iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** VI **Month of publication:** June 2021

DOI: https://doi.org/10.22214/ijraset.2021.35090

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VI Jun 2021- Available at www.ijraset.com

Character Chatbot - A Conversational AI Chatbot which can alter its Character according to User Specifications

Aanchal Malhotra[1], Ravi Gowda[2], Vishnuraj Yadav[3], Amarja Adgaonkar[4]
[1,2,3]*Student,* [4]*Professor, Information Technology Department, K C College of Engineering & Management Studies & Research, Thane (East), Maharashtra, India -400603*

Abstract*: *A chatbot is a type of AI software, developed for the purpose of simulating a conversation with the user. They converse in natural language via messaging applications, websites, or any other form of communication platform. Some chatbots use extensive word-classification processes, Natural Language processors, and sophisticated AI, while others simply scan for general keywords and generate responses using common phrases obtained from an associated library or database. In our project, we have developed a model where the user can create a chatbot specific to a character. The user will be able to interact with the chatbot by giving it a few personality traits and the chatbot will be able to hold a conversation accordingly. If a personality is not specified then the chatbot has been given a default character - 'Joey' from the T.V. show 'Friends'. We did this by using Simple Transformers Model along with the pre-trained model provided by Hugging Face. This includes the training dataset for various personality traits; but since we decided to give the chatbot a unique default character, we trained the model and fine-tuned it to a dataset we created ourselves in the same JSON format from every episode's script of the show. Lastly, we used Tkinter package to build the GUI for the chatbot and connected it to our trained model and the chatbot was ready to chat

Keywords: Chatbot, Personality, Training, Dataset, Transformers

I. INTRODUCTION

Chatbots are AI agents that can participate in a conversation with users. They can be built to cater to different needs and requirements and accordingly be used to accomplish various objectives. There are many chatbots which are part of a web application, these can be considered as customer-care chatbots. Their main aim is to aid the users and help them with their queries. So, these are informational chatbots or goal-based chatbots where they are provided with a manual containing the solutions to the most asked questions and with the help of it, they fetch the relevant answers and display them to the user.

These new age chatbots are increasingly expanding their presence in a number of businesses and are often used to perform automated tasks which require very less skill-based talents. Such chatbots can be made using frameworks like Google Dialogflow and IBM Watson, which help in making closed domain chatbots since they provide many user-friendly tools and use intents and contexts to build a conversation. Since such chatbots have started to make a prominent appearance in almost every domain and industry, our project focuses on the use of chatbot for helping the writing community of the world.

Writer's block can be a serious condition primarily associated with writing, it basically is the period of time in which the author loses all ability to produce new content, ultimately leading to a creative shutdown. Hence, we have thought of developing a system that lets writers create and explore the mind of their character by talking to them. To do that we use another way of developing chatbots i.e., by using open-source libraries and machine learning packages. These chatbots come under the open domain since they use Natural Language Processing to converse with the users. These AI bots make use of machine learning and generate responses based upon their training. Our project comes under the open domain where we do not use the typical rule-based means of creating a chatbot instead we implement Simple Transformers' ConvAIModel to build a conversational chatbot.

II. REVIEW OF LITERATURE

A Neural Chatbot with personality, Authors - Huyen Nguyen, David Morales, Tessera Chin, Year – 2017, This research paper from Stanford University goes through the process of building an open-domain response generator with personality and identity. They make use of sequence-to-sequence model for training their chatbot on a large dataset and eventually train the model based on particular characters from T.V. shows.

How I Used Deep Learning to Train a Chatbot to Talk Like Me, Author- Adit Deshpande, this blog talks about how the author used his daily conversations from various social media platforms and transformed them into a suitable dataset to train the chatbot on it so that the chatbot resembles the conversational style of the author.

A survey on social characteristics in human-chatbot, Authors: Ana Paula Chavesa, Marco Aurelio Gerosaa, Year: 2020, This paper talks about the characteristic features that are expected to be in a chatbot. Some of them are: Conversational Intelligence, how to keep visual elements consistent, the ability to include business integration and about Social Intelligence.

