarabhai dept',
 'space professor satish dhawan succeeded dr brahm prakash father nuclear material',
 'lucky worked three closely consider great opportunity life',
 'see four milestone career']
```

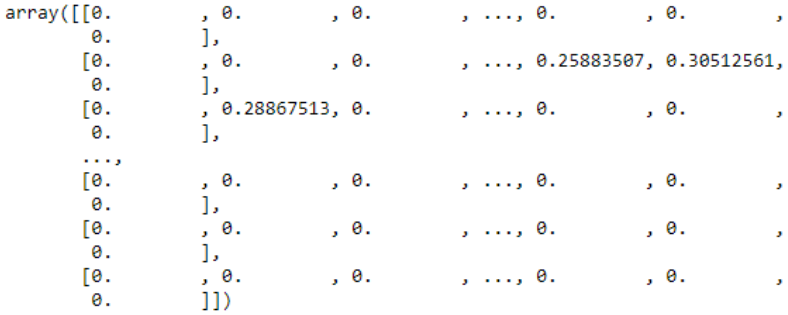
#Creating TFIDF model

from sklearn.feature_extraction.text import TfidfVectorizer

tfv=TfidfVectorizer()

X=tfv.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.        ]])
```

Q34. What are the problems in BOW and TFIDF?

A34.

1. In both BOW and TFIDF approach semantic information is not stored. TFIDF also gives importance to uncommon words.
2. There are also chances of over fitting in them.

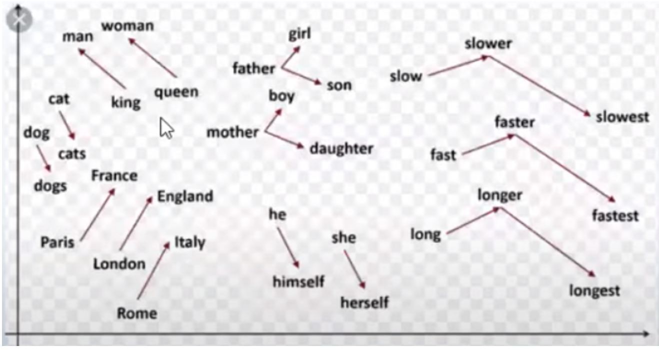
Q35. Why do we use Word2Vec?

A35. To overcome the above stated problems, we use Word2Vec. Word2Vec is also used to compute the model with big datasets.

Q36. What is Word2Vec?

A36. In Word2Vec, each word is basically represented as a vector of 32 or more dimensions instead of a single number.

In it the semantic information and relation between different words is also preserved. Due to which it performs very well with big dataset too.



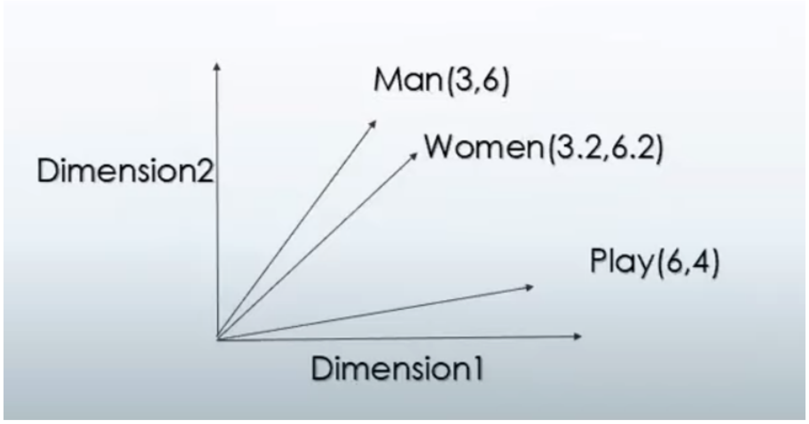
This is how words are represented into vectors, here we can see that the similar words are placed near to each other and different words are placed far from each other.

Q37. Give an example of representation of Word2Vec?

A37. So, let us understand the Word2Vec with a example,

Suppose we have 3 words “man” , ”women” , ”play” . We will be representing them in two dimensions only.

Here is how the representation looks like,



As from the above representation, we can see that the similar words "man" and "women" are placed near to each other. And the words different from them are placed far away from them. We can see that "play" is placed away from the other words.

Q38. Why is Word2Vec so famous nowadays?

A38. Word2Vec is a very hot topic nowadays in big companies, as it is very good with handling big datasets. Nowadays, many companies are doing research on it to make the best version of it. For example, some of the companies are trying to implement it with some kind of mathematical equation so that we can get a desired result.

King – Man +Women = Queen

Here the words in the above equation represent vectors.

Q39. What are the steps to create Word2Vec?

A39. The basic steps to implement Word2Vec are:

- Tokenize the raw sentences.
- Create histograms for the sentences
- Out of all words, take the most frequent words.
- Create a matrix with all the unique words. It also represents the occurrence relation between the words.

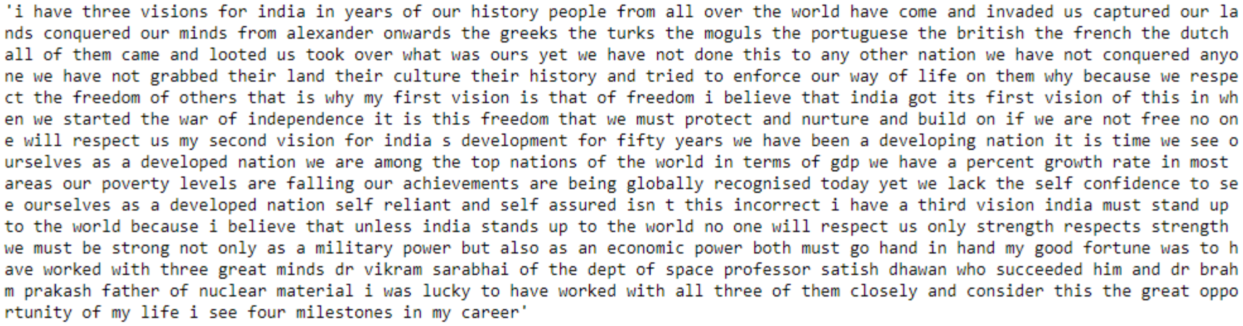
Q40. Which library is used to implement Word2Vec?
A40. Gensim library is used to implement Word2Vec. It creates upto 100 dimensions. To download it just write in your prompt - *pip install gensim.* It is a pretty amazing library with many useful features to find similar words, converting to vectors and many more.

