University of Essex

Project Report for CE903
Requirement Specification

Automatic Title Generation Using Encoder-Decoder Models

Team-9

Group Members: Karan Bhatt (2112102), Dhaval Patel (2111480), Ashish Gajera (2111045),
Vatsal Trivedi (2111154), Vipul Barot (2111824), Parth Patel (2111711)
Niaz Muhammad Umair (2111659),

Supervisor: Dr. Yu, Junto

Department of Computer Science and Electronic Engineering
May 24, 2022

Contents

1 Introduction

Title is the short capture summery of whole article and people always use it for search the article. However, a good title creates good impression of your article, our motive is to generate title from the given text data using python. In this document we discuss about our plan and method to execute this project in step by step.

Every researcher needs a good title for article, in which the summarization of whole article explained in short words. Title decide the what and which things are including in this article, by reading title person can imagine and understand the motive of the article.

Title generation using encoder-decoder system is where the complex text model feed as it is input data for capturing the meaning of the source input data. Which make easy abstractive text summarization.

1.1 Study Background

Title is focus part of any article. It makes long article content in a few words and captures readers' attention. A good title for a research article is the one which is able to introduce the research work to the full extent, but in a simple manner. Writing good titles that are informative and attractive is a challenging task. This project describes the importance of titles and the methods of creating titles for research papers.

Automatic title generate from text is unsupervised learning method and in this we don't required to perform any task after feeding input text to the machine. Machine or Local computer automatic take decision and give title as output.

We are going to use "arxiv.org" for sample title generation. On this website many research paper available. We ae use that research paper to perform title generation. One web application is construct for this in which we can execute python code.

1.2 Architecture of Development

Advance machine learning methods used to create informative title for article so it will easy for researchers as well for students who can easily find that documents by using key words. While doing this project we are going to use research papers of "arxiv.org". we generate new title for given abstract.

Use of encoding-decoding method help to deal with special characters and symbols with different language like Chinese, Korean, etc. we clean the given data and then encode it first, then sequence of byte data is maximum use is separate which are the key words of given dataset. Then it going to be decode at the end so we got the correct title of given text.

Web application using python create for this task where we can perform text input box and submit button to perform whole task. Flask is used to create web application in python.

Transformer is library is use for text summarize. Data clean and data train created for our NLP task. LSTM, RNN model will be use as machine learning method for title generate.

1.3 Summary

Text summarization is a problem in natural language processing of creating a short, accurate, and fluent summary of a source document. The Encoder-Decoder recurrent neural network architecture developed for machine translation has proven effective when applied to the problem of text summarization. It can be difficult to apply this architecture in the deep learning library, given some of the flexibility sacrificed to make the library clean, simple, and easy to use. In this application, we will discover how to implement the Encoder-Decoder architecture for text summarization in this app.

2 Requirements Analysis and Specification

2.1 Methodology

In general, there is two method of text summarize, Abstractive Summarization and Extractive Summarization. We are going to perform Abstractive Summarization. We are going to perform different machine learning method to implement this task and then after decide which one is good for title generation, the list is as bellow.

a) Model using Spacy
b) RNN
c) LSTM
d) LSTM with beam search

The reason we encode text is that computers do not understand characters, words, or sentences; Computers can only process numbers, therefore, if we wish to teach computers to understand natural language then we ought to encode the textual data into a numeric format before inputting it into any machine.

Therefore, text encoding may be defined as the process of converting text into meaningful numeric/vector representations. Within this process, we wish to preserve the context and dependencies between words and sentences so that the machine is capable of detecting patterns related with the text as well as understand the context of the article.

There are various ways to encode text:

- One-Hot Encoding
- Index-based Encoding
- Bag of Words (BOW)
- Term Frequency — Inverse Document Frequency (TF-IDF)
- Word Embeddings

2.2 System Architecture

Design

The Encoder-Decoder architecture is a good option of organizing neural networks for sequence prediction problems, which have a multiple number of inputs, outputs, or both inputs and outputs. Architecture will be construct of two components: an encoder and a decoder.

- **Encoder**: The encoder reads the entire input sequence and encodes it into an or a value that is useful in internal representation to understand by machine, it may be a fixed-length vector called the context vector or integer number.
- **Decoder**: The decoder reads the encoded input sequence from the encoder and generates the output sequence which will display exact sentences or characters that already feed to the machine.

