Application of Attention Based Models in a Text Summarizer – Generating Newspaper Headlines from Paragraphs

Enrollment No.	18104043	18104082	18104084
Name of Student	Priyanka Srivastava	Ishita Batra	Rudranil Ghosh

Name of Supervisor Dr. K. Vimal Kumar

December-2020

Report for Final Evaluation of Minor-I Project

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING & INFORMATION TECHNOLOGY
JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, NOIDA

Introduction

General Introduction

Newspapers play a significant role in our day to day life. In a news article, readers are attracted towards headlines. Headline creation is of utmost importance while preparing news as headlines are the first point of attraction for a reader. Our goal is to implement text summarization by generating headline for a news body using recurrent neural networks.

Headlines are the single most important factor when creating great content because headlines determine whether or not your target audience is going to read your article. A headline is the only impression you can make on an Internet surfer that can turn them into a potential reader.

An article, the first thing that catches users' attention is its headline. The more attractive the headline is, the more reader is bound to read it. If the article doesn't get a relevant headline, the user may not even consider reading the article, no matter how useful and knowledgeable the article might help it. So it is essential to choose a headline that effectively summarizes the whole article in a few words and also catches readers' attention. The aim of the project is to develop a generalized architecture that generates heading for the given text, which can be an article, a book's paragraph, newspaper contents, or any legal document. For this purpose, a model is proposed which uses an encoder-decoder recurrent neural network architecture consisting of LSTM and attention units.

Problem Statement

Given a news article text, we are going to summarize it and generate appropriate headlines using abstractive text summarisation

Whenever any media account shares a news story on Twitter or in any social networking site, they provide a crisp headlines /clickbait to make users click the link and read the article.

Often media houses provide sensational headlines that serve as a clickbait. This is a technique often employed to increase clicks to their site. We aim to curb this issue by generating genuine and apt headlines.

Our problem statement is to generate headlines given the article text/paragraphs or even generate relevant headlines from some given keywords, pertaining to real time data.

Process Outline

1. Data Preprocessing
   - Data Cleaning, handling missing values
   - Scraping real time data from multiple sources
2. Splitting the cleansed data into training and test set
3. Preprocess the extracted data and extract keywords
4. Building the model
5. Generate predicted headline
6. Build a web app prototype to demonstrate the same (if time permits)

Use Cases

1) **Headline generation helper:** Sometimes it is tricky for a person to think of a nice headline, this particular application could assist the user with the same, providing apt headlines. The more attractive the headline is, the more reader is bound to read it. If the article doesn't get a relevant headline, the user may not even consider reading the article, no matter how useful and knowledgeable the article might help it. So it is essential to choose a headline that effectively summarizes the whole article in a few words and also catches readers' attention. The aim of the project is to develop a generalized architecture that generates heading for the given text, which can be an article, a book's paragraph, newspaper contents, or any legal document.
2) **Attempt to curb click baits:** Social networks are generating huge amounts of complex textual data which is becoming increasingly difficult to process intelligently. Misinformation on social media networks, in the form of fake news, has the power to influence people, sway opinions and even have a decisive impact on elections. Clickbait does exactly this by using characteristics of natural language to entice users into clicking a link and can hence be classified as fake news
   There are misleading links which claim to talk about some news but end up describing something else. This application can prompt the user before clicking such links. (As it would be analyzing the text)

Background Study - Literature Survey

1. Summary of Papers

Paper 1

TITLE OF PAPER	Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond
AUTHORS	Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, Bing Xiang
YEAR OF PUBLICATION	2016
PUBLISHING DETAILS	Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning
SUMMARY	Abstractive text summarization is the task of generating a headline or a short summary consisting of a few sentences that captures the salient ideas of an article or a passage. Deep-learning based models that map an input sequence into another output sequence, called sequence-to-sequence models, have been successful in problems such as machine translation. In the framework of sequence-to-sequence models, a very relevant model to the task was the attentional Recurrent Neural Network (RNN) encoder-decoder model proposed, which produced state-of-the-art performance in machine translation (MT), which is also a natural language In this work, the authors have applied the attentional encoder-decoder for the task of abstractive summarization with very promising results, outperforming state-of-the-art results significantly on two different datasets. Each of the proposed novel models addressed a specific problem in abstractive summarization, yielding further improvement in performance.
Web Link	https://www.aclweb.org/anthology/K16-1028/

Table 1: Research Paper 1

Paper 2

TITLE OF PAPER	A Neural Attention Model for Abstractive Sentence Summarization
AUTHORS	Alexander M. Rush, Sumit Chopra, Jason Weston
YEAR OF PUBLICATION	2015
PUBLISHING DETAILS	Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing
SUMMARY	In this work, a fully data-driven approach to abstractive sentence summarization is proposed. The method utilizes a local attention-based model that generates each word of the summary conditioned on the input sentence. While the model is structurally simple, it can easily be trained end-to-end and scales to a large amount of training data. This probabilistic model is combined with a generation algorithm which produces accurate abstractive summaries.
Web Link	https://www.aclweb.org/anthology/D15-1044/

Table 2: Research Paper 2

Paper 3

TITLE OF PAPER	A Reinforced Topic-Aware Convolutional Sequence-to-Sequence Model for Abstractive Text Summarization
AUTHORS	Li Wang, Junlin Yao, Yunzhe Tao, Li Zhong, Wei Liu, Qiang Du
YEAR OF PUBLICATION	2018
PUBLISHING DETAILS	Published at Tencent AI Lab and Columbia University
SUMMARY	The paper proposed a deep learning approach to tackle the automatic summarization tasks by incorporating topic information into the convolutional sequence-to-sequence model and self-critical sequence training for optimization. In this work, the authors have proposed a model with reinforcement learning for abstractive text summarization. In addition, the model can produce summaries with better informativeness, coherence, and diversity.
Web Link	https://www.researchgate.net/publication/325053404_A_Reinforced_Topic-Aware_Convolutional_Sequence-to-Sequence_Model_for_Abstractive_Text_Summarization

Table 3: Research Paper 3

Paper 4

TITLE OF PAPER	Abstractive Sentence Summarization with Attentive Recurrent Neural Networks
AUTHORS	Sumit Chopra, Michael Auli, Alexander M. Rush
YEAR OF PUBLICATION	2016
PUBLISHING DETAILS	Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies
SUMMARY	The authors have used recurrent neural networks (RNN) to generate a summary of an input sentence. The conditioning is provided by a novel convolutional attention-based encoder which ensures that the decoder focuses on the appropriate input words at each step of generation. The RNN acts as a decoder to generate the summary of an input sentence, much like a standard recurrent language model. In addition, at every step the decoder also takes a conditioning input which is the output of an encoder module. Depending on the current state of the RNN, the encoder computes scores over the words in the input sentence.
Web Link	https://www.aclweb.org/anthology/N16-1012/

Table 4: Research Paper 4

Paper 5

TITLE OF PAPER	Effective Approaches to Attention-based Neural Machine Translation
AUTHORS	Minh-Thang Luong, Hieu Pham, Christopher D. Manning
YEAR OF PUBLICATION	2015
PUBLISHING DETAILS	Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing
SUMMARY	The authors focus on Neural machine translation(NMT) , that is often a large neural network that is trained in an end-to-end fashion and has the ability to generalize well to very long word sequences. Comparisons on various alignment functions have been made and this has shed light on which functions are best for which attentional models.
Web Link	https://arxiv.org/abs/1508.04025

Table 5: Research Paper 5

Paper 6

TITLE OF PAPER	Generating News Headlines with Recurrent Neural Networks
AUTHORS	Konstantin Lopyrev
YEAR OF PUBLICATION	2015
PUBLISHING DETAILS	Published at Standford.nlp.edu
SUMMARY	The author has described the application of an encoder-decoder recurrent neural network with LSTM units and attention to generating headlines from the text of news articles. The author has studied how the neural network decides which input words to pay attention to, and specifically identifies the function of the different neurons in a simplified attention mechanism. It has been observed that the network learns to detect linguistic phenomena, such as verbs, objects and subjects of a verb, ends of noun phrases, names, prepositions, negations, and so on.
Web Link	https://arxiv.org/abs/1512.01712

Table 6: Research Paper 6

Paper 7

TITLE OF PAPER	News Article Summarization with Attention-based Deep Recurrent Neural Networks
AUTHORS	Hujia Yu, Chang Yue, Chao Wang
YEAR OF PUBLICATION	2017
PUBLISHING DETAILS	Published at Stanford University
SUMMARY	This paper focuses on approaches to building a text automatic summarization model for news articles, generating a one-sentence summarization that mimics the style of a news title given some paragraphs. In this paper, the author has stressed over approaches to building a text automatic summarization model for news articles, and generating one-sentence summarization
Web Link	https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1174/reports/2746634.pdf

Table 7: Research Paper 7

Integrated Summary of the Literature Studied

Summarization is the task of condensing a piece of text to a shorter version, reducing the size of the initial text while at the same time preserving key informational elements and the meaning of content. Since manual text summarization is a time expensive and generally laborious task, the automatization of the task is gaining increasing popularity and therefore constitutes a strong motivation for academic research.

There are important applications for text summarization in various NLP related tasks such as text classification, question answering, legal texts summarization, news summarization, and headline generation.

Moreover, the generation of summaries can be integrated into these systems as an intermediate stage which helps to reduce the length of the document. The various techniques used for extraction processes include the use of boundary features, which is hard to be applied to complex images since they are too sensitive to unwanted edges.

Automatic text summarization is an exciting research area with several applications on the industry. By condensing large quantities of information into short, summarization can aid many downstream applications such as creating news digests, report generation, news summarization, and headline generation. There are two prominent types of summarization algorithms.

First, extractive summarization systems form summaries by copying and rearranging passages from the original text. Second, abstractive summarization systems generate new phrases, rephrasing or using words that were not in the original text. Due to the difficulty of abstractive summarization, the great majority of past work has been extractive.

The extractive approach is easier because copying large chunks of text from the source document ensures good levels of grammaticality and accuracy. On the other hand, sophisticated abilities that are crucial to high-quality summarization, such as paraphrasing, generalization, or the incorporation of real-world knowledge, are possible only in an abstractive framework. Even though abstractive summarization is a more challenging task, there has been a number of advances so far, thanks to recent developments in the deep learning area.

Neural Networks

Neural Networks are set of algorithms which closely resemble the human brain and are designed to recognize patterns. They interpret sensory data through a machine perception, labelling or clustering raw input. They can recognize numerical patterns, contained in vectors, into which all real-world data (images, sound, text or time series), must be translated. Artificial neural networks are composed of a large number of highly interconnected processing elements (neuron) working together to solve a problem

Recurrent Neural Networks(RNN)

Recurrent Neural Network is a generalization of feedforward neural network that has an internal memory. RNN is recurrent in nature as it performs the same function for every input of data while the output of the current input depends on the past one computation. After producing the output, it is copied and sent back into the recurrent network. For making a decision, it considers the current input and the output that it has learned from the previous input.

Unlike feedforward neural networks, RNNs can use their internal state (memory) to process sequences of inputs. This makes them applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition. In other neural networks, all the inputs are independent of each other. But in RNN, all the inputs are related to each other.

Long Short-Term Memory (LSTM) networks are a modified version of recurrent neural networks, which makes it easier to remember past data in memory. The vanishing gradient problem of RNN is resolved here. LSTM is well-suited to classify, process and predict time series given time lags of unknown duration. It trains the model by using back-propagation.

Text Summarisation

Text summarization refers to the technique of shortening long pieces of text. The intention is to create a coherent and fluent summary having only the main points outlined in the document.

Feature-Based Summarization: Elaborately speaking, the algorithm measures the frequency of words and phrases in the document and decides the importance of the sentence considering the words in the sentence and their frequencies. It states if the sentence has words with higher frequencies, It is important, but here we do not include the common words like “a”,” the”. Etc.

Extractive Summarizations: Extractive summarization techniques produce summaries by choosing a subset of the sentences in the original text

The Extractive Summarizers first create an ***intermediate representation*** that has the main task of highlighting or taking out the most important information of the text to be summarized based on the representations. There are two main types of representations:

1. **Topic representations:** It focuses on representing the topics represented in the texts. There are several kinds of approaches to get this representation. We here are going to talk about two of them. Others include **Latent Semantic Analysis and Bayesian Models.** If you want to study others as well I will encourage you to go through the references.

- **Frequency Driven Approaches**: In this approach, we assign weights to the words. If the word is related to the topic we assign 1 or else 0. The weights may be continuous depending on the implementation. Two common techniques for topic representations are:
- **Word Probability**: It simply uses the frequency of words as an indicator of the importance of the word. The probability of a word w is given by the frequency of occurrences of the word, f (w), divided by all words in the input which has a total of N words.

The use of texture features is able to detect the plate even if the boundary is deformed but is computationally complex. The use of character features is robust to rotation but is time consuming and produces errors when there is other text in the image. Most of these approaches model this problem as a classification problem which outputs whether to include a sentence in the summary or not. Other approaches have used topic information, Latent Semantic Analysis (LSA), Sequence to Sequence models, Reinforcement Learning and Adversarial processes.

To achieve even greater results , we can stack multiple RNN(LSTM or GRU or normal RNN) on top of each other, but we must take into consideration that they work with time.

Analysis, Design and Modelling

Neural networks are a specific set of algorithms that has revolutionized the field of machine learning. They are inspired by biological neural networks and the current so called deep neural networks have proven to work quite very well. This project would use **Recurrent Networks (RNNs) as they** have directed cycles in their connection graph. That means you we sometimes get back to where we started by following the arrows. However, They can have complicated dynamics and this can make them very difficult to train. For a project like this, Recurrent Neural networks were the first choice since there are a very natural way to model sequential data. They are equivalent to very deep nets with one hidden layer per time slice; except that they use the same weights at every time slice and they get input at every time slice.

There are mainly four types of summaries:

1. Single Document Summary: Summary of a Single Document
2. Multi-Document Summary: Summary from multiple documents
3. Query Focused Summary: Summary of a specific query
4. Informative Summary: It includes a summary of the full information.

Automatic text summarization is a common problem in machine learning and natural language processing (NLP). There are two approaches to this problem.

1. **Extractive Summarization**- Extractive text summarization done by picking up the most important sentences from the original text in the way that forms the final summary. We do some kind of extractive text summarization to solve our simple reading comprehension exercises. TextRank is a very popular extractive and unsupervised text summarization technique.

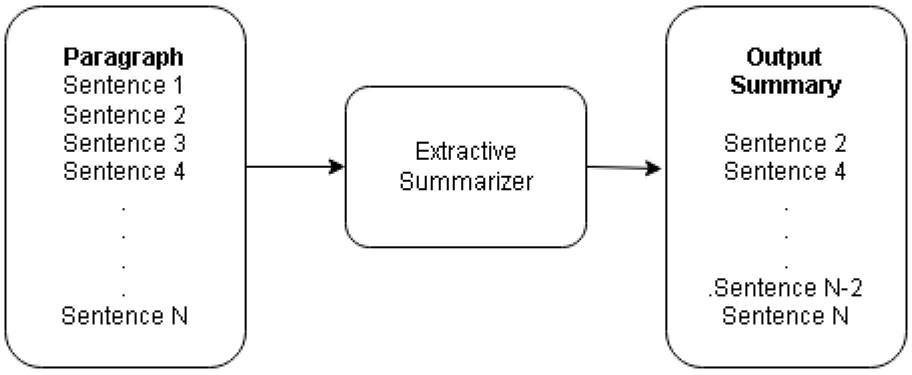


Fig 1: Extractive Text Summarisation – A visual (Created on draw.io)

2. **Abstractive Summarization**-Abstractive text summarization**,** on the other hand, is a technique in which the summary is generated by generating novel sentences by either rephrasing or using the new words, instead of simply extracting the important sentences. For example, some questions in the reading comprehension might not be straightforward in such cases we do rephrasing or use new words to answer such questions.

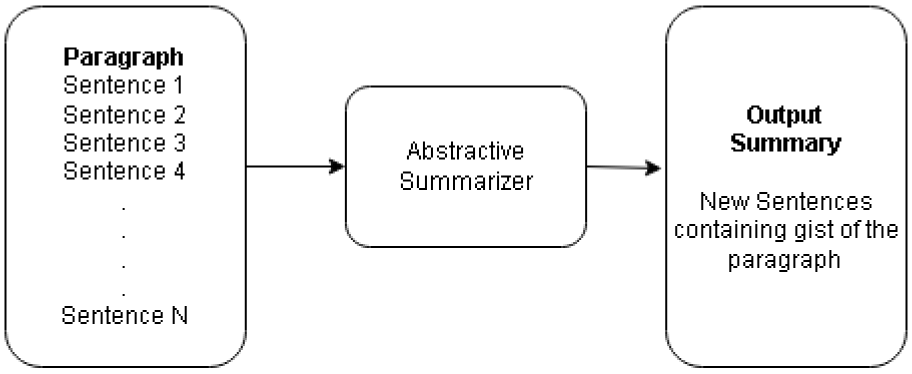


Fig 2: Abstractive Text Summarisation – A visual (Created on draw.io)

RNN Versus LSTM

RECURRENT NEURAL NETWORK (RNN)	LONG SHORT TERM MODEL (LSTM)
RNN is a generalization of feed forward neural network that has an internal memory.	LSTM networks are a modified version of recurrent neural networks, which makes it easier to remember past data in memory.
RNN is recurrent in nature as it performs the same function for every input of data while the output of the current input depends on the past one computation.	LSTM is well-suited to classify, process and predict time series given time lags of unknown duration. It trains the model by using back-propagation.
it is difficult to train standard RNNs to solve problems that require learning long-term temporal dependencies because the gradient of the loss function decays exponentially with time (called the vanishing gradient problem)	LSTM units include a 'memory cell' that can maintain information in memory for long periods of time.

