Chapter – 1

INTRODUCTION

1.1 Overview

In the times of today, the world is flocking with E commerce stores all around us. Nearly all business platforms are practically, in a way, an E commerce store. With easy access to the Internet everywhere and knowledge about the procedure, the market for E commerce has boomed to glorious heights in the recent past.

There are a number of parameters which contribute to define the success and credibility of an Ecommerce store. However, one very important factor in elevating the reputation, standard and evaluation of an Ecommerce store is Product Reviews. Not taken much into serious account, Product Reviews provide an Ecommerce store with one of the most valuable resources available: Customer Feedback. Merchants very frequently underestimate the importance of Product Reviews for an Ecommerce store, while paying more attention and being preoccupied by too many tasks to manage like, optimizing site designs, eliminating customer doubts, helping opportune customers decide which product to buy, customer services and administrative tasks.

One very important task for the Ecommerce store is to maintain its reputation in the online market. Quite naturally, it takes a lot of effort to gain that reputation but not much to lose it: Product Reviews are the best ways to maintain their winning streak. Product Reviews and feedbacks have changed the game for online market since internet has become a very household thing. The Product Reviews are the factors which either make or break the relationship of the consumer with the store – they help build loyalty and trust and lets the potential consumer know the product much more clearly and the aspects that differentiate it from the rest of the products elsewhere. An Ecommerce store which has a good compilation of consumer reviews for the products shows the wide consumer base it incapacitates. The store, thus, anticipates positive reviews to gain more customers in the future.[1]

Another key feature that Product Reviews provide us with are the branding, marketing and advertising of the Ecommerce store as well as the product in question. With the right push at the correct angle, an Ecommerce store can very easily advertise its prospects in the market via the customer feedbacks. Say, a product has a number of good reviews on a certain Ecommerce store, consequently the user interest in the product rises on a general scale – consumers talk more about the product with each other when they meet personally due to the available online reviews which catches their eyes and hence, directly or indirectly the advertising and marketing of the product extends like a chain reaction, a forest fire. This is the reason why brands like Samsung, Levis, etc. have gained a huge market in a very short time – apart from the quality of the products they sell. Positive Product Reviews help establish a favorable image to the product which actually persists long after the product has already been launched. Also, for the serious marketer, Product Reviews are a great way to interact with the consumer directly and know the details of their product from the eyes of the consumer and make developments wherever necessary. With the correct follow up of the Product Reviews, the seller can accurately know the loopholes of its products and the areas in which it can improve its product to meet the needs of the end consumer. This leads to a positive experience for the consumer which ultimately helps the business in the future.[2]

However, the biggest advantage Product Reviews provide for an Ecommerce store is the increase in its Sales or the increase in the number of purchases from the consumers. Online reviews are so important to businesses because they ultimately increase the sales by giving the consumers the information they need to make the decision to purchase the product. People are always more likely to buy the products which has already been recommended by other users. When customer reviews about the product has been added to the Ecommerce store, 42% of the site administrators have reported increases in average order value, versus only 6% that report a decrease with inclusion of reviews.

Okay, so having mentioned above clearly about the uses and advantages Product Reviews provide the administrator and the Ecommerce store with, viz – building up online reputation, advertising & branding of the store and the products and making developments in the products wherever necessary based on customer feedback- there is one very important role played by Product Reviews that highly benefit the consumer who wishes to buy products online.

Nearly all the stores claim and commit that their product is the best. We cannot trust their words as a parameter of evaluating different products from different sites. The best judges for a product are the consumer of the product themselves. But all the consumers cannot even theoretically use all the products: this is where Product Reviews come into play. With the help of Product Reviews, a customer knows exactly what he is buying, what he is being offered and what are the things he can expect from the product. Product Reviews have become more important due to the lack of the ability to 'test' the product prior to its purchase online; whereas checking and 'testing' the product physically before buying them is feasible in brick-and-mortar concept of business – naturally, if a consumer who wishes to buy a product online is unable to practically try the product in the beginning and is at risk at accesing the quality and buying a new product, the only way he can be confident about the usability of the item is by going through the reviews it has received – by having the full accessibility and knowledge about all the pros and cons of the specified product. According to stats, nearly 40% of the consumers say that they wouldn't buy electronic items without reading online reviews about the product. Needless to say, the immediate benefit of Product Reviews is that they can make the potential future customers feel much more confident. The more positive reviews a product has, the more convinced the consumer will be in buying that product and believe in the idea that they are making the right decision. Nowadays, apart from Product Reviews, even product ratings prove to be an important aspect for the consumer while purchasing the online product. Potential customers these days, first check the product rating (how many stars the product has acquired out of five) and then see if the Product Reviews authenticate the product or not. However, negative reviews are also helpful for the consumers and the business enterprise: for a consumer, a negative review provides him with the worst case scenario he can expect from the product – he would have the exact knowledge about what loopholes he might encounter with the product and make prior precautions; for an Ecommerce enterprise, a negative review aids in maintaining the credibility and authenticity of the product – if all the reviews are positive then the probable customer tends to trust the product less (since everyone knows every product must have some drawbacks). Most reviewbased portals, like Yelp or Amazon, contain thousands of user-generated reviews. It is impossible for any human reader to process even the most relevant of these documents. The most promising tool to solve this task is a text summarization.

1.2 Content Summarization

- Summarization is the task of condensing a piece of text to a shorter version that contains the main information from the original. There are two broad approaches to summarization: extractive and abstractive. Extractive methods assemble summaries exclusively from passages (usually whole sentences) taken directly from the source text, while abstractive methods may generate novel words and phrases not featured in the source text – as a human-written abstract usually does.
- The extractive approach is easier, because copying large chunks of text from the source document ensures baseline 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.

1.3 Need of Content Summarization in E Commerce Industry

- With the increasing use of e-commerce websites our project, we can try to improve the shopping experience for users. The domain of this project lies under Natural Language Processing (NLP) which basically includes analysis, classification and summarization of raw text obtained from customer reviews. The review for a product on the Amazon.in website consists of an overall rating of the product which is obtained from a statistics of each individual's customer rating out of 5 stars and a customer review section where customers drop in their experience on buying that particular product. While the overall rating gives a vague idea of the product's genuineness, the customer review section gives a potentially elaborated idea.

- 61% of customers read online reviews before making a purchase decision, and they are now essential for e-commerce sites. User reviews are proven sales drivers, and something the majority of customers will want to see before deciding to make a purchase.[6]
- There have been so many positive recommendations of the value of reviews for ecommerce, that the case doesn't really need to be made anymore, though I'll make it again anyway. Quite simply, user reviews increase conversions. They can eliminate any doubts potential customers may have about a product, or can help product selection.

1.4 SEO Benefits of Reviews

Improving conversions and improving customer experience should be the main purpose of user reviews, but let's not forget the considerable SEO benefits.

- Fresh, unique content for search engines. Search engine spiders like unique content that is regularly updated, and user reviews are a great way to attract more content.
- When many ecommerce sites just use the same standard manufacturer descriptions and product specifications, user-generated content can differentiate a product page in the search results.
- This is important as it makes pages more useful to customers, and also increases the chance of ranking highly in the SERPs.
- Reviews are an increasingly important part of the purchase journey for online consumers. Indeed, a recent survey found that 61% of consumers would read online reviews when purchasing technology items such as MP3 players and cameras. This also means that more consumers will be searching for the name of the product plus the word 'review', or related words such as 'ratings'.
- If review content is correctly formatted, then these rich snippets can help increases clickthroughs from search engine results pages.
- Long tail targeting : The additional content generated by user reviews increases the chance of ranking well for long tail searches.

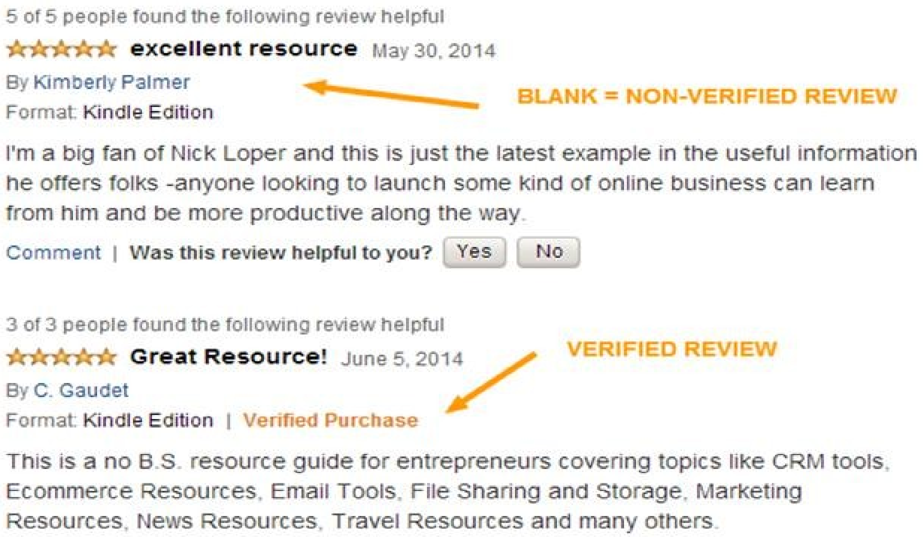
1.5 Negative Reviews and Customers

1.5.1 Bad Reviews are valuable too...

All reviews are valuable, and a mix of positive and negative reviews helps to improve consumer trust in the opinions they read. Indeed, recent stats from Reevoo suggest that the presence of bad reviews actually improves conversions by 67%. Reevoo found that people that seek out and read bad reviews convert better, as the very fact that they are paying such close attention means they are more likely to be in purchase mode. 68% of consumers trust reviews more when they see both good and bad scores, while 30% suspect censorship or faked reviews when they don’t see any negative opinions on the page.

1.5.2 Too many bad reviews aren’t good for business

The benefits of bad reviews very much depends on the proportion of good to bad. The negative reviews make the positive ones more believable, but there is a point at which they ring alarm bells for consumers. If, for instance, a product page contains 15 reviews, and two are negative, then the other 13 look trustworthy. If that proportion changes, it’s a different matter. Recent research from Lightspeed found that between one and three bad online reviews would be enough to deter the majority (67%) of shoppers from purchasing a product or service. [5]



5 of 5 people found the following review helpful

★★★★★ **excellent resource** May 30, 2014

By Kimberly Palmer

Format: Kindle Edition

BLANK = NON-VERIFIED REVIEW

I'm a big fan of Nick Loper and this is just the latest example in the useful information he offers folks -anyone looking to launch some kind of online business can learn from him and be more productive along the way.

3 of 3 people found the following review helpful

★★★★★ **Great Resource!** June 5, 2014

By C. Gaudet

Format: Kindle Edition | Verified Purchase

VERIFIED REVIEW

This is a no B.S. resource guide for entrepreneurs covering topics like CRM tools, Ecommerce Resources, Email Tools, File Sharing and Storage, Marketing Resources, News Resources, Travel Resources and many others.

Figure 1.1 Amazon Reviews

1.6 Python

- Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

Figure 1.2 Python Logo

- **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to perl and php.
- **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.[10]

1.6.1 History of Python

- Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.
- Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

- Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.[10]

1.6.2 Python Features

- **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
- **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** − Python provides interfaces to all major commercial databases.
- **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** − Python provides a better structure and support for large programs than shell scripting.[10]

1.7 Machine Learning

- Machine learning is a method of data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention.

- The name *machine learning* was coined in 1959 by Arthur Samuel. Machine learning explores the study and construction of algorithms that can learn from and make predictions on data– such algorithms overcome following strictly static program instructions by making data-driven predictions or decisions, through building a model from sample inputs. Machine learning is employed in a range of computing tasks where designing and programming explicit algorithms with good performance is difficult or infeasible; example applications include email filtering, detection of network intruders, and computer vision.[10]

Figure 1.3: The world of Machine Learning

1.7.1 Machine learning techniques

Machine learning techniques are typically classified into several broad categories:

- Supervised learning: The computer is presented with example inputs and their desired outputs, given by a "teacher", and the goal is to learn a general rule that maps inputs to outputs. As special cases, the input signal can be only partially available, or restricted to special feedback.
- Semi-supervised learning: The computer is given only an incomplete training signal: a training set with some (often many) of the target outputs missing.
- Unsupervised learning: No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself

(discovering hidden patterns in data) or a means towards an end (feature learning).