Code to implement Word2Vec

```
#Import Libraries
import nltk
from gensim.models import Word2Vec
from nltk.corpus import stopwords
import re

#Paragraph
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
```

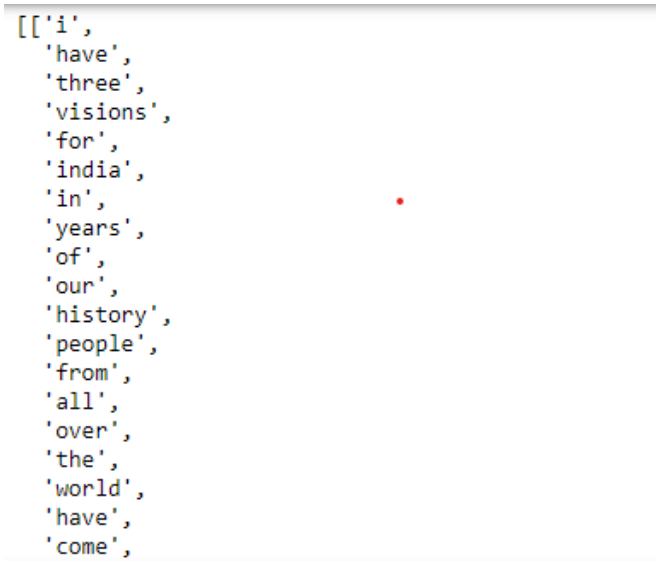
```
change is real, it is happening right now. It is the most
urgent threat facing our entire species, and we need to work
collectively together and stop procrastinating. We need to
support leaders around the world who do not speak for the
big polluters, but who speak for all of humanity, for the
indigenous 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."""
```

```
#removing all the things other than alphabets
review=re.sub("[^a-zA-Z]"," ",paragraph)
review=review.lower()
#removing all the unnecessary spaces
review = re.sub(r'\s+',' ',review)
review
```



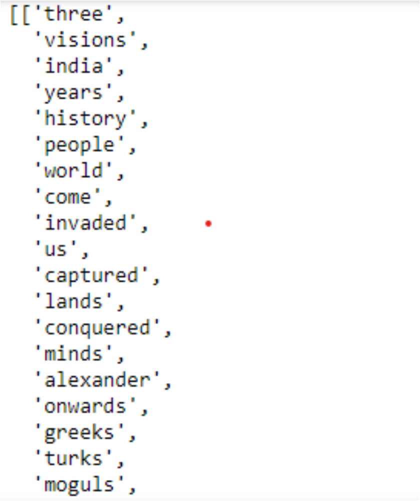
```
'i have three visions for india in years of our history people from all over the world have come and invaded us captured our la
nds conquered our minds from alexander onwards the greeks the turks the moguls the portuguese the british the french the dutch
all of them came and looted us took over what was ours yet we have not done this to any other nation we have not conquered anyo
ne we have not grabbed their land their culture their history and tried to enforce our way of life on them why because we respe
ct 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 wh
en 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 on
e will respect us my second vision for india s development for fifty years we have been a developing nation it is time we see o
urselves as a developed nation we are among the top nations of the world in terms of gdp we have a 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 se
e 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 h
ave worked with three great minds dr vikram sarabhai of the dept of space professor satish dhawan who succeeded him and dr brah
m prakash father of nuclear material i was lucky to have worked with all three of them closely and consider this the great oppo
rtunity of my life i see four milestones in my career'
```

```
#Tokenizing the sentence into words
sentences = [nltk.word_tokenize(sentence) for sentence in sentences]
sentences
```



```
[['i',
  'have',
  'three',
  'visions',
  'for',
  'india',
  'in',
  'years',
  'of',
  'our',
  'history',
  'people',
  'from',
  'all',
  'over',
  'the',
  'world',
  'have',
  'come',
```

```
#Removing stop words
for i in range(len(sentences)):
   sentences[i]=[word for word in sentences[i] if not word in set(stopwords.words("english"))]
print(sentences)
```



```
[['three',
  'visions',
  'india',
  'years',
  'history',
  'people',
  'world',
  'come',
  'invaded',
  'us',
  'captured',
  'lands',
  'conquered',
  'minds',
  'alexander',
  'onwards',
  'greeks',
  'turks',
  'moguls',
```

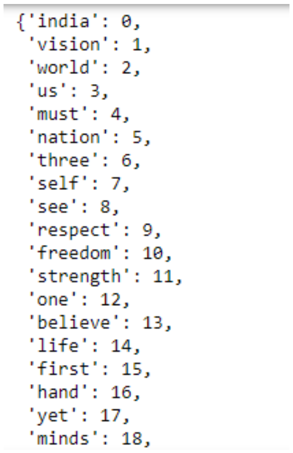
#Training the word2vec model

model = Word2Vec(sentences,min_count=1)   (min_count represents that if a word is present less than it, that word will be skipped)

#vocabularies found out in this paragraph

words = model.wv.key_to_index

print(words)

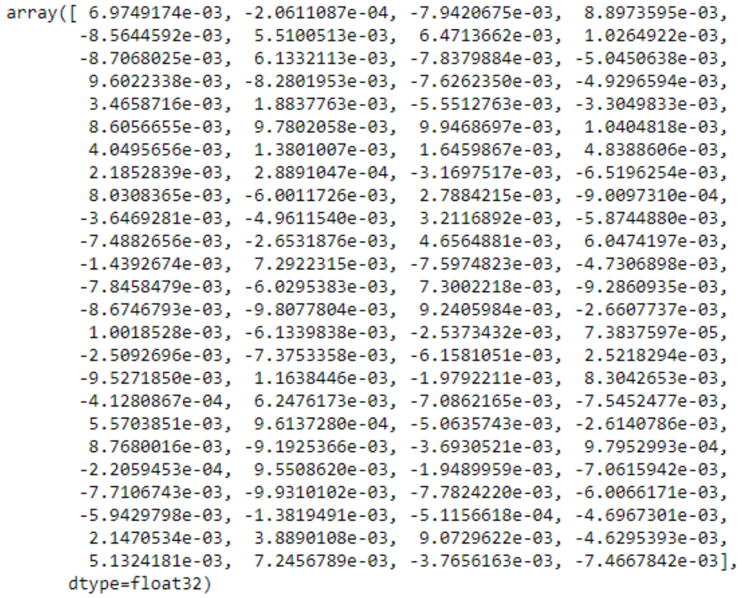


```
{'india': 0,
 'vision': 1,
 'world': 2,
 'us': 3,
 'must': 4,
 'nation': 5,
 'three': 6,
 'self': 7,
 'see': 8,
 'respect': 9,
 'freedom': 10,
 'strength': 11,
 'one': 12,
 'believe': 13,
 'life': 14,
 'first': 15,
 'hand': 16,
 'yet': 17,
 'minds': 18,
```