Deep Learning Based Chatbot Models, Authors: Richard Kriszti´an Cs´aky, this report talks about the history of chatbots and different models that can be used to develop one. It gives information about pre-trained models, attention mechanism and Encoder – Decoder model.

A Neural Chatbot with Personality, Author: Huyen Nguyen, David Morles, Tessera Chin, 2017, To build an open-domain response generator with personality and identity, The bot can talk reasonably fluently, having distinct personalities, and seems to have learned certain aspects of their identity It was also proved that a human is more than 50% likely to believe that the response actually came from the real character.

Improved Sequence Generation Models for Dialogue: Author: Jason Weston, Emily Dinan, Alexander H. Miller, 2018, Sequence generation models for dialogue are known to have several problems: they tend to produce short, generic sentences that are uninformative and unengaging. Retrieval models on the other hand can surface interesting responses

III. REPORT ON PRESENT INVESTIGATION

A. *Requirement Analysis*

1) *Scope:* The scope of the project is to build a model which will be able to replicate the behaviour of a decided character via conversation. It should help the writers to understand their character's personality via an actual conversation with it, rather than having to rely solely upon their imagination. This will help the writers to get to know their characters better by exploring its psyche and eventually can use it to complete their creative work which can be a book, novel, screenplay etc
2) *Feasibility Study:* The sole purpose of the project is to give the creative writers a platform to experience a new way of content writing. The platform will be absolutely feasible to its targeted audience with the aim of aiding them in their creative writing process.
3) *Hardware Requirements*

a) *RAM:* Minimum 8GB but generally, 16GB or more is recommended
b) *CPU:* Our system had 1.60GHZ - 1,80GHz but recommended base clock is 3.7GHz; for better functionality one can even use CUDA which is a parallel computing platform by Nvidia to enable GPU

4) *Software Requirements*

a) *Anaconda:* We used it to manage packages required for the project. This platform is mainly used for data science and ML applications. The virtual environment for our project is created here
b) *Jupyter Notebook:* We installed it for our environment and trained and tested our model in this notebook using Python
c) *PyTorch:* This is an open-source Machine Learning library used for natural language processing. We used it since it has an optimized tensor library that can be used for deep learning using CPUs
d) *Simple Transformers:* It is based on the Transformers library by Hugging Face. We use the ConvAIModel class to train our chatbot and interact with it
e) *Tkinter:* This is Python library for building a GUI. We used it to build the interface for our chatbot. 'Friends' dataset: In order to create the dataset for our default chatbot character ('Joey' form 'Friends)', we downloaded all the scripts of the show from Kaggle.

B. *Problem Statement*

Content writers go through rough patches where it becomes difficult for them to be creative and imagine story-lines. Such an occurrence of experiencing a writer's block can impact their productivity drastically. If we even look up what 'Writer's block' means the search results say: 'when your imaginary friends refuse to talk to you'. Hence, we thought of building a model that enabled the user to do just that. The user will be able to talk to their designated imaginary friends by defining its personality traits into the system. Character chatbot then, will be able to resemble those traits during the interactions. In a case where the personality traits are not mentioned the chatbot will have a default character- 'Joey' from the famous T.V. show 'Friends'.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VI Jun 2021- Available at www.ijraset.com