- Reinforcement learning: Data (in form of rewards and punishments) are given only as feedback to the program's actions in a dynamic environment, such as driving a vehicle or playing a game against an opponent.[10]

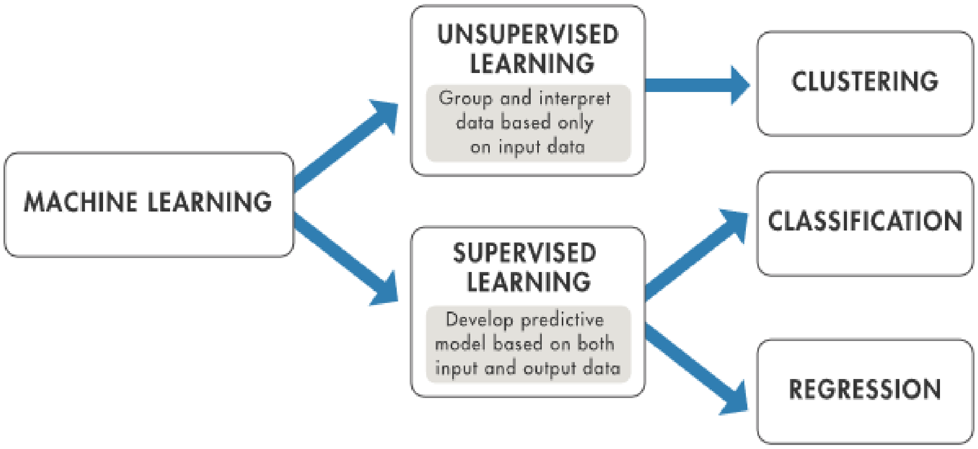

MACHINE LEARNING
UNSUPERVISED LEARNING
Group and interpret data based only on input data
SUPERVISED LEARNING
Develop predictive model based on both input and output data
CLUSTERING
CLASSIFICATION
REGRESSION

Figure 1.4 Classification of Machine Learning

1.7.2 Evolution of machine learning

- Because of new computing technologies, machine learning today is not like machine learning of the past. It was born from pattern recognition and the theory that computers can learn without being programmed to perform specific tasks; researchers interested in artificial intelligence wanted to see if computers could learn from data. The iterative aspect of machine learning is important because as models are exposed to new data, they are able to independently adapt. They learn from previous computations to produce reliable, repeatable decisions and results. It's a science that's not new – but one that has gained fresh momentum.

- While many machine learning algorithms have been around for a long time, the ability to automatically apply complex mathematical calculations to big data –

over and over, faster and faster – is a recent development. Here are a few widely publicized examples of machine learning applications you may be familiar with:

- The heavily hyped, self-driving Google car? The essence of machine learning.
- Online recommendation offers such as those from Amazon and Netflix? Machine learning applications for everyday life.
- Knowing what customers are saying about you on Twitter? Machine learning combined with linguistic rule creation.
- Fraud detection? One of the more obvious, important uses in our world today.

1.8 Deep Learning

Deep Learning is a subfield of machine learning concerned with algorithms inspired by the structure and function of the brain called artificial neural networks. It imitates the workings of the human brain in processing data and creating patterns for use in decision making. Deep learning is a subset of machine learning in artificial intelligence (AI) that has networks capable of learning unsupervised from data that is unstructured or unlabeled. Also known as deep neural learning or deep neural network.

Deep learning has evolved hand-in-hand with the digital era, which has brought about an explosion of data in all forms and from every region of the world. This data, known simply as big data, is drawn from sources like social media, internet search engines, e-commerce platforms, and online cinemas, among others. This enormous amount of data is readily accessible and can be shared through fintech applications like cloud computing. However, the data, which normally is unstructured, is so vast that it could take decades for humans to comprehend it and extract relevant information. Companies realize the incredible potential that can result from unraveling this wealth of information and are increasingly adapting to AI systems for automated support.[10]

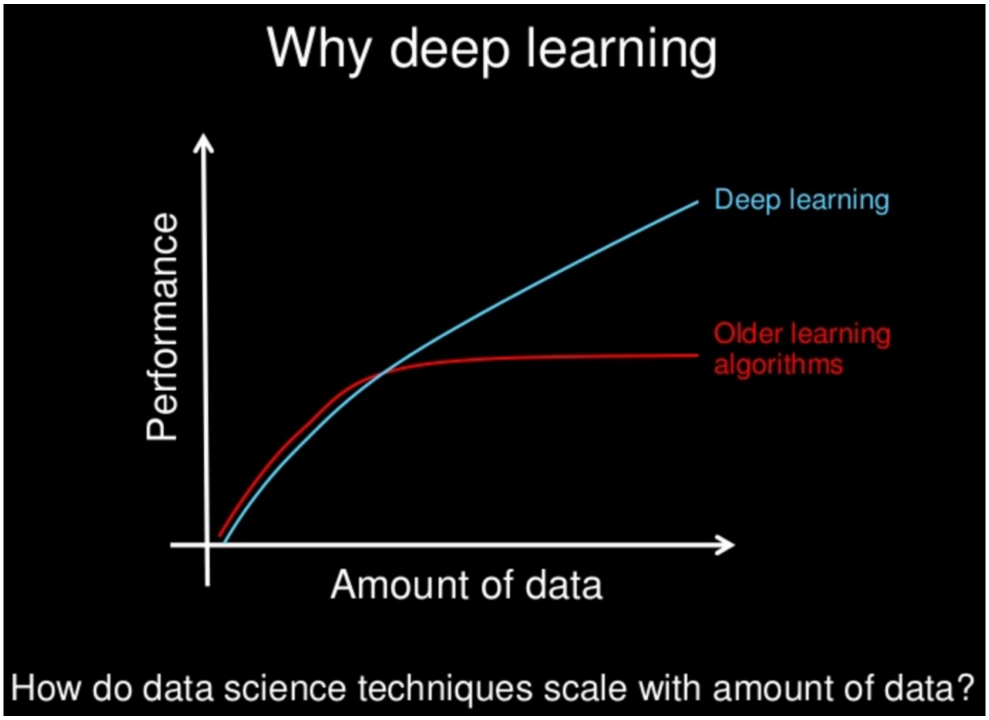


Figure 1.5 Classification of Machine Learning

1.9 Natural Language Processing

Natural Language Processing is the technology used to aid computers to understand the human's natural language. It's not an easy task teaching machines to understand how we communicate. Natural Language Processing, usually shortened as NLP, is a branch of artificial intelligence that deals with the interaction between computers and humans using the natural language. The ultimate objective of NLP is to read, decipher, understand, and make sense of the human languages in a manner that is valuable.

Most NLP techniques rely on machine learning to derive meaning from human languages. NLP entails applying algorithms to identify and extract the natural language rules such that the unstructured language data is converted into a form that computers can understand. When the text has been provided, the computer will utilize algorithms to extract meaning associated with every sentence and collect the essential data from them. Sometimes, the computer may fail to understand the meaning of a sentence well, leading to obscure results. Syntactic analysis and semantic analysis are the main techniques used to complete Natural Language Processing tasks.

Named entity recognition (NER): It involves determining the parts of a text that can be identified and categorized into preset groups. Examples of such groups include names of people and names of places.

Word sense disambiguation: It involves giving meaning to a word based on the context.

Natural language generation: It involves using databases to derive semantic intentions and convert them into human language.

Natural Language Processing plays a critical role in supporting machine-human interactions. As more research is being carried in this field, we expect to see more breakthroughs that will make machines smarter at recognizing and understanding the human language.[10]

Chapter - 2

SOFTWARE REQUIREMENT SPECIFICATION

- A software requirements specification (SRS) is a detailed description of a software system to be developed with its functional and non-functional requirements. The SRS is developed based the agreement between customer and contractors. It may include the use cases of how user is going to interact with software system. The software requirement specification document consistent of all necessary requirements required for project development. To develop the software system we should have clear understanding of Software system. To achieve this we need to continuous communication with customers to gather all requirements.
- A good SRS defines the how Software System will interact with all internal modules, hardware, communication with other programs and human user interactions with wide range of real life scenarios. Using the Software requirements specification (SRS) document on QA lead, managers creates test plan.
- It is very important that testers must be cleared with every detail specified in this document in order to avoid faults in test cases and its expected results. It is highly recommended to review or test SRS documents before start writing test cases and making any plan for testing. Let’s see how to test SRS and the important point to keep in mind while testing it.

2.1 <u>Problem Statement</u>

Reviews and Ratings fed by the customers on various e-commerce sites, in some way, provides a great deal of information for the users to judge the quality and features of the products they see online. The major hitch is that the popular products have thousands of reviews. We do not have the time or patience to read all these reviews. Although reading through the customer reviews gives a comprehensible picture, it might be very time consuming in some cases where the product has thousands of reviews listed.

Hence, our application eases this task by analyzing and summarizing the reviews which will help the user decide what other buyers have experienced on buying this product. We carry out this process by a number of modules that include feature extraction and opinion mining to improve the process of analysis and helps in the formation of an efficient summary. In this project, the design of a unified opinion mining and propose a feature and opinion mining system for customer sentiment analysis framework is presented with natural language processing and Named entity recognition approach. As the number of reviews are in terms of hundreds on certain products and in terms of thousands on popular products it is evident that the user may not read all the reviews and might miss out on some critical reviews that concern his needs. Hence we provide a solution to summarize it based on the product's features. This saves the time and energy of the users which would rather be well spent. The user will be able to decide on one look of the graphical outcome of the summarization.

As far as the feasibility study is concerned, the reviews extraction must be real-time so we are using scrapy and selenium to extract reviews in real-time. But there might be some fake reviews also. The effect of these fake opinions might not be nullified here but it will be added as the future scope of the project.

2.2 **Scope**

We all interact with applications which uses text summarization. Many of those applications are for the platform which publishes articles on daily news, entertainment, sports. With our busy schedule, we prefer to read the summary of those article before we decide to jump in for reading entire article. Reading a summary help us to identify the interest area, gives a brief context of the story. Summarization can be defined as a task of producing a concise and fluent summary while preserving key information and overall meaning.Summarization systems often have additional evidence they can utilize in order to specify the most important topics of document(s). Moved by the cutting edge advancement and Innovation, Data is to this century what oil was to the last one. Today, our reality is parachuted by the gathering and dissemination of huge amounts of data.

In fact, the International Data Corporation that the total amount of digital data circulating annually around the world would sprout from 4.4 zettabytes in 2013 to hit 180 zettabytes in 2025. That's a lot of data!

With such a huge amount of data circulating in the digital space, there is a need to develop algorithms that can automatically shorten large huge texts and summaries that information that can fluently pass the intended messages.

2.3 <u>**Purpose**</u>

Summarization of opinions from product reviews is the most common example of opinion summarization and content summarization. These reviews often come from stores of electronics like Amazon. Basically, we can regard the "summarization" as the "function" its input is document and output is summary.

2.4 <u>**Definitions and Acronyms**</u>

- **Single document summarization**

  [*summary = summarize(document)]*

- **Multi-document summarization**

  [*summary = summarize(document_1, document_2, …)]*

  We can take the query to add the viewpoint of summarization.

- **Query focused summarization**

  summary = summarize(document, query)

  This type of summarization is called "Query focused summarization" on the contrary to the "Generic summarization". Especially, a type that set the viewpoint to the "difference" (update) is called "Update summarization".

- **Update summarization**

summary = summarize(document, previous_document_or_summary)

And the "*summary*" itself has some variety.

- **Indicative summary**

It looks like a summary of the book. This summary describes what kinds of the story, but not tell all of the stories especially its ends (so indicative summary has only partial information).

- **Informative summary**

In contrast to the indicative summary, the informative summary includes full information of the document.

- **Keyword summary**

Not the text, but the words or phrases from the input document.

- **Headline summary**

Only one line summary.