# Finding Word Vectors

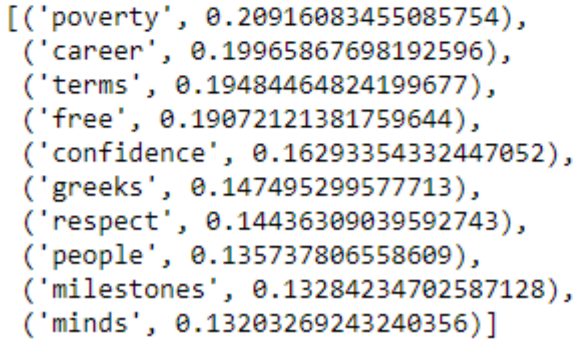
vector = model.wv['war']

vector



```
array([ 6.9749174e-03, -2.0611087e-04, -7.9420675e-03,  8.8973595e-03,
       -8.5644592e-03,  5.5100513e-03,  6.4713662e-03,  1.0264922e-03,
       -8.7068025e-03,  6.1332113e-03, -7.8379884e-03, -5.0450638e-03,
        9.6022338e-03, -8.2801953e-03, -7.6262350e-03, -4.9296594e-03,
        3.4658716e-03,  1.8837763e-03, -5.5512763e-03, -3.3049833e-03,
        8.6056655e-03,  9.7802058e-03,  9.9468697e-03,  1.0404818e-03,
        4.0495656e-03,  1.3801007e-03,  1.6459867e-03,  4.8388606e-03,
        2.1852839e-03,  2.8891047e-04, -3.1697517e-03, -6.5196254e-03,
        8.0308365e-03, -6.0011726e-03,  2.7884215e-03, -9.0097310e-04,
       -3.6469281e-03, -4.9611540e-03,  3.2116892e-03, -5.8744880e-03,
       -7.4882656e-03, -2.6531876e-03,  4.6564881e-03,  6.0474197e-03,
       -1.4392674e-03,  7.2922315e-03, -7.5974823e-03, -4.7306898e-03,
       -7.8458479e-03, -6.0295383e-03,  7.3002218e-03, -9.2860935e-03,
       -8.6746793e-03, -9.8077804e-03,  9.2405984e-03, -2.6607737e-03,
        1.0018528e-03, -6.1339838e-03, -2.5373432e-03,  7.3837597e-05,
       -2.5092696e-03, -7.3753358e-03, -6.1581051e-03,  2.5218294e-03,
       -9.5271850e-03,  1.1638446e-03, -1.9792211e-03,  8.3042653e-03,
       -4.1280867e-04,  6.2476173e-03, -7.0862165e-03, -7.5452477e-03,
        5.5703851e-03,  9.6137280e-04, -5.0635743e-03, -2.6140786e-03,
        8.7680016e-03, -9.1925366e-03, -3.6930521e-03,  9.7952993e-04,
       -2.2059453e-04,  9.5508620e-03, -1.9489959e-03, -7.0615942e-03,
       -7.7106743e-03, -9.9310102e-03, -7.7824220e-03, -6.0066171e-03,
       -5.9429798e-03, -1.3819491e-03, -5.1156618e-04, -4.6967301e-03,
        2.1470534e-03,  3.8890108e-03,  9.0729622e-03, -4.6295393e-03,
        5.1324181e-03,  7.2456789e-03, -3.7656163e-03, -7.4667842e-03],
      dtype=float32)
```

```
# Most similar words
similar = model.wv.most_similar('vikram')
similar
```



```
[('poverty', 0.20916083455085754),
 ('career', 0.19965867698192596),
 ('terms', 0.19484464824199677),
 ('free', 0.19072121381759644),
 ('confidence', 0.16293354332447052),
 ('greeks', 0.147495299577713),
 ('respect', 0.14436309039592743),
 ('people', 0.135737806558609),
 ('milestones', 0.13284234702587128),
 ('minds', 0.13203269243240356)]
```

Q41. What is word embedding?

A41. Word embedding is a technique which used to represent features in a unique way. It is basically converting word to numbers.

Q42. What is one-hot representation?

A42. This technique will be better understood with the help of an example. Firstly, we will initialize a dictionary of words,

|v|=10,000

Now, suppose we are representing the word “Man” in one hot representation. So, ”man” will be assigned a unique position in the dictionary. Let’s say “man” is stored at 5000 location in the dictionary.

So, in one-hot representation we will assign every word a special location in the dictionary.

Q43. What are the disadvantages of one-hot representation?

A43. As, we can see that the size of dictionary is 10,000, which means that every word will be having 10,000 dimensions, and if we have many words than all of them will be having same big matrix. So, it becomes very difficult to understand it. And when we apply, ml model on it then we will not be able to generalize the matrices easily. Due to which no similarity can be find out and no semantic will be found out. Due to these problems, our model will not be able to give results and accuracy.

Q44. How do we overcome the problems in one-hot representation?
A44. To overcome the problems in one-hot representation, there is new technique introduced word embedding. In word embedding, we assign every word a value according to the feature. We will see it in depth onwards.

Q45. How does Word embedding technique works?
A45. Suppose we have words “boy, girl, king, queen, apple, mango”, now we have to represent it in the form of word embedding. So, in word embedding we assign each word a value according to the feature present in that location. So, let me simplify it for you by making the table as an example,

features	boy	girl	king	queen	apple	Mango
Gender	-1	1	-0.92	0.92	0	0.1
Royal	0.01	0.02	0.95	0.96	-0.02	0.01
Age	0.03	0.02	0.7	0.6	0.95	0.92
Fruit	0	0.01	0.02	0.03	-1	1

So from the above table, we can see that according to the feature we have assigned the value to every word. We see if the feature is related to the word it will be having a high value and if not related it will be having low value almost 0. And also in above example table, I have taken only 4 features to make you understand the concept but in real they are around 300-n. You can choose them according to your needs.
The above representation is done with features, due to which we can find similarity between the words. And also now we have less dimensions and dense matrix.

Q46. How do we find relationship between words?

A46. Suppose we want to find the relationship between the words "boy" and "girl" on the feature "Gender". So, for that we will just subtract the matrices of both the words on a particular feature. We get the results as they both are having similar values, when we subtract them on any feature, than other all will almost be zero and that one will be 2 or ~2. We use cosine similarity to find the similar word. When we get the first matrix of boy, girl then we will compare it with king. If the distance is close then it is a similar word to king otherwise not.

When we convert the matrix of 300 dimensions to 2 dimensions, than we will see that similar words are placed close to each other.

Q47. What are the steps to implement word embedding in Keras?

A47.In keras, we have a feature called as an Embedding layer. But before passing to it we will do some preprocessing,

Suppose we have a sentence [Boy is Good]

Firstly, we will create a dictionary of words → |v|=10,000

Then we will do one-hot representation (convert word into dimensions) → [2000,4000,5000].

Now, we will pass it to embedding layer, in embedding layer we define the first parameter as "dimensions", in how much dimensions we want the output.

Q48. Why is RNN so popular?

A48. RNN is widely popular in the technical field because it has an internal memory due to which it can remember important things about the input they received, which allows them to be very precise in predicting what's coming next.

Q49. Why RNN is used in NLP?

A49. In NLP, we convert words to vectors so that our machine learning model can understand it. But when we apply techniques like BOW, TFIDF, Word2Vec to convert the word to vector. The sequence information is discarded. For example, there is a sentence "My name is Dev Chauhan", here we can see that sentence is making a sense and has a meaning but when the above techniques are applied the sequence information will be lost and sentence will not be meaningful.

Q50. What is forward propagation?