The information retained from the last state is simply forwarded to the next state.	A set of gates is used to control when information enters the memory, when it's output, and when it's forgotten.
They only consist of hidden states which serve as the memory of RNNs.	They consist of hidden as well as cell states .

Table 8: A brief difference in RNN and LSTM models.

All RNNs have feedback loops in the recurrent layer. This lets them maintain information in 'memory' over time. But, it can be difficult to train standard RNNs to solve problems that require learning long-term temporal dependencies. This is because the gradient of the loss function decays exponentially with time (called the vanishing gradient problem). LSTM networks are a type of RNN that uses special units in addition to standard units. LSTM units include a 'memory cell' that can maintain information in memory for long periods of time. A set of gates is used to control when information enters the memory, when it's output, and when it's forgotten. This architecture lets them learn longer-term dependencies. GRUs are similar to LSTMs, but use a simplified structure. They also use a set of gates to control the flow of information, but they don't use separate memory cells, and they use fewer gates.

Standard RNNs (Recurrent Neural Networks) suffer from vanishing and exploding gradient problems. LSTMs (Long Short Term Memory) deal with these problems by introducing new gates, such as input and forget gates, which allow for a better control over the gradient flow and enable better preservation of “long-range dependencies”. The long range dependency in RNN is resolved by increasing the number of repeating layer in LSTM. LSTMs are often referred to as fancy RNNs. Vanilla RNNs do not have a cell state. They only have hidden states and those hidden states serve as the memory for RNNs.

One of the appeals of RNNs is the idea that they might be able to connect previous information to the present task, such as using previous video frames might inform the understanding of the present frame. If RNNs could do this, they’d be extremely useful. But can they? It depends.

Sometimes, we only need to look at recent information to perform the present task. For example, consider a language model trying to predict the next word based on the previous ones. If we are trying to predict the last word in “the clouds are in the *sky*,” we don’t need any further context – it’s pretty obvious the next word is going to be sky. In such cases, where the gap between the relevant information and the place that it’s needed is small, RNNs can learn to use the past information.

But there are also cases where we need more context. Consider trying to predict the last word in the text “I grew up in France I speak fluent *French*.” Recent information suggests that the next word is probably the name of a language, but if we want to narrow down which language, we need the context of France, from further back. It’s entirely possible for the gap between the relevant information and the point where it is needed to become very large.

Meanwhile, LSTM has both cell states and a hidden states. The cell state has the ability to remove or add information to the cell, regulated by "gates". And because of this "cell", in theory, LSTM should be able to handle the long-term dependency (in practice, it's difficult to do so.)

Long Short Term Memory networks – usually just called “LSTMs” – are a special kind of RNN, capable of learning long-term dependencies. They work tremendously well on a large variety of problems, and are now widely used.

LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behaviour, not something they struggle to learn! All recurrent neural networks have the form of a chain of repeating modules of neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer.

The cell state is kind of like a conveyor belt. It runs straight down the entire chain, with only some minor linear interactions. It’s very easy for information to just flow along it unchanged.

LSTMs were a big step in what we can accomplish with RNNs. It’s natural to wonder: is there another big step? A common opinion among researchers is: “Yes! There is a next step and it’s attention!” The idea is to let every step of an RNN pick information to look at from some larger collection of information. For example, if you are using an RNN to create a caption describing an image, it might pick a part of the image to look at for every word it outputs

Problem With Long Sequences

The encoder-decoder recurrent neural network is an architecture where one set of LSTMs learn to encode input sequences into a fixed-length internal representation, and second set of LSTMs read the internal representation and decode it into an output sequence.

This architecture has shown state-of-the-art results on difficult sequence prediction problems like text translation and quickly became the dominant approach.

The encoder-decoder architecture still achieves excellent results on a wide range of problems. Nevertheless, it suffers from the constraint that all input sequences are forced to be encoded to a fixed length internal vector.

This is believed to limit the performance of these networks, especially when considering long input sequences, such as very long sentences in text translation problems.

Attention within Sequences

Attention is the idea of freeing the encoder-decoder architecture from the fixed-length internal representation.

This is achieved by keeping the intermediate outputs from the encoder LSTM from each step of the input sequence and training the model to learn to pay selective attention to these inputs and relate them to items in the output sequence.

Put another way, each item in the output sequence is conditional on selective items in the input sequence.

Each time the proposed model generates a word in a translation, it (soft-)searches for a set of positions in a source sentence where the most relevant information is concentrated. The model then predicts a target word based on the context vectors associated with these source positions and all the previous generated target words, it encodes the input sentence into a sequence of vectors and chooses a subset of these vectors adaptively while decoding the translation. This frees a neural translation model from having to squash all the information of a source sentence, regardless of its length, into a fixed-length vector.

This increases the computational burden of the model, but results in a more targeted and better-performing model.

In addition, the model is also able to show how attention is paid to the input sequence when predicting the output sequence. This can help in understanding and diagnosing exactly what the model is considering and to what degree for specific input-output pairs.

The proposed approach provides an intuitive way to inspect the (soft-)alignment between the words in a generated translation and those in a source sentence. This is done by visualizing the annotation weights… Each row of a matrix in each plot indicates the weights associated with the annotations. From this we see which positions in the source sentence were considered more important when generating the target word.

The Concept of Attention

1. When humans read any lengthy paragraph, they pay attention to certain words then they change their attention to the next few words and so on.
2. Intuitively think of your teacher correcting your 10 mark answer in your History paper. The teacher would have a key with them in which certain important points /words are there. So in your answer sheet your teacher would look for these important words in the key. More attention is paid to the keywords.
3. Hence humans change their attention from one sequence of words to another sequence of words when given a lengthy input sequence
4. **So, instead of looking at all the words in the source sequence, we can increase the importance of specific parts of the source sequence that result in the target sequence.** This is the basic idea behind the attention mechanism.
5. Attention mechanism makes use of bidirectional RNNs The regular RNN is unidirectional as the sequence is processed from the first word to the last word. In bidirectional RNN we will have a connection in forward and reverse direction.
6. So in addition to the forward connection, there is also a backward connection for each of the neurons.

There are 2 different classes of attention mechanism depending on the way the attended context vector is derived:

- Global Attention-Here, the attention is placed on all the source positions. In other words, **all the hidden states of the encoder are considered for deriving the attended context vector.**
- Local Attention-Here, the attention is placed on only a few source positions. **Only a few hidden states of the encoder are considered for deriving the attended context vector.**

What is Attention?

“Can I have your Attention please!”. The introduction of the Attention Mechanism in deep learning has improved the success of various models in recent years and continues to be an omnipresent component in state-of-the-art models. Therefore, it is vital that it is understood how it goes about achieving its effectiveness and thus used in this project.

When it is thought about the English word “Attention”, it is known that it means directing focus at something and taking greater notice. The Attention mechanism in Deep Learning is based off this concept of directing the focus, and it pays greater attention to certain factors when processing the data. In broad terms, Attention is one component of a network’s architecture, and is in charge of managing and quantifying the interdependence between the input and output elements or within the input elements. There are two different major types of Attention:

Bahdanau Attention	Luong Attention
It is commonly referred to as Additive Attention, came from a paper by Dzmitry Bahdanau.	It is often referred to as Multiplicative Attention and was built on top of the Attention mechanism proposed by Bahdanau.
It takes concatenation of forward and backward source hidden state.	It uses top hidden layer states in both of encoder and decoder.
But in the Bahdanau at time t we consider about t-1 hidden state of the decoder. Then we calculate alignment, context vectors as above. But then we concatenate this context with hidden state of the decoder at t-1.	In Luong attention they get the decoder hidden state at time t. Then calculate attention scores and from that get the context vector which will be concatenated with hidden state of the decoder and then predict.
Bahdanau has only concatenated score alignment model.	Luong has different types of alignments

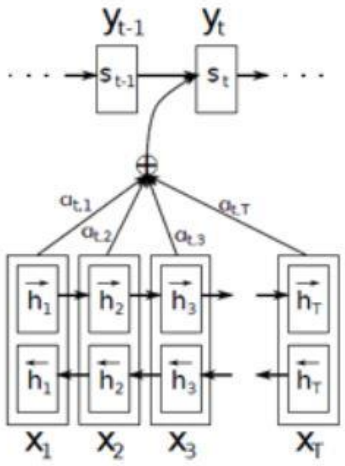
Importance of Attention based models in an abstractive summarizer?

The attention mechanism was employed for neural machine translation before being utilized for NLP tasks such as text summarization. A basic encoder-decoder architecture may fail when given long sentences since the size of encoding is fixed for the input string; thus, it cannot consider all the elements of a long input. To remember the input that has a significant impact on the summary, the attention mechanism was introduced. The attention mechanism is employed at each output word to calculate the weight between the output word and every input word; the weights add to one. The advantage of using weights is to show which input word must receive attention with respect to the output word. The weighted average of the last hidden layers of the decoder in the current step is calculated after passing each input word and fed to the SoftMax layer along the last hidden layers.

Lopyrev proposed a simplified attention mechanism that was utilised in an encoder-decoder RNN to generate headlines for news articles. The news article was fed into the encoder one word at a time and then passed through the embedding layer to generate the word representation. The experiments were conducted using simple and complex attention mechanisms. In the simple attention mechanism, the last layer after processing the input in the encoding was divided into two parts: one part for calculating the attention weight vector, and one part for calculating the context vector. However, in the complex attention mechanism, the last layer was employed to calculate the attention weight vector and context vector without fragmentation. In both figures, the solid lines indicate the part of the hidden state of the last layer that is employed to compute the context vector, while the dashed lines indicate the part of the hidden state of the last layer that is applied to compute the attention weight vector. The same difference was seen on the decoder side: in the simple attention mechanism, the last layer was divided into two parts (one part was passed to the SoftMax layer, and the other part was applied to calculate the attention weight), while in the complex attention mechanism, no such division was made. A beam search at the decoder side was performed during testing to extend the sequence of the probability.

Advantages of Attention in general?

Recurrent Neural Networks (RNNs) are powerful neural network architectures used for modeling sequences. LSTM (Long Short-Term Memory) based RNNs are surprisingly good at capturing long-term dependencies in the sequences. A barebones sequence-to-sequence/encoder-decoder architecture performs incredibly well in tasks like Machine Translation. This is the diagram of the Attention model shown in Bahdanau’s paper.

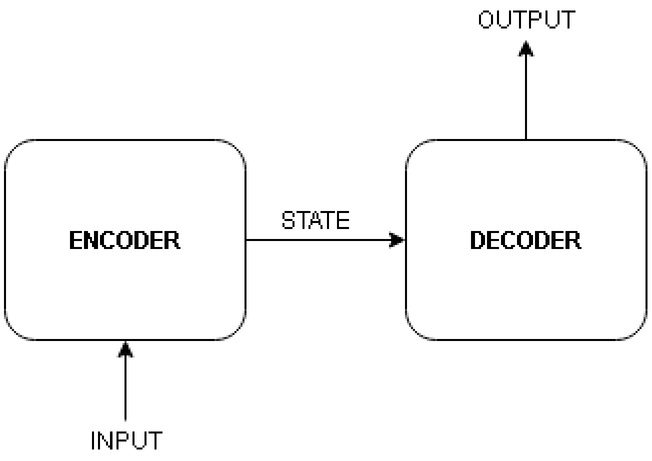


A typical sequence-sequence architecture consists of an encoder and a decoder RNN. The encoder processes a source sequence and reduces it into a fixed length vector – the context, and the decoder generates a target sequence, token by token, conditioned on the context. The context is usually the final state of the encoder RNN. The final state of the encoder RNN ultimately decides the decoding process, or at least heavily influences it, while the previous states of the RNN, {h0, h1, . . h5} do not have any influence over the decoding process. Next observation is that the final encoder state, a fixed length vector, represents the essence or the meaning of the source sequence in the context of translation. When the source sequence is too long and contains multiple information-rich phrases apart from each other, the burden of capturing and condensing the information into a fixed length vector representation becomes too much for the encoder to bear. This inability leads to loss of information which gets reflected in the generated target sequence.

Thus, we need a mechanism to allow the decoder to selectively (dynamically) focus on the information-rich phrases in the source sequence. The decoder would take a peek at the encoder states that are relevant to the current decoding step. This relevance sorting is precisely what the Attention Mechanism does by allowing the decoder to pay attention to different parts of the source sequence at different decoding steps.

What is seq2seq learning with Attention?

Recurrent Neural Networks (or more precisely LSTM/GRU) have been found to be very effective in solving complex sequence related problems given a large amount of data. They have real time applications in speech recognition, Natural Language Processing (NLP) problems, time series forecasting, etc. Sequence to Sequence (often abbreviated to seq2seq) models are a special class of Recurrent Neural Network architectures typically used (but not restricted) to solve complex Language related problems like Machine Translation, Question Answering, creating Chat-bots or as used in this project - Text Summarization. Attention was originally introduced as a solution to address the main issue surrounding seq2seq models, and to great success. Seq2seq model is also known as the Encoder-Decoder model.



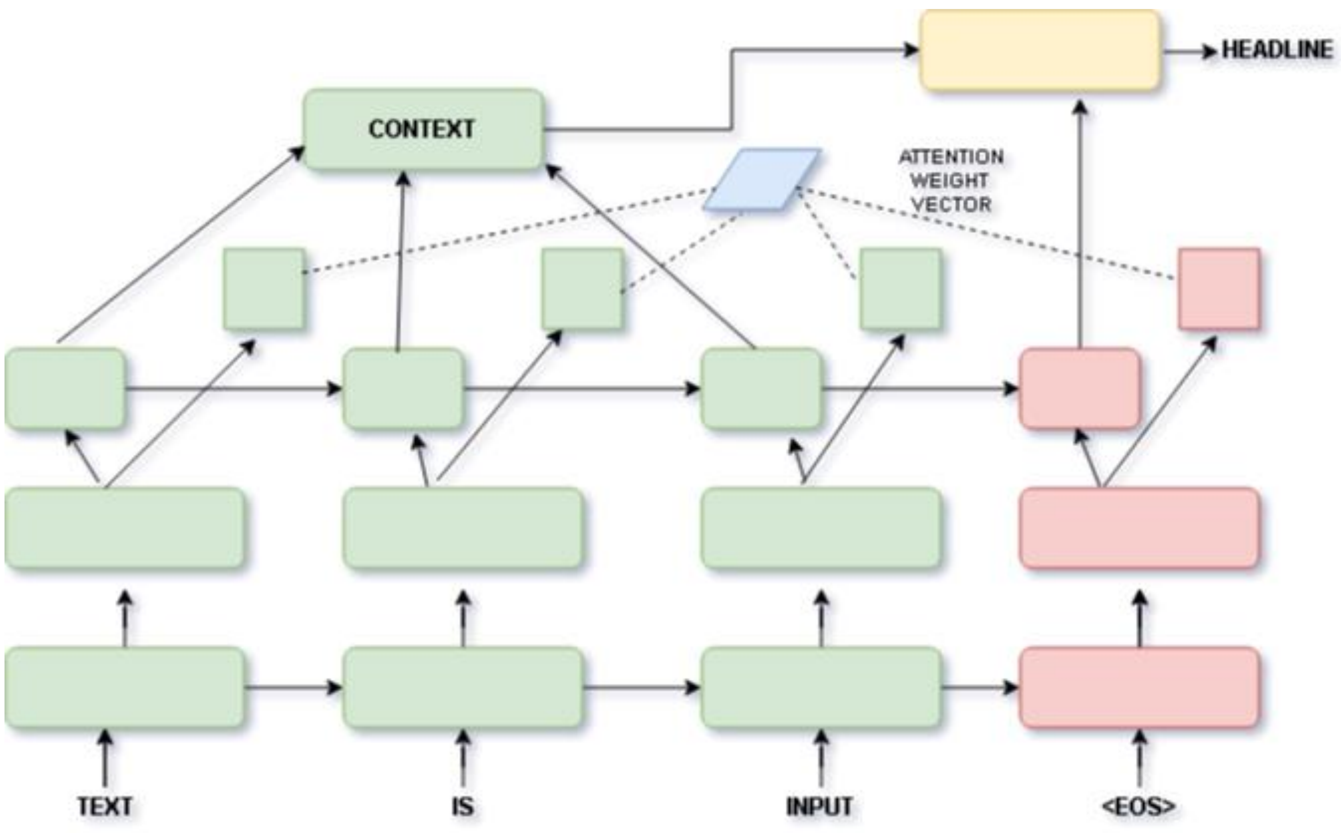
The standard seq2seq model is generally unable to accurately process long input sequences, since only the last hidden state of the encoder RNN is used as the context vector for the decoder. On the other hand, the Attention Mechanism directly addresses this issue as it retains and utilizes all the hidden states of the input sequence during the decoding process. It does this by creating a unique mapping between each time step of the decoder output to all the encoder hidden states. This means that for each output that the decoder makes, it has access to the entire input sequence and can selectively pick out specific elements from that sequence to produce the output.
Therefore, the mechanism allows the model to focus and place more “Attention” on the relevant parts of the input sequence as needed.