In Encoder-Decoder network, Both the encoder and the decoder sub models are trained same time. Natural language problems required the development of separate models that were later pass into a pipeline of the machine and help to resolve errors during the sequence generation process.

Encoded output is used as part, generating each text input in the simplified output which is use for NLP. Data cleaning is main part of this process where we use machine learning technique to remove unnecessary text data and it will generate clean text data which is in encoded form. Fixed-length encoding of the input text, limits the length of output sequences which make simplified output.

An extension of the Encoder-Decoder architecture is to provide a more expressive form of the encoded input sequence and allow the decoder to learn where to provide attention to the encoded input, when generating each step of the output sequence.

Encoder-Decoder architecture with attention is popular for a suite of natural language processing problems that generate variable length output sequences, such as text summarization.

The application of architecture to text summarization is as follows:

- **Encoder**: The encoder is responsible for reading the source document and encoding it to an internal simple representation which is understand by machine.
- **Decoder**: The decoder is a language model responsible for generating each word in the output summary using the encoded representation of the source document.

Pseudo Code

Pseudo code is a high-level algorithmic code designing, which use for checking grammar similarity with all the special characters. In this respect its aim is to create simple sentence sequence which is easier to read by humans rather than the computer, so it is always easier to understand. It is useful for designing algorithms with other system in simple manner and can be used as an intermediate step between two systems and the code itself. We can use periods and delimiters where we required is benefit of pseudo code. BERT pseudo code library can be directly installed on python for our text analysis.

System Map

System map helps to create planning of the sequence of task given. Simple graphical presentation given bellow figure. Our input data is text, which is article, research paper or any short story. Language of this text is not sure by us in current potion so we use encoding-decoding method where encoding method converts the text into binary/integer array. Some ignorable part of text like space, dot, explanation mark etc. is clean by using data wrangling method and then pass to the machine. At the other hand when we got conclusion after neural language processes, output is again decoded to the original format.

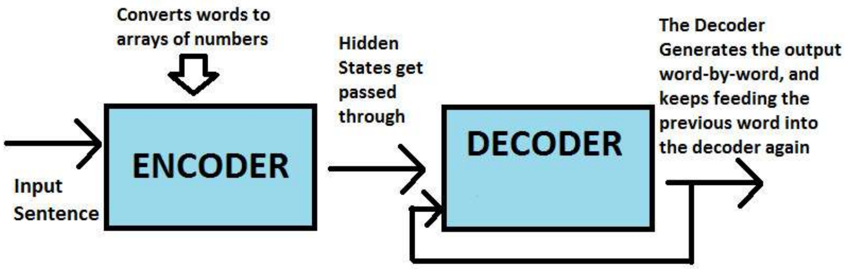
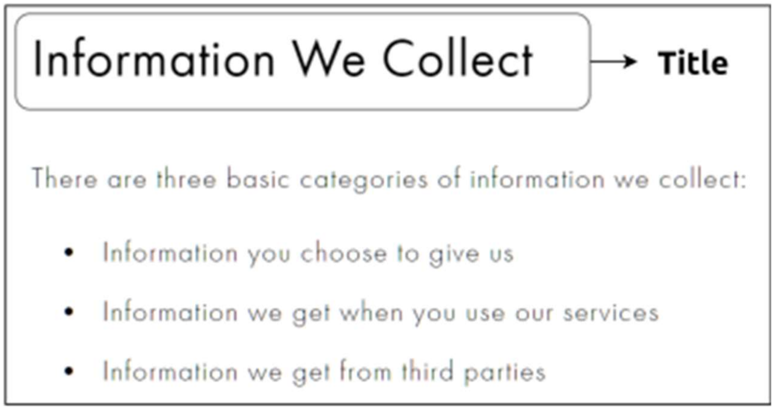

Converts words to arrays of numbers
Input Sentence
ENCODER
Hidden States get passed through
DECODER
The Decoder Generates the output word-by-word, and keeps feeding the previous word into the decoder again

Figure 1: System Map

2.3 Functional Requirements

Functional requirements of the project is to design a simple system that give abstractive summarize output from the given input data. For this we are going to use python as our key programing language and our functional input-output model where user will interface and task will perform create on HTML. “Flask” will be our framework that generate a web application for our task.



