Report On

Automatic Text Generation Using Tensorflow

Submitted in partial fulfillment of the requirements of the Course project in Semester VIII of Final Year Computer Engineering

by
Yash Raul (Roll No. 70)
Sonit Shetty (Roll No. 74)

Mentor
Dr. Tatwadarshi Nagarhalli

University of Mumbai

Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

(A.Y. 2021-22)

Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

CERTIFICATE

This is to certify that the Mini Project entitled **"Automatic Text Generation using Tensorflow"** is a bonafide work of **Yash Raul (Roll no. 70), Sonit Shetty (Roll no. 74)** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **"Bachelor of Engineering"** in Semester VIII of Final Year **"Computer Engineering" .**

_____

Dr. Tatwadarshi Nagarhalli
Mentor

_____

Dr. Megha Trivedi
Head of Department

_____

Dr. H.V. Vankudre
Principal

Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

Course Project Approval

This Mini Project entitled **"Automatic Text Generation Using Tensorflow"** by **Yash Raul (Roll no. 70), Sonit Shetty (Roll no. 74)** is approved for the degree of **Bachelor of Engineering** in Semester VIII of Final Year **Computer Engineering.**

Examiners

1.....
(Internal Examiner Name & Sign)

2.....
(External Examiner name & Sign)

Date:

Place:

Contents

ABSTRACT

Our goal here is to assess the state of the art of text generation for two purposes: to help people who intend to apply that art in the near future and to aid in the design or selection of appropriate research. This assessment covers all of the technical methods by which computer programs create and present English text in their outputs. (For simplicity we always call the output language English.) Because text generation has not always been taken seriously from a technical point of view, it has been actively pursued only recently as a topic in artificial intelligence. As a result of this late start, much of the technology available for application today is still rather superficial. However, text generation is now such an active research topic that this superficial technology will soon be surpassed. (The last part of this report contains an extensive bibliography on the subject.)

1.1 INTRODUCTION

TensorFlow and Keras can be used for some amazing applications of natural language processing techniques, including the generation of text. TensorFlow is one of the most commonly used machine learning libraries in Python, specializing in the creation of deep neural networks. Deep neural networks excel at tasks like image recognition and recognizing patterns in speech. TensorFlow was designed by Google Brain, and its power lies in its ability to join together many different processing nodes. Keras' foundational principles are modularity and user-friendliness, meaning that while Keras is quite powerful, it is easy to use and scale. Natural language processing(NLP) is exactly what it sounds like, the techniques used to enable computers to understand natural human language, rather than having to interface with people through programming languages. Natural language processing is necessary for tasks like the classification of word documents or the creation of a chatbot.This report consists of two documents describing the state of the art of computer generation of natural language text. Both were prepared by a panel of individuals who are active in research on text generation. The first document assesses the state of the art, identifying four kinds of technical developments which will shape the art in the coming decade: linguistically justified grammars, knowledge representation methods, models of the reader, and models of discourse. The second document is a comprehensive bibliography on text generation, the first of its kind. In addition to citations of documents, it includes descriptions of ongoing research efforts.

1.2 PROBLEM STATEMENT

The main challenge is information overload, which poses a big problem to access a specific, important piece of information from vast datasets.
Semantic and context understanding is essential as well as challenging for summarisation systems due to quality and usability issues.

OBJECTIVES

The goal of text-to-text generation is to make machines express like a human in many applications such as conversation, summarization, and translation. It is one of the most important yet challenging tasks in natural language processing (NLP).

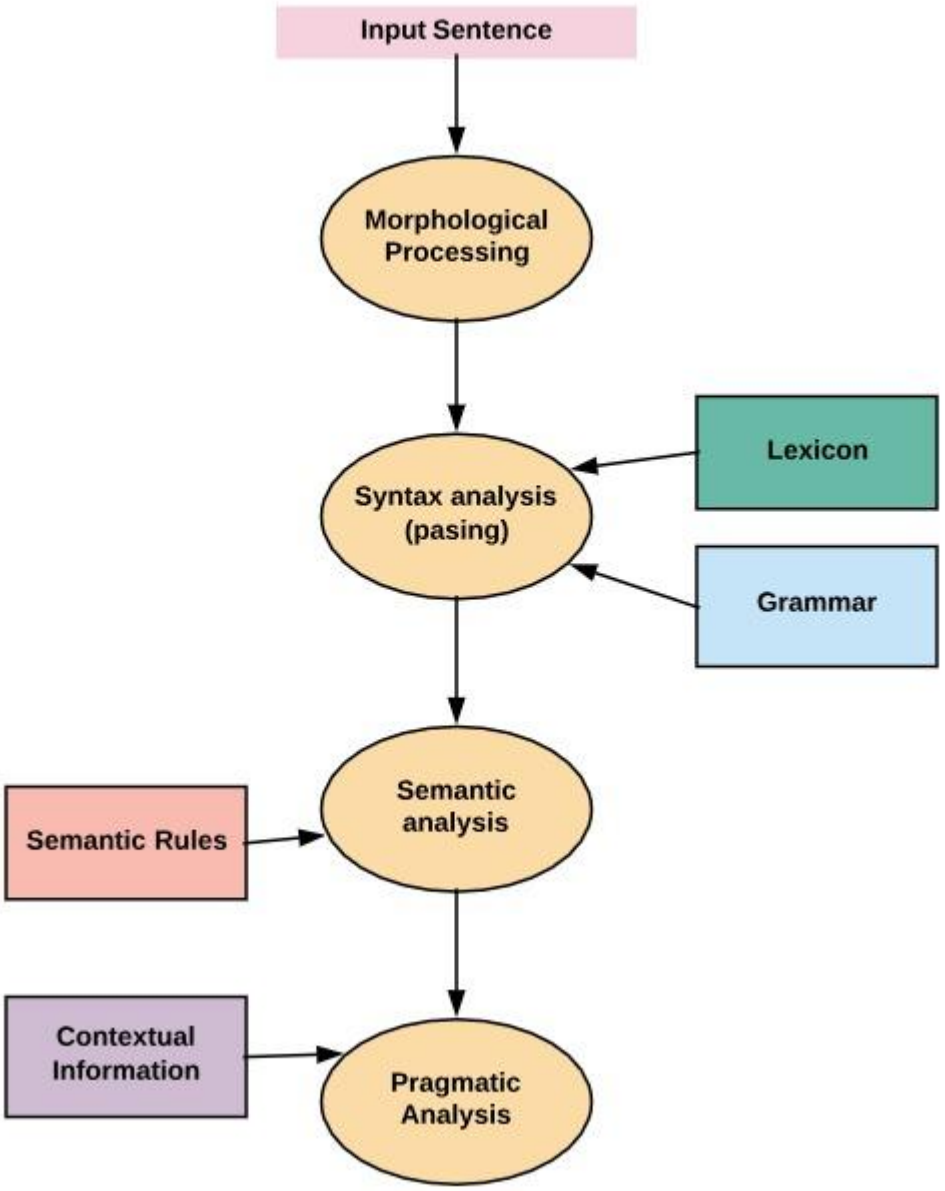
1.3 SCOPE

Developers can make use of NLP to perform tasks like speech recognition, sentiment analysis, translation, auto-correct of grammar while typing, and automated answer generation. NLP is a challenging field since it deals with human language, which is extremely diverse and can be spoken in a lot of ways.

2.1 Introduction

Text generation is a subfield of natural language processing. It leverages knowledge in computational linguistics and artificial intelligence to automatically generate natural language texts, which can satisfy certain communicative requirements.The five phases of NLP involve lexical (structure) analysis, parsing, semantic analysis, discourse integration, and pragmatic analysis. Some well-known application areas of NLP are Optical Character Recognition (OCR), Speech Recognition, Machine Translation, and Chatbots.NLP is important because it helps resolve ambiguity in language and adds useful numeric structure to the data for many downstream applications, such as speech recognition or text analytics.

2.2 Architecture / Framework / Block diagram



2.3Algorithm and Process Design

Long Short-Term Memory (LSTMs) networks are a specific type of Recurrent Neural Networks. LSTMs have advantages over other recurrent neural networks. While recurrent neural networks can usually remember previous words in a sentence, their ability to preserve the context of earlier inputs degrades over time.

The longer the input series is, the more the network "forgets". Irrelevant data is accumulated over time and it blocks out the relevant data needed for the network to make accurate predictions about the pattern of the text. This is referred to as the vanishing gradient problem.