A50. In recurrent neural network, the forward propagation is done with respect to time. Due to which sequence information is kept. Suppose you have a sentence with 4 words. We will break the sentence into 4 parts/inputs.

$$X = (x1 , x2 , x3 , x4)$$

$$T = (t , t+1 , t+2 , t+3)$$

So, we will pass x1 to the hidden layer, with an input

At t=1, 01 = f(x1*weight + O0*weight').

In RNN when we start passing the inputs we provide an output to the input layer too with some weight. After passing it we will get the output O1. Now, when we pass the next input i.e. x2, we will pass it to the next hidden layer as

At t=2, 02 = f(x2*weight+O1*weight') .

And like this all the future inputs will be passed. Due to this, we have the information of previous output too in the current input and it will give better results. Similarly the output for further nodes are given below:

At t=3, O3 = f(x3*weight + O2*weight')

At t=4, O4 = f(x4*weight + O3*weight')

Now, the predicted result will be Y = f(O4*weight'')

(Here, in above equations f represents activation function)

So, we will compare the predicted output with the actual output and calculate the loss and try to minimize the loss with the help of optimizers.

Q51. What is backward propagation?

A51. After the forward propagation, we received a predicted output. Now we will compare the predicted output with the actual output .If the difference between the predicted output and actual output is a large value than we will try to minimize. To minimize the difference we use optimizers, the weights are now adjusted in such a manner that loss in minimum.

Q52. How are weights adjusted in backward propagation?

A52. The weights are adjusted with the help of optimizer function. We compare the predicted output with the actual output. We get a difference and according to this difference we adjust the weights such that this difference becomes minimum.

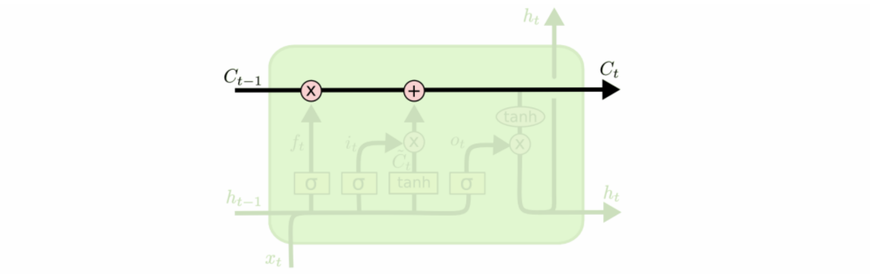
Q53. What are the different cells and steps in LSTM RNN?

A53. In RNN, we have different cells/states:

- Memory Cell
- Forget Cell
- Input Gate
- Output Gate

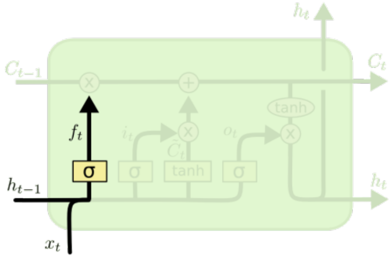
Q54. Explain memory cell?

A54. This cell is used for remembering and forgetting information based on the context of the input. Now, suppose we have a use case where we have to generate text, suppose we have our text as “Hi my name is dev and I am 21 years old”. Now I have to generate the text that will come after it. So, our RNN after reaching the end of the sentence will be referring to my name (noun). This is my context. And now if I want to change the context, my next sentence can be “My brother’s name is Kshitiz”. Now when I am training my RNN, it should forget some information about me and print some information about my brother



So, here “C(t-1)” is the input from previous state and is passed through the point wise operator with current input .(A point wise operator multiplies the corresponding values regarding the position, for example a=[1,2,3,4,5,6] , b=[1,1,0,0,0,1]. After point wise operation we will have the output as c=[1,2,0,0,0,6]). After passing through point wise operation we are having an output as [1,2,0,0,0,6], where “0” stands for the forgetting that particular information.

Q55. Explain forget cell/ gate?



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

Where, W(f) is the weight

H(t-1) is the previous output

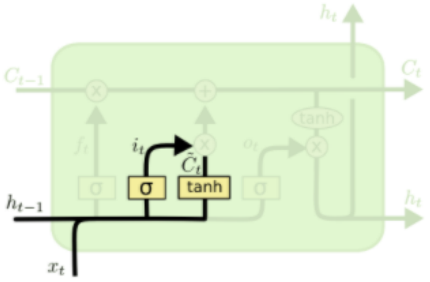
X(t) is the current input

B(f) is the bias

A55. After this, the above equation is passed through the sigmoid activation function which transforms it to 0-1. If the matrix contains more number of “0” then we say that there is a change of change of context, for example, previous output was “king” and current output is “poor”. If the matrix contains less number of “0” then we can say that there is no change in the context, for example, previous output was “king” and current output is “queen”. When we pass the matrix with more zeroes to the point wise operator, there will be more number of zeroes i.e. there will be more forgetting of data because the context has been changed. It will forget information (not all information).This state is called cell state.

Q56. What is input gate layer?

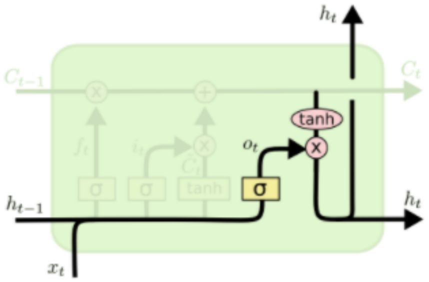
A56. This layer is used to decide what information is going to be stored in memory cell. Firstly, a sigmoid activation function is applied to decide which of the data will be updated. Secondly, we will pass the information through tanh function and then we will pass both the outputs through the point wise operator. And after that we will be adding that information to the memory cell.



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

Q57. What is output gate layer?

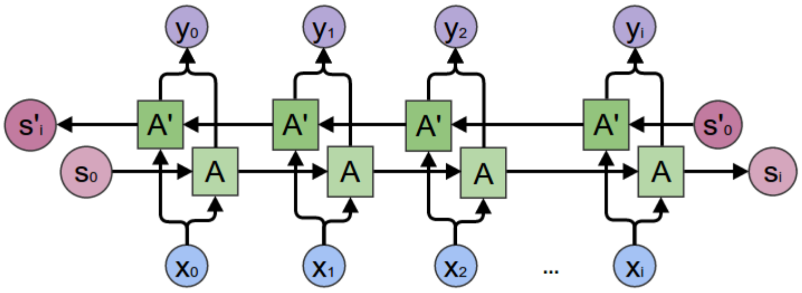
A57. Firstly, the concatenation of previous output and current input will pass through the sigmoid activation function. Secondly, whatever is present in the memory cell will be passed through a tanh function. After these two steps, they will be passed through point wise operator and this will be our output for this cell. This will be passed on to the next cell.



$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Q58. What is a bidirectional RNN?

A58. In a Bidirectional RNN, we have hidden layer for forward propagating RNN and a hidden layer for backward propagating RNN. Due to which the output at a single time is the combination of outputs from forward propagation and backward propagation.



Q59. How do we use bidirectional RNN?