Information We Collect

There are three basic categories of information we collect:

- Information you choose to give us
- Information we get when you use our services
- Information we get from third parties

Figure 2: Functional Requirements

We need sentence-title and paragraph-title pairs to train a supervised model for title generation. However, there are no readily available datasets in this form. We make use suitable dataset or suggested web page in our task “arXiv.org” were we can found many research papers.

Text generation techniques make use of an encoder-decoder architecture, wherein the encoder computes a representation of the input, and the decoder generates the output. Prior work employs sequence to sequence based Long Short Term Memory (LSTM) celled encoder-decoder architecture to generate headlines for news articles.

many sequence to sequence models fail while they try to generate title, resulting in poor headlines for long articles. To overcome this problem, attention layers are added to the encoder-decoder networks, which gives transparent, short, key-oriented data and that data is given to transformers module which is special model library used for text analysis and generating text from the decoder.

2.4 Non-Functional Requirements

Our motive is to generate title from the text and we are going to design a web application for this, user have to enter the article or text data on our web application, for that user have to copy the text or have to write text on our text box which we are going to design on our system, to take input text. Here a non-functional task we have to perform which is part of our system and user have to perform this task which may be our goal to improve after getting positive result of current system.

- **Time**
  When user enter the text on machine, our system give respond to the enter text data and title will be generated after the encoding-decoding and prediction process. For this we have to create system in such manner that it time saving to generate meaningful title.

- **User Interface**
  Create user friendly system should be important. For this we will create one text box where user can easily write or past the text and then user will click on button "Predict Title" and user get the generated output in one display area.

- **Portability and Flexibility**
  Our web-application should be portable for all platforms where user want to perform the task of title generation and it should be faster as per user's requirement.

- **Scalability**
  It is important as a part of function where our system can control server load of multiple user and when user increase, performance of system should not drop. Delay in output generation should not be tolerated by user when user traffic increase on our system.

- **Safety and Security**

  Use data is very valuable so our system should be as much secure that user data will not lick at any cost and should maintain the law of research and privacy also. It should prevent attack of unauthorized access.

2.5 System Models

Below figure show the system model where our flow is design and decided. It is a internal model for the title generation where flow starts with user interface with text data then it is processed for data cleaning and pre-processing for our core data analysis.

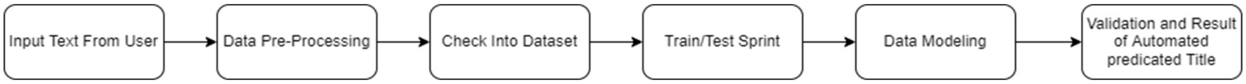


Figure 3: System Flow Diagram

3 Testing Procedure

3.1 Process

Quality check is important part of any Product and our Software should meat this level of clarity and purity, where our system always performs accurate either the environment is worst still should not show wrong output to the user. For this our system checked before putting in the market. To test the software which is designed by our team, has to meet the specific requirements and ensure the all functions work properly, for that we need to perform bellow testing method on our system:

- **Unit Testing**

  Unit testing is a task were all the individual part of software is preforming and tested in real time application. When we are creating big system, we need to test that system in many parts like we have to check data accepting, data cleaning, encoding of text etc. which are the different small parts to test in this system. We need to check that our all individual part of software work smoothly while they are combine in our machine to perform our objective task. Any un-sequential task can degrade our system performance. To overcome the failure, we will perform every part of our software should perform properly and pass through our quality system.
  We will perform below unit parts of system.

  - Application launch
  - All libraries function check
  - Data accept and feed to system
  - Data cleaning and wrangling
  - Data encoding
  - Neural network based machine learning process on filter data
  - Data decoding
  - Data execution on our system for user interface.

- **System Testing**
  After testing all individual test, it is also very important that this all individual unit will work nicely together. Sometime, communication between two modules/parts not performing o then it will make our system failure. Some software bugs are overflow, data loss type which we should be very careful while we are going to perform system.