You don't need to understand the algorithms that deal with the vanishing gradient problem (although you can read more about it here), but know that an LSTM can deal with this problem by selectively "forgetting" information deemed non essential to the task at hand. By suppressing nonessential information, the LSTM is able to focus on only the information that genuinely matters, taking care of the vanishing gradient problem. This makes LSTMs more robust when handling long strings of text.

When it comes to implementing an LSTM in Keras, the process is similar to implementing other neural networks created with the sequential model. You start by declaring the type of model structure you are going to use, and then add layers to the model one at a time. LSTM layers are readily accessible to us in Keras, we just have to import the layers and then add them with model.add.
In between the primary layers of the LSTM, we will use layers of dropout, which helps prevent the issue of overfitting. Finally, the last layer in the network will be a densely connected layer that will use a sigmoid activation function and output probabilities.

2.4 Details of Hardware & Software

Hardware

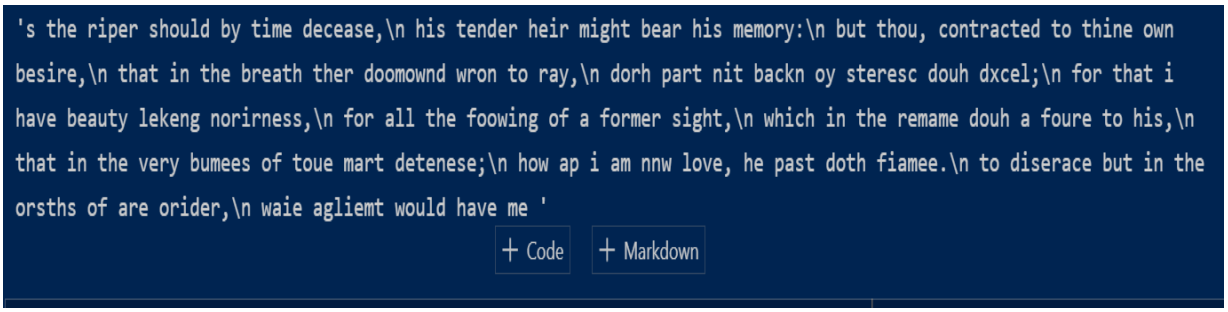
- Intel i5 processor
- RAM – 8GB
- Hard disk – 10GB
- Web browser
- Internet Connection

Software

- Jupyter Notebook
- Python — a programming language
- Pandas — data manipulation and analysis library
- NumPy — scientific computing library

2.5 Experiment and Results for Validation and Verification

This time we trained our model for 100 epochs and a batch size of 50. We at least obtained a non-repetitive sequence of characters, which contains a decent number of legitimate words. Also, the model learnt to produce a sonnet-like word structure.



2.6 Conclusion and Future work.

NLP is all about analyzing and representing human language computationally. It equips computers to respond using context clues just like a human would. Some everyday applications of NLP around us include spell check, autocomplete, spam filters, voice text messaging, and virtual assistants like Alexa, Siri, etc.

You'll want to increase the number of training epochs to improve the network's performance. However, you may also want to use either a deeper neural network (add more layers to the network) or a wider network (increase the number of neurons/memory units) in the layers.
You could also try adjusting the batch size, one hot-encoding the inputs, padding the input sequences, or combining any number of these ideas.

References

[1]Berry, M. 1977 Introduction to Systemic Linguistics: Levels and Links. B.T. Batsford, Ltd., London.

[2]Birnbaum, L., Flowers, M., and McGuire, R. 1980 Towards an A.I. model of argumentation. In Proceedings of the First National Conference on Artificial Intelligence.

[3]Stanford University (August). Boguraev, B.K. 1979 Automatic Resolution of Linguistic Ambiguities. Computational Laboratory, Cambridge University, England, Technical Report 11 (August).

[4]Bossie, S. 1981 A Tactical Component for Text Generation: Sentence Generation Using a Functional Grammar. University of Pennsylvania, Technical Report MS-CIS-81-5.

[5]Brown, G.P. 1974 Some problems in German to English Machine Translation. Massachusetts Institute of Technology, Technical Report 142 (December). Project MAC.