C. Problem Design

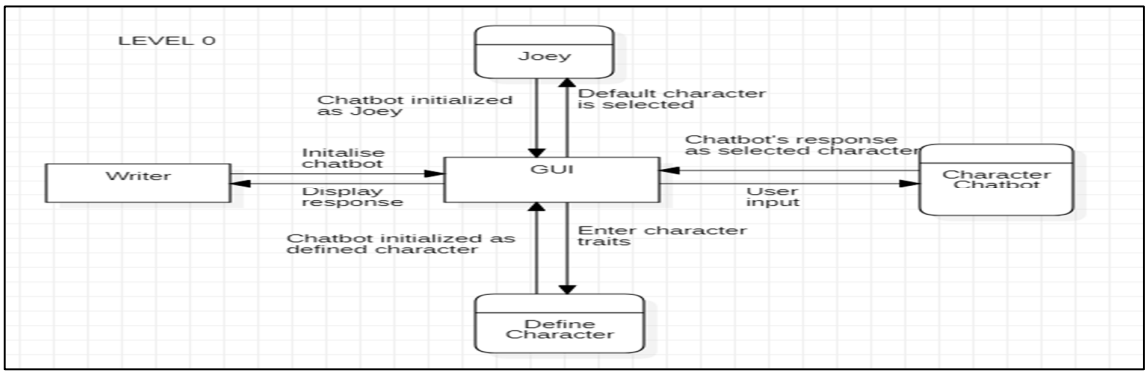


Fig. 1. DFD Level 0

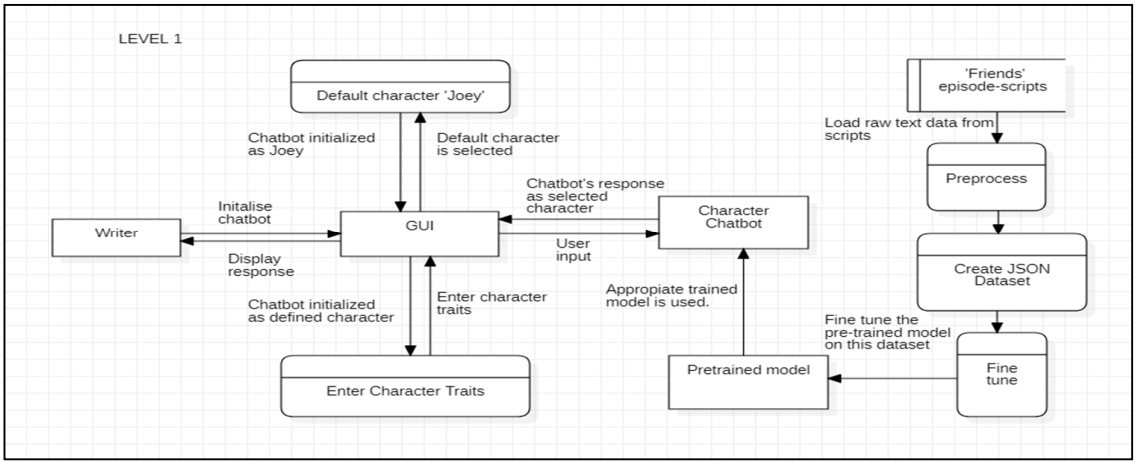


Fig. 2. DFD Level 2

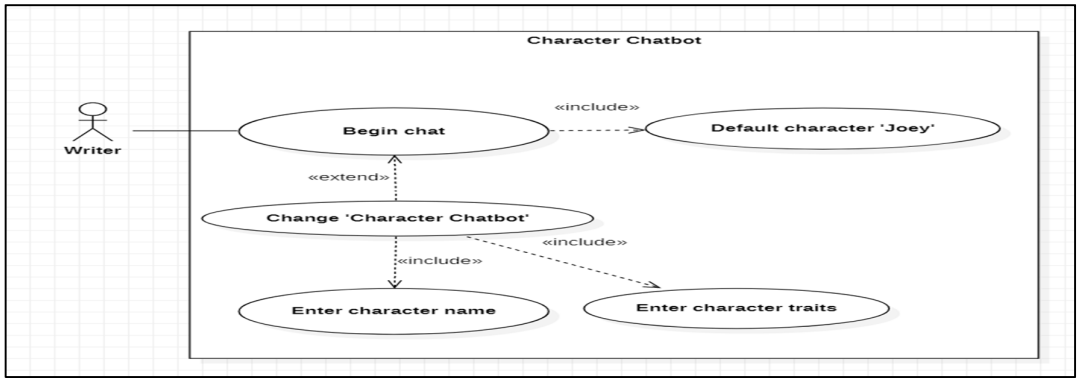


Fig. 3. Use Case Diagram

D. Methodology

In order to build a chatbot with good conversational skills, developers need to build up its vocabulary and teach it basic sentence formation and grammar; just like how a child is taught a language. So, to do that, we made use of a library - Simple Transformers which enables you to train and evaluate transformer model quickly. Transformers are a collection of pre-trained models which makes the whole process of NLP much easier.