2.5 **Software Perspective**

The project needed several requirements to be gathered before proceeding to design and develop this tool. One among them included gathering information about how does NER works and what are the different modules required for creating this tool.

We would require a great deal of knowledge and research in the NLP fields of POS tagging, tokenization and understanding about the models proposed in the research work provided in the references [1],[2],[3] and [4] that are required for the content and reviews summarization.

2.6 **Software Functioning**

The basic functioning of the **Reviews Summarization** system is to get the name or Product ASIN of any product information along with many other factors like frequency of reviews etc. and as a result, it will display a proper analysis of the reviews for that product that were given to it as input. The following are some of the functions or operations that the Review Summarizer system performs:

- Giving an overall review analysis of the product that was given to the system as an input.
- Determine the polarity (Positive/Negative) of the reviews and also the overall polarity of the product reviews.
- Providing the user with the proper summary of these reviews

2.7 **User Classes and Characteristics**

The various user classes that you anticipate will use this product; includes every e commerce freaks out there. Not only those but also a large group of reviewers and daily shoppers. The users can themselves understand the customer based reviews of those products accordingly.

2.8 **Assumptions and Dependencies**

Although the review summarization system tries to summarize and analyze the reviews as accurate as possible, there are some assumptions made while generating the target feature and outputs. Some of these assumptions include:

- It might not be necessary that the user select all the reviews of the product.
- The number of reviews to provide the summary can be chosen by the users.
- The number of reviews must be greater than 100 so as to perform a valid and efficient analysis.
- The reviews on Amazon or any other site can be fake. A fake review can be posted by a company employee, paid individual or anyone else with a vested

interest in selling more product. This system does not detect the fake reviews and will include all the reviews whether fake or genuine.

2.9 <u>External Interface Requirements</u>

2.9.1 User Interface

- A basic web application or a website is used for user interface. The user interface is an interactive flask and web based tool which implements all the functionalities specified in section 2.6. It consists of the main functionality of the system using the NLP techniques and Deep Learning models where user fills given the product exact name or Product ASIN as input and get the NLP based analysis and summary as output. The user interface uses a combination of three web-based language:
  - ✓ HTML
  - ✓ CSS
  - ✓ JavaScript

2.9.2 Software Interfaces

- ▪ A full stack working system is not complete until a back end is provided for implementing all the functionality and storing the data. The user interface described in the previous section takes help of the most important backend framework Flask along with python programming language for operating algorithm and NLP models on the data stored in the database.

<u>2.10 Software Requirements</u>

This particular section contains all the software requirements at a level of detail. The following points describe each of the system feature in detail:

- <u>Review Scraping System</u>: The Review Scraping System is used to scrape all product reviews for a particular product or products. Web Scraping (also termed Screen Scraping, Web Data Extraction, Web Harvesting etc.) is a technique employed to extract large amounts of data from websites whereby the data is extracted and saved to a local file in your computer or to a database in table (spreadsheet) format. Data displayed by most websites can only be viewed using a web browser. They do not offer the functionality to save a copy of this data for personal use. The only option then is to manually copy and paste the data - a very tedious job which can take many hours or sometimes days to complete. Web Scraping is the technique of automating this process, so that instead of manually copying the data from websites, the Web Scraping software will perform the same task within a fraction of the time. In the scraping tool, we have used scrapy as well as selenium using Python for interactive data gathering.

- Inputs to the Prediction System: The inputs that the system takes from the user:

  - ✓ Product ASIN
  - ✓ Product Exact Name
  - ✓ Product Amazon Link
  - ✓ Number of reviews to take

- Output from the Prediction System: The prediction system basically gives the product other details, complete analysis according to the reviews and the summary generated for the considered reviews.
- Reviews analyzer and summarizer: The following are the requirements and objectives of this system:

- ➢ To build an algorithm for summarization of customer reviews.
- ➢ To extract reviews, perform analysis on them, classify them based on polarity and produce a summary.
- ➢ To implement a unique ‘feature’ and ‘opinion’ based analysis to produce a more critical review summary.
- ➢ To provide a feature based rating on the respective product

2.11 **Non-Functional Requirements**

2.11.1 Performance Requirements

The project must meet the end user requirements. Accuracy and fast outcome must be imposed on the Project. The project is developed as easy as possible for the sake of end user. The project has to be developed with view of satisfying the future requirements and future enhancement. The tool has been finally implemented satisfying the needs and requirements specified. As per the performance is concerned, this system is said to perform the processing of the data as well as generate well and good results, which are also important to be considered even if large amount of data was used.

2.11.2 Quality and Reliability Requirements

A software component that is developed for reusability would be correct and contain no defects. In reality, formal verification is not carried out routinely, and defects can occur. However, with each reuse, defects are found and eliminated; as a result the component quality improves. Over time, the components become virtually defect free. Software reliability is defined in statistical term as" the probability of faultier-free operation of a computer program in a specified environment for specified time". For software quality and reliability, failure is non-conformance to software requirements. One failure can be corrected within seconds while another requires week and even months to correct. Complicating the issue even further, the correction of one failure may in fact result in the introduction of the errors that ultimately result in other failure.

2.11.3 Maintainability

Maintainability is the relative cost of fixing, updating, extending, operating and servicing an entity over its lifetime. An entity with relatively low cost in these areas is considered maintainable whereas an entity with high costs may be considered non-maintainable or "high maintenance." A scheduled maintenance of the system and database is required for future adaptation of the system in the box office world.

2.11.4 Feasibility

The feasibility study addresses the subject of the availability of resources and software/hardware requirements, and assesses the operating costs for obtaining these supplies.

2.12 **Software and Hardware Requirements**

2.12.1 Software Requirements

Below are the software requirements for the project: -

Operating System: Linux

Technical Skills:

Server Side Libraries: Tensor Flow, Keras, NumPy, NLTK, Natural Language Processing, scikit-learn, PyTorch, SpaCy.

Other requirements: Scrapy, Selenium and Flask

Tools: VSCode or any other IDE like Jupyter Notebook.

2.12.2 Hardware Requirements

Minimum RAM Space Required: 512MB

Chapter - 3

DIAGRAMS

3.1 Use Case Diagram

- The purpose of use case diagram is to capture the dynamic aspect of a system. However, this definition is too generic to describe the purpose, as other four diagrams (activity, sequence, collaboration, and State chart) also have the same purpose. Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analysed to gather its functionalities, use cases are prepared and actors are identified. When the initial task is complete, use case diagrams are modelled to present the outside view.

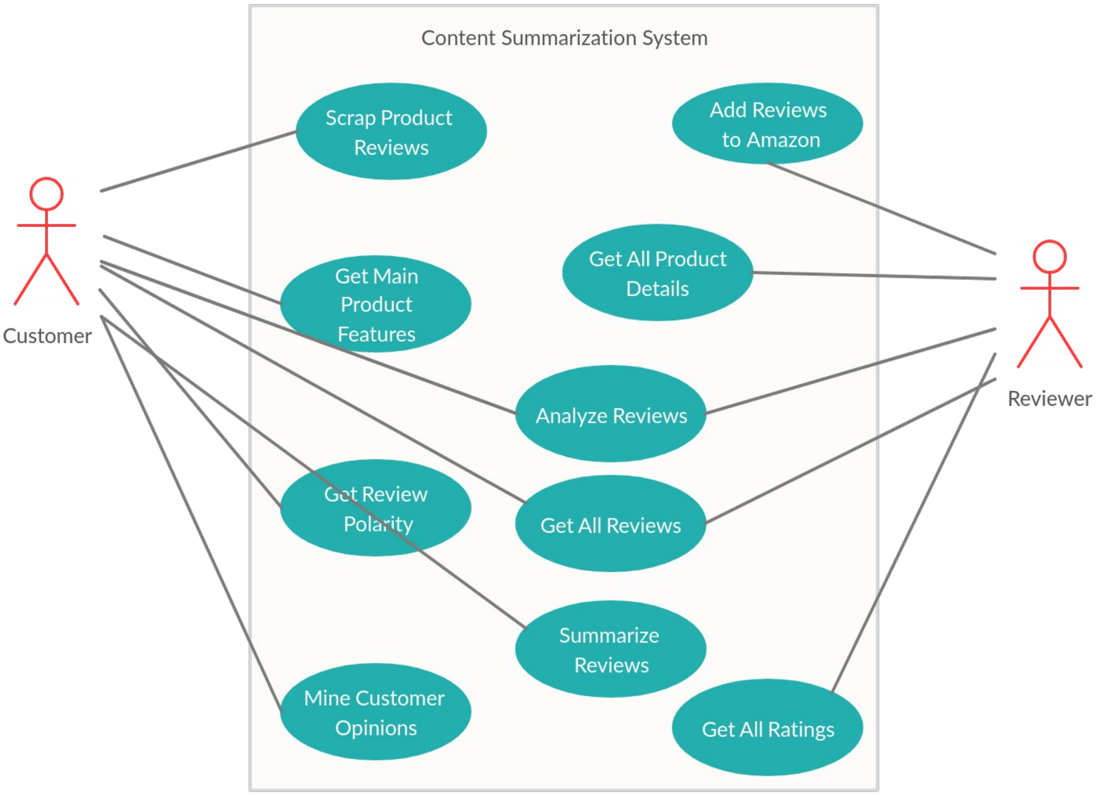


Figure 3.1: Use Case Diagram for Content Summarizer

3.2 Data Flow Diagram

- A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyse an existing system or model a new one.

✓ DFD Level 0 is also called a Context Diagram. It’s a basic overview of the whole system or process being analysed or modelled. It’s designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities.

✓ DFD Level 1 provides a more detailed breakout of pieces of the Context Level Diagram. You will highlight the main functions carried out by the system, as you break down the high-level process of the Context Diagram into its sub-processes.

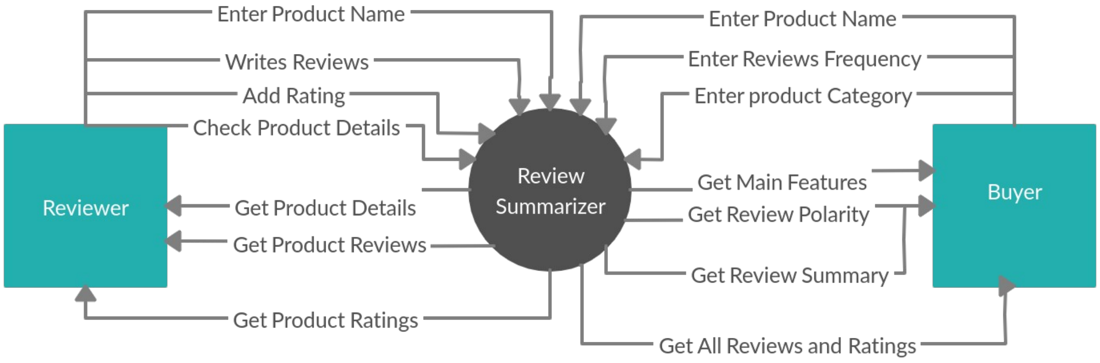


Figure 3.2: Level 0 DFD for Content Summarizer

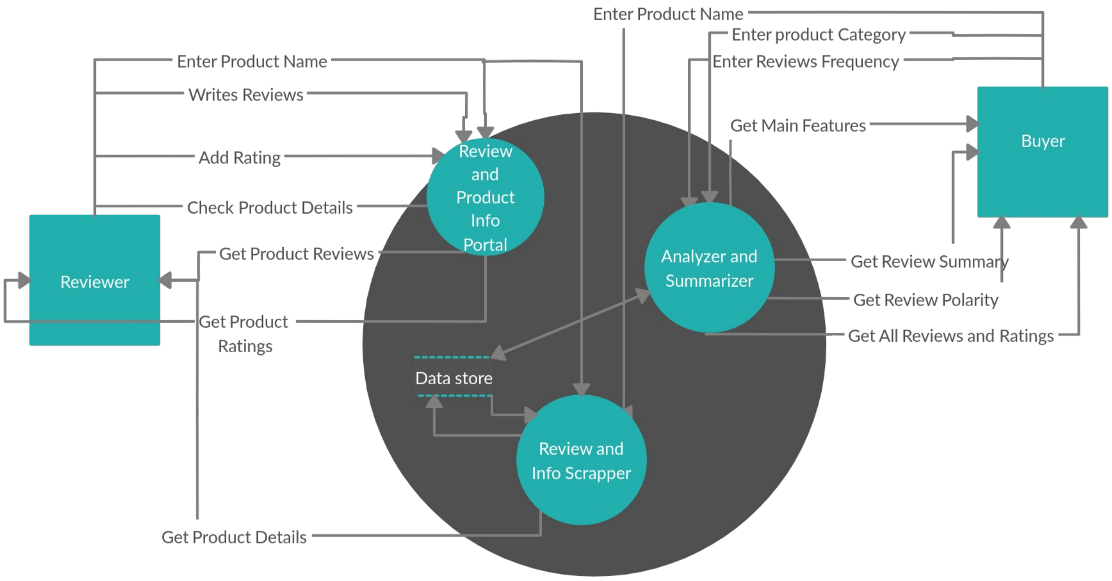


Figure 3.3: Level 1 DFD for Content Summarizer

3.3 Entity Relationship(ER) Diagram

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes.

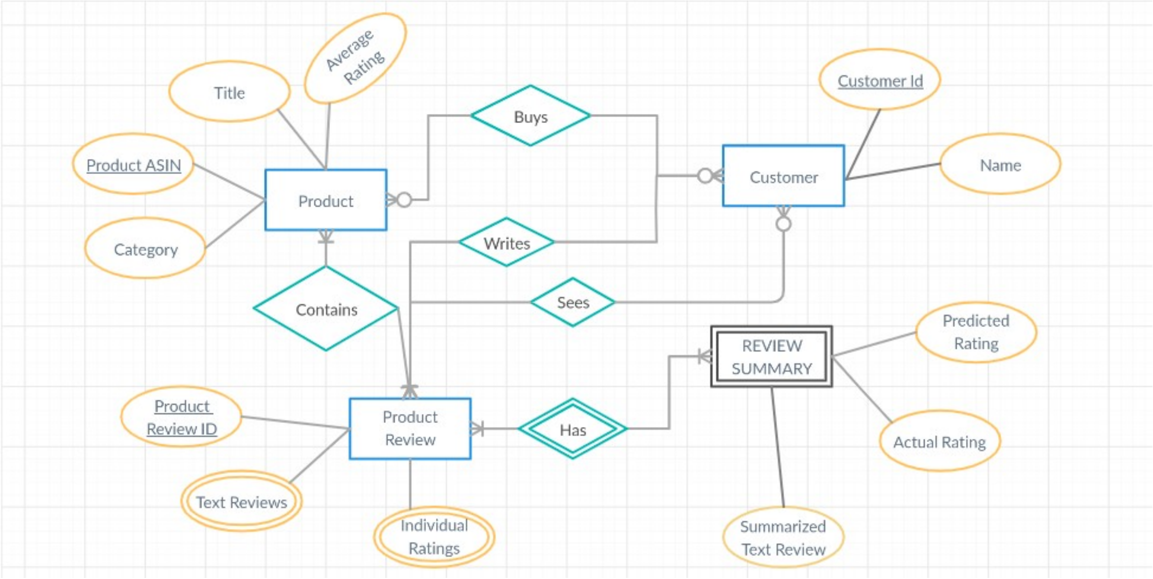


Figure 3.4: ER Diagram for Content Summarizer

Chapter - 4

PROCESS SELECTION AND IMPLEMENTATION

This chapter is meant to give a higher understanding of the different parts of the project in detail, it will include the actual data set used, the ways to acquire the data, cleaning and preprocessing of the acquired data, extracting and visualizing the required features and the different methods of analyzing the data in order to get the best presentable way of showing the output analysis and reviews summary.

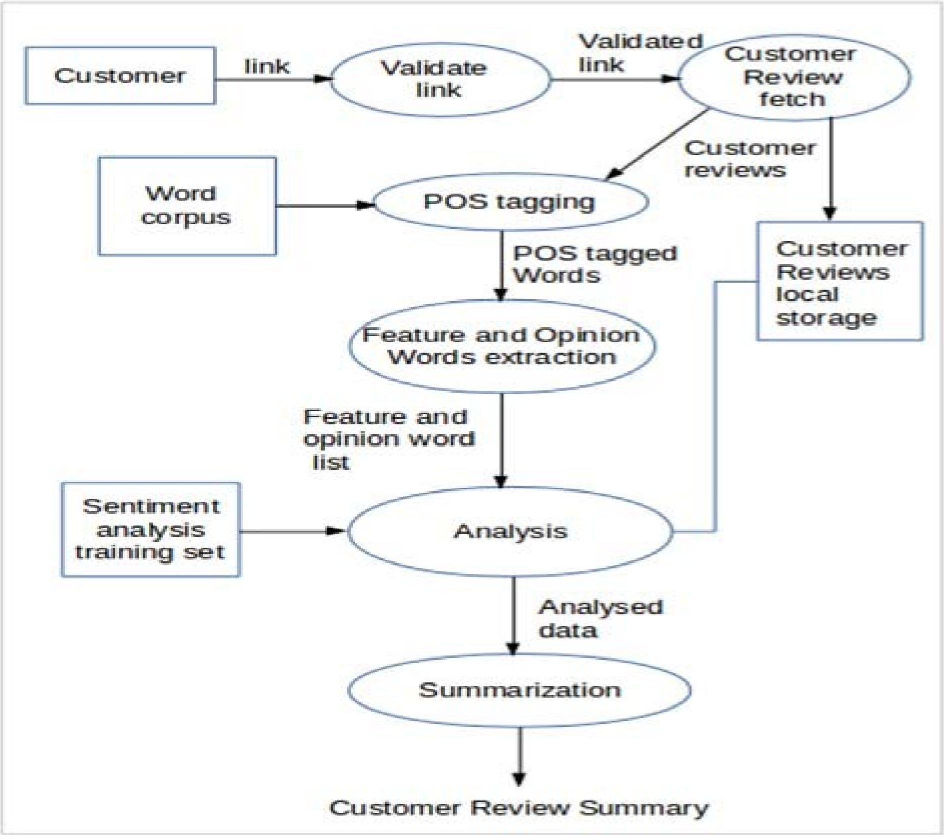


Figure 4.1 Basic Steps of Content Summarizer

4.1 <u>**Data Acquisition**</u>

- The data set is actually a mixture of the online amazon data available as well as the scraped data (made from scratch using Web Scraping). This data set consists of a few million Amazon customer reviews (input text) and star ratings (output labels) for learning the model for sentiment analysis and opinion mining. The idea here is a dataset is more than a toy - real business data on a reasonable scale - but can be trained in minutes on a modest laptop.[3]

4.1.1 Web Scraping

Web scraping is about downloading structured data from the web, selecting some of that data, and passing along what you selected to another process. Nowadays data is everything and if someone wants to get data from web pages then one way to use an API or implement Web Scraping techniques. In Python, Web scraping can be done easily by using scraping tools like BeautifulSoup. But what if the user is concerned about performance of scraper or need to scrape data efficiently.

To overcome this problem, one can make use of MultiThreading/Multiprocessing with BeautifulSoup module and he/she can create spider, which can help to crawl over a website and extract data. In order to save the time we use Scrapy. With the help of Scrapy one can :

1. Fetch millions of data efficiently
2. Run it on server
3. Fetching data
4. Run spider in multiple processes

Scrapy supports both versions of Python 2 and 3. If you’re using Anaconda, you can install the package from the conda-forge channel, which has up-to-date packages for Linux, Windows and OS X.

To install Scrapy using conda, run:

```
conda install -c conda-forge scrapy
```

Alternatively, if you’re on Linux or Mac OSX, you can directly install scrapy by:

```
pip install scrapy
```

Scrapy Shell

I love the python shell, it helps me "try out" things before I can implement them in detail. Similarly, scrapy provides a shell of its own that you can use to experiment. To start the scrapy shell in your command line type:

```
scrapy shell
```

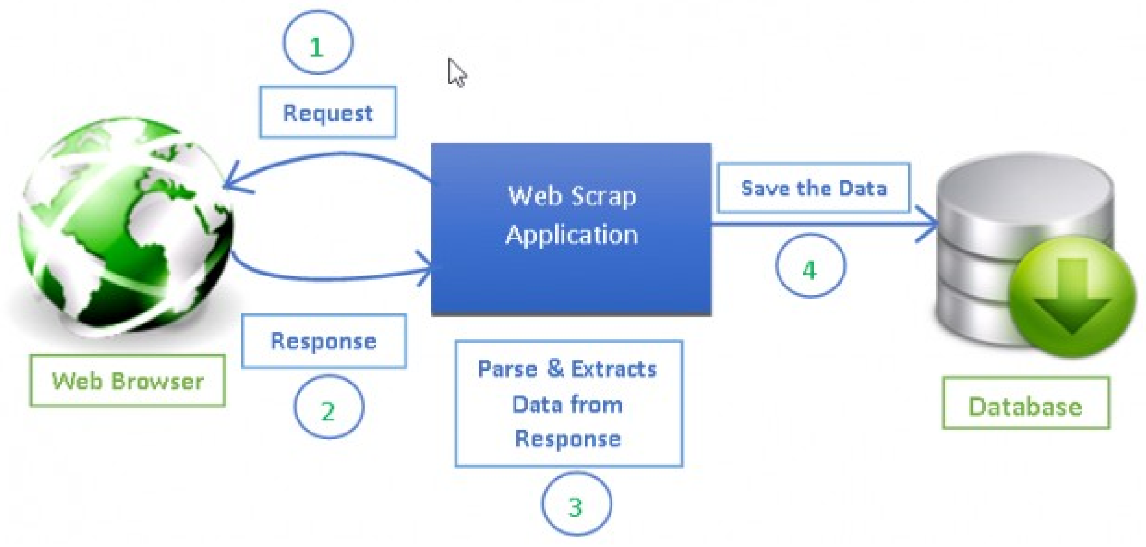


Figure 4.2 Web Scraping Process

You have to run a crawler on the web page using the fetch command in the Scrapy shell. A crawler or spider goes through a web page downloading its text and metadata.

```
fetch(https://www.amazon.in/slp/all-product/2k82po2tj6z984r)
```

The crawler returns a response which can be viewed by using the view(response) command on shell:

```
view(response)
```

You can view the raw HTML script by using the following command in Scrapy shell:

```
print(response.text)
```

Using CSS Selectors for Extraction

You can extract this using the element attributes or the css selector like classes. Write the following in the Scrapy shell to extract the product name:

```
response.css(".product::text").extract_first()
```

Using XPath for Extraction

XPath is a query language for selecting nodes in an XML document [7]. You can navigate through an XML document using XPath. Behind the scenes, Scrapy uses Xpath to navigate to HTML document items. The CSS selectors you used above are also converted to XPath, but in many cases, CSS is very easy to use. But you should know how the XPath in Scrapy works.

Creating a Scrapy project and Custom Spider

Web scraping can be used to make an aggregator that you can use to compare data. For example, you want to buy a tablet, and you want to compare products and prices together you can crawl your desired pages and store in an excel file. Here you will be scraping aliexpress.com for tablets information.[4]

Now, you will create a custom spider for the same page. First, you need to create a Scrapy project in which your code and results will be stored. Write the following command in the command line or anaconda prompt.

```
scrapy startproject  ReviewScraper
```

This will create a hidden folder in your default python or anaconda installation. ReviewScraper will be the name of the folder. You can give any name. You can view the folder contents directly through explorer. Following is the structure of the folder:

file/folder	**Purpose**
scrapy.cfg	deploy configuration file
ReviewScraper/	Project's Python module, you'll import your code from here
__init.py__	Initialization file
items.py	project items file
pipelines.py	project pipelines file
settings.py	project settings file
spiders/	a directory where you'll later put your spiders

file/folder	**Purpose**
__init.py__	Initialization file

Table 4.1 Scrapy File Structure

4.1.2 Other Data Sources

The non-categorical or the content for the reviews modeling and analysis of the Reviews is obtained from Amazon Website as well as Kaggle. KAGGLE **i**s an online community of data scientists and machine learners, owned by Google LLC. Kaggle allows users to find and publish data sets, explore and build models in a web-based data-science environment, work with other data scientists and machine learning engineers, and enter competitions to solve data science challenges. Kaggle got its start by offering machine learning competitions and now also offers a public data platform, a cloud-based workbench for data science, and short form AI education.

4.2 **Data Cleaning**

Performing basic preprocessing steps is very important before we get to the model building part. Using messy and uncleaned text data is a potentially disastrous move. So in this step, we will drop all the unwanted symbols, characters, etc. from the text that do not affect the objective of our problem.

Here is the dictionary that we will use for expanding the contractions:

contraction_mapping = {"ain't": "is not", "aren't": "are not","can't": "cannot", "'cause": "because", "could've": "could have", "couldn't": "could not", "didn't": "did not", "doesn't": "does not", "don't": "do not", "hadn't": "had not", "hasn't": "has not", "haven't": "have not","he'd": "he would","he'll": "he will", "he's": "he is", "how'd": "how did", "how'd'y": "how do you", "how'll": "how will", "how's": "how is", “I'd": "I would", "I'd've": "I would have", "I'll": "I will", "I'll've": "I will have","I'm": "I am", "I've": "I have", "i'd": "i would", “i'd've": "i would have", "i'll": "i will", "i'll've": "i will have","i'm": "i am", "i've": "i have", "isn't": "is not", "it'd": "it would", "it'd've": "it would have", "it'll": "it will", "it'll've": "it will have","it's": "it is", "let's": "let us", "ma'am": "madam", "mayn't": "may not", "might've": "might have","mightn't": "might not","mightn't've": "might not have", "must've": "must have", "mustn't": "must not", "mustn't've": "must not have", "needn't": "need not", "needn't've": "need not have","o'clock": "of the clock", "oughtn't": "ought not", "oughtn't've": "ought not have", "shan't": "shall not", "sha'n't": "shall not", "shan't've": "shall not have", "she'd": "she would", "she'd've": "she would have", "she'll": "she will", "she'll've": "she will have", "she's": "she is", "should've": "should have", "shouldn't": "should not", "shouldn't've": "should not have", "so've": "so have","so's": "so as", "this's": "this is","that'd": "that would", "that'd've": "that would have", "that's": "that is", "there'd": "there would", “there'd've": "there would have", "there's": "there is", "here's": "here is","they'd": "they would", "they'd've": "they would have", "they'll": "they will", "they'll've": "they will have", "they're": "they are", "they've": "they have", "to've": "to have", "wasn't": "was not", "we'd": "we would", "we'd've": "we would have", "we'll": "we will", "we'll've": "we will have", "we're": "we are", "we've": "we have", "weren't": "were not", "what'll": "what will", "what'll've": "what will have", "what're": "what are", "what's": "what is", "what've": "what have", "when's": "when is", "when've": "when have", "where'd": "where did", "where's": "where is", “where've": "where have", "who'll": "who will", "who'll've": "who will have", "who's": "who is", "who've": "who have", "why's": "why is", "why've": "why have", "will've": "will have", "won't": "will not", "won't've": "will not have", "would've": "would have", "wouldn't": "would not", "wouldn't've": "would not have", "y'all": "you all", "y'all'd": "you all would","y'all'd've": "you all would have","y'all're": "you all are","y'all've": "you all have", “you'd": "you would", "you'd've": "you would have", "you'll": "you will", "you'll've": "you will have","you're": "you are", "you've": "you have"}

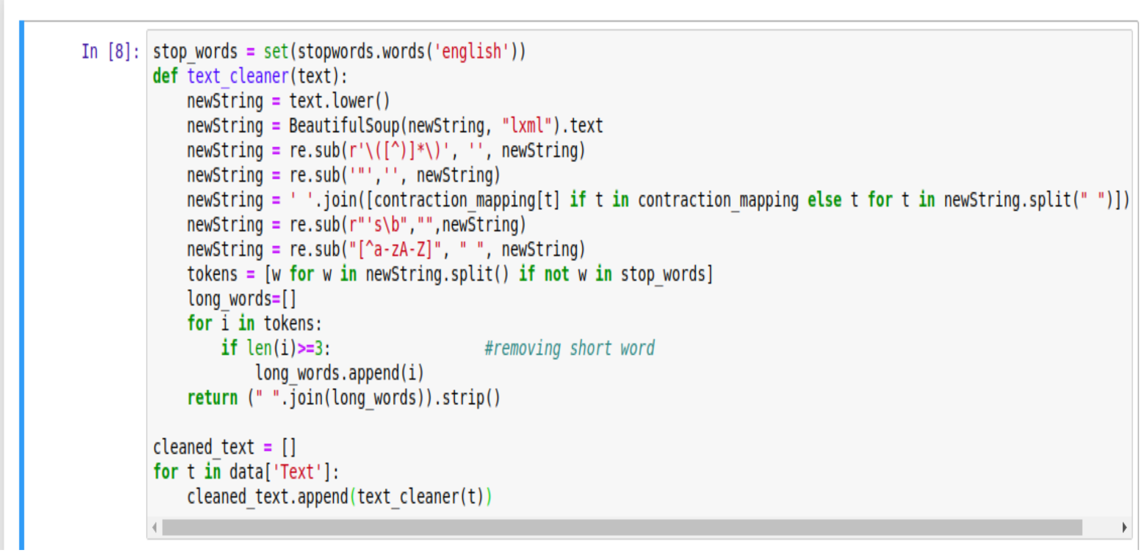
Next, We need to define two different functions for preprocessing the reviews and generating the summary since the preprocessing steps involved in text and summary differ slightly.

4.2.1 Text Cleaning

We will perform the below preprocessing tasks for our data:

- Convert everything to lowercase
- Remove HTML tags

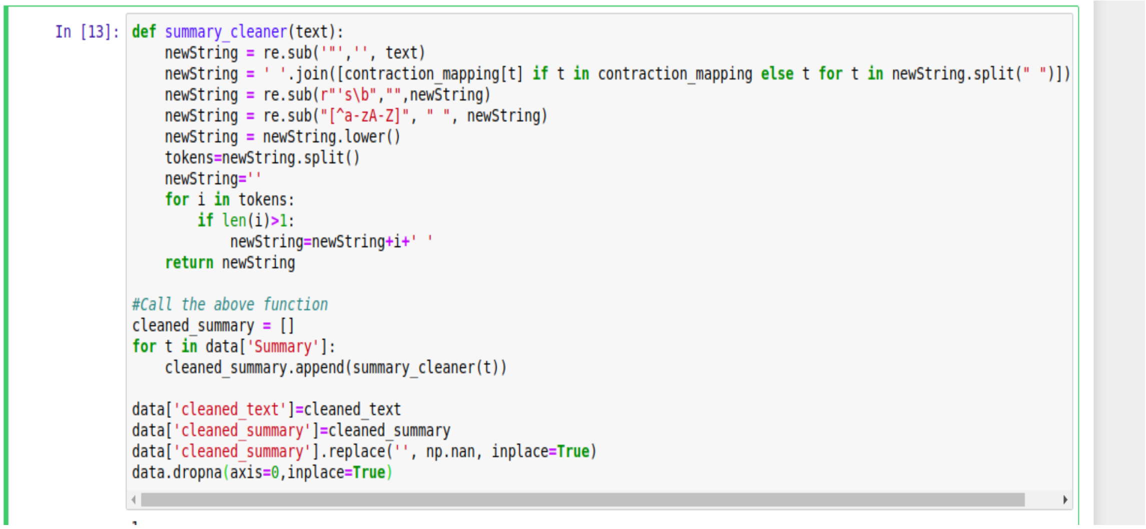
- Contraction mapping
- Remove (‘s)
- Remove any text inside the parenthesis ( )
- Eliminate punctuations and special characters
- Remove stopwords
- Remove short words



```
stop_words = set(stopwords.words('english'))
def text_cleaner(text):
    newString = text.lower()
    newString = BeautifulSoup(newString, "lxml").text
    newString = re.sub(r'\([^)]*\)', '', newString)
    newString = re.sub('"','', newString)
    newString = ' '.join([contraction_mapping[t] if t in contraction_mapping else t for t in newString.split(" ")])
    newString = re.sub(r"'s\b","",newString)
    newString = re.sub("[^a-zA-Z]", " ", newString)
    tokens = [w for w in newString.split() if not w in stop_words]
    long_words=[]
    for i in tokens:
        if len(i)>=3:                  #removing short word
            long_words.append(i)
    return (" ".join(long_words)).strip()

cleaned_text = []
for t in data['Text']:
    cleaned_text.append(text_cleaner(t))
```

Figure 4.3 Text and Dataset cleaning code

4.2.2 Summary Cleaning

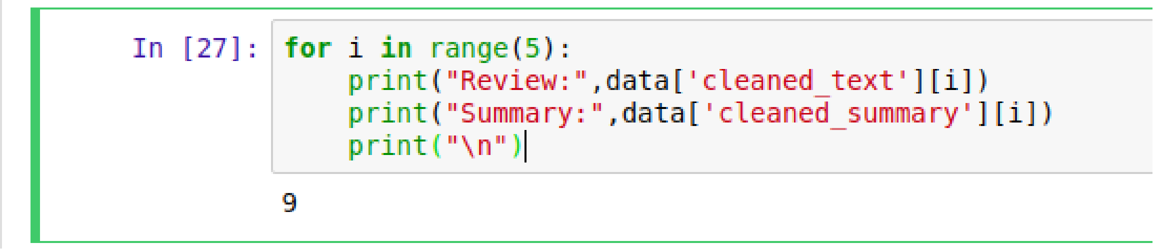


```
In [13]: def summary_cleaner(text):
             newString = re.sub('"','', text)
             newString = ' '.join([contraction_mapping[t] if t in contraction_mapping else t for t in newString.split(" ")])
             newString = re.sub(r"'s\b","",newString)
             newString = re.sub("[^a-zA-Z]", " ", newString)
             newString = newString.lower()
             tokens=newString.split()
             newString=''
             for i in tokens:
                 if len(i)>1:
                     newString=newString+i+' '
             return newString

         #Call the above function
         cleaned_summary = []
         for t in data['Summary']:
             cleaned_summary.append(summary_cleaner(t))

         data['cleaned_text']=cleaned_text
         data['cleaned_summary']=cleaned_summary
         data['cleaned_summary'].replace('', np.nan, inplace=True)
         data.dropna(axis=0,inplace=True)
```

Figure 4.4 Summary or Target Dataset Cleaning code

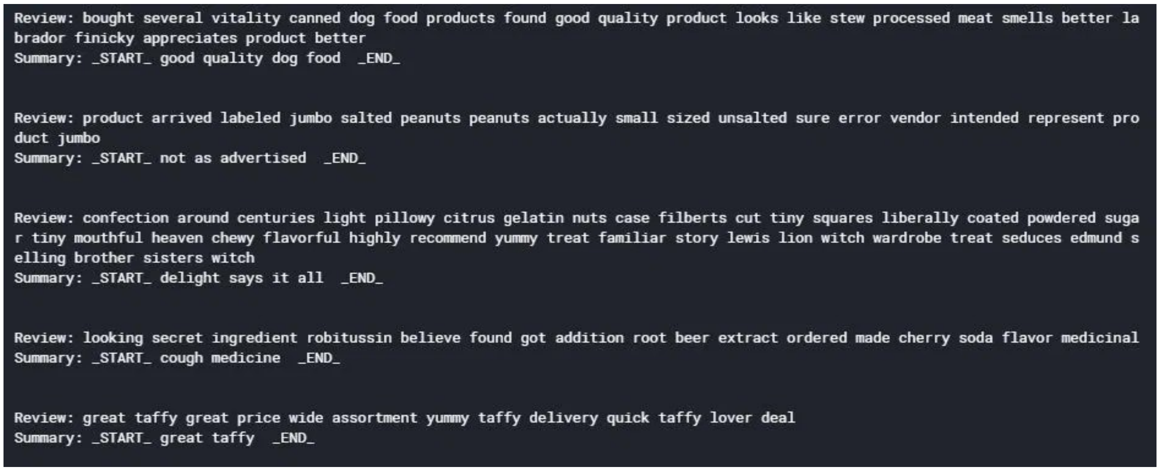
Remember to add the START and END special tokens at the beginning and end of the summary:

data['cleaned_summary'] = data['cleaned_summary'].apply(lambda x : '_START_ '+ x + ' _END_')



```
In [27]: for i in range(5):
             print("Review:",data['cleaned_text'][i])
             print("Summary:",data['cleaned_summary'][i])
             print("\n")

         9
```