Model Architecture

For implementing our neural network we would use *keras* library of python along with the TensorFlow backend and other libraries required for data pre-processing and designing. After loading the data we use the contraction mapping in order to deal with the contracted words of the language. It helps in better understanding(intended meaning) and improved accuracy of the model.

Moving on to the data pre-processing step, data cleaning is implemented along with removal of stop-words in order to make data ready for our model. This phase is implemented separately for both summary and complete text of the training model. In the next phase model building is implemented. 3 Layers of LSTM are added(can be increased or decreased accordingly) along with encoder and decoder. It is followed by attention layer and finally we complete the headline prediction from the paragraphs.

Our model follows the sequence-to-sequence learning framework with attention. It has three components: an encoder network, a decoder network, and an attention network as shown in the figure.



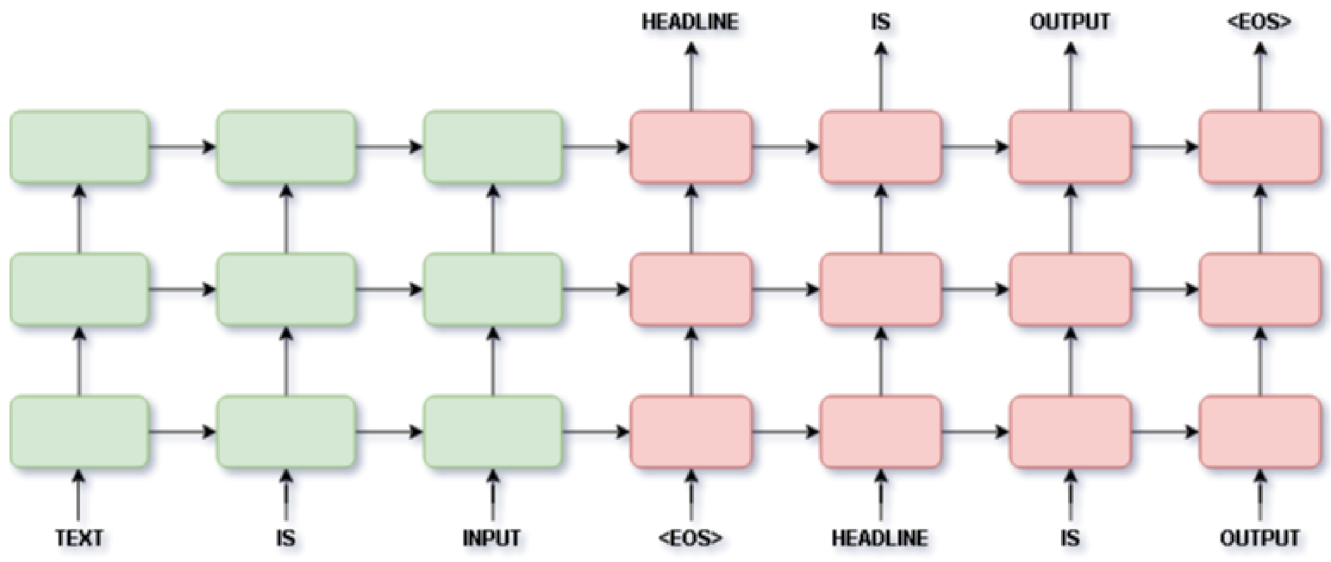


Fig 3: Model Architecture for Attention based models.

(Inspired from: https://arxiv.org/abs/1512.01712, created on draw.io)

Deep learning architectures/models to be used

Recurrent Neural Networks: Humans don't start their thinking from scratch every second. As you read this essay, you understand each word based on your understanding of previous words. You don't throw everything away and start thinking from scratch again. Your thoughts have persistence.Traditional neural networks can't do this, and it seems like a major shortcoming. For example, imagine you want to classify what kind of event is happening at every point in a movie. It's unclear how a traditional neural network could use its reasoning about previous events in the film to inform later ones.

Recurrent neural networks address this issue. They are networks with loops in them, allowing information to persist. A **recurrent neural network** (**RNN**) is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. This allows it to exhibit temporal dynamic behavior. Derived from feedforward neural networks, RNNs can use their internal state (memory) to process variable length sequences of inputs. he main specialty in RNNs is the ability to model short term dependencies. This is due to the hidden state in the RNN. It retains information from one time step to another flowing through the unrolled RNN units. Each unrolled RNN unit has a hidden state. The current time steps hidden state is calculated using information of the previous time step's hidden state and the current input. This process helps to retain information on what the model saw in the previous time step when processing the current time steps information.

Now, the loops make recurrent neural networks seem kind of mysterious. However, if we think a bit more, it turns out that they aren't all that different than a normal neural network. A recurrent neural network can be thought of as multiple copies of the same network, each passing a message to a successor. These loops make recurrent neural networks seem kind of mysterious. However, if you think a bit more, it turns out that they aren't all that different than a normal neural network. A recurrent neural network can be thought of as multiple copies of the same network, each passing a message to a successor.

One of the appeals of RNNs is the idea that they might be able to connect previous information to the present task, such as using previous video frames might inform the understanding of the present frame. If RNNs could do this, they'd be extremely useful. But can they? It depends.

Sometimes, we only need to look at recent information to perform the present task. For example, consider a language model trying to predict the next word based on the previous ones. If we are trying to predict the last word in "the clouds are in the *sky*," we don't need any further context – it's pretty obvious the next word is going to be sky. In such cases, where the gap between the relevant information and the place that it's needed is small, RNNs can learn to use the past information.

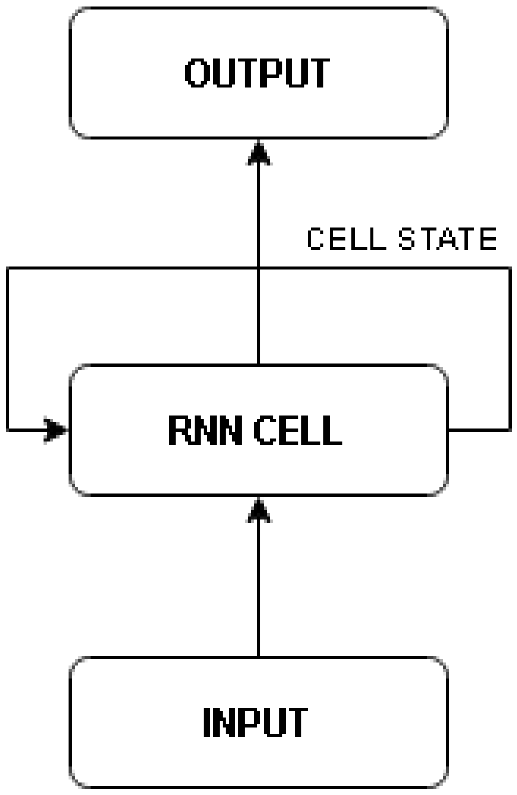


Fig 4: Rolled version of an RNN (Created on draw.io)

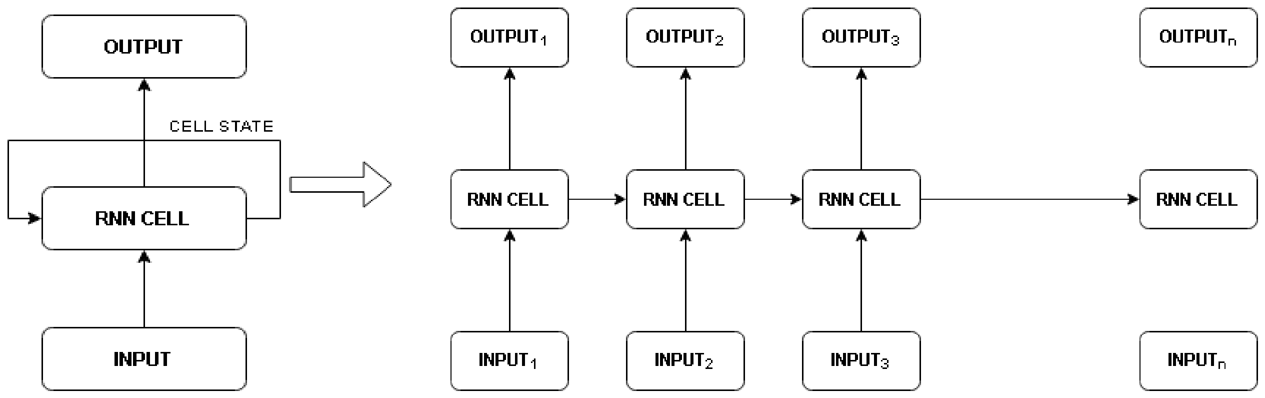


Fig. 5: An unrolled RNN (Created on draw.io)

Seq-2Seq Model – Now the question is, What if the input x is a sequence? What if x is a sequence of words. In most languages sequence of words matters a lot. We need to somehow preserve the sequence of words.The core idea here is if output depends on a sequence of inputs then we need to build a new type of neural network which gives importance to sequence information, which somehow retains and leverages the sequence information. Seq2Seq is a method of encoder-decoder based machine translation that maps an input of sequence to an output of sequence with a tag and attention value.

The idea is to use 2 RNNs that will work together with a special token and trying to predict the next state sequence from the previous sequence. Seq2Seq Model is a kind of model that use Encoder and a Decoder on top of the model.

We use embedding, so we have to first compile a “vocabulary” list containing all the words we want our model to be able to use or read. The model inputs will have to be tensors containing the IDs of the words in the sequence.

The Encoder will encode the sentence word by words into an indexed of vocabulary or known words with index, and the decoder must generate each word in the output sequence given two sources of information: **Context Vector**: The encoded representation of the source document provided by the encoder, and G**enerated Sequence**: The word or sequence of words already generated as a summary.

The context vector may be a fixed-length encoding as in the simple Encoder-Decoder architecture, or may be a more expressive form filtered via an attention mechanism.

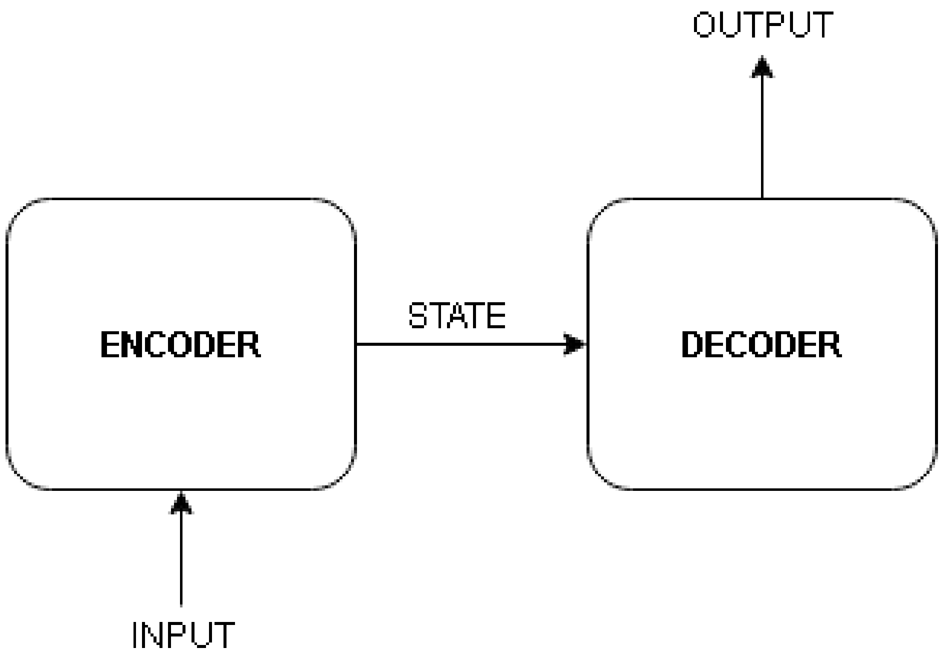


Fig 6: Basic Functioning of an Encoder-Decoder (Created on draw.io)

Long Short-Term Memory (LSTM): Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems. Long Short Term Memory networks – usually just called "LSTMs" – are a special kind of RNN, capable of learning long-term dependencies. They work tremendously well on a large variety of problems, and are now widely used.

LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behaviour, not something they struggle to learn. All recurrent neural networks have the form of a chain of repeating modules of neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer.

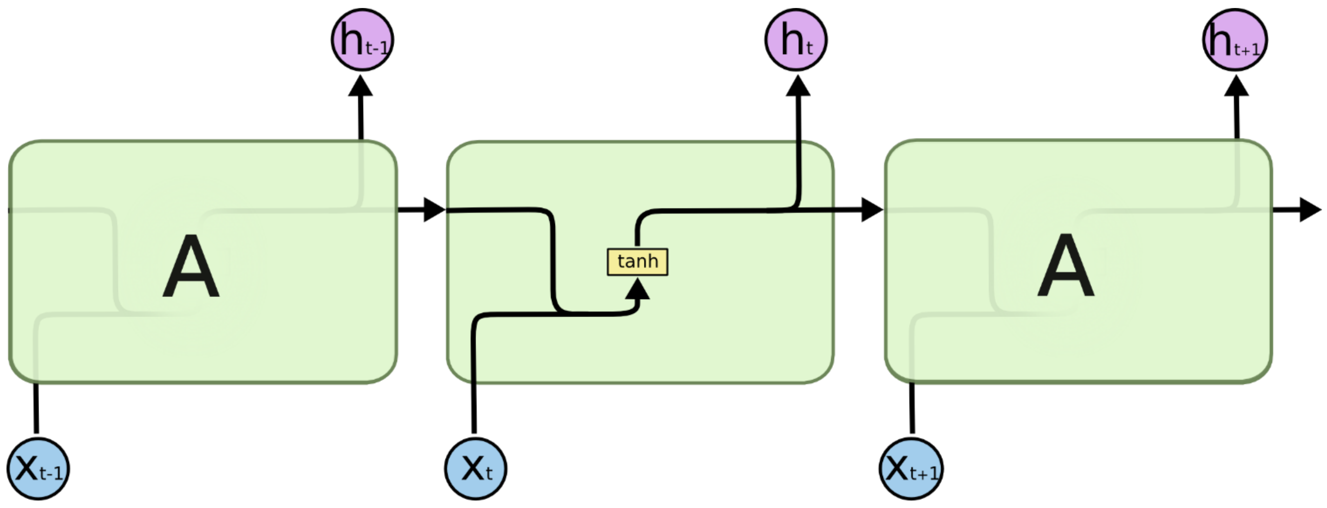


Fig 7: The repeating module in a standard RNN contains a single layer. (Source – Hackernoon)

LSTMs also have this chain like structure, but the repeating module has a different structure. Instead of having a single neural network layer, there are four, interacting in a very special way.

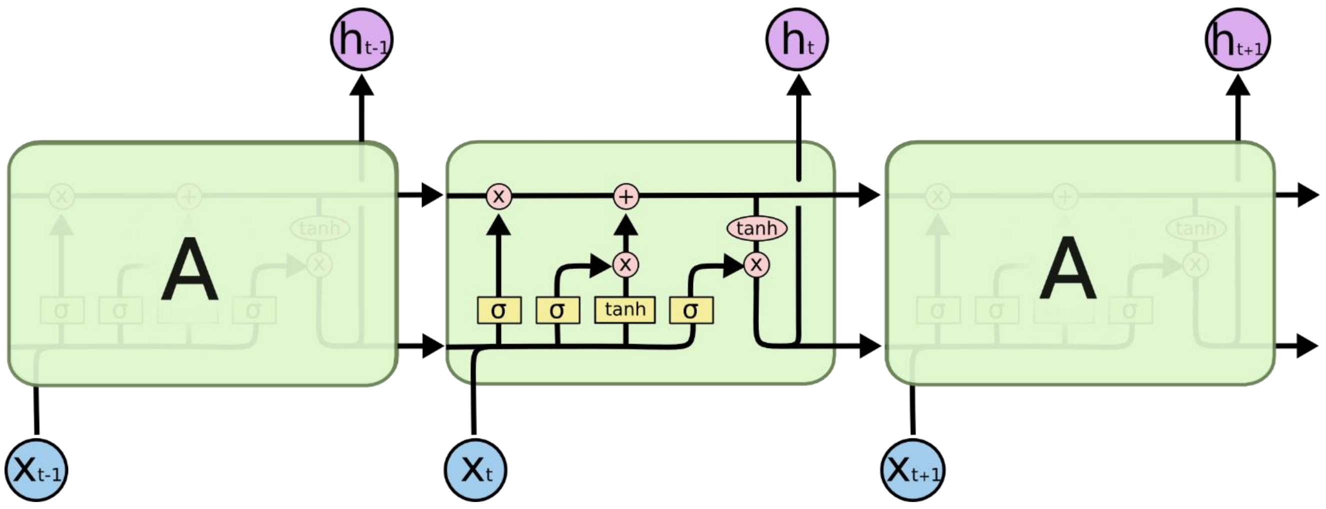


Fig 8: The repeating module in an LSTM contains four interacting layers. (Source – Hackernoon)

Tools and Technologies

Python: It is an interpreted, high-level, general programming language. It is powerful, fast and open source. With a rich support for tons of libraries and frameworks related to Deep Learning & Computer Vision.

Matplotlib/Seaborn: Theis would the primary plotting library for visualization of results, data (Images) and it’s general statistics.

Recurrence Neural Networks: A recurrent neural network (RNN) is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. This allows it to exhibit temporal dynamic behavior.

NLP: Natural language processing (NLP) is a subfield of linguistics, computer science, and artificial intelligence concerned with the interactions between computers and human language, in particular how to program computers to process and analyze large amounts of natural language data.

NLTK: Natural Language Toolkit is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum.

Keras: Keras is a neural networks library written in Python that is high-level in nature – which makes it extremely simple and intuitive to use. It works as a wrapper to low-level libraries like TensorFlow or Theano high-level neural networks library, written in Python that works as a wrapper to TensorFlow or Theano.

Genism: A Python library for topic modelling, document indexing and similarity retrieval with large corpora. Target audience is the natural language processing (NLP) and information retrieval (IR) community.

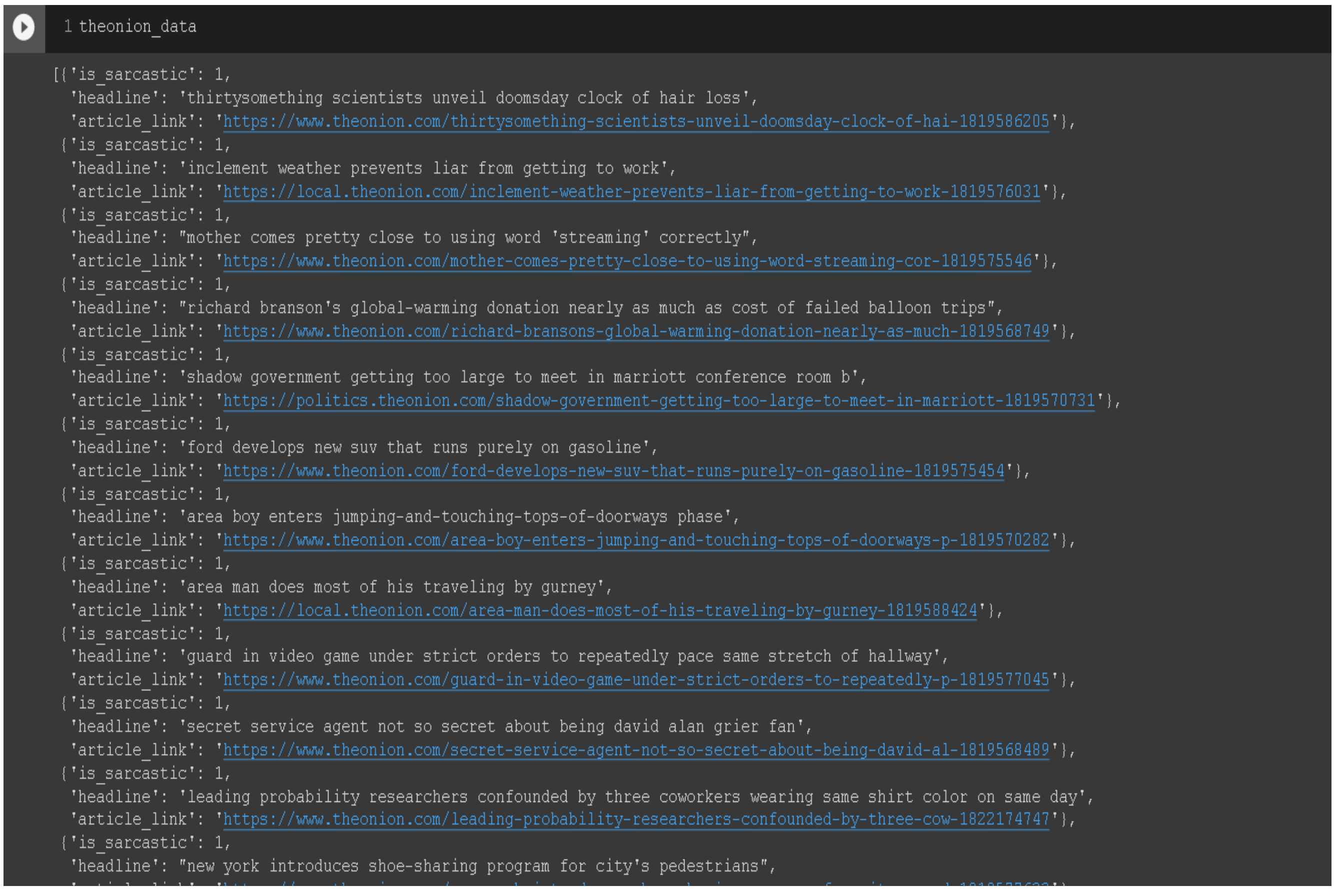
Newspaper3k: Newspaper is a python library for extracting & curating articles. Newspaper has a seamless language extraction and detection. If no language is specified, Newspaper will attempt to auto detect a language.

Pickle: Python pickle module is used for serializing and de-serializing python object structures. The process to converts any kind of python objects (list, dict, etc.) into byte streams (0s and 1s) is called pickling or serialization or flattening or marshalling.

Word Embeddings: An **embedding** is a relatively low-dimensional space into which you can translate high-dimensional vectors. Embeddings make it easier to do machine learning on large inputs like sparse vectors representing words. Word embeddings are computed by applying dimensionality reduction techniques to datasets of co-occurrence statistics between words in a corpus of text.

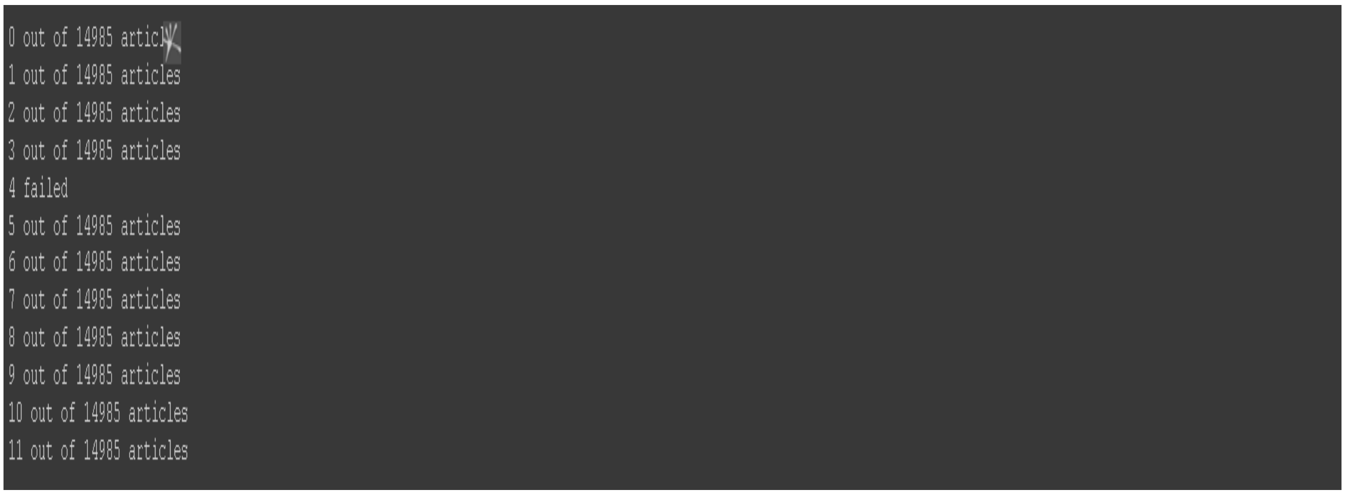
Tokenization Any Machine learning or deep learning model cannot understand text directly. We need to convert it into numbers. We can do this through tokenization. These pieces are called **tokens**. For example, we can divide a chunk of text into words, or we can divide it into sentences. A tokenizer builds the vocabulary and converts a word sequence to an integer sequence.

Screenshots of the outputs



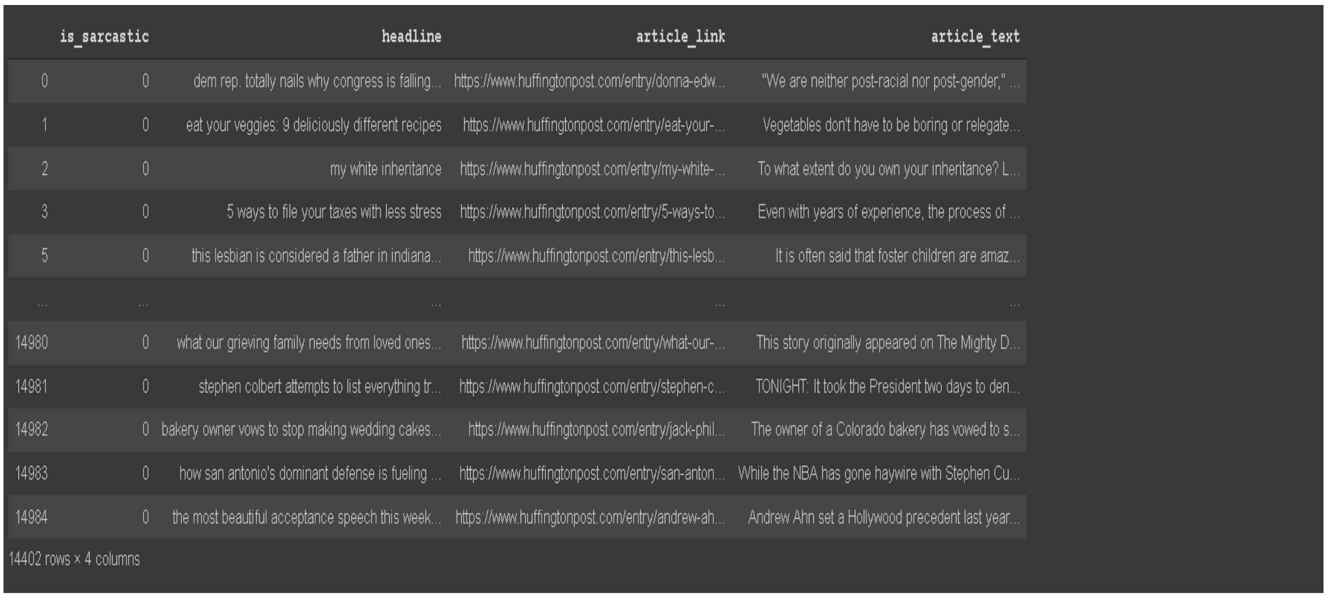
```
theonion_data
```

```
[{'is_sarcastic': 1,
  'headline': 'thirtysomething scientists unveil doomsday clock of hair loss',
  'article_link': 'https://www.theonion.com/thirtysomething-scientists-unveil-doomsday-clock-of-hai-1819586205'},
 {'is_sarcastic': 1,
  'headline': 'inclement weather prevents liar from getting to work',
  'article_link': 'https://local.theonion.com/inclement-weather-prevents-liar-from-getting-to-work-1819576031'},
 {'is_sarcastic': 1,
  'headline': "mother comes pretty close to using word 'streaming' correctly",
  'article_link': 'https://www.theonion.com/mother-comes-pretty-close-to-using-word-streaming-cor-1819575546'},
 {'is_sarcastic': 1,
  'headline': "richard branson's global-warming donation nearly as much as cost of failed balloon trips",
  'article_link': 'https://www.theonion.com/richard-bransons-global-warming-donation-nearly-as-much-1819568749'},
 {'is_sarcastic': 1,
  'headline': 'shadow government getting too large to meet in marriott conference room b',
  'article_link': 'https://politics.theonion.com/shadow-government-getting-too-large-to-meet-in-marriott-1819570731'},
 {'is_sarcastic': 1,
  'headline': 'ford develops new suv that runs purely on gasoline',
  'article_link': 'https://www.theonion.com/ford-develops-new-suv-that-runs-purely-on-gasoline-1819575454'},
 {'is_sarcastic': 1,
  'headline': 'area boy enters jumping-and-touching-tops-of-doorways phase',
  'article_link': 'https://www.theonion.com/area-boy-enters-jumping-and-touching-tops-of-doorways-p-1819570282'},
 {'is_sarcastic': 1,
  'headline': 'area man does most of his traveling by gurney',
  'article_link': 'https://local.theonion.com/area-man-does-most-of-his-traveling-by-gurney-1819588424'},
 {'is_sarcastic': 1,
  'headline': 'guard in video game under strict orders to repeatedly pace same stretch of hallway',
  'article_link': 'https://www.theonion.com/guard-in-video-game-under-strict-orders-to-repeatedly-p-1819577045'},
 {'is_sarcastic': 1,
  'headline': 'secret service agent not so secret about being david alan grier fan',
  'article_link': 'https://www.theonion.com/secret-service-agent-not-so-secret-about-being-david-al-1819568489'},
 {'is_sarcastic': 1,
  'headline': 'leading probability researchers confounded by three coworkers wearing same shirt color on same day',
  'article_link': 'https://www.theonion.com/leading-probability-researchers-confounded-by-three-cow-1822174747'},
 {'is_sarcastic': 1,
  'headline': "new york introduces shoe-sharing program for city's pedestrians",
```



```
0 out of 14985 articles
1 out of 14985 articles
2 out of 14985 articles
3 out of 14985 articles
4 failed
5 out of 14985 articles
6 out of 14985 articles
7 out of 14985 articles
8 out of 14985 articles
9 out of 14985 articles
10 out of 14985 articles
11 out of 14985 articles
```