Now in this library we particularly used the ConvAIModel which is specifically designed to train Conversational AIs i.e., chatbots. This class helped us to fine-tune our dataset even further and performed Transfer Learning form the pre-trained model to the new dataset. To fine-tune our model on a particular dataset we created one of our own for the default character that was assigned to the chatbot ('Joey' from 'Friends'). But this dataset had to be in a defined format. The ConvAIModel follows the Facebook Persona-Chat format which is as follows:

It is in a form of a dictionary where the first key is the personality and its value is the list of traits for that personality. The second key is utterances, the value for which is a list containing a dictionary of its own having two more keys. The first one being candidates, the value for this is a list of strings all acting as distractors (possible replies by the chatbot) except for the last one which is to be used by the chatbot as the answer. The second key is history, the value for it is a list containing the user input as well as the previous conversation between the user and the chatbot

The dataset format design is as follows:

```
[
        {
        'personality': [
                'Joey', 'I am an actor',
                ],
        'utterances': [
                {
                'candidates': [
                'distractor1',
                'distractor2',
                 'last string which is the reply'
                ],
                'history': [
                'user input'
                ]
                },
                ]
        }
]
```

So, we created the dataset by defining a few traits of Joey in the personality. Then we took all of Joey's dialogues as the different candidates and the conversation that has happened previous to it in the script as it's corresponding history. We made it using a Python script which fetched the scripts from the given path and processed those conversations to fit into the format. At the end we converted the result into a JSON file. Finally, we used this JSON dataset to fine-tune our model and trained it even further on a number of epochs. After testing it in the Jupyter notebook itself and getting satisfying results, we then decided to finish off the project by making a GUI for the chatbot.

We designed the GUI of our chatbot's interface using the Tkinter package. We designed it in such a way that there would be two entry boxes along with their respective buttons namely - 'Character' and 'Chat'.

The former enabling the user to enter all the characteristics the user wants the chatbot to have in a form of strings separated by a space and a period. These sentences have to be entered as a single entry, the first string being the name of the character they want to assign to the chatbot.

Eg: Aanchal. I am a student. I live in India . The reason for it is that the dataset follows such a format and so when we interact with the model it needs to follow the same style. The latter performs the simple of task of chatting with the Character Chatbot. If the character button has not been used to assign any character previously then the chat button automatically assumes the default character 'Joey' and begins conversing with the user.

E. Implementation Plan

To achieve our project's goal, we initially began researching various papers and existing Machine Learning models to decide which route was best suited for our plan. We went through the following ideas:

1) *Neural Machine Translation*: It is a sequence-to-sequence model that is mainly used for machine translation with the source as the input language and target as the output language. In the model it was used to translate Spanish into English and so we thought of switching up that dataset to meet our requirements. Instead of the dataset consisting of lists where Spanish sentences were the elements after their equivalent English translations, we put Joey's dialogues as the output and the dialogue just before it as the input. Unfortunately, the output of the model after interacting with it did not make proper sense indicating we needed a larger database for it to get trained upon.