Figure 4.5 Output of Cleaning dataset

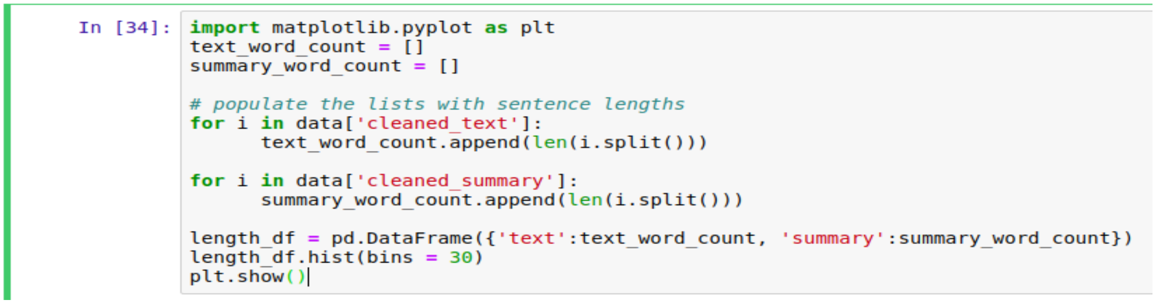


```
Review: bought several vitality canned dog food products found good quality product looks like stew processed meat smells better la
brador finicky appreciates product better
Summary: _START_ good quality dog food  _END_


Review: product arrived labeled jumbo salted peanuts peanuts actually small sized unsalted sure error vendor intended represent pro
duct jumbo
Summary: _START_ not as advertised  _END_


Review: confection around centuries light pillowy citrus gelatin nuts case filberts cut tiny squares liberally coated powdered suga
r tiny mouthful heaven chewy flavorful highly recommend yummy treat familiar story lewis lion witch wardrobe treat seduces edmund s
elling brother sisters witch
Summary: _START_ delight says it all  _END_


Review: looking secret ingredient robitussin believe found got addition root beer extract ordered made cherry soda flavor medicinal
Summary: _START_ cough medicine  _END_


Review: great taffy great price wide assortment yummy taffy delivery quick taffy lover deal
Summary: _START_ great taffy  _END_
```

Figure 4.6 Output of above code

4.2.3 Understanding the distribution of the sequences

Here, we will analyze the length of the reviews and the summary to get an overall idea about the distribution of length of the text. This will help us fix the maximum length of the sequence:



```
In [34]: import matplotlib.pyplot as plt
         text_word_count = []
         summary_word_count = []

         # populate the lists with sentence lengths
         for i in data['cleaned_text']:
               text_word_count.append(len(i.split()))

         for i in data['cleaned_summary']:
               summary_word_count.append(len(i.split()))

         length_df = pd.DataFrame({'text':text_word_count, 'summary':summary_word_count})
         length_df.hist(bins = 30)
         plt.show()
```

Figure 4.7 Code for sequence distrubution

OUTPUT:

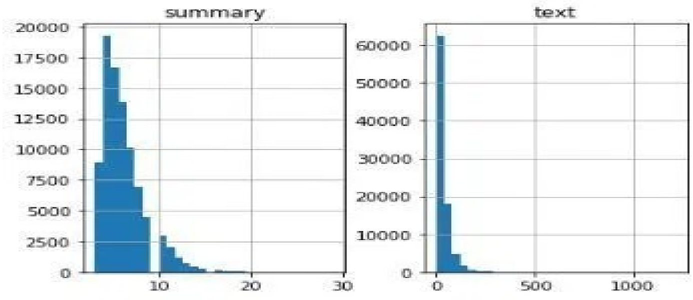


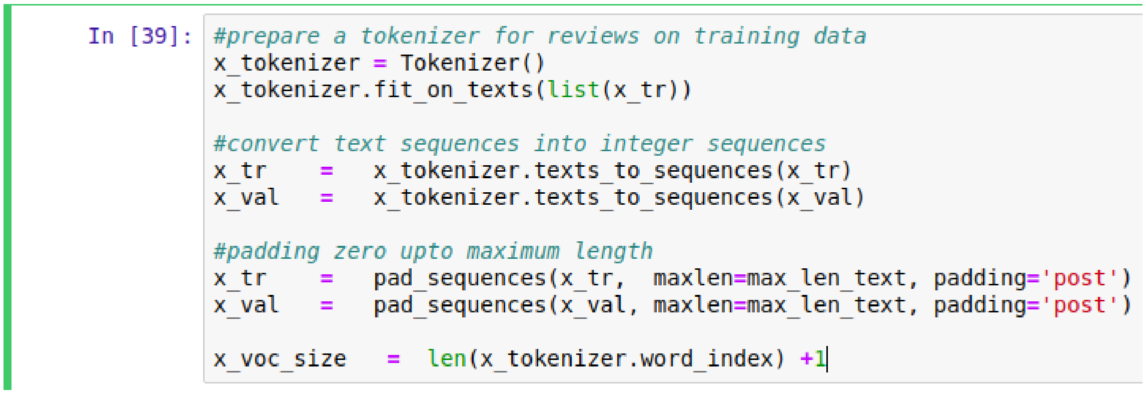
Figure 4.8 Output plot of distribution of sequences

4.3 Preparing the Tokenizer

A tokenizer builds the vocabulary and converts a word sequence to an integer sequence. Go ahead and build tokenizers for text and summary:

a) Text Tokenizer

b) Summary Tokenizer



In [39]:

```
#prepare a tokenizer for reviews on training data
x_tokenizer = Tokenizer()
x_tokenizer.fit_on_texts(list(x_tr))

#convert text sequences into integer sequences
x_tr    =   x_tokenizer.texts_to_sequences(x_tr)
x_val   =   x_tokenizer.texts_to_sequences(x_val)

#padding zero upto maximum length
x_tr    =   pad_sequences(x_tr,  maxlen=max_len_text, padding='post')
x_val   =   pad_sequences(x_val, maxlen=max_len_text, padding='post')

x_voc_size   =  len(x_tokenizer.word_index) +1
```

Figure 4.9 Code for Tokenizer

4.4 Model building

We are finally at the model building part. But before we do that, we need to familiarize ourselves with a few terms which are required prior to building the model.

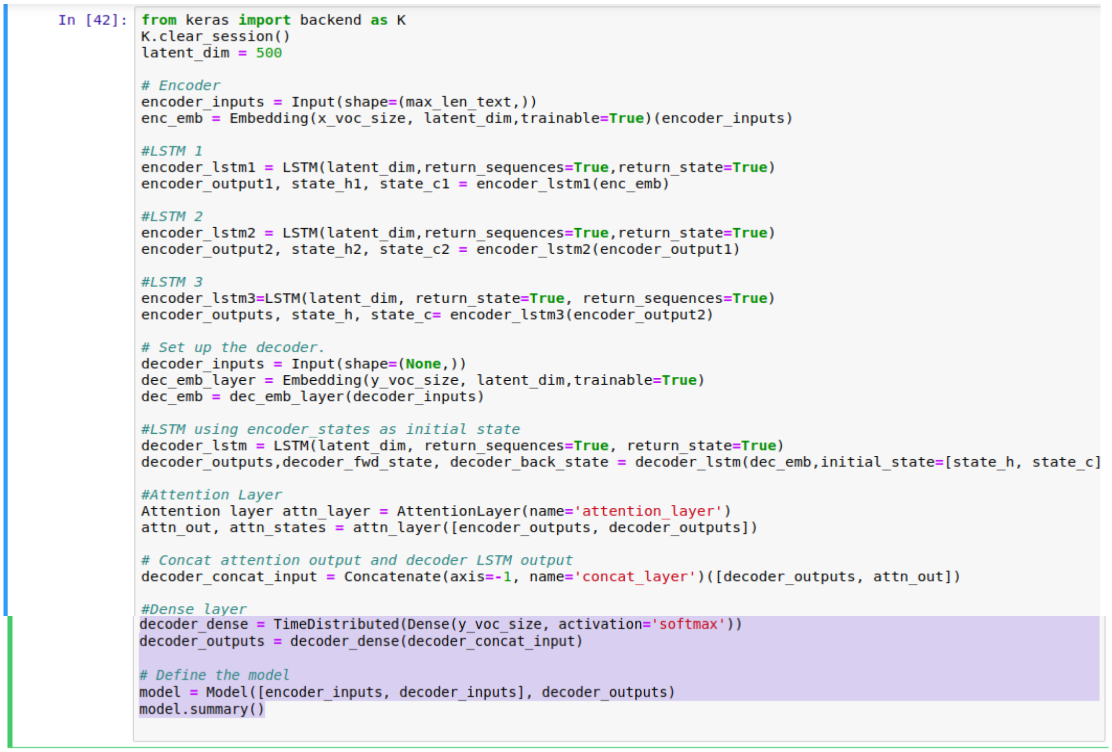
- **Return Sequences = True:** When the return sequences parameter is set to **True**, LSTM produces the hidden state and cell state for every timestep

- **Return State = True:** When return state = **True**, LSTM produces the hidden state and cell state of the last timestep only

- **Initial State:** This is used to initialize the internal states of the LSTM for the first timestep

- **Stacked LSTM:** Stacked LSTM has multiple layers of LSTM stacked on top of each other. This leads to a better representation of the sequence. I encourage you to experiment with the multiple layers of the LSTM stacked on top of each other (it’s a great way to learn this)

Here, we are building a 3 stacked LSTM for the encoder:



```
In [42]: from keras import backend as K
         K.clear_session()
         latent_dim = 500

         # Encoder
         encoder_inputs = Input(shape=(max_len_text,))
         enc_emb = Embedding(x_voc_size, latent_dim,trainable=True)(encoder_inputs)

         #LSTM 1
         encoder_lstm1 = LSTM(latent_dim,return_sequences=True,return_state=True)
         encoder_output1, state_h1, state_c1 = encoder_lstm1(enc_emb)

         #LSTM 2
         encoder_lstm2 = LSTM(latent_dim,return_sequences=True,return_state=True)
         encoder_output2, state_h2, state_c2 = encoder_lstm2(encoder_output1)

         #LSTM 3
         encoder_lstm3=LSTM(latent_dim, return_state=True, return_sequences=True)
         encoder_outputs, state_h, state_c= encoder_lstm3(encoder_output2)

         # Set up the decoder.
         decoder_inputs = Input(shape=(None,))
         dec_emb_layer = Embedding(y_voc_size, latent_dim,trainable=True)
         dec_emb = dec_emb_layer(decoder_inputs)

         #LSTM using encoder_states as initial state
         decoder_lstm = LSTM(latent_dim, return_sequences=True, return_state=True)
         decoder_outputs,decoder_fwd_state, decoder_back_state = decoder_lstm(dec_emb,initial_state=[state_h, state_c]

         #Attention Layer
         Attention layer attn_layer = AttentionLayer(name='attention_layer')
         attn_out, attn_states = attn_layer([encoder_outputs, decoder_outputs])

         # Concat attention output and decoder LSTM output
         decoder_concat_input = Concatenate(axis=-1, name='concat_layer')([decoder_outputs, attn_out])

         #Dense layer
         decoder_dense = TimeDistributed(Dense(y_voc_size, activation='softmax'))
         decoder_outputs = decoder_dense(decoder_concat_input)

         # Define the model
         model = Model([encoder_inputs, decoder_inputs], decoder_outputs)
         model.summary()
```

Figure 4.10 Model Building

Concept of early stopping: It is used to stop training the neural network at the right time by monitoring a user-specified metric.

4.5 <u>Understanding the Encoder-Decoder Architecture</u>

Let’s understand this from the perspective of text summarization. The input is a long sequence of words and the output will be a short version of the input sequence.

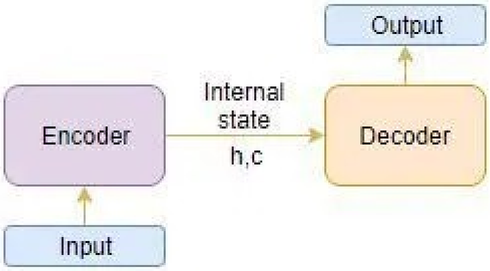


Figure 4.11 Basic Encoder Decoder Architecture

Generally, variants of Recurrent Neural Networks (RNNs), i.e. Gated Recurrent Neural Network (GRU) or Long Short Term Memory (LSTM), are preferred as the encoder and decoder components. This is because they are capable of capturing long term dependencies by overcoming the problem of vanishing gradient.

We can set up the Encoder-Decoder in 2 phases:

- Training phase
- Inference phase

Let’s understand these concepts through the lens of an LSTM model.

4.5.1 Training phase

In the training phase, we will first set up the encoder and decoder. We will then train the model to predict the target sequence offset by one timestep. Let us see in detail on how to set up the encoder and decoder.

4.5.2 Encoder

An Encoder Long Short Term Memory model (LSTM) reads the entire input sequence wherein, at each timestep, one word is fed into the encoder. It then processes the information at every timestep and captures the contextual information present in the input sequence.

I’ve put together the below diagram which illustrates this process:

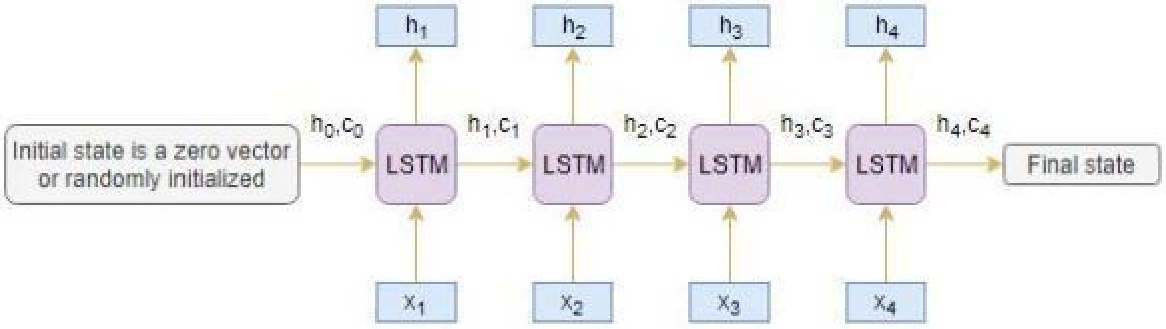


Figure 4.12 LSTM Based Encoder Architechture

The hidden state (hi) and cell state (ci) of the last time step are used to initialize the decoder. Remember, this is because the encoder and decoder are two different sets of the LSTM architecture.

4.5.3 Decoder

The decoder is also an LSTM network which reads the entire target sequence word-by-word and predicts the same sequence offset by one timestep. The decoder is trained to predict the next word in the sequence given the previous word.

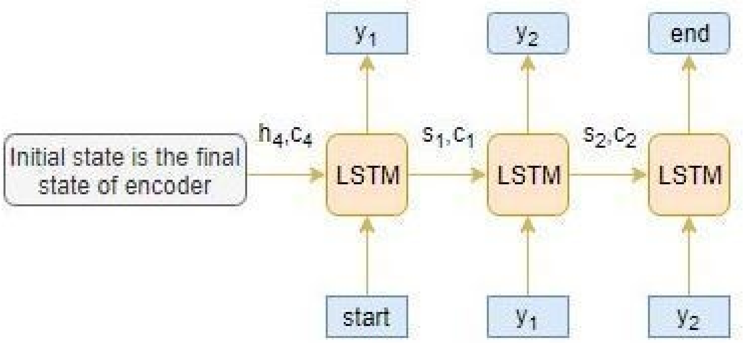


Figure 4.13 LSTM Based Decoder Architechture

<start> and <end> are the special tokens which are added to the target sequence before feeding it into the decoder. The target sequence is unknown while decoding the test sequence. So, we start predicting the target sequence by passing the first word into the decoder which would be always the <start> token. And the <end> token signals the end of the sentence.

4.5.4 Inference Phase

After training, the model is tested on new source sequences for which the target sequence is unknown. So, we need to set up the inference architecture to decode a test sequence:

Figure 4.13 Combined LSTM Based Encoder Decoder Architechture

How does the inference process work?

Here are the steps to decode the test sequence:

1.Encode the entire input sequence and initialize the decoder with internal states of the encoder

2.Pass <start> token as an input to the decoder

3.Run the decoder for one timestep with the internal states

4.The output will be the probability for the next word. The word with the maximum probability will be selected

5.Pass the sampled word as an input to the decoder in the next timestep and update the internal states with the current time step

6.Repeat steps 3 – 5 until we generate <end> token or hit the maximum length of the target sequence

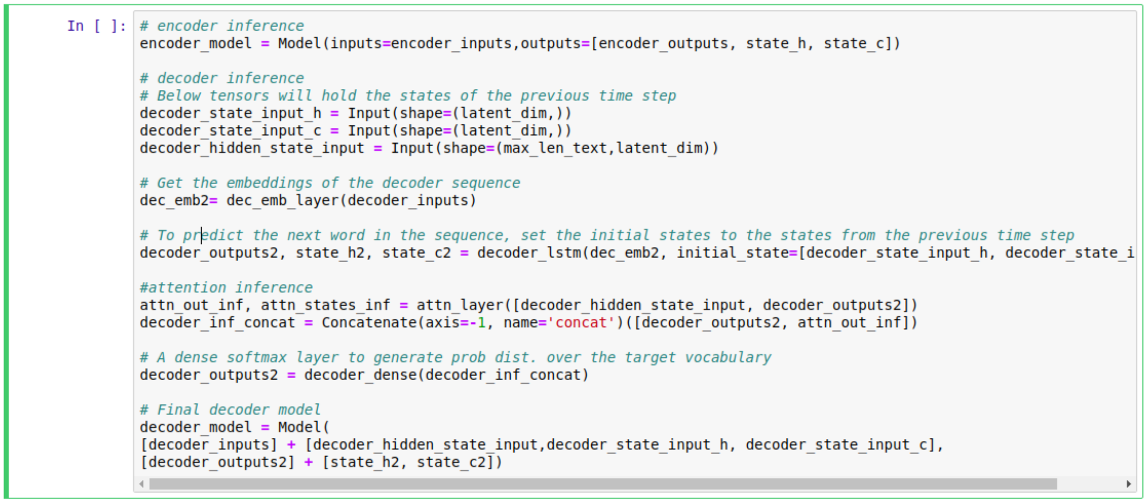
Let’s take an example where the test sequence is given by [x1, x2, x3, x4]. How will the inference process work for this test sequence? I want you to think about it before you look at my thoughts below.

1.Encode the test sequence into internal state vectors

2.Observe how the decoder predicts the target sequence at each timestep

4.5.5 Inference

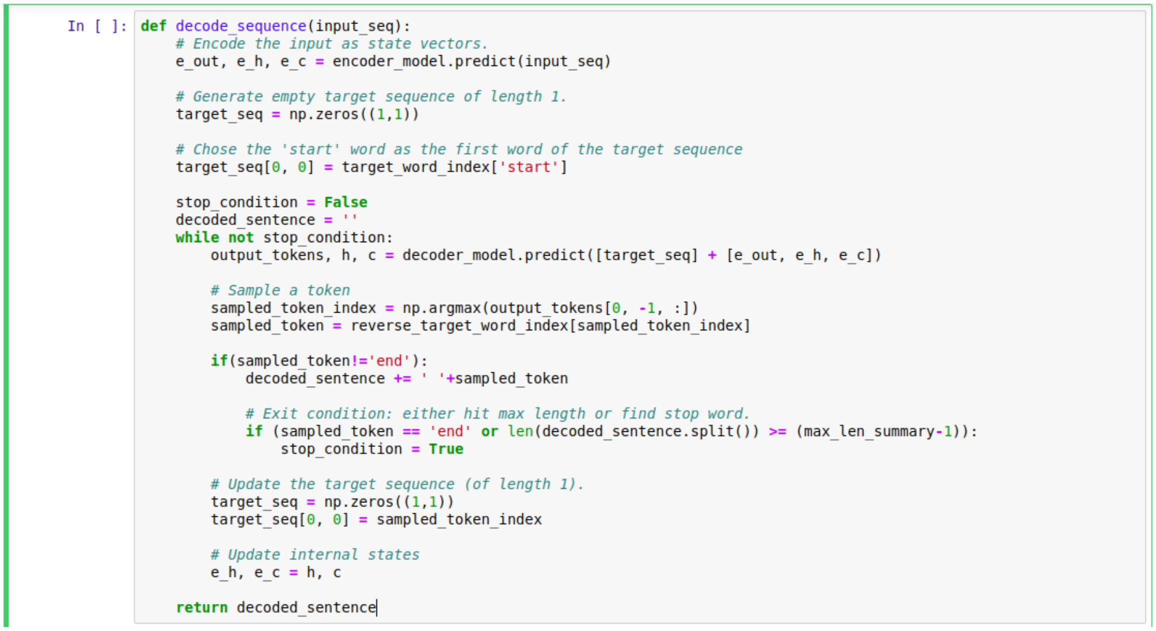
Set up the inference for the encoder and decoder:



```python
# encoder inference
encoder_model = Model(inputs=encoder_inputs,outputs=[encoder_outputs, state_h, state_c])

# decoder inference
# Below tensors will hold the states of the previous time step
decoder_state_input_h = Input(shape=(latent_dim,))
decoder_state_input_c = Input(shape=(latent_dim,))
decoder_hidden_state_input = Input(shape=(max_len_text,latent_dim))

# Get the embeddings of the decoder sequence
dec_emb2= dec_emb_layer(decoder_inputs)

# To predict the next word in the sequence, set the initial states to the states from the previous time step
decoder_outputs2, state_h2, state_c2 = decoder_lstm(dec_emb2, initial_state=[decoder_state_input_h, decoder_state_i

#attention inference
attn_out_inf, attn_states_inf = attn_layer([decoder_hidden_state_input, decoder_outputs2])
decoder_inf_concat = Concatenate(axis=-1, name='concat')([decoder_outputs2, attn_out_inf])

# A dense softmax layer to generate prob dist. over the target vocabulary
decoder_outputs2 = decoder_dense(decoder_inf_concat)

# Final decoder model
decoder_model = Model(
[decoder_inputs] + [decoder_hidden_state_input,decoder_state_input_h, decoder_state_input_c],
[decoder_outputs2] + [state_h2, state_c2])
```

Figure 4.14 Code for process of Inference

We are defining a function below which is the implementation of the inference process:



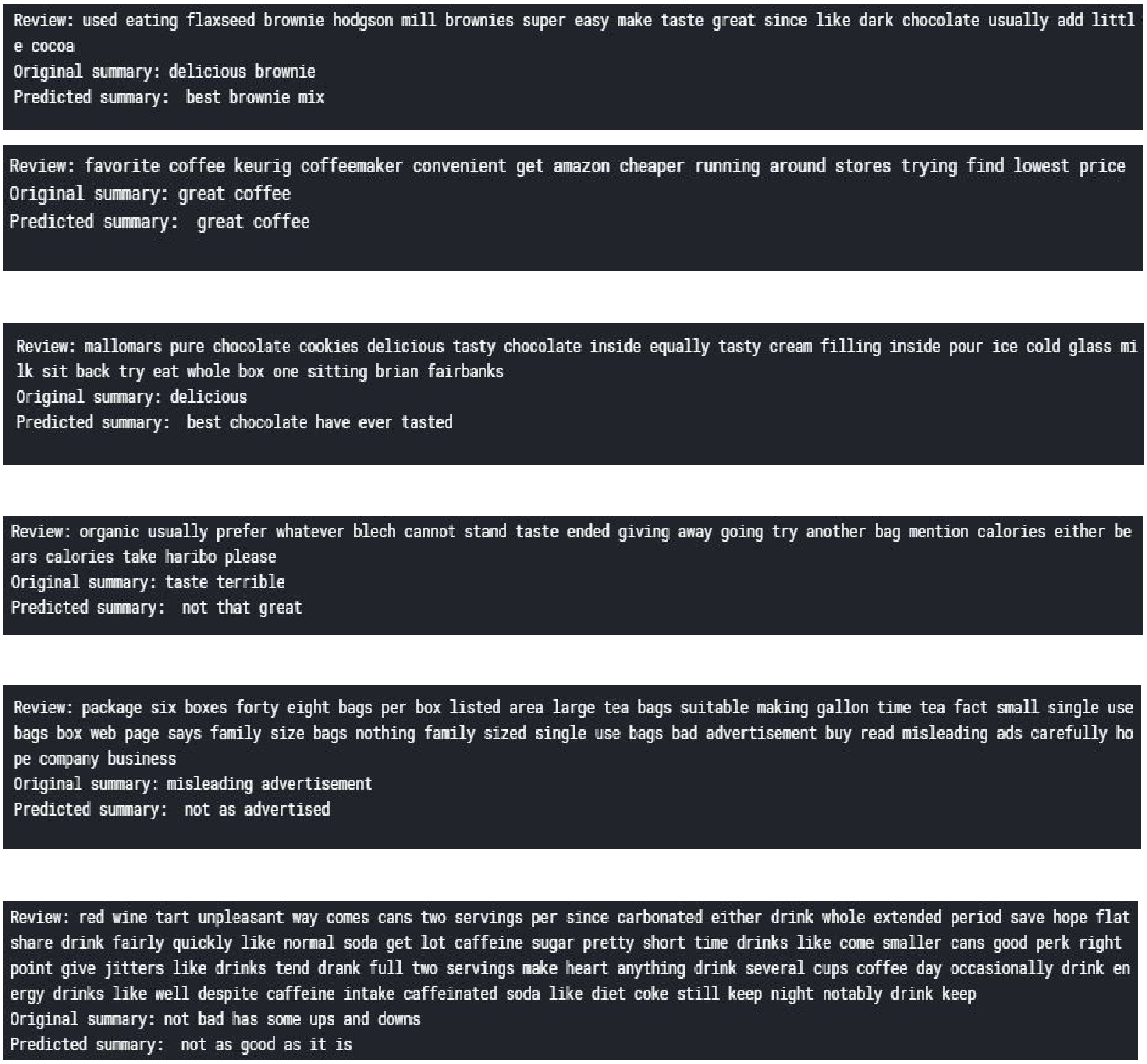
```python
def decode_sequence(input_seq):
    # Encode the input as state vectors.
    e_out, e_h, e_c = encoder_model.predict(input_seq)

    # Generate empty target sequence of length 1.
    target_seq = np.zeros((1,1))

    # Chose the 'start' word as the first word of the target sequence
    target_seq[0, 0] = target_word_index['start']

    stop_condition = False
    decoded_sentence = ''
    while not stop_condition:
        output_tokens, h, c = decoder_model.predict([target_seq] + [e_out, e_h, e_c])

        # Sample a token
        sampled_token_index = np.argmax(output_tokens[0, -1, :])
        sampled_token = reverse_target_word_index[sampled_token_index]

        if(sampled_token!='end'):
            decoded_sentence += ' '+sampled_token

            # Exit condition: either hit max length or find stop word.
            if (sampled_token == 'end' or len(decoded_sentence.split()) >= (max_len_summary-1)):
                stop_condition = True

        # Update the target sequence (of length 1).
        target_seq = np.zeros((1,1))
        target_seq[0, 0] = sampled_token_index

        # Update internal states
        e_h, e_c = h, c

    return decoded_sentence
```

Figure 4.15 Code for process of inference

Chapter - 5

RESULTS

After performing the complete process and creating a model for reviews summarization, we tested the model on various reviews on Amazon that were linked to the Electronic and fashion industry (as in our dataset) as well as some other reviews too:



```
Review: used eating flaxseed brownie hodgson mill brownies super easy make taste great since like dark chocolate usually add littl
e cocoa
Original summary: delicious brownie
Predicted summary:  best brownie mix
```

```
Review: favorite coffee keurig coffeemaker convenient get amazon cheaper running around stores trying find lowest price
Original summary: great coffee
Predicted summary:  great coffee
```

```
Review: mallomars pure chocolate cookies delicious tasty chocolate inside equally tasty cream filling inside pour ice cold glass mi
lk sit back try eat whole box one sitting brian fairbanks
Original summary: delicious
Predicted summary:  best chocolate have ever tasted
```

```
Review: organic usually prefer whatever blech cannot stand taste ended giving away going try another bag mention calories either be
ars calories take haribo please
Original summary: taste terrible
Predicted summary:  not that great
```

```
Review: package six boxes forty eight bags per box listed area large tea bags suitable making gallon time tea fact small single use
bags box web page says family size bags nothing family sized single use bags bad advertisement buy read misleading ads carefully ho
pe company business
Original summary: misleading advertisement
Predicted summary:  not as advertised
```

```
Review: red wine tart unpleasant way comes cans two servings per since carbonated either drink whole extended period save hope flat
share drink fairly quickly like normal soda get lot caffeine sugar pretty short time drinks like come smaller cans good perk right
point give jitters like drinks tend drank full two servings make heart anything drink several cups coffee day occasionally drink en
ergy drinks like well despite caffeine intake caffeinated soda like diet coke still keep night notably drink keep
Original summary: not bad has some ups and downs
Predicted summary:  not as good as it is
```

Figure 5.1 Individual Reviews Summary

Tested Reviews and their outputs:

Reviews	Summary
This sound track was beautiful! It paints the senery in your mind so well I would recomend it even to people who hate vid. game music! I have played the game Chrono Cross but out of all of the games I have ever played it has the best music!	Stuning even for the non-gamer
I loved Whisper of the wicked saints. The story was amazing and I was pleasantly surprised at the changes in the book. I am not normaly someone who is into romance novels, but the world was raving about this book and so I bought it.	Glorious Story
I bought this to replace a 13" tube tv in my kitchen. It was a larger screen plus relcaiming the counter space. The picture is not good and not as expected. Bad Quality also.	Tube TV in kitchen not good quality
Bought this camera for Christmas. Had her open it first so she could film the nights party & NO TAPES ARE INCLUDED! Of course all stores are closed Christmas day so the thing is USELESS until the day after. Shame on Sony. How much would a tape included in the packace cost them ?	Costly sony product and useless
My daughter wanted this for a few months before I finally gave in and bought it for her. She was so excited. We got home and played it for one hour hoping that something different or better would happen. The game was so so boring. Waste of time and money	Different hour game boring and waste

Table 5.1 Review Summary for Products of other category

As we can see above that the model was quite efficient in creating the summary of a particular review as we fed it in the model. Although, it was seen above that the model is not able to handle negation or out of the context sentences very well. Now, we have the results shown from overall review analysis using this model:

Product ASIN	Product Title		Outputs	Polarity
B071FF8LMK	Allen Solly Men's Polo		Very soft on heel. Comfortable to walk. Height increasing. Poor quality. Only half part of sole can be used.	>0
B07FQX8C3K	Amazon Brand - Symbol Men's Plain Formal Shirt	-	Formal Shirt at cheap price. Poor quality. Smelling bad	<0
B014VFXU1U	Clarks Women's Leather Pumps	-	Fitting is not perfect. Duplicate product. Cheap	<0
B018TM9FWY	Combo Of Belt,Watch,Sunglass,Wallet And Cardholder Gift Pack Of 5	-	Good product as shown. Belt and watch is ok. Best Complimentary gift. Watch not working.	>0

Table 5.2 Review Summarization for product inputs

Chapter - 6

COMPARISON AND ANALYSIS

6.1 BRIEF STUDY OF OTHER SUMMARIZATION SYSTEMS

We look at the other summarization systems that are available today in the market with their disadvantages.

Automatic summarization systems

Automatically summarizes a single text file into a paragraph by picking up random sentences. The drawbacks of such applications are there is no analysis is performed on the content and there is no application on Customer reviews.

Summarization systems that use feature selection methods

They use feature selection techniques such as Document Frequency Thresholding (DFT), Information Gain (IG) and Mutual Information (MI). The drawback being there is no application on customer reviews.

Mining and summarization of customer reviews

Mine the features of the product on which the customers have expressed their opinions. The drawback being there is no ready application present for use of customers.

Opinion Observer:

Analysing and comparing opinions on the web. Provides a comparative analysis on the strengths and weaknesses of products in terms of various product features. It basically uses machine learning algorithms. The drawback being that it is a comparison methodology that compares products across the web which is not the objective of our project.

6.2 Existing vs Proposed System

In order to know the genuineness and quality of the products online, the users, as a matter of fact, tend to go through the customer reviews and decide based on those reviews. Sometimes it is time consuming as there are hundreds and thousands of reviews. As a result, users might miss out on some critical reviews. In the proposed system, we are able to get the reviews summary not only by the complete review text but also by just specifying the name or Product ASIN or product link of the desired review of the product. We have also used the concept of Inference approach and tried to do both

Extractive and Abstractive Summarization. This is a very interesting approach. Here, we generate new sentences from the original text. This is in contrast to the extractive approach we saw earlier where we used only the sentences that were present. The sentences generated through abstractive summarization might not be present in the original text. The next new approach that is included in this system is Attention Mechanism. Let’s consider a simple example to understand how Attention Mechanism works:

- Source sequence: “Which sport do you like the most?

- Target sequence: “I love cricket”

The first word ‘I’ in the target sequence is connected to the fourth word ‘you’ in the source sequence, right? Similarly, the second-word ‘love’ in the target sequence is associated with the fifth word ‘like’ in the source sequence.

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.

Chapter - 7

CONCLUSION AND FUTURE SCOPE

7.1 Final Conclusion

After studying the existing systems, we conclude that our solution provides a more realistic and efficient summarization of user opinions. Also building an application that uses this solution has brought about its use in an apt manner. This can be scaled into other domains data analytics applications that concern analysis of raw text data and summarization.

7.2 Future Scope

There’s a lot more we can do to play around and experiment with the model:

- First of all, we can try to increase the training dataset size and build the model.
- The generalization capability of a deep learning model enhances with an increase in the training dataset size.
- Also, we can try implementing Bi-Directional LSTM which is capable of capturing the context from both the directions and results in a better context vector.
- Use the beam search strategy for decoding the test sequence instead of using the greedy approach (argmax).
- Evaluating the performance of your model based on the BLEU score.
- Also, we can implement pointer-generator networksand coverage mechanisms.

Chapter - 8

REFERENCES

For Links:

1. https://www.analyticsvidhya.com/blog/2019/06/comprehensive-guide-text-summarization-using-deep-learning-python/

2. https://towardsdatascience.com/understand-text-summarization-and-create-your-own-summarizer-in-python-b26a9f09fc70

3. https://towardsdatascience.com/text-summarization-in-python-76c0a41f0dc4

4. https://towardsdatascience.com/text-summarization-with-amazon-reviews-41801c2210b

5. https://dziganto.github.io/cross-validation/data%20science/machine%20learning/model%20tuning/python/Model-Tuning-with- Validation-and-Cross-Validation/

For Research Papers and Thesis:

6. Summarization of Customer Reviews for a Product on a website using Natural Language Processing: Conference Paper · September 2016, DOI: 10.1109/ICACCI.2016.7732392

7. Get To The Point: Summarization with Pointer-Generator Networks: Abigail See, Stanford University, axHiv039555.1704.04368v2 April 2017

For Articles:

8. https://pdfs.semanticscholar.org/d6be/438df373dedf06d6c062596d4e40f59b022c.pdf

9. https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/

10. https://www.geeksforgeeks.org/