A59. Suppose we have a RNN with hidden layers and we want to predict the output at t=t+2 , we will be having an input value from t,t+1 but what if the output of t+2 is dependent on t+3. So, we will be needing the output from t+3 too. But with unidirectional RNN we cannot achieve that. So, we will use bidirectional RNN, where we will setup a backward RNN, where number of hidden layers are same. The only difference is that, it is starting from the end to start. All the outputs of backward RNN are combined with the forward RNN of that particular time to predict the output.

Now, if we want to predict the output for t=t+2, it will be having the inputs from t, t+1 and t+3. Due to which the accuracy of predicted output will increase.

Q60. Where do we use Bidirectional RNN?

A60. We use the bidirectional RNN mostly in NLP. When we have to predict a word we will get the output from the past words as well as from the future words which will help to increase the accuracy.

Q61. Where does Bidirectional RNN do not work?

A61. The Bidirectional RNN will not work well with Speech Recognition because we may not get all the input at once. The input is provided with respect to time due to which our bidirectional RNN will lack.

Q62. What is the limitation of Bidirectional RNN?

A62. As we are using exact replica of hidden layers bidirectionally. So, it will take more time to compute the output. For example, when are using stacking hidden layers it will take more time to compute.

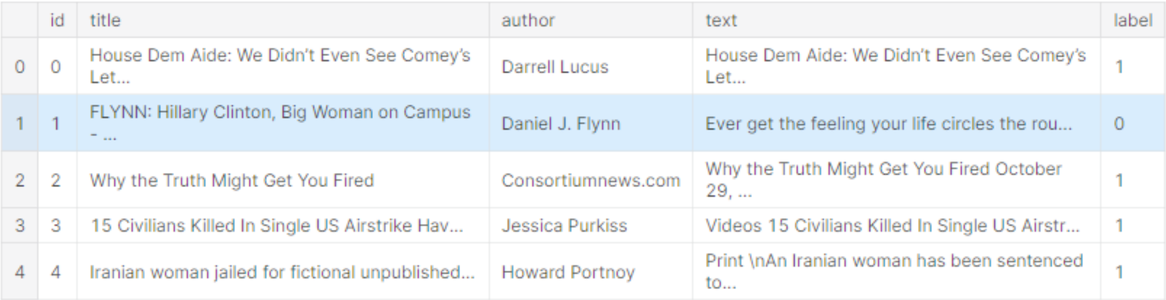
CODE FOR FAKE NEWS CLASSIFICATION USING LSTM

Problem: There are many fake news going on nowadays on social media. So we have a dataset where we have some texts, we have to classify which of them are real news and which of them are fake news.

Dataset: https://www.kaggle.com/saratchendra/fake-news

Code

```
# Importing dataset
df=pd.read_csv("/kaggle/input/fake-news/fake_train.csv")
df.head()
```

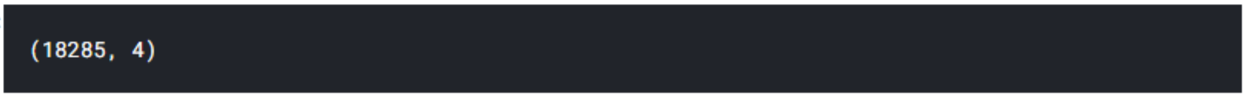


	id	title	author	text	label
0	0	House Dem Aide: We Didn't Even See Comey's Let...	Darrell Lucus	House Dem Aide: We Didn't Even See Comey's Let...	1
1	1	FLYNN: Hillary Clinton, Big Woman on Campus - ...	Daniel J. Flynn	Ever get the feeling your life circles the rou...	0
2	2	Why the Truth Might Get You Fired	Consortiumnews.com	Why the Truth Might Get You Fired October 29, ...	1
3	3	15 Civilians Killed In Single US Airstrike Hav...	Jessica Purkiss	Videos 15 Civilians Killed In Single US Airstr...	1
4	4	Iranian woman jailed for fictional unpublished...	Howard Portnoy	Print \nAn Iranian woman has been sentenced to...	1

```
#Removing all Null Values
df=df.dropna()
```

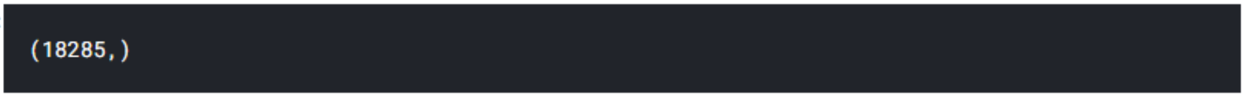
```
#Splitting the dataset as:
#independent feature
X=df.iloc[:,:-1]
#dependant feature
y=df.iloc[:,-1]
```

```
X.shape
```



```
(18285, 4)
```

```
y.shape
```



```
(18285,)
```

#Importing Libraries

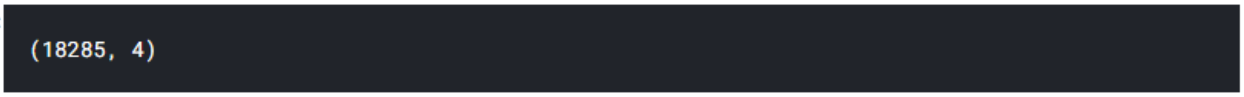
```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding
#pad_sequence is used to make sure that input length of every sentence is same
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.preprocessing.text import one_hot
from tensorflow.keras.layers import LSTM
from tensorflow.keras.layers import Dense
```

#Setting the vocabulary size

```
voc_size=5000
```

#Creating a copy so that any changes in copy does not affect the original data

```
messages = X.copy()
messages.shape
```



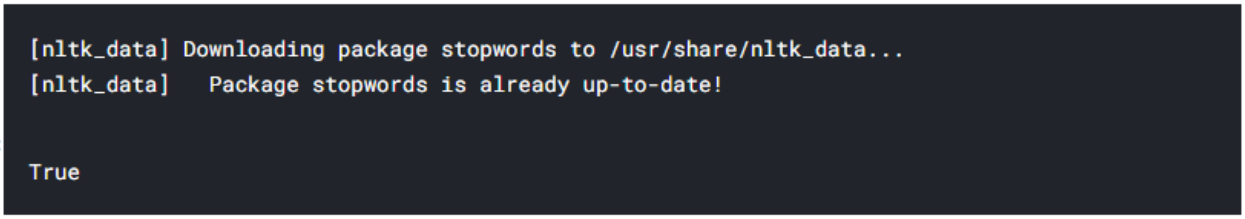
```
(18285, 4)
```

#resetting the index as we have dropped the nan values

```
messages.reset_index(inplace=True)
```