2) *Transformer Model to NMT Model*: Since the dataset for Joey was not enough to train the chatbot upon, we decided to use Transformer Model for training and used the Cornell Movie-Dialogs Corpus. This is a well formatted dataset consisting of 220,579 conversations involving 9,035 characters from 617 movies having a total of 304,713 utterances. Transformer basically uses stacks of self-attention layers instead of CNNs and RNNs. Upon testing the outcome of the model was quite satisfactory and it was able to give decent replies. We decided to save the weights of the training done in Transformer Model and load them in the NMT Model advertently carrying out Transfer Learning so that the chatbot can provide better outputs. But we realized soon enough that the weights saved in the Transformer Model could not be loaded in an NMT Model.
3) *Transformer Model to Transformer Model*: Since we could not transfer the weights of the Transformer Model to the NMT Model, we then trained the dataset we created for 'Joey' in a Transformer Model so that the weights of the pre-trained model (model trained on the Cornell-Movie Dialogue Corpus) could be transferred easily. After completing the whole Transfer Learning process when we ultimately loaded the weights in the Transformer Model to further train it on Joey's dataset the chatbot's output on interaction was still not up to mark. Also, we could not specify a character for the chatbot through this method. We concluded that the initial training required an even larger dataset which was not a practical option for us since training on such large dataset required bigger RAMs.
4) *Final implementation*: To overcome the problem of not having sufficient resources for training the model on a large dataset we decided to look for already available modules which provided pre-trained datasets which offered fine tuning. So, we came across the Simple Transformers Model and used it as the pre-trained model instead of having to train the model on a large corpus of text. This was proved to be beneficial to us since it was backed by PyTorch library which made the whole training process much easier. We began with installing Anaconda and created a virtual environment on it named 'transformers. After that we began downloading all the packages required for the project namely PyTorch and Simple Transformers. We created a Jupyter note book for this environment and started training our chatbot.

   For the dataset we extracted all the scripts of the show and saved it on our local machine. We then created a code in python which would basically take all of the scripts of each episode one after the other and put them in to the specified format of the dataset mentioned earlier. The JSON dataset formed for the whole length of the scripts was of a very large size which gave us error during training. So, we had to reduce the size of the dataset considerably to continue with the process. After reducing the length of the dataset, we trained the model on it and saved the outputs of this training in a folder so that the next time we run the command we do not have to train the model all over again. We can just load the weights form the previous training and initialize the chatbot. Now finally during testing the chatbot, we gave it a few traits of Joey's character to see if it could depict his personality in the chat.

   To our relief the chatbot was successful in achieving the essence of the character and the only thing that remained to be done was to create a GUI for it. So, we made use of the Tkinter module and began designing the interface. We made two different buttons and entry boxes respectively, first one being the button with which you can directly start chatting with the default character- 'Joey'. The other one being an entry box in which you can give your desired traits and the name of the character and the Character Chatbot would accordingly alter its personality.

IV. RESULTS & DISCUSSIONS

The final output is as follows:
Conversation with Character Chatbot as default character - 'Joey'