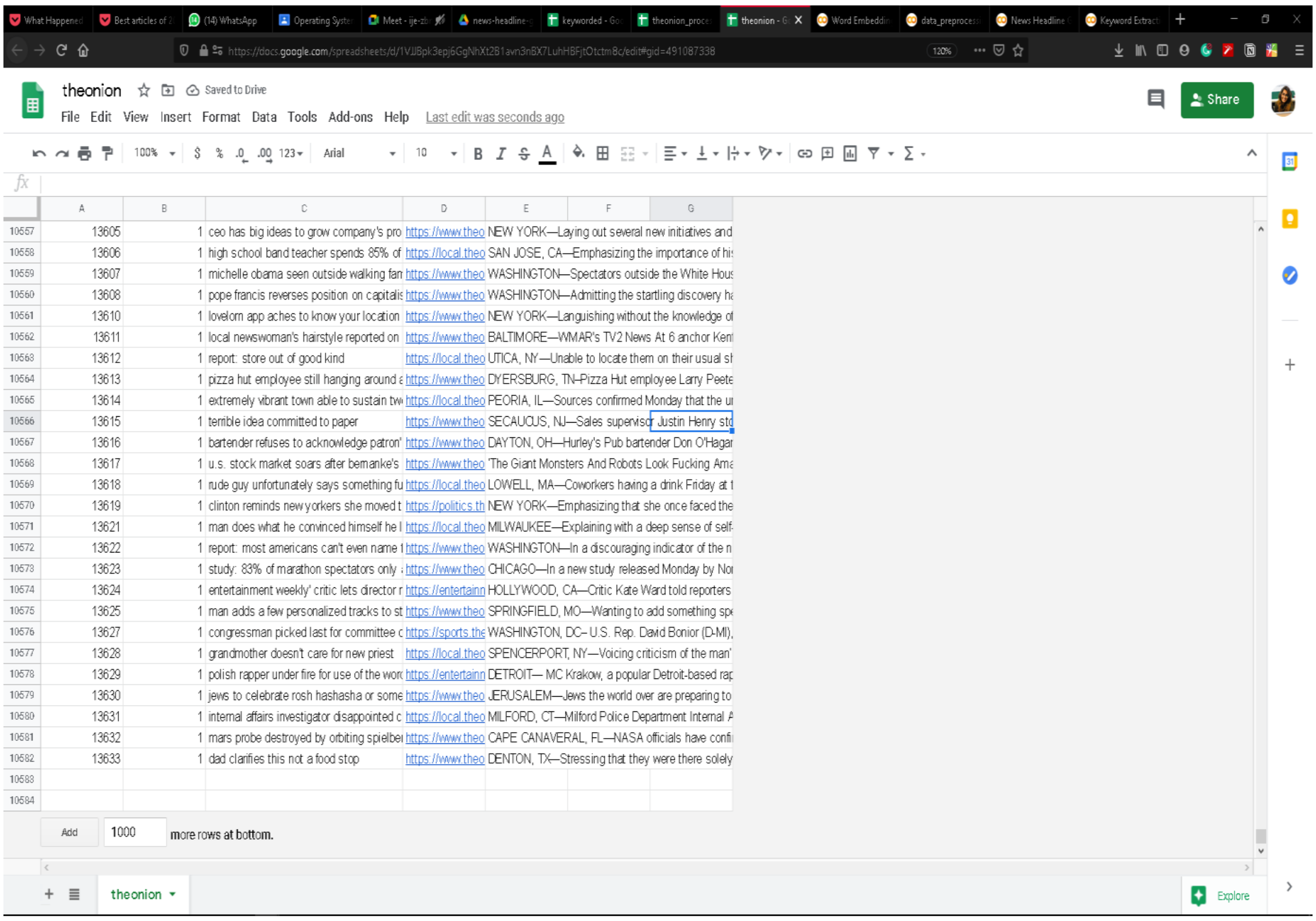
The Onion Scraping



	is_sarcastic	headline	article_link	article_text
0	0	dem rep. totally nails why congress is falling...	https://www.huffingtonpost.com/entry/donna-edw...	"We are neither post-racial nor post-gender," ...
1	0	eat your veggies: 9 deliciously different recipes	https://www.huffingtonpost.com/entry/eat-your-...	Vegetables don't have to be boring or relegate...
2	0	my white inheritance	https://www.huffingtonpost.com/entry/my-white-...	To what extent do you own your inheritance? L...
3	0	5 ways to file your taxes with less stress	https://www.huffingtonpost.com/entry/5-ways-to...	Even with years of experience, the process of ...
5	0	this lesbian is considered a father in indiana...	https://www.huffingtonpost.com/entry/this-lesb...	It is often said that foster children are amaz...
...	...	...	...	...
14980	0	what our grieving family needs from loved ones...	https://www.huffingtonpost.com/entry/what-our-...	This story originally appeared on The Mighty D...
14981	0	stephen colbert attempts to list everything tr...	https://www.huffingtonpost.com/entry/stephen-c...	TONIGHT: It took the President two days to den...
14982	0	bakery owner vows to stop making wedding cakes...	https://www.huffingtonpost.com/entry/jack-phil...	The owner of a Colorado bakery has vowed to s...
14983	0	how san antonio's dominant defense is fueling ...	https://www.huffingtonpost.com/entry/san-anton...	While the NBA has gone haywire with Stephen Cu...
14984	0	the most beautiful acceptance speech this week...	https://www.huffingtonpost.com/entry/andrew-ah...	Andrew Ahn set a Hollywood precedent last year...

14402 rows × 4 columns

Huffington Post

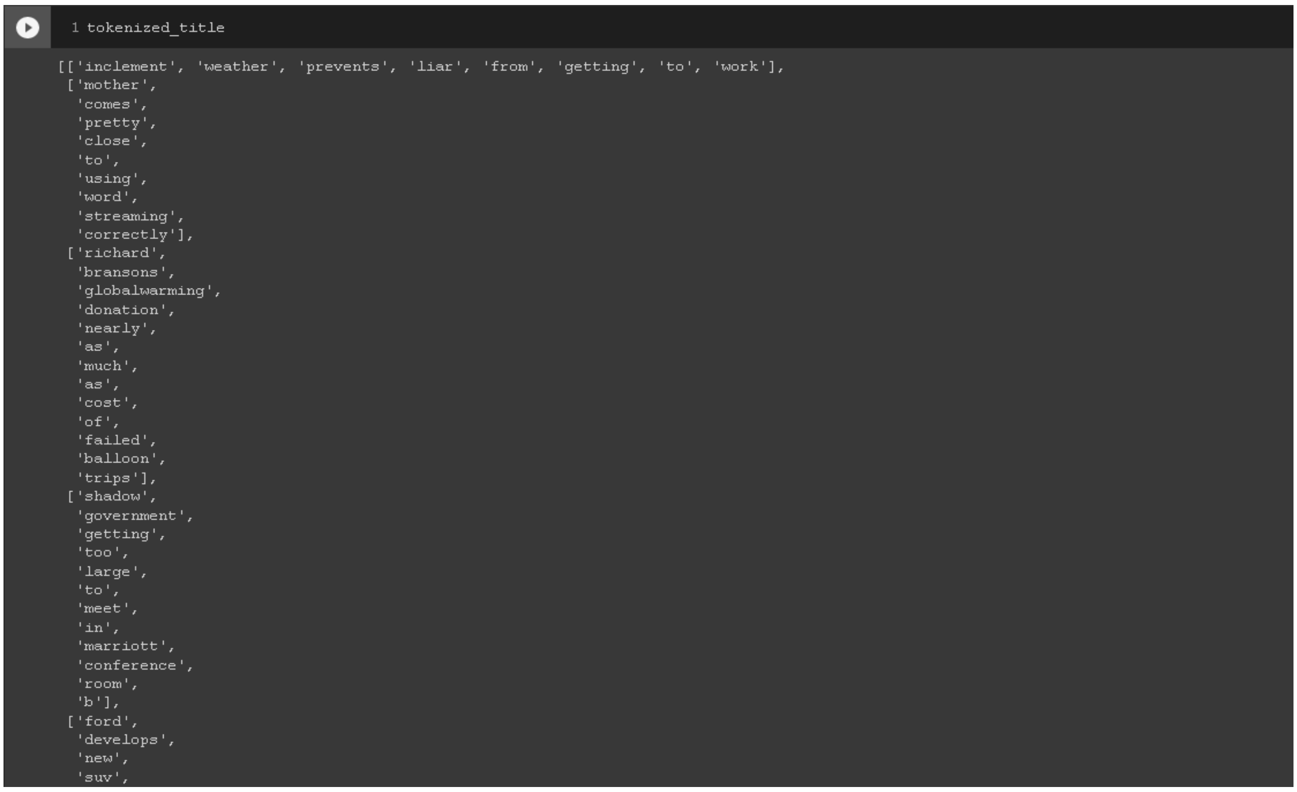


theonion

	A	B	C	D	E	F	G
10557	13605	1	ceo has big ideas to grow company's pro	https://www.theo	NEW YORK—Laying out several new initiatives and		
10558	13606	1	high school band teacher spends 85% of	https://local.theo	SAN JOSE, CA—Emphasizing the importance of his		
10559	13607	1	michelle obama seen outside walking fan	https://www.theo	WASHINGTON—Spectators outside the White Hous		
10560	13608	1	pope francis reverses position on capitalis	https://www.theo	WASHINGTON—Admitting the startling discovery ha		
10561	13610	1	lovelorn app aches to know your location	https://www.theo	NEW YORK—Languishing without the knowledge of		
10562	13611	1	local newswoman's hairstyle reported on	https://www.theo	BALTIMORE—WMAR's TV2 News At 6 anchor Kent		
10563	13612	1	report: store out of good kind	https://local.theo	UTICA, NY—Unable to locate them on their usual sh		
10564	13613	1	pizza hut employee still hanging around a	https://www.theo	DYERSBURG, TN—Pizza Hut employee Larry Peete		
10565	13614	1	extremely vibrant town able to sustain tw	https://local.theo	PEORIA, IL—Sources confirmed Monday that the ur		
10566	13615	1	terrible idea committed to paper	https://www.theo	SECAUCUS, NJ—Sales supervisor Justin Henry sto		
10567	13616	1	bartender refuses to acknowledge patron'	https://www.theo	DAYTON, OH—Hurley's Pub bartender Don O'Hagar		
10568	13617	1	u.s. stock market soars after bernanke's	https://www.theo	'The Giant Monsters And Robots Look Fucking Ama		
10569	13618	1	rude guy unfortunately says something fu	https://local.theo	LOWELL, MA—Coworkers having a drink Friday at t		
10570	13619	1	clinton reminds new yorkers she moved t	https://politics.th	NEW YORK—Emphasizing that she once faced the		
10571	13621	1	man does what he convinced himself he l	https://local.theo	MILWAUKEE—Explaining with a deep sense of self-		
10572	13622	1	report: most americans can't even name t	https://www.theo	WASHINGTON—In a discouraging indicator of the n		
10573	13623	1	study: 83% of marathon spectators only :	https://www.theo	CHICAGO—In a new study released Monday by Nor		
10574	13624	1	entertainment weekly' critic lets director r	https://entertainn	HOLLYWOOD, CA—Critic Kate Ward told reporters		
10575	13625	1	man adds a few personalized tracks to st	https://www.theo	SPRINGFIELD, MO—Wanting to add something spe		
10576	13627	1	congressman picked last for committee c	https://sports.the	WASHINGTON, DC– U.S. Rep. David Bonior (D-MI),		
10577	13628	1	grandmother doesn't care for new priest	https://local.theo	SPENCERPORT, NY—Voicing criticism of the man'		
10578	13629	1	polish rapper under fire for use of the worc	https://entertainn	DETROIT— MC Krakow, a popular Detroit-based rap		
10579	13630	1	jews to celebrate rosh hashasha or some	https://www.theo	JERUSALEM—Jews the world over are preparing to		
10580	13631	1	internal affairs investigator disappointed c	https://local.theo	MILFORD, CT—Milford Police Department Internal A		
10581	13632	1	mars probe destroyed by orbiting spielber	https://www.theo	CAPE CANAVERAL, FL—NASA officials have confi		
10582	13633	1	dad clarifies this not a food stop	https://www.theo	DENTON, TX—Stressing that they were there solely		
10583							
10584							

Add 1000 more rows at bottom.

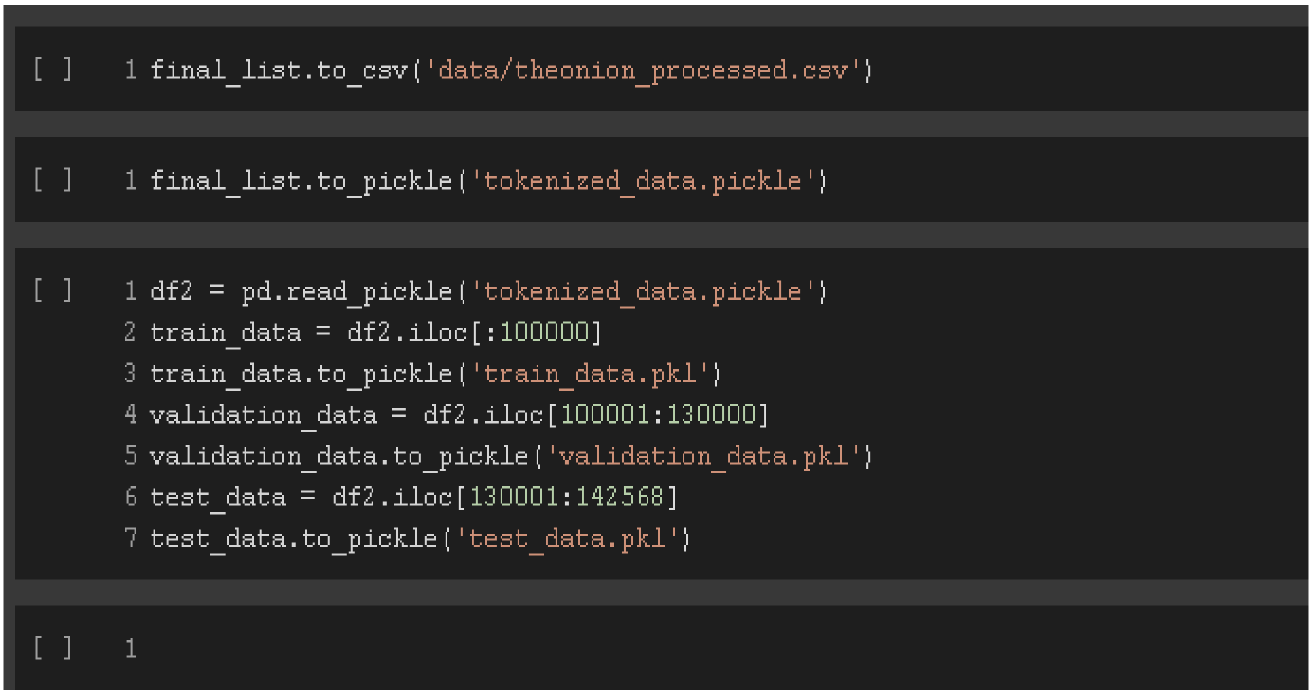
Raw data extracted



```
tokenized_title
```

```
[['inclement', 'weather', 'prevents', 'liar', 'from', 'getting', 'to', 'work'],
 ['mother',
  'comes',
  'pretty',
  'close',
  'to',
  'using',
  'word',
  'streaming',
  'correctly'],
 ['richard',
  'bransons',
  'globalwarming',
  'donation',
  'nearly',
  'as',
  'much',
  'as',
  'cost',
  'of',
  'failed',
  'balloon',
  'trips'],
 ['shadow',
  'government',
  'getting',
  'too',
  'large',
  'to',
  'meet',
  'in',
  'marriott',
  'conference',
  'room',
  'b'],
 ['ford',
  'develops',
  'new',
  'suv',
```

Pre-processing



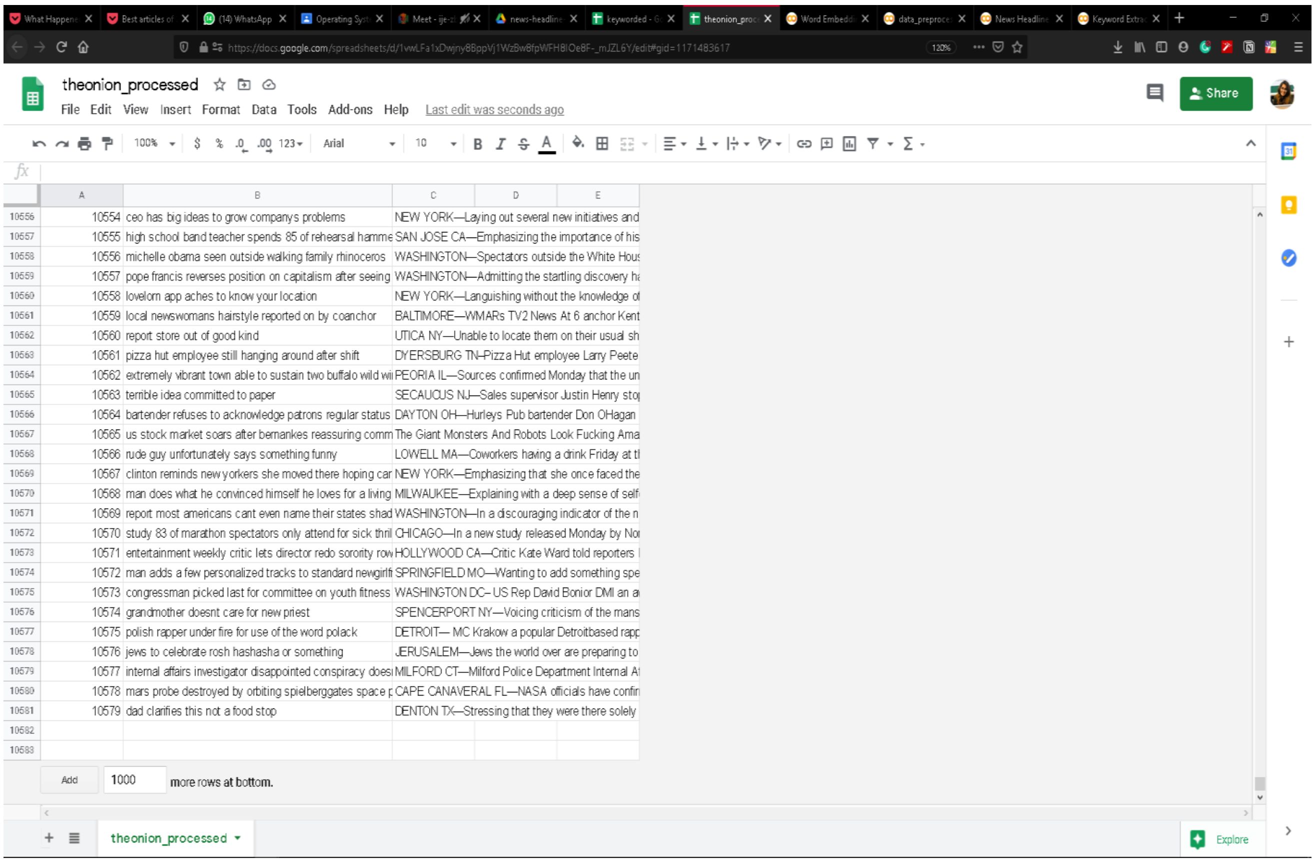
```
[ ]  final_list.to_csv('data/theonion_processed.csv')
```

```
[ ]  final_list.to_pickle('tokenized_data.pickle')
```

```
[ ]  df2 = pd.read_pickle('tokenized_data.pickle')
     train_data = df2.iloc[:100000]
     train_data.to_pickle('train_data.pkl')
     validation_data = df2.iloc[100001:130000]
     validation_data.to_pickle('validation_data.pkl')
     test_data = df2.iloc[130001:142568]
     test_data.to_pickle('test_data.pkl')
```

```
[ ]
```

Saving new file after filtering and tokenising


theonion_processed

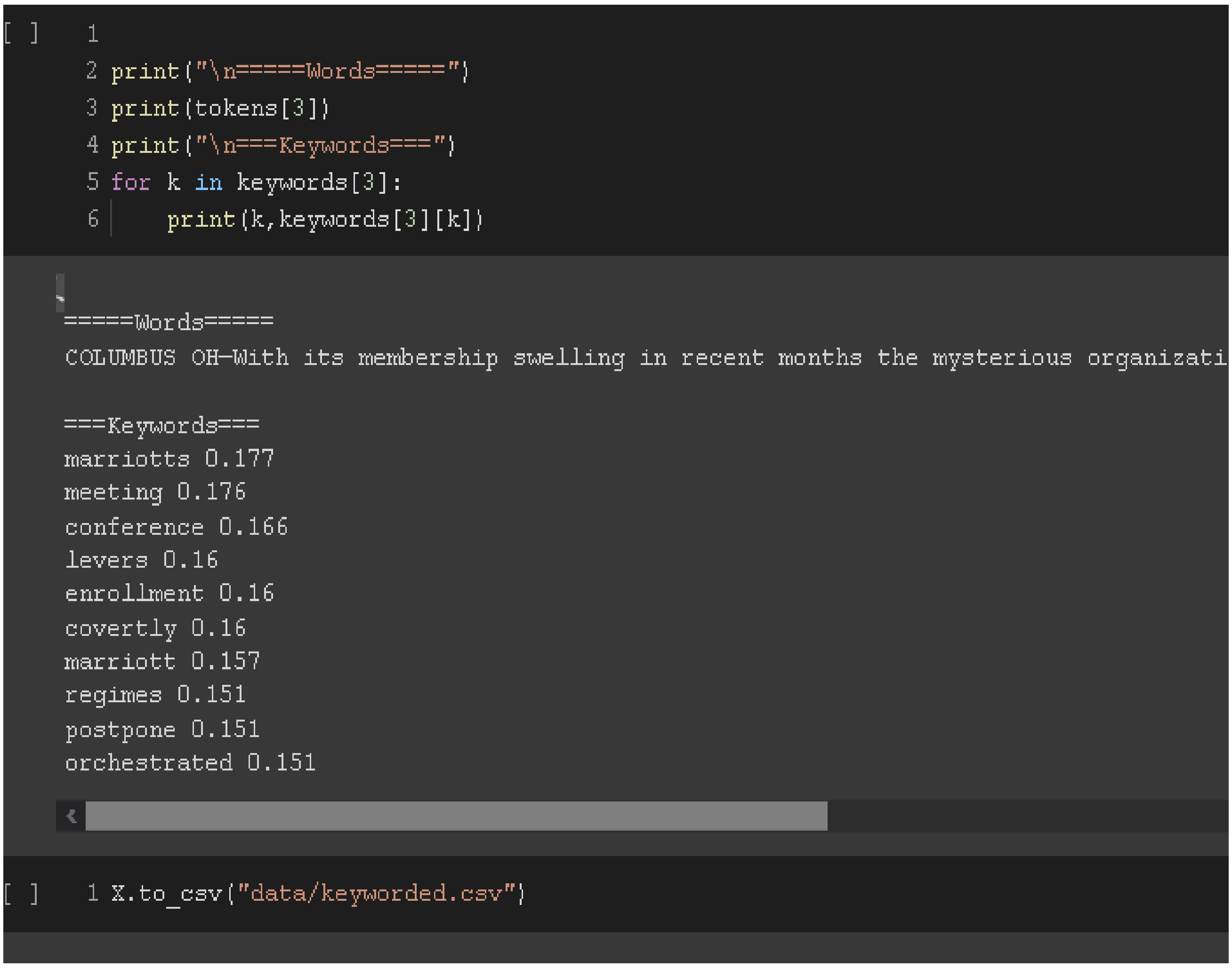
File Edit View Insert Format Data Tools Add-ons Help Last edit was seconds ago

	A	B	C	D	E
10556	10554	ceo has big ideas to grow companys problems	NEW YORK—Laying out several new initiatives and		
10557	10555	high school band teacher spends 85 of rehearsal hamme	SAN JOSE CA—Emphasizing the importance of his		
10558	10556	michelle obama seen outside walking family rhinoceros	WASHINGTON—Spectators outside the White Hous		
10559	10557	pope francis reverses position on capitalism after seeing	WASHINGTON—Admitting the startling discovery ha		
10560	10558	lovelorn app aches to know your location	NEW YORK—Languishing without the knowledge of		
10561	10559	local newswomans hairstyle reported on by coanchor	BALTIMORE—WMARs TV2 News At 6 anchor Kent		
10562	10560	report store out of good kind	UTICA NY—Unable to locate them on their usual sh		
10563	10561	pizza hut employee still hanging around after shift	DYERSBURG TN—Pizza Hut employee Larry Peete		
10564	10562	extremely vibrant town able to sustain two buffalo wild wi	PEORIA IL—Sources confirmed Monday that the un		
10565	10563	terrible idea committed to paper	SECAUCUS NJ—Sales supervisor Justin Henry sto		
10566	10564	bartender refuses to acknowledge patrons regular status	DAYTON OH—Hurleys Pub bartender Don OHagan		
10567	10565	us stock market soars after bernankes reassuring comm	The Giant Monsters And Robots Look Fucking Ama		
10568	10566	rude guy unfortunately says something funny	LOWELL MA—Coworkers having a drink Friday at t		
10569	10567	clinton reminds new yorkers she moved there hoping car	NEW YORK—Emphasizing that she once faced the		
10570	10568	man does what he convinced himself he loves for a living	MILWAUKEE—Explaining with a deep sense of self		
10571	10569	report most americans cant even name their states shad	WASHINGTON—In a discouraging indicator of the n		
10572	10570	study 83 of marathon spectators only attend for sick thril	CHICAGO—In a new study released Monday by No		
10573	10571	entertainment weekly critic lets director redo sorority row	HOLLYWOOD CA—Critic Kate Ward told reporters		
10574	10572	man adds a few personalized tracks to standard newgirlfi	SPRINGFIELD MO—Wanting to add something spe		
10575	10573	congressman picked last for committee on youth fitness	WASHINGTON DC– US Rep David Bonior DMI an a		
10576	10574	grandmother doesnt care for new priest	SPENCERPORT NY—Voicing criticism of the mans		
10577	10575	polish rapper under fire for use of the word polack	DETROIT— MC Krakow a popular Detroitbased rapp		
10578	10576	jews to celebrate rosh hashasha or something	JERUSALEM—Jews the world over are preparing to		
10579	10577	internal affairs investigator disappointed conspiracy does	MILFORD CT—Milford Police Department Internal A		
10580	10578	mars probe destroyed by orbiting spielberggates space p	CAPE CANAVERAL FL—NASA officials have confir		
10581	10579	dad clarifies this not a food stop	DENTON TX—Stressing that they were there solely		
10582					
10583					

Add 1000 more rows at bottom.

theonion_processed

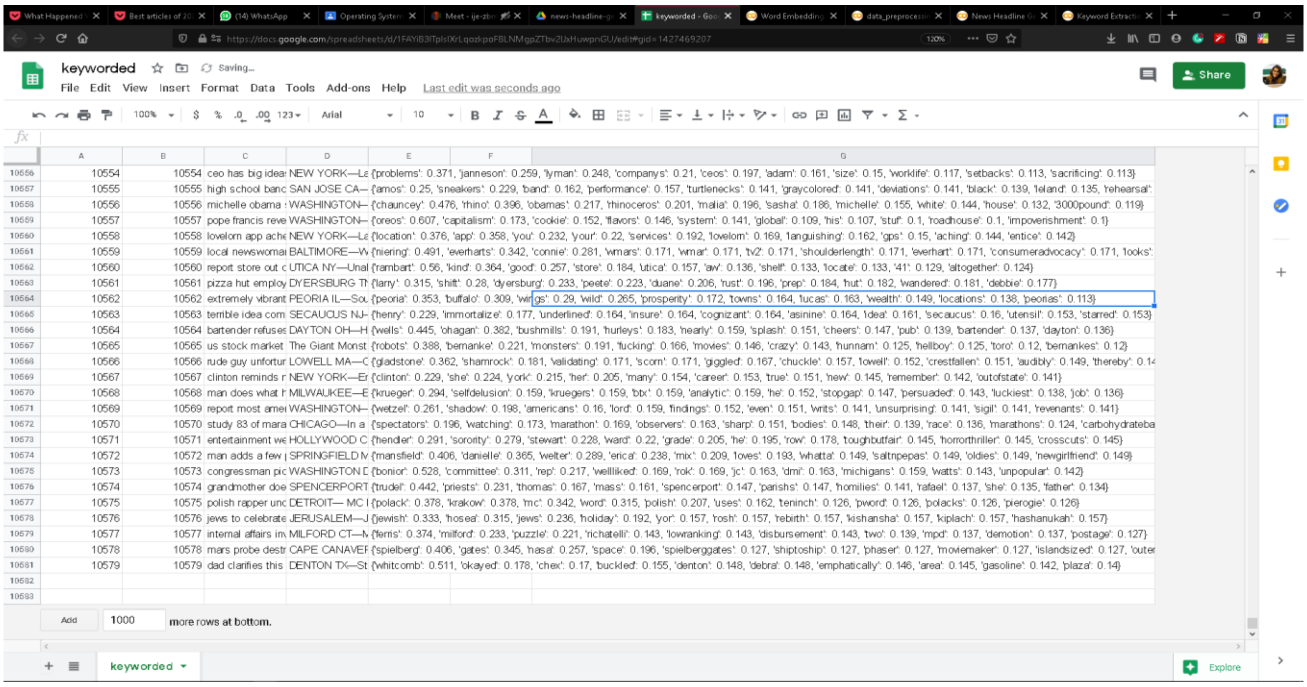
Processed data after splitting into headline and paragraph


```
[ ]

    print("\n=====Words=====")
    print(tokens[3])
    print("\n===Keywords===")
    for k in keywords[3]:
        print(k,keywords[3][k])
```

```

=====Words=====
COLUMBUS OH—With its membership swelling in recent months the mysterious organizati

===Keywords===
marriotts 0.177
meeting 0.176
conference 0.166
levers 0.16
enrollment 0.16
covertly 0.16
marriott 0.157
regimes 0.151
postpone 0.151
orchestrated 0.151
```

```
[ ]  X.to_csv("data/keyworded.csv")
```

Extracting keywords by keeping a track of feature name and their corresponding score



keyworded

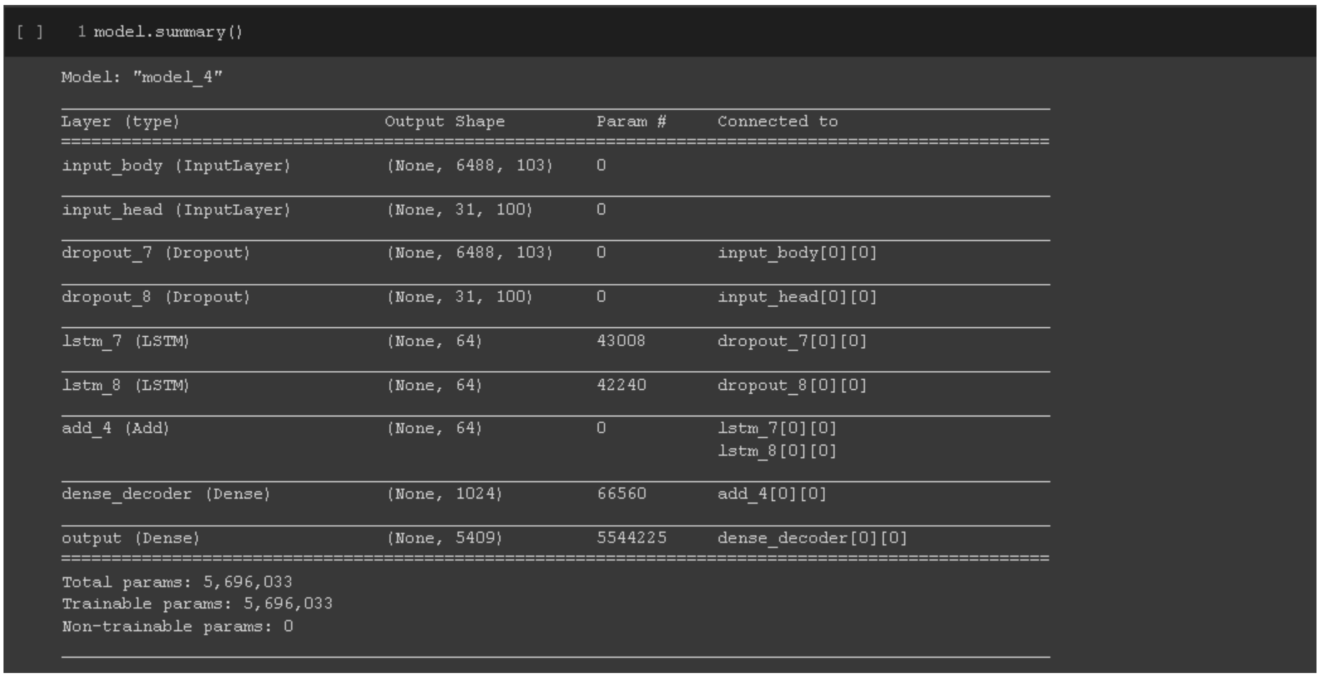
File Edit View Insert Format Data Tools Add-ons Help Last edit was seconds ago

	A	B	C	D	E	F	G
10556	10554	10554	ceo has big idea:	NEW YORK—La	{'problems': 0.371, 'janneson': 0.259, 'lyman': 0.248, 'companys': 0.21, 'ceos': 0.197, 'adam': 0.161, 'size': 0.15, 'worklife': 0.117, 'setbacks': 0.113, 'sacrificing': 0.113}		
10557	10555	10555	high school banc	SAN JOSE CA—	{'amos': 0.25, 'sneakers': 0.229, 'band': 0.162, 'performance': 0.157, 'turtlenecks': 0.141, 'graycolored': 0.141, 'deviations': 0.141, 'black': 0.139, 'leland': 0.135, 'rehearsal':		
10558	10556	10556	michelle obama :	WASHINGTON—	{'chauncey': 0.476, 'rhino': 0.396, 'obamas': 0.217, 'rhinoceros': 0.201, 'malia': 0.196, 'sasha': 0.186, 'michelle': 0.155, 'white': 0.144, 'house': 0.132, '3000pound': 0.119}		
10559	10557	10557	pope francis reve	WASHINGTON—	{'oreos': 0.607, 'capitalism': 0.173, 'cookie': 0.152, 'flavors': 0.146, 'system': 0.141, 'global': 0.109, 'his': 0.107, 'stuf': 0.1, 'roadhouse': 0.1, 'impoverishment': 0.1}		
10560	10558	10558	lovelorn app ache	NEW YORK—La	{'location': 0.376, 'app': 0.358, 'you': 0.232, 'your': 0.22, 'services': 0.192, 'lovelorn': 0.169, 'languishing': 0.162, 'gps': 0.15, 'aching': 0.144, 'entice': 0.142}		
10561	10559	10559	local newswomai	BALTIMORE—W	{'niering': 0.491, 'everharts': 0.342, 'connie': 0.281, 'wmars': 0.171, 'wmar': 0.171, 'tv2': 0.171, 'shoulderlength': 0.171, 'everhart': 0.171, 'consumeradvocacy': 0.171, 'looks':		
10562	10560	10560	report store out c	UTICA NY—Unal	{'rambart': 0.56, 'kind': 0.364, 'good': 0.257, 'store': 0.184, 'utica': 0.157, 'aw': 0.136, 'shelf': 0.133, 'locate': 0.133, '41': 0.129, 'altogether': 0.124}		
10563	10561	10561	pizza hut employ	DYERSBURG TI	{'larry': 0.315, 'shift': 0.28, 'dyersburg': 0.233, 'peete': 0.223, 'duane': 0.206, 'rust': 0.196, 'prep': 0.184, 'hut': 0.182, 'wandered': 0.181, 'debbie': 0.177}		
10564	10562	10562	extremely vibrant	PEORIA IL—Sou	{'peoria': 0.353, 'buffalo': 0.309, 'wir	gs': 0.29, 'wild': 0.265, 'prosperity': 0.172, 'towns': 0.164, 'lucas': 0.163, 'wealth': 0.149, 'locations': 0.138, 'peorias': 0.113}	
10565	10563	10563	terrible idea com	SECAUCUS NJ-	{'henry': 0.229, 'immortalize': 0.177, 'underlined': 0.164, 'insure': 0.164, 'cognizant': 0.164, 'asinine': 0.164, 'idea': 0.161, 'secaucus': 0.16, 'utensil': 0.153, 'starred': 0.153}		
10566	10564	10564	bartender refuses	DAYTON OH—H	{'wells': 0.445, 'ohagan': 0.382, 'bushmills': 0.191, 'hurleys': 0.183, 'nearly': 0.159, 'splash': 0.151, 'cheers': 0.147, 'pub': 0.139, 'bartender': 0.137, 'dayton': 0.136}		
10567	10565	10565	us stock market	The Giant Monst	{'robots': 0.388, 'bernanke': 0.221, 'monsters': 0.191, 'fucking': 0.166, 'movies': 0.146, 'crazy': 0.143, 'hunnam': 0.125, 'hellboy': 0.125, 'toro': 0.12, 'bernankes': 0.12}		
10568	10566	10566	rude guy unfortur	LOWELL MA—C	{'gladstone': 0.362, 'shamrock': 0.181, 'validating': 0.171, 'scorn': 0.171, 'giggled': 0.167, 'chuckle': 0.157, 'lowell': 0.152, 'crestfallen': 0.151, 'audibly': 0.149, 'thereby': 0.14		
10569	10567	10567	clinton reminds r	NEW YORK—Er	{'clinton': 0.229, 'she': 0.224, 'york': 0.215, 'her': 0.205, 'many': 0.154, 'career': 0.153, 'true': 0.151, 'new': 0.145, 'remember': 0.142, 'outofstate': 0.141}		
10570	10568	10568	man does what h	MILWAUKEE—E	{'krueger': 0.294, 'selfdelusion': 0.159, 'kruegers': 0.159, 'btx': 0.159, 'analytic': 0.159, 'he': 0.152, 'stopgap': 0.147, 'persuaded': 0.143, 'luckiest': 0.138, 'job': 0.136}		
10571	10569	10569	report most amei	WASHINGTON—	{'wetzel': 0.261, 'shadow': 0.198, 'americans': 0.16, 'lord': 0.159, 'findings': 0.152, 'even': 0.151, 'writs': 0.141, 'unsurprising': 0.141, 'sigil': 0.141, 'revenants': 0.141}		
10572	10570	10570	study 83 of mara	CHICAGO—In a	{'spectators': 0.196, 'watching': 0.173, 'marathon': 0.169, 'observers': 0.163, 'sharp': 0.151, 'bodies': 0.148, 'their': 0.139, 'race': 0.136, 'marathons': 0.124, 'carbohydrateba		
10573	10571	10571	entertainment we	HOLLYWOOD C	{'hendler': 0.291, 'sorority': 0.279, 'stewart': 0.228, 'ward': 0.22, 'grade': 0.205, 'he': 0.195, 'row': 0.178, 'toughbutfair': 0.145, 'horrorthriller': 0.145, 'crosscuts': 0.145}		
10574	10572	10572	man adds a few	SPRINGFIELD M	{'mansfield': 0.406, 'danielle': 0.365, 'welter': 0.289, 'erica': 0.238, 'mix': 0.209, 'loves': 0.193, 'whatta': 0.149, 'saltnpepas': 0.149, 'oldies': 0.149, 'newgirlfriend': 0.149}		
10575	10573	10573	congressman pic	WASHINGTON D	{'bonior': 0.528, 'committee': 0.311, 'rep': 0.217, 'wellliked': 0.169, 'rok': 0.169, 'jc': 0.163, 'dmi': 0.163, 'michigans': 0.159, 'watts': 0.143, 'unpopular': 0.142}		
10576	10574	10574	grandmother doe	SPENCERPORT	{'trudel': 0.442, 'priests': 0.231, 'thomas': 0.167, 'mass': 0.161, 'spencerport': 0.147, 'parishs': 0.147, 'homilies': 0.141, 'rafael': 0.137, 'she': 0.135, 'father': 0.134}		
10577	10575	10575	polish rapper unc	DETROIT— MC I	{'polack': 0.378, 'krakow': 0.378, 'mc': 0.342, 'word': 0.315, 'polish': 0.207, 'uses': 0.162, 'teninch': 0.126, 'pword': 0.126, 'polacks': 0.126, 'pierogie': 0.126}		
10578	10576	10576	jews to celebrate	JERUSALEM—J	{'jewish': 0.333, 'hosea': 0.315, 'jews': 0.236, 'holiday': 0.192, 'yor': 0.157, 'rosh': 0.157, 'rebirth': 0.157, 'kishansha': 0.157, 'kiplach': 0.157, 'hashanukah': 0.157}		
10579	10577	10577	internal affairs in	MILFORD CT—M	{'ferris': 0.374, 'milford': 0.233, 'puzzle': 0.221, 'richatelli': 0.143, 'lowranking': 0.143, 'disbursement': 0.143, 'two': 0.139, 'mpd': 0.137, 'demotion': 0.137, 'postage': 0.127}		
10580	10578	10578	mars probe destr	CAPE CANAVEF	{'spielberg': 0.406, 'gates': 0.345, 'nasa': 0.257, 'space': 0.196, 'spielberggates': 0.127, 'shiptoship': 0.127, 'phaser': 0.127, 'moviemaker': 0.127, 'islandsized': 0.127, 'outer		
10581	10579	10579	dad clarifies this	DENTON TX—St	{'whitcomb': 0.511, 'okayed': 0.178, 'chex': 0.17, 'buckled': 0.155, 'denton': 0.148, 'debra': 0.148, 'emphatically': 0.146, 'area': 0.145, 'gasoline': 0.142, 'plaza': 0.14}		
10582							
10583							

Add 1000 more rows at bottom.

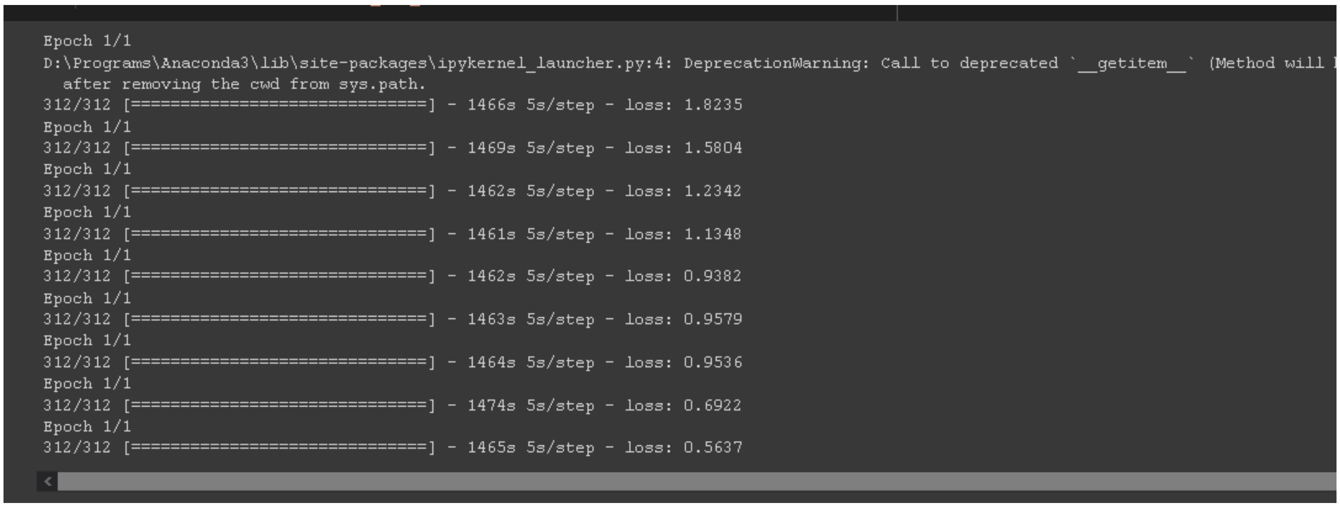
keyworded

Dataset after generating keywords and relevant weightage



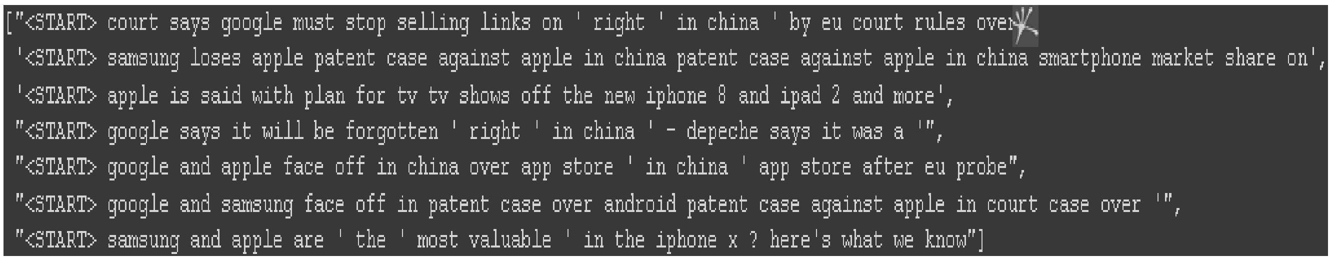
```
[ ]   1 model.summary()

Model: "model_4"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_body (InputLayer)         (None, 6488, 103)    0
__________________________________________________________________________________________________
input_head (InputLayer)         (None, 31, 100)      0
__________________________________________________________________________________________________
dropout_7 (Dropout)             (None, 6488, 103)    0           input_body[0][0]
__________________________________________________________________________________________________
dropout_8 (Dropout)             (None, 31, 100)      0           input_head[0][0]
__________________________________________________________________________________________________
lstm_7 (LSTM)                   (None, 64)           43008       dropout_7[0][0]
__________________________________________________________________________________________________
lstm_8 (LSTM)                   (None, 64)           42240       dropout_8[0][0]
__________________________________________________________________________________________________
add_4 (Add)                     (None, 64)           0           lstm_7[0][0]
                                                                 lstm_8[0][0]
__________________________________________________________________________________________________
dense_decoder (Dense)           (None, 1024)         66560       add_4[0][0]
__________________________________________________________________________________________________
output (Dense)                  (None, 5409)         5544225     dense_decoder[0][0]
==================================================================================================
Total params: 5,696,033
Trainable params: 5,696,033
Non-trainable params: 0
__________________________________________________________________________________________________
```

A brief summary of the model



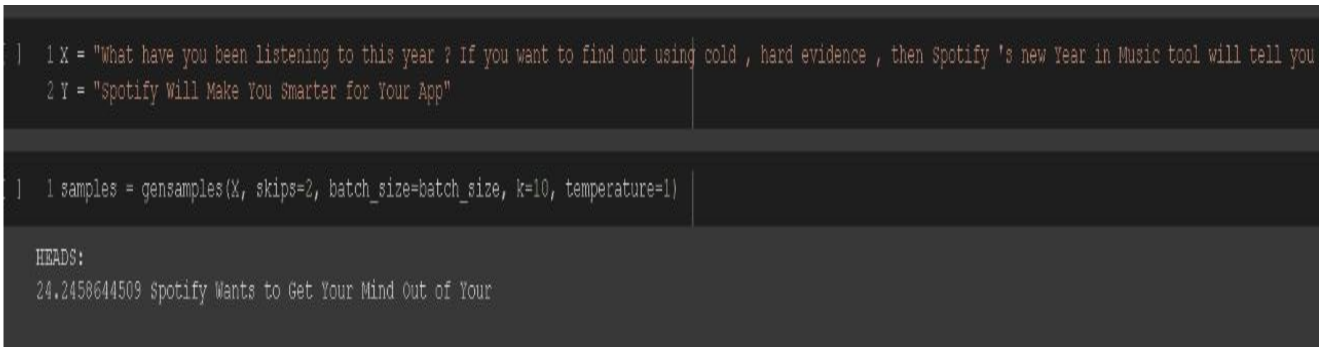
```
Epoch 1/1
D:\Programs\Anaconda3\lib\site-packages\ipykernel_launcher.py:4: DeprecationWarning: Call to deprecated `__getitem__` (Method will
  after removing the cwd from sys.path.
312/312 [==============================] - 1466s 5s/step - loss: 1.8235
Epoch 1/1
312/312 [==============================] - 1469s 5s/step - loss: 1.5804
Epoch 1/1
312/312 [==============================] - 1462s 5s/step - loss: 1.2342
Epoch 1/1
312/312 [==============================] - 1461s 5s/step - loss: 1.1348
Epoch 1/1
312/312 [==============================] - 1462s 5s/step - loss: 0.9382
Epoch 1/1
312/312 [==============================] - 1463s 5s/step - loss: 0.9579
Epoch 1/1
312/312 [==============================] - 1464s 5s/step - loss: 0.9536
Epoch 1/1
312/312 [==============================] - 1474s 5s/step - loss: 0.6922
Epoch 1/1
312/312 [==============================] - 1465s 5s/step - loss: 0.5637
```

Model training



```
["<START> court says google must stop selling links on ' right ' in china ' by eu court rules over",
 '<START> samsung loses apple patent case against apple in china patent case against apple in china smartphone market share on',
 '<START> apple is said with plan for tv tv shows off the new iphone 8 and ipad 2 and more',
 "<START> google says it will be forgotten ' right ' in china ' - depeche says it was a '",
 "<START> google and apple face off in china over app store ' in china ' app store after eu probe",
 "<START> google and samsung face off in patent case over android patent case against apple in court case over '",
 "<START> samsung and apple are ' the ' most valuable ' in the iphone x ? here's what we know"]
```

Outputs Predicted

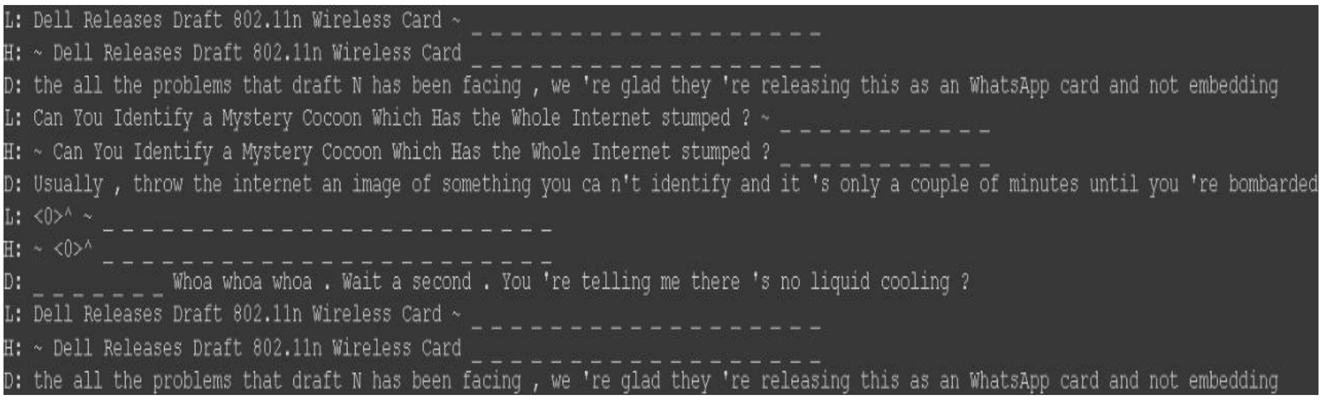


```
[ ]   X = "What have you been listening to this year ? If you want to find out using cold , hard evidence , then Spotify 's new Year in Music tool will tell you
      Y = "Spotify Will Make You Smarter for Your App"
```

```
[ ]   samples = gensamples(X, skips=2, batch_size=batch_size, k=10, temperature=1)
```

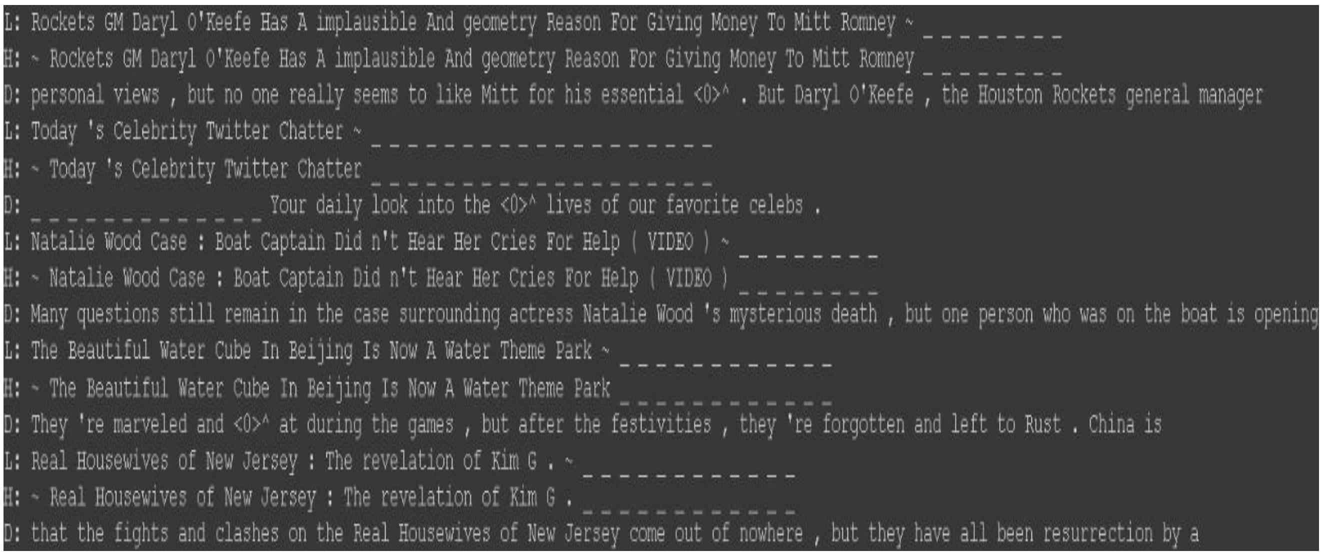
```
HEADS:
24.2458644509 Spotify Wants to Get Your Mind Out of Your
```

Outputs Predicted



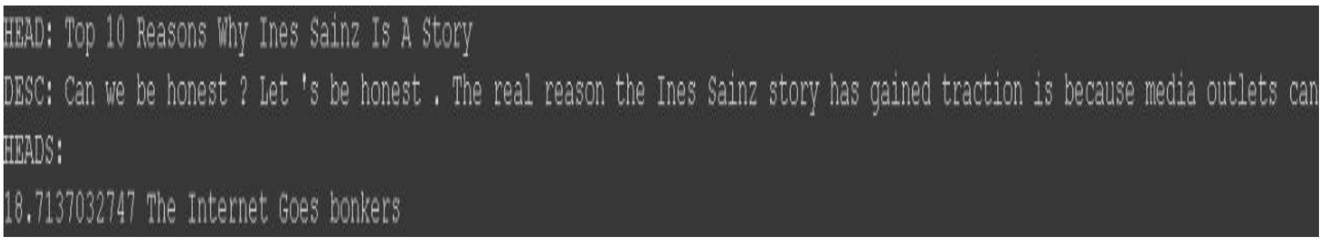
```
L: Dell Releases Draft 802.11n Wireless Card ~ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
H: ~ Dell Releases Draft 802.11n Wireless Card _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
D: the all the problems that draft N has been facing , we 're glad they 're releasing this as an WhatsApp card and not embedding
L: Can You Identify a Mystery Cocoon Which Has the Whole Internet stumped ? ~ _ _ _ _ _ _ _ _ _ _ _
H: ~ Can You Identify a Mystery Cocoon Which Has the Whole Internet stumped ? _ _ _ _ _ _ _ _ _ _ _
D: Usually , throw the internet an image of something you ca n't identify and it 's only a couple of minutes until you 're bombarded
L: <0>^ ~ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
H: ~ <0>^ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
D: _ _ _ _ _ _ _ Whoa whoa whoa . Wait a second . You 're telling me there 's no liquid cooling ?
L: Dell Releases Draft 802.11n Wireless Card ~ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
H: ~ Dell Releases Draft 802.11n Wireless Card _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
D: the all the problems that draft N has been facing , we 're glad they 're releasing this as an WhatsApp card and not embedding
```

Outputs Predicted



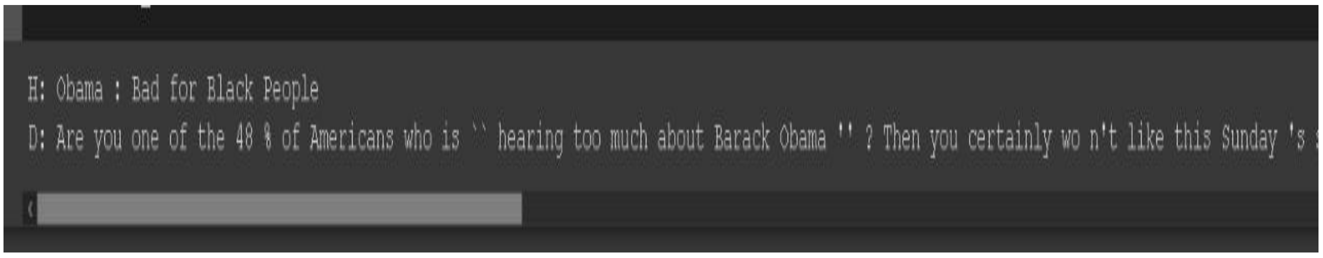
```
L: Rockets GM Daryl O'Keefe Has A implausible And geometry Reason For Giving Money To Mitt Romney ~ _ _ _ _ _ _ _ _
H: ~ Rockets GM Daryl O'Keefe Has A implausible And geometry Reason For Giving Money To Mitt Romney _ _ _ _ _ _ _ _
D: personal views , but no one really seems to like Mitt for his essential <0>^ . But Daryl O'Keefe , the Houston Rockets general manager
L: Today 's Celebrity Twitter Chatter ~ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
H: ~ Today 's Celebrity Twitter Chatter _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
D: _ _ _ _ _ _ _ _ _ _ _ _ _ Your daily look into the <0>^ lives of our favorite celebs .
L: Natalie Wood Case : Boat Captain Did n't Hear Her Cries For Help ( VIDEO ) ~ _ _ _ _ _ _ _ _
H: ~ Natalie Wood Case : Boat Captain Did n't Hear Her Cries For Help ( VIDEO ) _ _ _ _ _ _ _ _
D: Many questions still remain in the case surrounding actress Natalie Wood 's mysterious death , but one person who was on the boat is opening
L: The Beautiful Water Cube In Beijing Is Now A Water Theme Park ~ _ _ _ _ _ _ _ _ _ _ _ _
H: ~ The Beautiful Water Cube In Beijing Is Now A Water Theme Park _ _ _ _ _ _ _ _ _ _ _ _
D: They 're marveled and <0>^ at during the games , but after the festivities , they 're forgotten and left to Rust . China is
L: Real Housewives of New Jersey : The revelation of Kim G . ~ _ _ _ _ _ _ _ _ _ _ _ _
H: ~ Real Housewives of New Jersey : The revelation of Kim G . _ _ _ _ _ _ _ _ _ _ _ _
D: that the fights and clashes on the Real Housewives of New Jersey come out of nowhere , but they have all been resurrection by a
```

Outputs Predicted



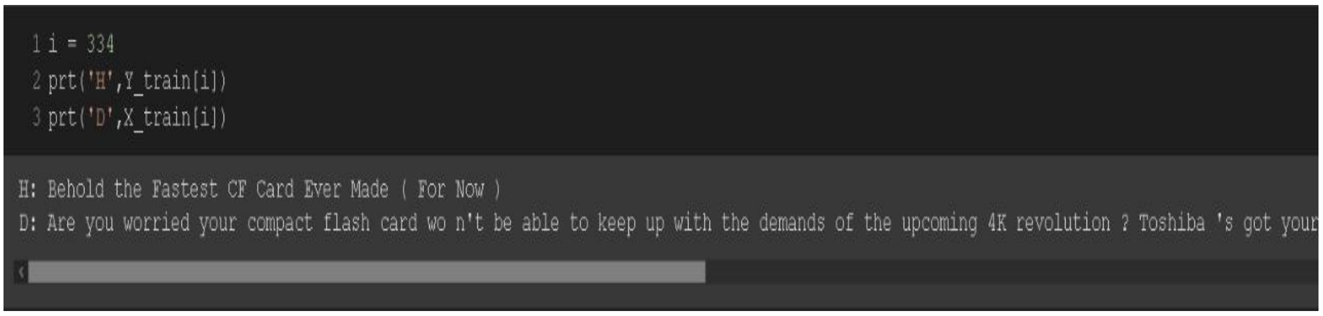
```
HEAD: Top 10 Reasons Why Ines Sainz Is A Story
DESC: Can we be honest ? Let 's be honest . The real reason the Ines Sainz story has gained traction is because media outlets can
HEADS:
18.7137032747 The Internet Goes bonkers
```

Outputs Predicted



```
H: Obama : Bad for Black People
D: Are you one of the 48 % of Americans who is `` hearing too much about Barack Obama '' ? Then you certainly wo n't like this Sunday 's
```

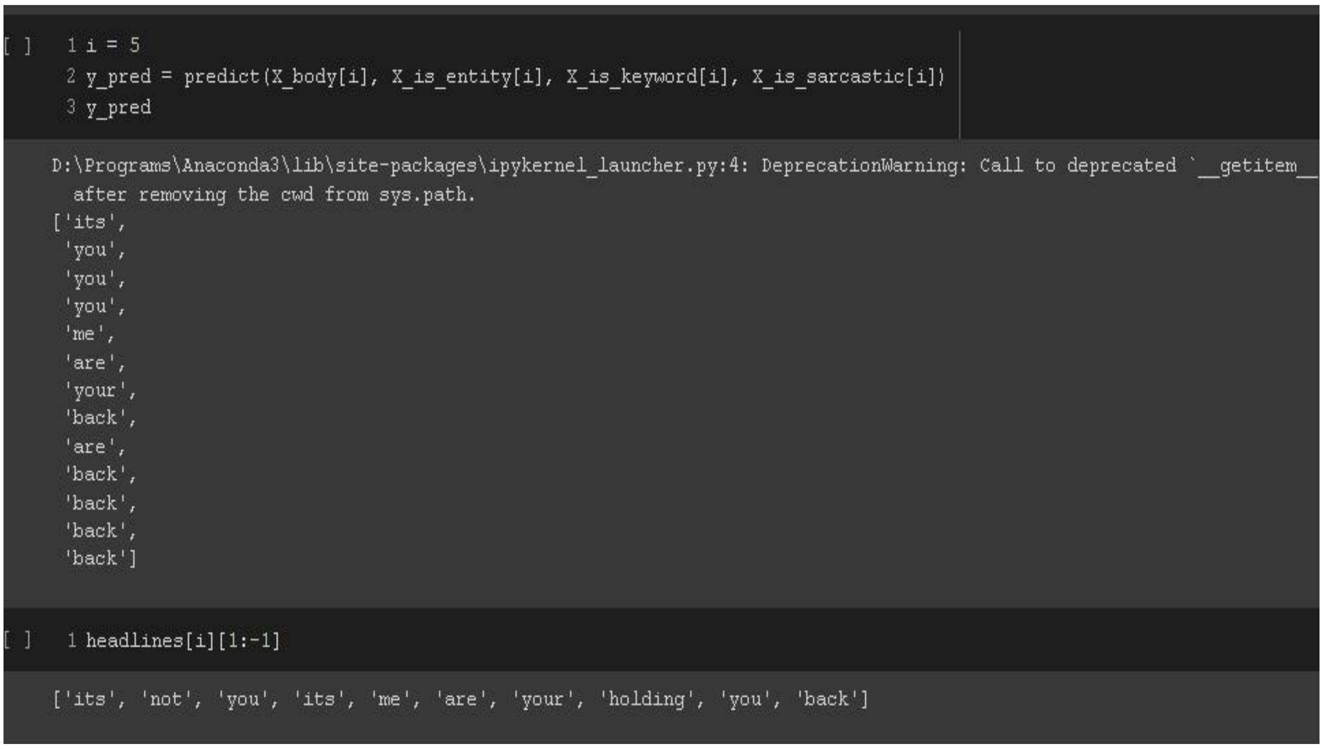
Outputs Predicted



```
i = 334
prt('H',Y_train[i])
prt('D',X_train[i])
```

```
H: Behold the Fastest CF Card Ever Made ( For Now )
D: Are you worried your compact flash card wo n't be able to keep up with the demands of the upcoming 4K revolution ? Toshiba 's got your
```

Outputs Predicted



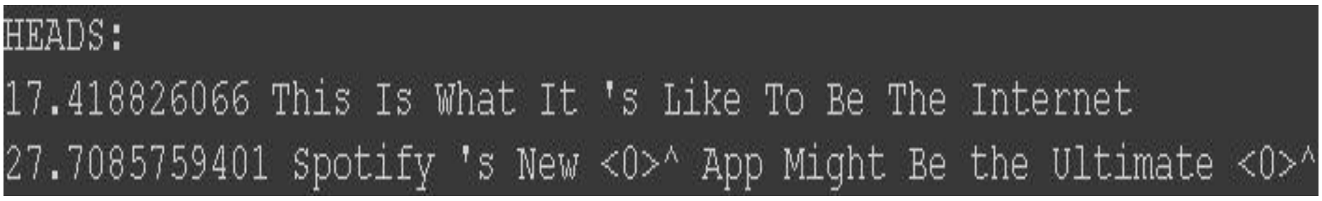
```
[ ]  i = 5
     y_pred = predict(X_body[i], X_is_entity[i], X_is_keyword[i], X_is_sarcastic[i])
     y_pred
```

```
D:\Programs\Anaconda3\lib\site-packages\ipykernel_launcher.py:4: DeprecationWarning: Call to deprecated `__getitem__
  after removing the cwd from sys.path.
['its',
 'you',
 'you',
 'you',
 'me',
 'are',
 'your',
 'back',
 'are',
 'back',
 'back',
 'back',
 'back']
```

```
[ ]  headlines[i][1:-1]
```

```
['its', 'not', 'you', 'its', 'me', 'are', 'your', 'holding', 'you', 'back']
```

Outputs Predicted



```
HEADS:
17.418826066 This Is What It 's Like To Be The Internet
27.7085759401 Spotify 's New <0>^ App Might Be the Ultimate <0>^
```

Outputs predicted

Conclusion

A news article often has a brief title that summarizes its content for readers to decide whether they want to read further. Therefore, a well-summarized title is crucial in delivering the main point of the news article to its potential readers. As a result, automatic text summarization techniques have a huge potential for news articles in that it expedites the process of summarizing a given documents for humans and, if models are well trained, generates the summary with a high accuracy. Our goal is to build a text automatic summarization model with deep learning algorithms that can output a one-sentence summarization given an article.

Models developed for text auto summarization has immediate applications in news articles title generations and beyond, such as machine translation, image captioning, as well as video summarization.

There are in general two types of summarization techniques: extractive summarization and abstractive summarization, where the former summarizes articles by selecting a subset of words that retains the most important points, and the latter generates a summary based on semantic understanding of the input. In this project we focus on abstractive approaches.

Significant amount of work has been done on automatic summarization. Since earlier models greatly focused on extractive methods, abstractive summarization is a relatively new research area with less experimentations. However, recent advances in research using abstractive methods has made it quite a popular field. As a next step we would like to further improve the grammaticality of the summaries in a data-driven way, as well as scale this system to generate paragraph-level summaries. Both pose additional challenges in terms of efficient alignment and consistency in generation. We have trained an encoder-decoder recurrent neural network with LSTM units and attention for generating news headlines using the texts of news articles from the real world data. Using only the first few words of a news article, the model would generate a concise summary of those particular words, and most of the time the summary would be valid and grammatically correct, given the apt pre - processing that would be done.

Future Scope of the Project

- We aim to Prepare a full stack GUI based cross platform application using this project so that the user can enter any text/paragraph and the application generates a single line summary of it.
- This would be extended so that the user can themselves specify the word limit for the headline/summary.
- The user would have a choice to enter the sentiment of the headline they want to generate. For e.g. the headline could have a tragic sentiment to it, or it could be politicized. This aspect right now would depend on the attention layer, but in future we aim to work on it so that custom headlines are generated.
- We aim to increase the size of the dataset by extracting data from various sources and languages.
- A language translator can be built so that the headline is generated in custom language as specified by the user.

References

[1] Nallapati, Ramesh, et al. "Abstractive text summarization using sequence-to-sequence rnns and beyond." *arXiv preprint arXiv:1602.06023* (2016).

[2] Rush, Alexander M., Sumit Chopra, and Jason Weston. "A neural attention model for abstractive sentence summarization." *arXiv preprint arXiv:1509.00685* (2015).

[3] Wang, Li, et al. "A reinforced topic-aware convolutional sequence-to-sequence model for abstractive text summarization." *arXiv preprint arXiv:1805.03616* (2018).

[4] Chopra, Sumit, Michael Auli, and Alexander M. Rush. "Abstractive sentence summarization with attentive recurrent neural networks." *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2016.

[5] Luong, Minh-Thang, Hieu Pham, and Christopher D. Manning. "Effective approaches to attention-based neural machine translation." *arXiv preprint arXiv:1508.04025* (2015).

[6] Lopyrev, Konstantin. "Generating news headlines with recurrent neural networks." *arXiv preprint arXiv:1512.01712* (2015).

[7] Yu, Hujia, Chang Yue, and Chao Wang. *News article summarization with attention-based deep recurrent neural networks*. Stanford University, Tech. Rep.