#Importing Some more Libraries

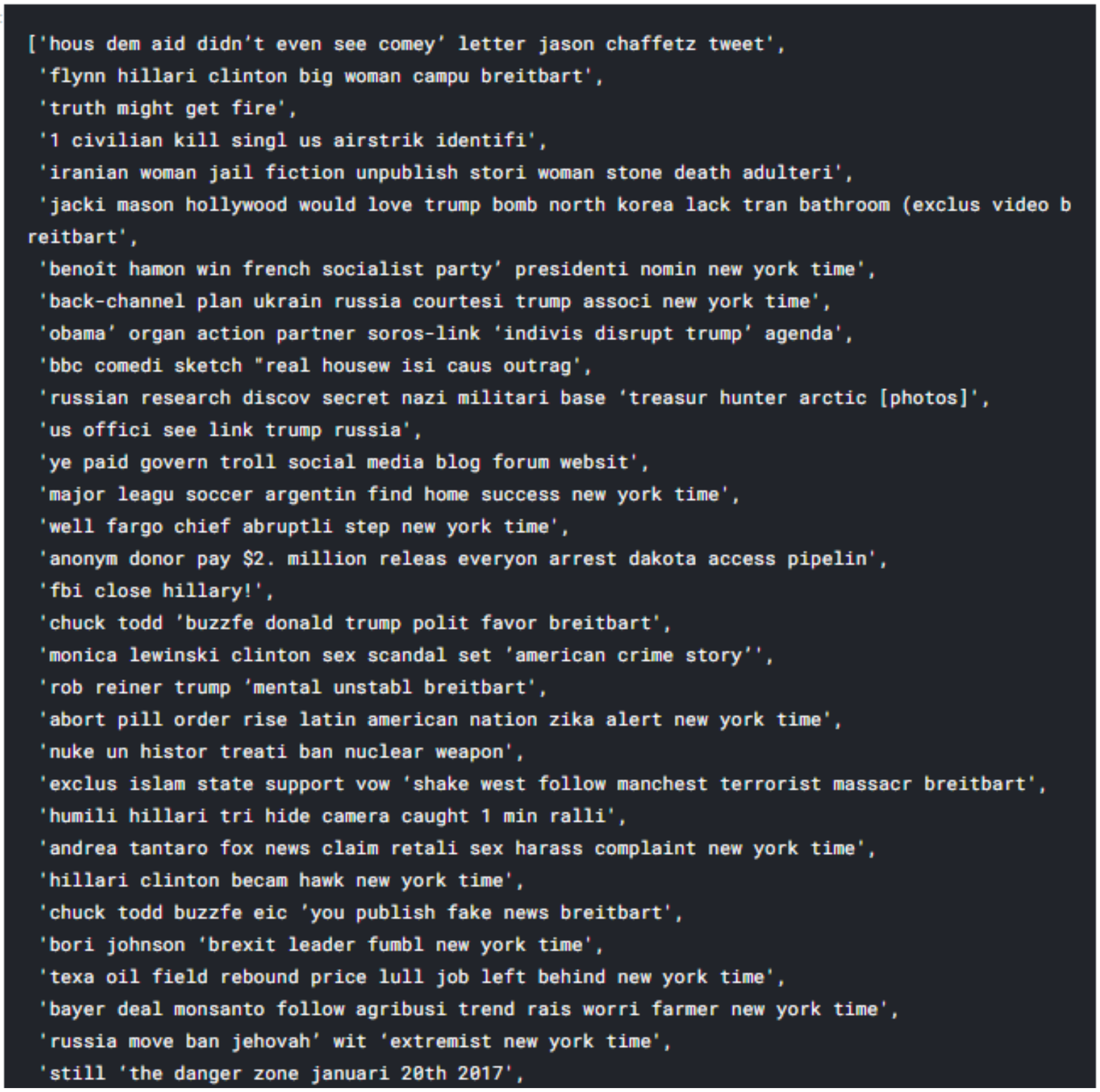
```
import nltk
import re
from nltk.corpus import stopwords

#Stopwords are the words that are not important in a sentence
nltk.download("stopwords")
```



```
[nltk_data] Downloading package stopwords to /usr/share/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!


True
```

#Data Preprocessing

```
from nltk.stem.porter import PorterStemmer
ps=PorterStemmer()
corpus=[]
for i in range(len(messages)):
   #subtituting everythin other than letter "a-z" and "A-Z" with a space
   review=re.sub("[^a-zA-Z] "," ",messages["title"][i])
   #converting all the words in lower case
   review=review.lower()
   review=review.split()

   review=[ps.stem(word) for word in review if not word in stopwords.words("english")]
   review=" ".join(review)
   corpus.append(review)
corpus
```



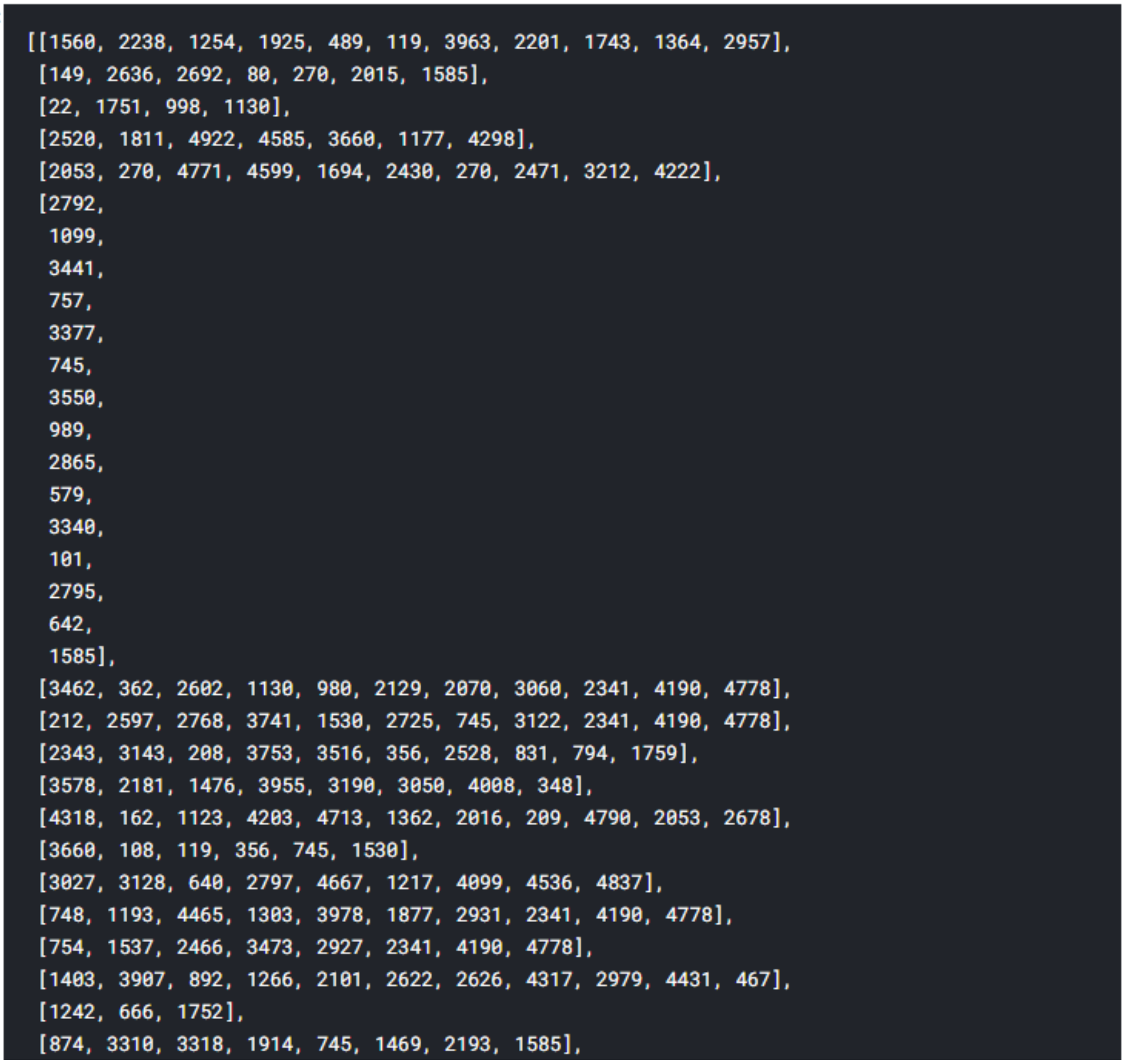
```
['hous dem aid didn’t even see comey’ letter jason chaffetz tweet',
 'flynn hillari clinton big woman campu breitbart',
 'truth might get fire',
 '1 civilian kill singl us airstrik identifi',
 'iranian woman jail fiction unpublish stori woman stone death adulteri',
 'jacki mason hollywood would love trump bomb north korea lack tran bathroom (exclus video b
reitbart',
 'benoît hamon win french socialist party’ presidenti nomin new york time',
 'back-channel plan ukrain russia courtesi trump associ new york time',
 'obama’ organ action partner soros-link ‘indivis disrupt trump’ agenda',
 'bbc comedi sketch "real housew isi caus outrag',
 'russian research discov secret nazi militari base ‘treasur hunter arctic [photos]',
 'us offici see link trump russia',
 'ye paid govern troll social media blog forum websit',
 'major leagu soccer argentin find home success new york time',
 'well fargo chief abruptli step new york time',
 'anonym donor pay $2. million releas everyon arrest dakota access pipelin',
 'fbi close hillary!',
 'chuck todd ’buzzfe donald trump polit favor breitbart',
 'monica lewinski clinton sex scandal set ’american crime story’',
 'rob reiner trump ‘mental unstabl breitbart',
 'abort pill order rise latin american nation zika alert new york time',
 'nuke un histor treati ban nuclear weapon',
 'exclus islam state support vow ‘shake west follow manchest terrorist massacr breitbart',
 'humili hillari tri hide camera caught 1 min ralli',
 'andrea tantaro fox news claim retali sex harass complaint new york time',
 'hillari clinton becam hawk new york time',
 'chuck todd buzzfe eic ’you publish fake news breitbart',
 'bori johnson ‘brexit leader fumbl new york time',
 'texa oil field rebound price lull job left behind new york time',
 'bayer deal monsanto follow agribusi trend rais worri farmer new york time',
 'russia move ban jehovah’ wit ‘extremist new york time',
 'still ‘the danger zone januari 20th 2017',
```

#One Hot Representation

#we will do one hot encoding for the corpus. It is alloting every word an index according to the vocabulary size

onehot=[one_hot(words,voc_size) for words in corpus]

onehot



```
[[1560, 2238, 1254, 1925, 489, 119, 3963, 2201, 1743, 1364, 2957],
 [149, 2636, 2692, 80, 270, 2015, 1585],
 [22, 1751, 998, 1130],
 [2520, 1811, 4922, 4585, 3660, 1177, 4298],
 [2053, 270, 4771, 4599, 1694, 2430, 270, 2471, 3212, 4222],
 [2792,
  1099,
  3441,
  757,
  3377,
  745,
  3550,
  989,
  2865,
  579,
  3340,
  101,
  2795,
  642,
  1585],
 [3462, 362, 2602, 1130, 980, 2129, 2070, 3060, 2341, 4190, 4778],
 [212, 2597, 2768, 3741, 1530, 2725, 745, 3122, 2341, 4190, 4778],
 [2343, 3143, 208, 3753, 3516, 356, 2528, 831, 794, 1759],
 [3578, 2181, 1476, 3955, 3190, 3050, 4008, 348],
 [4318, 162, 1123, 4203, 4713, 1362, 2016, 209, 4790, 2053, 2678],
 [3660, 108, 119, 356, 745, 1530],
 [3027, 3128, 640, 2797, 4667, 1217, 4099, 4536, 4837],
 [748, 1193, 4465, 1303, 3978, 1877, 2931, 2341, 4190, 4778],
 [754, 1537, 2466, 3473, 2927, 2341, 4190, 4778],
 [1403, 3907, 892, 1266, 2101, 2622, 2626, 4317, 2979, 4431, 467],
 [1242, 666, 1752],
 [874, 3310, 3318, 1914, 745, 1469, 2193, 1585],
```

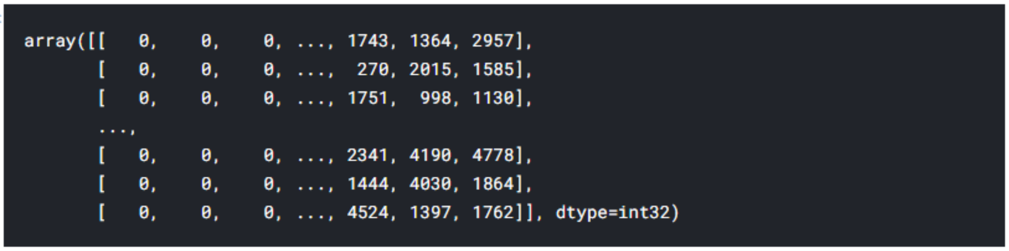
#Embedding Representation

sent_len=20

#if length of sentence is not 20 than it will ad 0 in front of sentence such that length becomes 20

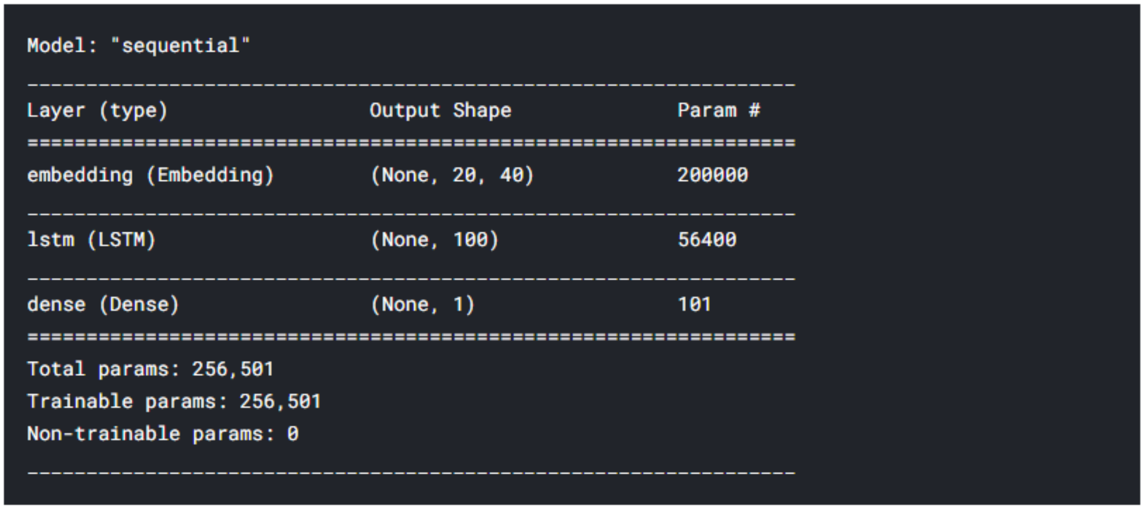
embedded_docs=pad_sequences(onehot,padding="pre",maxlen=sent_len)