Fig. 4. Final Output 1

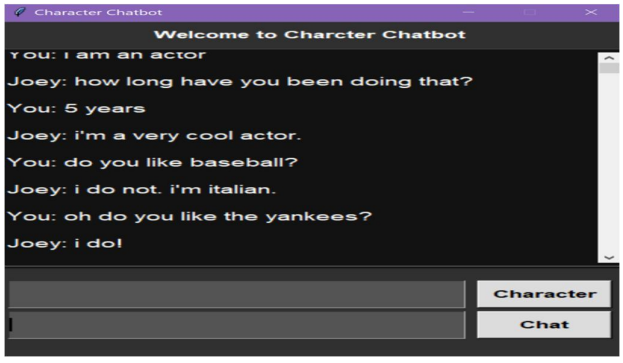


Fig. 5. Final output

Conversation with Character Chatbot as the character defined by the user

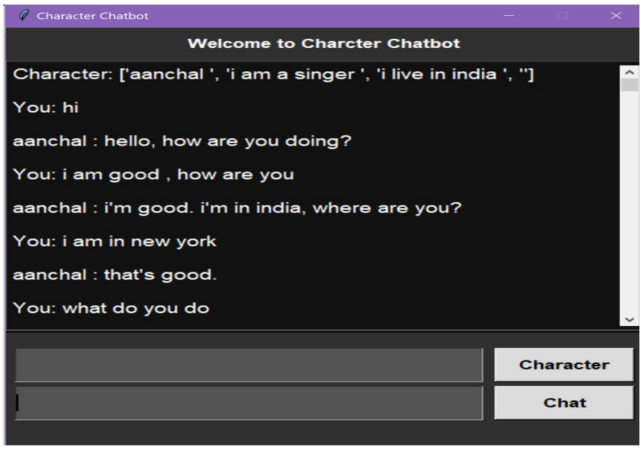


Fig. 6. Final output 3

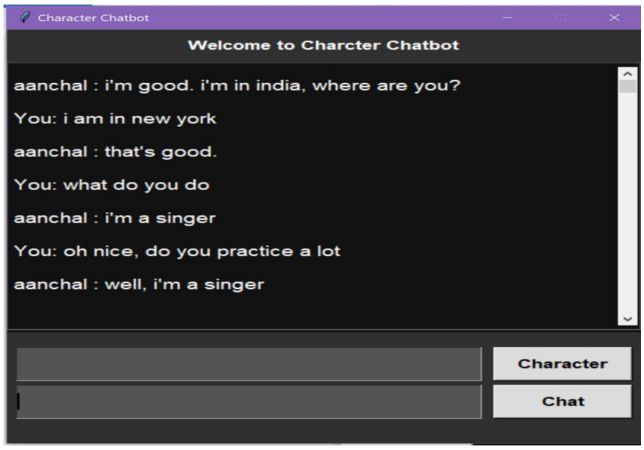


Fig. 7. Final output 4

A special feature of the chatbot is that the user can change the character by entering the new personality traits in the middle of the ongoing chat.