  We have to make a check list for the system ok performance and while we are performing this task, we have to check all the points should fulfill our system requirement. Any error or any function not fulfill the requirement that we decided previously is not acceptable and we have to solve that query for best performance of system

  After achieve all the requirement we can process further for our function improvement task. As our system is unsupervised so we need to cross check all the possible side of system output that will harm our system requirement. One of the popular automated software testing tools is *Selenium* that could be used in this project.

- **Performance Testing**

  Successful execution of software is important but need to check performance of the system over critical parameters, like speed, user traffic etc. if your system work properly but output generation give a lot time delay in prediction task then the module selected as a machine earning is not good. We have to take care of this type of problem in the task to give our best output.

  Meaning full title is main part of output otherwise random generated output is not good for any article so we have to make proper prediction for title and it should decode properly and show on the system for user. Any wrong step of user must not be create system bug and it should display proper error for wrong step of use.

  Performance testing give satisfactory report of whether software speed, scalability and stability requirements under expected workloads meets to user satisfaction level.

- **Functional Requirements**

  System should be able to handle all defined task and avoid malfunction while use, if the input options is invalid, the system need to return an error message without break the program. otherwise, if the expected output is different.

  User may enter any language and any character but our system meet all the required function that it can omit not required word and data cleaning is perform well.

  Character limit is also important, we have to decide maximum character limit for text input, otherwise our machine learning model may be go in infinite prediction loop so we have to limit our system input in trial base and the we need to improve our maximum text input limit.

  If user will reach to maximum character limit then system should display and error to use for "Text Overflow"

  The result should be simple and short otherwise user will get long title and that's didn't make and kind of sense.

  Meaning full title is also important for our system requirement.

  Encoding and decoding must be secure and user details not be explicated to console or terminal it must be confidential.

  Data privacy and research law should be maintained by our system for every user.

- **Non-Functional Requirements**
  Performance testing will be good to check non-functional requirement.

  Our system should be fast, not consume time delay in prediction. It will make user traffic as well user negative impression.

  Graphical design is also important part of system design. If user will not accept the look of your system then it will also decrease our product market value and uses.

User interface is also important. System meet user friendly and easily accessible and it can perform stress free operation for user

It should fulfill the main object of the system it meet all the satisfactory and similary result of title that is predicted and it meets abstractive summery of whole entered text by user.

Non- functional requirements will be tested manually using excel and testing it manually. For example, in this project if article chosen is of one sentence or not.

Check if the algorithm for generating title is working or not by entering one sentence and then checking whether it provides correct output or not.

As the main testing will be done of the algorithm and the UI created. These are the important factors of testing non-functional requirements.

3.2 Schedule

To fulfill the timeline of project we should make a schedule of our all task and start work in team.

The testing progress will divide in multiple stages during the project development. In general, the testing will start from small units of the codes and functions of the system. The schedule will be followed as:

- System model work
- System individual unit code design
- Unit testing
- Unit combining up to the user interface level
- System testing
- Performance testing

To complete task with given timeline we need to check all unit testing after completing the individual unite code design. It will make parallel work of unit design and testing, for that we need good team communication and project planning tools like jira, Microsoft team management etc. For this project we are going to use Jira for our planning tool.

4 Project Management

To be productive and creative in current society, a product must fulfil a variety of standards, which we may achieve through well-structured project management. Project management is the process of designing, executing, managing, and ending a project in order to acquire a competitive advantage. set objectives and achieve requirements at specified periods This is frequently accomplished by dividing the job into five parts. The sections are as follows: conception & initiation, definition & planning, launch & execute, performance & control, and project closing.

Conception & Initiation-Typically, this is the first stage of any project management and includes generating and enhancing the project's primary concept through in-depth analyses about the project's major concept and the deployment of boards, spider diagrams, flowcharts, and storyboards. It is about bringing the project's major goal and purpose into action.

Definition & Planning – This phase is the major body of project management and contains time management, evenly dividing effort among personnel, and finding issues and solutions prior to commencing real development. There are several methods to graphically organize all of this content, such as utilizing a 'Phase model.' This approach creates all or most of the critical phases of the project development.

Launch & Execution – Into this phase, one generates graphical tables for the project's study estimates and highlights the actual status of the project as well as if the desired quality has been fulfilled or not.

Performance & Control – This step includes determining if the project will meet its aims and how well it works in reality. In this phase of project management, the quality is also checked to see if it is genuinely deliverable to the customer, and tests are done.

Project closing – Project closure report must be created, which covers every aspect about what went well and what went poorly throughout project management.

To The usage of a development methodology is required to assure the release of a successful project within the deadline specified, in order to guarantee a better approach of verifying and staying up with the essential requirements. Some typical sorts of such approaches include 'Waterfall Methodology' and 'Agile Methodology,' and after careful study, we have opted to use the former for this project.

4.1 Agile Methodology

The Agile methodology is concepts that give special important to adaptation and flexibility, as well as ongoing communication and constant improvement throughout all phases of software development. Once the project has been chosen, the members of the team would go through the Planning, Planning execution, and Planning evaluation.

An 'Agile Manifesto' establishes the framework for this technique, with four major values: humans and interactions over procedures and technologies, functioning software over exhaustive documentation, customer participation over negotiating the contract, and adapting to change over implementing a strategy.

The Agile methodology in project management provides several frameworks through which its ideals can be maintained, like Scrum, Kanban, Extreme Programming, and Adaptive Project Framework.

We would use Scrum, which seems to be a technique that involves having daily stand-ups, sprints, reviews, and retrospectives. The team is directed by a Scrum master, whose major responsibility would be to ensure that almost all tasks are performed without failure, removing all barriers.

Daily stand-ups – It is also called as a 'Scrum Meeting,' and it lets team members discussing their completed work, which improvements needs to be done, and future goals.

Sprints – It is a set of time frames that includes an information about all of the project's major events, as well as their major deliverables and deadlines.

Sprint Review – This gives the team members the opportunity to showcase their progress so far in this, often by providing a minimal viable product.

Sprint Retrospective – This is a chance for the team to identify topics which might be enhanced in the upcoming sprint.

4.2 Jira

Jira is a software program designed to assist groups to manage their thoughts and the procedure of building a project. It is a robust project management tool with several applications, few of them are shown below.

Management of requirements and test cases, agile and project management teams [19]. Jira offers Scrum and Kanban for groups that have opted to work using agile approaches. boards, in which tasks are assigned to configurable processes, allowing for more insight over the process of Every work item's status, as well as teamwork transparency.

We would be adopting a Scrum board to monitor issues, customize processes, estimate and track work, and report progress.

At the present, our board consists solely of the records of all previous meetings, as well as a backlog of important lessons and materials that will aid in future implementation.

4.3 GitLab

GitLab is a free and open - source programmer that create capability to design, safety, and management all in one software. It is an open-core model of development, with basic functionality published within an open-source license.

Once done, we will use GitLab to contribute our Minimal Viable Product and also the source for our Final Project.

References

[1] Quick tour (huggingface.co)

[2] Natural Language Processing with Python by Steven Bird, Ewan Klein and Edward Loper

[3] Applied Text Analysis with Python: Enabling Language-Aware Data Products with Machine Learning Benjamin Bengfort, Rebecca Bilbro, Tony Ojeda

[4] https://scikit-learn.org/

[5] Introduction to TensorFlow

[6] Long Short-Term Memory Networks With Python by Jason brownlee (Develop Deep Learning Models for your Sequence Prediction Problems)

[7] Python Natural Language Processing: Explore tools and techniques to analyze and process text with a view to building real-world NLP applications by Aman Kedia

[8] Introduction to Machine Learning with Python: A Guide for Data Scientists by Andreas C. Mueller

[9] Python for Data Analysis, 2e: Data Wrangling with Pandas, Numpy, and Ipython by Wes Mckinney

[10] Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems by Aurelien Geron

[11] Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. From neural sentence summarization to headline generation: A coarse-to-fine approach. In IJCAI,pages 4109–4115, 2017.

[12] Yuko Hayashi and Hidekazu Yanagimoto. Headline generation with recurrent neural network. In New Trends in E-service and Smart Computing, pages 81–96. Springer, 2018.

[13] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1724–1734, 2014.

[14] Abhijith Athreya Mysore Gopinath, Shomir Wilson, and Norman Sadeh. Supervised and unsupervised methods for robust separation of section titles and prose text in web documents. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 850–855, 2018.

[15] https://tutorials.one/encoder-decoder-models-for-text-summarization-in-keras/