embedded_docs



```
array([[   0,    0,    0, ..., 1743, 1364, 2957],
       [   0,    0,    0, ...,  270, 2015, 1585],
       [   0,    0,    0, ..., 1751,  998, 1130],
       ...,
       [   0,    0,    0, ..., 2341, 4190, 4778],
       [   0,    0,    0, ..., 1444, 4030, 1864],
       [   0,    0,    0, ..., 4524, 1397, 1762]], dtype=int32)
```

```
#Model Creation
embedding_vector_features=40
model=Sequential()
model.add(Embedding(voc_size,embedding_vector_features,input_length=sent_len))
model.add(LSTM(100))
model.add(Dense(1,activation="sigmoid"))
model.compile(loss="binary_crossentropy",optimizer="adam",metrics=["accuracy"])

model.summary()
```



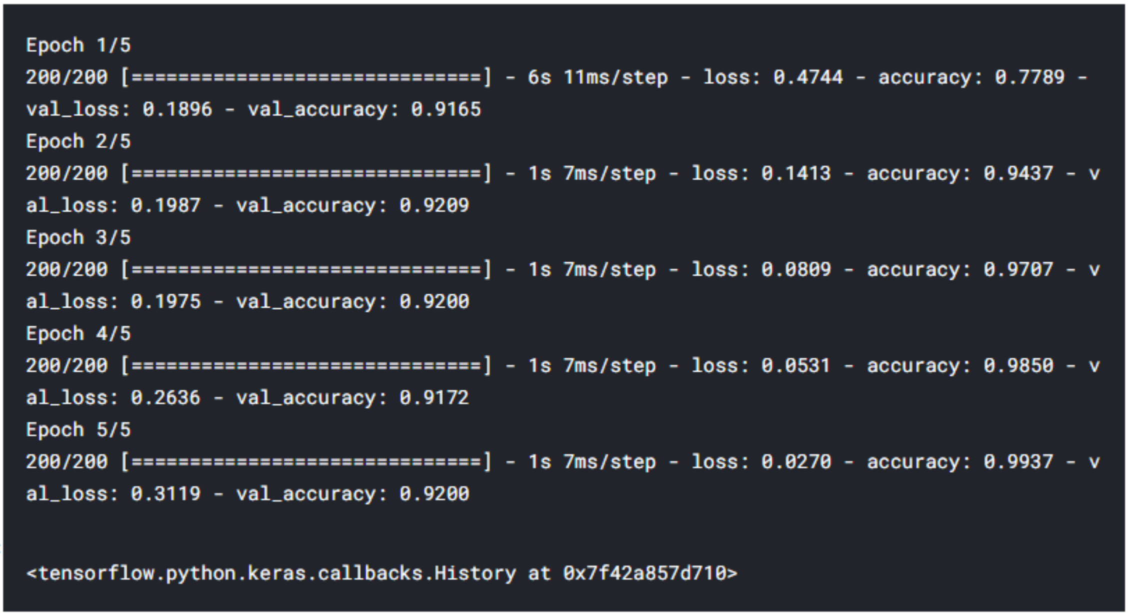
```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 20, 40)            200000
_________________________________________________________________
lstm (LSTM)                  (None, 100)               56400
_________________________________________________________________
dense (Dense)                (None, 1)                 101
=================================================================
Total params: 256,501
Trainable params: 256,501
Non-trainable params: 0
_________________________________________________________________
```

```
#Creating new independent and dependent variables
import numpy as np
X_final=np.array(embedded_docs)
y_final=np.array(y)
```

```
#Splitting the dataset into train and test set
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X_final,y_final,test_size=0.3,random_state=0)
```

#Model Training

```
model.fit(X_train,y_train,validation_data=(X_test,y_test),epochs=5,batch_size=64
)
```

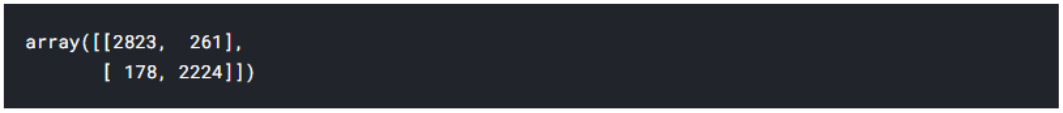


```
Epoch 1/5
200/200 [==============================] - 6s 11ms/step - loss: 0.4744 - accuracy: 0.7789 -
val_loss: 0.1896 - val_accuracy: 0.9165
Epoch 2/5
200/200 [==============================] - 1s 7ms/step - loss: 0.1413 - accuracy: 0.9437 - v
al_loss: 0.1987 - val_accuracy: 0.9209
Epoch 3/5
200/200 [==============================] - 1s 7ms/step - loss: 0.0809 - accuracy: 0.9707 - v
al_loss: 0.1975 - val_accuracy: 0.9200
Epoch 4/5
200/200 [==============================] - 1s 7ms/step - loss: 0.0531 - accuracy: 0.9850 - v
al_loss: 0.2636 - val_accuracy: 0.9172
Epoch 5/5
200/200 [==============================] - 1s 7ms/step - loss: 0.0270 - accuracy: 0.9937 - v
al_loss: 0.3119 - val_accuracy: 0.9200

<tensorflow.python.keras.callbacks.History at 0x7f42a857d710>
```

```
y_pred=model.predict_classes(X_test)
```

#Creating Confusion Matrix

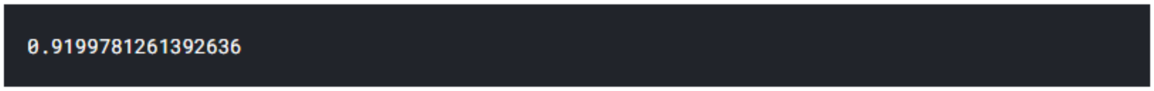
```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,y_pred)
```



```
array([[2823,  261],
       [ 178, 2224]])
```

#Accuracy

```
from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_pred)
```



```
0.9199781261392636
```

So by using LSTM model we are able to classify the fake and real news with a very high accuracy.