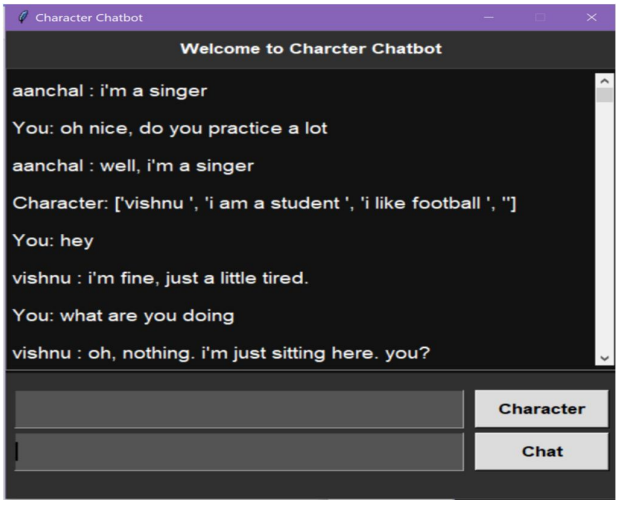


Fig. 8. Final Output 5

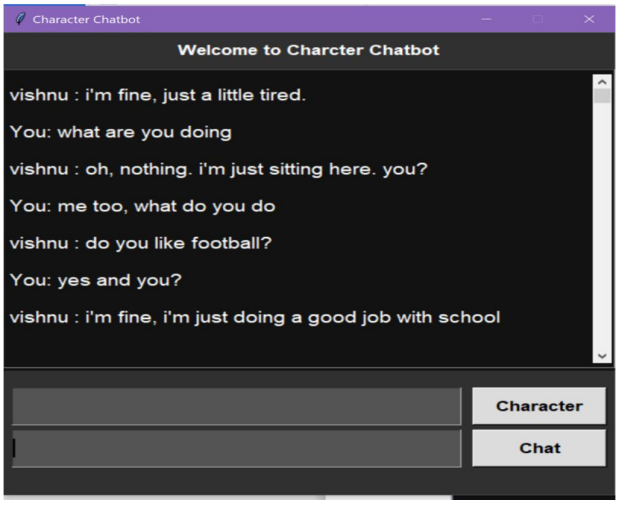


Fig. 9. Final output 6

The aim of the project had always been to create a chatbot with a controllable personality so that writers and creative producers could use it to their advantage. As our model focuses on giving the users the ability to alter the character of the chatbot, we can say that the resultant model follows that theme and produces response coherent to the explicitly defined personality. The interface is user-friendly and easy to understand where one can have a proper grasp over the chatbot's responses. To evaluate the success of the chatbot, the only viable option was to let people test it since the parameters to have such evaluations are very subjective and differs from one person to another. So according to it the resultant conversations with the Character Chatbot definitely make sense and it tries it's best to stick to the personality assigned to it.

V. CONCLUSIONS

After going through an exhaustive research and planning, we finally stumbled upon the method which catered to the needs of our project quite fittingly. Right from installing the pre-trained model, to creating our own dataset and using it to fine-tune the model and then designing the interface for the chatbot, we can conclude that we successfully implemented the project design. As for the future scope of the project, additional features like giving the chatbot access to a camera and seeing how it makes observations of its surroundings through a visual tour can be interesting. In this way the writers can get into the depth of the chatbot's personality even more and understand how it perceives different notions Last but not the least we can create a chatbot which provides more than one default character and allow the users to eavesdrop on a conversation among those characters

REFERENCES

[1] Thomas Wolf et al. “Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations”, pp. 38-45, Nov 2020.

[2] Denis Rothman, “Transformers for Natural Language Processing”, Jan 2021, ISBN:9781800565791

[3] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever , “Language Models are Unsupervised Multitask Learners”, 2018 (https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf).

[4] Ari Holtzman, Jan Buys, Maxwell Forbes, Yejin Choi, “The Curious Case of Neural Text Degeneration”, Published in ICLR, 2020 (https://arxiv.org/abs/1904.09751).

[5] Thomas Wolf et al. “Transformers: State-of-the-Art Natural Language Processing”, pp. 38-45, Hugging Face, Brooklyn, USA, Nov 2020 (https://www.aclweb.org/anthology/2020.emnlp-demos.6.pdf).

[6] Emily Dinan et al. “The Second Conversational Intelligence Challenge (ConvAI2)”, Jan 2019 (https://arxiv.org/abs/1902.00098).

[7] Ilya Kulikov, Alexander H. Miller, Kyunghyun Cho, Jason Weston, “Importance of a Search Strategy in Neural Dialogue Modelling”, iNLG Nov 2019 (https://arxiv.org/abs/1811.00907).

[8] Kenton Murray, David Chiang, “Correcting Length Bias in Neural Machine Translation”, WMT Aug 2018 (http://arxiv.org/abs/1808.10006).

[9] Yilin Yang, Liang Huang, Mingbo Ma, “Breaking the Beam Search Curse: A Study of (Re-)Scoring Methods and Stopping Criteria for Neural Machine Translation”, EMNLP Oct 2018 (https://arxiv.org/abs/1808.09582).

[10] Jason Weston, Emily Dinan, Alexander H. Miller, “Retrieve and Refine: Improved Sequence Generation Models For Dialogue”, Sep 2018 (https://arxiv.org/abs/1808.04776).

[11] Angela Fan, Mike Lewis, Yann Dauphin, “Hierarchical Neural Story Generation”, May 2018 (https://arxiv.org/abs/1805.04833).

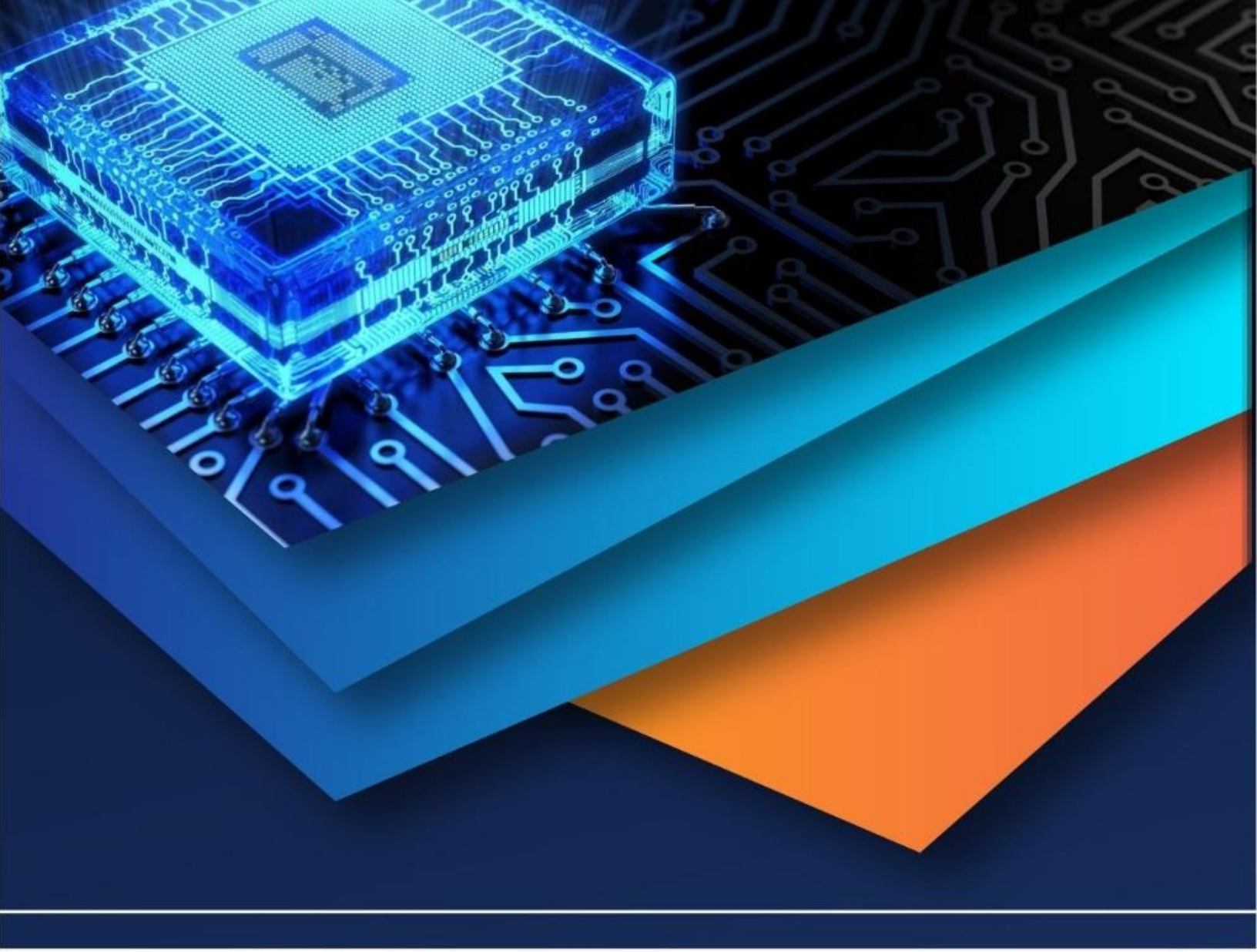
[12] Anna Goldie, “Massive Exploration of Neural Machine Translation Architectures”, Mar 2017 (https://arxiv.org/abs/1703.03906).

[13] Jeremy Howard, “Universal Language Model Fine-tuning for Text Classification”, Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Jan 2018.

[14] Aurélien Geron, “Natural language processing using transformer architectures”, Oct 2019

[15] Chenguang Wang, Mu Li, Alexander J. Smola, “Language Models with Transformers”, Oct 2019 (https://arxiv.org/abs/1904.09408).

[16] Ashish Vaswani et al. “Transformer: A Novel Neural Network Architecture for Language Understanding”, Aug 2017.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY