ACIT5900

MASTER THESIS

in

Applied Computer and Information Technology (ACIT)

Aug 2023

Applied Artificial Intelligence

GenerativeAutoAi - A generative model for cars

Shiladitya Chatterjee

S366270

Department of Computer Science

Faculty of Technology, Art and Design

Acknowledgement

This project report is part of the course ACIT5900-1 21H Master thesis(short) Research. This report includes detailed research on creation of a new dataset which contains the details of small or hatchback cars in the Indian automobile space, utilizing the data to fine tune a language model and cater to the end consumers. Being a student of Artificial Intelligence, it gives me an opportunity to explore the realm of data creation, gaining more insight on deep learning technologies, especially areas of Natural Language Processing. I would like to express my gratitude to my professor Raju Shrestha for providing me with invaluable guidance during this entire journey. Weekly planned meetings and discussions helped me immensely to proceed further with my project. He has helped in providing valuable input, especially during important decision-making phases in the project. I would also like to extend my gratitude to hugging face community for answering queries and providing with the pre-trained models. I would also like to thank Oslo Met for giving me the opportunity to study the application of Artificial Intelligence, thereby helping me gain valuable insight into the world of Artificial Intelligence.

Shiladitya Chatterjee

Oslo, Norway

Aug 2023

Abstract

The car industry at the Indian subcontinent has been a booming industry over the decade. As new and new cars have arrived at the Indian market it becomes extremely difficult for a consumer to keep track of the cars launched and their details. The paper outlines the attempt to provide a solution to the end-user/customer in the form of generative model containing all the relevant information of all the hatchback cars available in the Indian automobile market. The distributed data was collated and carefully curated to create a unique dataset containing details of 23 cars which are now present in the Indian market. The dataset used for this experiment has been curated from data taken from over 10 websites primarily containing the different data of each car. A generative model based purely on cars has been developed and it is called GenerativeAutoAi. The power of transfer learning has been leveraged to train the language model GPT-2. The GPT-2 had a BLEU score of 0.78. The new innovative approach of fine tuning was used to tune a large language model Falcon7B. The Falcon 7B model was trained with the help of QLora. This model has just been launched in the hugging face family for commercial use hence a lot of research has not gone into how it can be further leveraged. It has primarily been used to showcase the use of the created data and with just 650+ pairs of Question and Answers it has showed remarkable fluency.

Keywords: GPT-2, Fine-Tuning, Falcon 7B, Qlora

Contents

List of symbols and abbreviations

SUV – Sports Utility Vehicle

MUV – Multi Utility Vehicle

LoRA – Low rank adoption

QLoRA – Quantized Low rank adoption

QnA – Question and Answers.

GPT-2 - Generative Pre-trained Transformer 2

BLEU: Bilingual Evaluation Understudy

TER: Translation Edit Rate

Rouge - Recall-Oriented Understudy for Gisting Evaluation

List of figures

List of tables

1. Introduction

1.1 Background: A brief view of the Indian Automobile Market.

The Indian subcontinent with over 1.5 billion population is one of the biggest markets to any consumer goods. As a massive market for consumer goods car manufacturers across the world have queued up to manufacture, assemble or rebadge cars in the Indian market. This has seen an unprecedented number of brands and cars across various segments being available. A close look at the segment would reveal that it's being divided into primarily 3 segments which are:

- Hatchback or small car segment.
- Sedan
- SUV / MUV

These prominent demarcations have taken place in the last 15 - 20 years. According to the blog post (GoMechanic ,2020), the first small car or hatch back was launched in 1983 by Maruti (owned by the gov of India) and Suzuki of Japan. Since the launch of Maruti 800 (the first hatch back) there has been numerous car manufacturers like Fiat, Tata, Hyundai, Honda, Daewoo, Chevrolet, Ford. Over the years, apart from the existing car manufacturers, a lot of new players have entered space. At present there are over 21 cars from various manufacturers being offered in the hatch back segment. This segment has the most cars and varieties offered across all the segments. The reason for the launch and upgrade (facelift and engine change). According to the blogpost (GoMechanic ,2020) the primary reason for the popularity of hatchback cars can be attributed to the following reasons: -

- Affordability – Hatchback or small cars which are available across the various price ranges. The price bracket has been kept such that a hatch back can serve both as a first small budget car, a second family car for daily commute or a fully loaded luxury small car with all the latest features. The competitive price bracket has made these brands of small cars more appealing to the Indian consumer.

- Small but practical size - A country of 1.5 billion people, a practical approach for any family is to have a car which is small but has sufficient space to accommodate a small family yet maneuver freely in the city traffic. A car which can take up less parking space and is more practical for the daily commute.
- Safety - Safety has not become one of the most important aspects in terms of presenting a car in the Indian market. Every car now comes with air bags, ABS and EBD. This has been a game changer as more and more people are trying out small cars instead of big cars. Small entry level hatchbacks are now considered as an alternative to high end motorcycles as they can accommodate both families and provide a safer ride than motorcycles.
- Mileage – Fuel expenses are a major concern to every family. Increased fuel price would mean the cost of maintaining a car increases. A small car in such gives a nice value proposition to the consumer. Most small cars or hatchbacks have an attractive fuel consumption of more than 20 km/liters. This is one of the major reasons why the Indian consumer find buying a hatchback more beneficial.
- Availability of EV: In addition to petrol / diesel, the green energy initiative has paved way for numerous premier companies as well as numerous startups to venture into the EV space. Automobile companies like Maruti Suzuki, PV, MG, Tata, Hyundai have launched their EV cars which are cheaper to maintain thereby making them lucrative to the customers.
- Variety of cars – Hatchback were introduced as an entry level segment to the Indian automobile market. However, owing to the overwhelming response more and more companies have come up with more varieties in terms of specifications and features. Hatchback cars now boast not only a significant mileage but also features like apple and android car play, parking assist, automatic transmission, hill assist, infotainment with state-of-the-art display system.

According to the Wikipedia (“Automotive industry in India”, n.o) the Indian subcontinent as of 2023 is the world’s 3 largest producer of automobiles which as of April 2022 is worth US $ 100 billion. It accounts to 8% of the total India’s export. However, as there are 1,5 billion people the statistics stand at 22 automobiles per 1000 people. This scope has lured investors and manufacturers set up assembling and manufacturing plants in India. This sudden increase of options in terms of hatch back cars has left the consumers gasping for information on the available cars.

Research questions

1. How can a language model cater the consumer of automobile industry in the Indian subcontinent?

2. What can be a possible data source which can provide all information related to cars at one place?

1.2 Problem Statement

As discussed earlier, the sudden influx of many automobile companies has left the consumers confused when it comes to choosing a car. At present a total of 21 cars are available in the market with a minimum of 4 variants per car to a maximum of 11 variants. Therefore, a total of at least 60 cars and a max of 200 cars are available. There are over 10 – 12 prominent internet sites which host almost all the details of the cars. Hosting the details of the new cars available is for information purposes only. These sites primarily cater to the used cars market. The details of the cars are also available on the respective companies' website. Typically, these websites host a common pattern of information. The specifications of cars, features of cars, expert reviews and user reviews are typically the data present on these websites. The information is present in abundance but in a scattered manner which is time consuming to retrieve. A consumer must browse through various websites patiently and curate information to decide on a car. This comes with its own challenges as the consumer is greeted with a huge amount of information which is sometimes repetitive in nature. A classic example being the specification, features and ex-showroom price which will be the same across all the websites. The first phase of the project is to answer the important que. Exploring the options of generative language model are the most obvious options since the introduction of language models have proved remarkable success in generating human like responses which are coherent and meaningful in nature while maintaining the contexts related to the information. Open AI's ChatGPT has proved the power and utility of the large language models. An AI assistant like Siri, Alexa has seamlessly integrated into our daily lives. The choice of chat assistant is more prudent as it can be used as an app on the phone. The user can provide prompts or contexts and the chatbot can provide the answer quickly. This will help a huge pool of consumers to make a more prudent decision while buying a hatchback. To achieve this a new data set needs to be collated and curated to ready it for the language model. The project has 2 parts, one of which is data creation and curation and the language model which is the engine. The technologies used in the entire project would be Natural Language processing, Transfer Learning, Web Scraping. The next sections will outline some of the technologies used in the project.

1.2.1 Natural Language Processing

Natural Language Processing is a branch of applied artificial intelligence which has evolved from computational linguistics. The resources present in DeepLearning.Ai (DeepLearningAi,2023) provides a deeper insight into the world of NLP. It can be termed as a discipline of engineering which builds machines which can interpret and understand human languages and accomplish tasks. NLP can be divided into 2 segments which overlap with each other. They are as follows:

- Natural Language Understanding – Natural Language Understanding or NLU deals with analysis and understanding the textual meaning.
- Natural Language Generation – Natural Language Generation or NLG deals with generating texts.

NLU and NLG are not distinctive, rather overlapping segments. In this paper both the techniques have been used. A chatbot essentially takes as input the sentence which is written in several different languages and then generates responses abiding by the context. This is thus utilizing both the aspect of NLU and NLG. Speech recognition and its various applications also leverage the power of NLP. The applications of NLP are widespread, and its instances can be found in daily life. Some of the most common applications encountered in daily life are as follows: -

- Customer Service chatbot – These chatbots are domain specific and have found their usage in multiple industries like FMCG, Banking, fin-techs, Insurance as well as retail chains. These chatbots can be used for customer service, to enhance the customer experience which can range from complaints to helping in choosing articles.

- Conversational agents – These are the most accessible ones as they are present on any phone. Apple’s Siri, Amazon’s Alexa are some common examples. These are voice assistants which talk to humans and perform certain tasks based on instructions.

The power of NLP has also been used by Google to refine their search engine results and by Facebook to filter hate speech. As mentioned earlier, NLP with its branches of NLU and NLG are used to perform various tasks. Some of the most common tasks performed by NLP are as follows.

- Sentiment Analysis – The purpose of the model is to convey an emotional intent of a piece of text. The model takes as input a piece of text processes it and returns the sentiment in the form of ‘positive’, ‘negative’ or ‘neutral’. These have a widespread usage in customer reviews in various industries.
- Toxicity Classification - This can be interpreted as a branch of sentiment analysis which takes in a piece of text and categorizes into specific categories like hate speech, insults, use of obscene language, racial discrimination, discrimination against certain identities. This classification is most effectively used to make online conversations more pleasant in social media platforms.
- Machine translation - Machine translation can be simply referred to as a translator which can take a piece of text or sometimes just a single word and translate it to another language while preserving the meaning of the original word. One of the most used applications is ‘Google Translate’.
- Name Entity Recognition - This extracts information from a piece of text. This is used in text summarization tasks, for example – summarization of news articles.
- Spam Detection - A major application of this NLP task is to detect spam emails. The model takes as input an email and then classifies whether it is spam or not. Major email providers use this as one of the means of spam detection.
- Grammatical Error Correction - With the introduction of new and powerful deep learning models performing NLP tasks, commercial day to day applications have been created to

help in daily life of consumers. ‘Grammarly’ is one such application which does online grammar check. Microsoft Word also leverages on the same technology.

- Topic Modeling - This task takes input a corpus of text and brings out abstract topics from the corpus. The output is the list of topics which defines the words of each topic. The most popular topic modeling technique is Latent Dirichlet Allocation. LDA breaks the document into a series of topics and the topic into series of words. An important application is the usage of finding evidence in legal documents by the lawyers.
- Text Generation - Text generation or Natural Language generation generates texts which are like texts written by humans. These have many applications and can be used to generate tweets, blogs, stories, emails, songs, scripts, or simply conversational texts targeted to specific domain or causal language modeling. Some of the latest models can generate code in specific programming languages. There are various text generation models available in the market. Some successful models being Bert, GPT2, GPT3, GPT3.5, Bloom, Falcon, Llama. The primary usages are autocompleted and chatbots.
    - Auto Complete - Auto complete predicts the next word in the sequence. GPT-2 is one of the famous auto-complete models. The functionality of Auto Complete is used in WhatsApp, Gmail and in various social media applications. Other important applications are writing stories, lyrics of a song, review.
    - Chat bots - These can be termed as a conversational agent where a machine talks to either a human or a machine. These can be divided into database queries which are predominantly used to query answers from a question – answer database. The other is conversation generation where the bot can generate responses like humans. Open AI’s ChatGPT, Google’s LaMDA are some of the widely used conversational agents in the present world.
- Information retrieval – The model focuses on retrieving information. The primary goal is to retrieve relevant pieces of information from a collection of texts. Once the relevant paragraph or document has been identified then the pretrained LLM or large language model is used to create the response. The document retrieval system works on the processes of matching and indexing. The input is often a collection of pdfs which are

further broken down into small chunks of texts. These chunks of text are then stored in a vector database. As a result, each chunk of data is now represented in a vectorized format. Two tower network architecture is used where the matching is done using distance score. Searches in google, YouTube for text and image data are one of the prime applications of information retrieval models. As of date langchain is one of the revolutionary technologies which helps a user to talk to their own custom documents.

- Summarization – This is a task of highlighting the most important text by shortening the text. Summarization tasks can be divided into Extractive summarization which extracts the most important sentences. It uses a scoring system to score every sentence and then picks the ones with the highest scores. The other form of summarization is abstractive summarization where in the model paraphrases to create a summary. They are usually used in sequence-to-sequence tasks. The output in this case is a long form task and the input is typically a paragraph, or a long form task and the output is a summary of the same.

Deep Learning NLP Techniques: As we have established the power and the utility of Natural Language processing, we can now explore some Deep Learning Architectures for Natural Language processing. Some of the most used architectures are as follows:

- Convolution Neural Network – Traditionally used in the field of image processing it was first proposed in the paper by Yoon Kim to use Convolution Neural Network. In the paper sentences or documents are used instead of pixels and are represented as matrices.

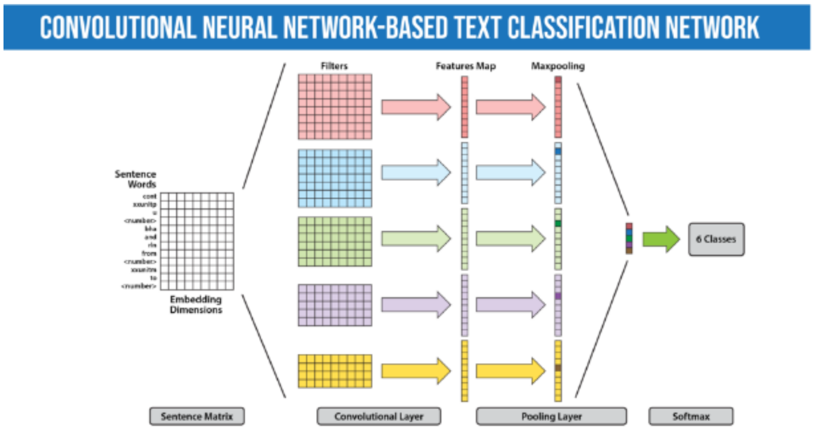


Figure 1.1: Structure of Convolutional Neural Network, Image source DeepLearningAi,2023

Sentences are used as input and the model uses convolutional layers to perform feature extractions before flattening out the last layer to perform the classification.

- Recurrent Neural Network (RNN): Recurrent Neural Network or RNNs have been primarily used for performing Natural Language Processing tasks since its inception. This is because the architecture of the RNN helps it to remember the previous information with the help of hidden state. The information of the previous state is used to predict the next text. Architectures like CNN does not preserve the past information which is essential in creating coherent and meaningful text. The variations of recurrent neural networks like Gated Recurrent Unit (GRU) and long short-term memory (LSTM) can retain past information over an extended context thereby aiding in more accurate text generation. Bi-directional LSTM /GRU helps retain information in both directions which aids in text generation and text classification.

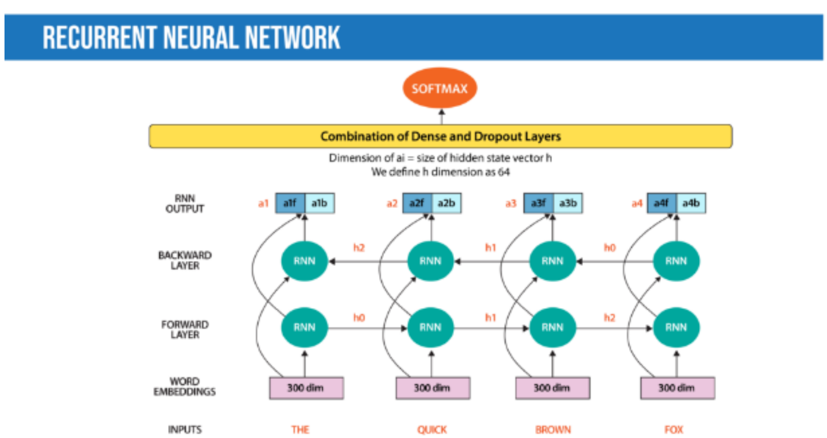


Figure 1.2: Bi-directional recurrent neural network. Image source DeepLearningAi,2023

- Autoencoders – These are deep learning encoders and decoders which approximate a mapping from input to output such that input = output. The input is taken and compressed to a low dimension vector. From there the input is learned by reconstructing it. This technique can be used to enforce dimensionality reduction. Autoencoders have found widespread applications in the field of genetics.

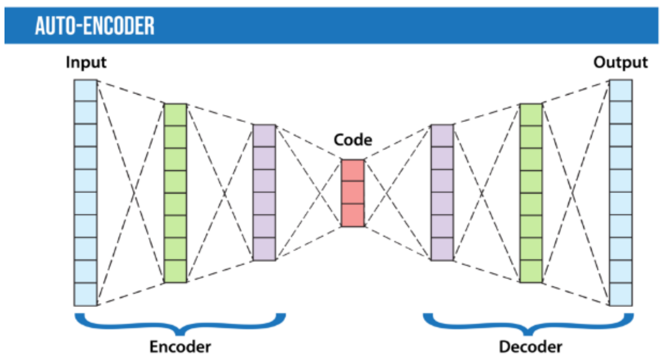


Figure 1.3: Auto Encoder-Decoder. Image source DeepLearningAi,2023

- Encoder-Decoder Sequence to Sequence – The most successful deep learning model prior to the introduction of transformers are the Seq2Seq models. These can be thought of as an adapted version of auto encoders which can perform tasks like summarization and translation. Like the autoencoders in Seq2Seq models the encoders encode the input into a vector representation. This is passed to the decoder which performs various tasks for example summarization or translation.

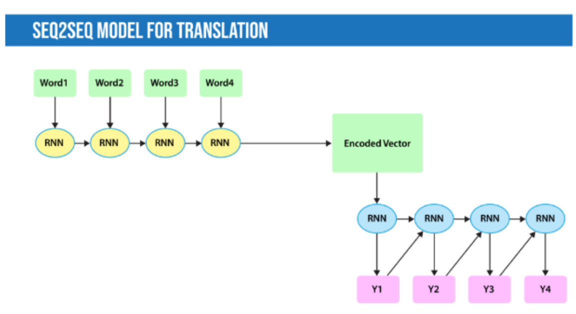


Figure 1.4: Structure of Seq2Se2. Image source DeepLearningAi,2023

- Transformers – The deep learning model which is the most used are the transformers. The architecture was proposed in 2017 paper "Attention is all you need "by (Vaswani et al., 2017). Instead of the recurrence it relies on self-attention mechanism. The self-attention mechanism helps in retaining contextual information. All the inputs are processed at once which helps in reducing infrastructure cost since it reduces the training speed and inference cost when compared to RNN's. The transformer architecture has revolutionized the way NLP tasks were solved. As compared to the previous architecture of RNN, Seq2Seq, transformers have made NLP applications not only versatile but accessible to most people. Based on the popularity of the transformer due to its versatility and performance there are large language models which are built. These are models trained on billion parameters and produce astounding results. Large language models (LLM) like BLOOM, GPT-3.5, GPT-4, Falcon 40B, Llama 7B are some common LLMs which are trained on more than billion parameters and generate meaningful and coherent results.

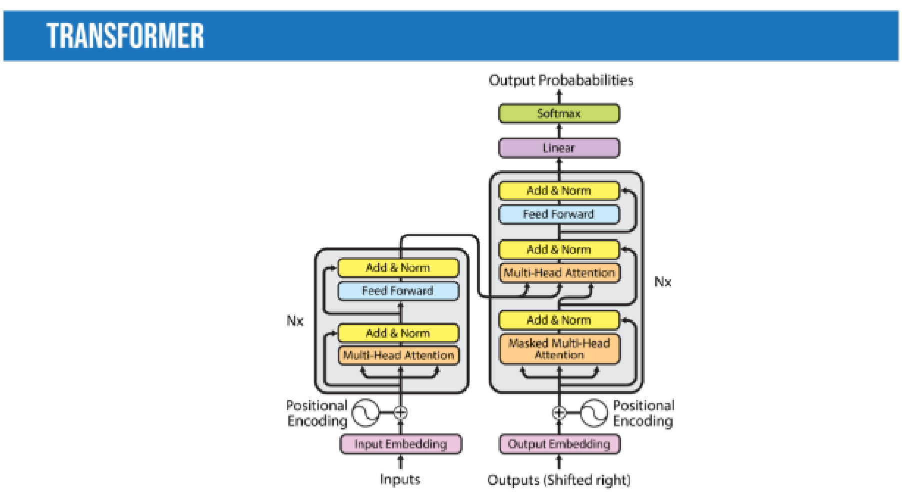


Figure 1.5: Structure of Transformers. Image source DeepLearningAi,2023

The above gives a comprehensive description of the evolution of natural language processing as a field of applied artificial intelligence as outlined in the resources of DeepLearning.AI (Deep Learning AI, 2023)

1.2.2 Transfer learning

According to Transfer learning (Torrey & Shavlik, 2009), Transfer learning as the name suggests the utilization of knowledge acquired from one task to other tasks. The comprehensive survey (Zhuang et al., 2020) outlined the mechanism of transfer learning and this has been explained in this part. In an ideal world the environment useful for machine learning applications to thrive is the availability of huge amount of labeled training data. This problem to some extent was solved by semi supervised learning. The semi-supervised learning approach requires a limited amount of trained data and uses vast amounts of unlabeled data to improve the training accuracy. Data collection is the most important and difficult job owing to various constraints like searching for the right kind of data, authorization of using the same, downloading or scraping the data, preprocessing the data. These are true for both labelled and unlabeled data. This is one of the reasons why with limited data the semi supervised learning models struggle with accuracy. The reason for such a design is because deep learning requires huge amounts of data. The vast amount of labeled data required to train a deep learning model is extremely difficult. The dependency of the target domain data can be effectively reduced using Transfer learning. Computation is another important aspect which has a profound effect in application and research. Most large models both in natural language processing and vision are extremely difficult to train from scratch using the commonly available resources. Most new large language models have well over 100 million trainable parameters and some even have 40 billion parameters. The data as well as the computation power required to train such models are extremely difficult to find. Using transfer learning a pre-trained model which has been trained on other similar tasks can be used to trained on the new dataset. An example can be the ResNet, VGANET which is a pretrained convolution neural network architecture which is pre-trained on ImageNet dataset which contains large amounts of images. A simple task of then identifying a cat vs a dog with the help of a handful of labeled pictures of cats and dogs can easily be achieved by using any of the pretrained models. While defining transfer learning it is important to define a task and domain. Mathematically as explained in the comprehensive survey (Zhuang et al., 2020) a domain can be defined as a domain $\mathcal{D}$ composed of two parts which are the feature space X and a marginal distribution $\mathcal{P}(X)$.This can be summed up as $D = \{X, \mathcal{P}(X)\}$ where X denotes the instance set which is defined as $X = \{x | x_i \in X_i \; i = 1 \ldots., n\}$.

A Task, $\mathcal{T}$ on the other hand contains a label space $\mathcal{Y}$ and a decision function f which can be defined as $\mathcal{T} = \{y, f\}$. The function f is an implicit function which is learned from the sampled data. The models which output the predicted conditional distribution of instances. In this case the decision function can be defined as $f(x_j) = \{P(y_k | x_j) | y_k \in Y, k = 1, .., |y|\}$. In most cases the domain can be observed with or without the field information. Transfer learning can be described as given some observations corresponding to $m^s \in N^+$ source domains and tasks and some observations about $m^T \in N^+$ target domains and tasks, transfer learning uses the knowledge of the source domain to improve the performance of the learned decision function as $f^{T_j}(j = 1, \ldots.., m^T)$ on target domains. This is the definition of multisource transfer learning. If m^S equals 1, then the scenario is called single source transfer learning. m^T represents the number of transfer learning tasks. Although a few studies are centered around the setting of $m^T \geq 2$, the current research focus on the setting of $m^T = 1$ which can then be further translated into $m^S = m^T = 1$. Typically, a classic scenario would be a trained model on some domain and a supply of a smaller number of target domain data. In this scenario, the observations are instances and models, and the objective is to learn a decision function of better accuracy. Domain adoption is another process that works with adapting one or more source domains to transfer knowledge thereby aiding in the improvement of the performance of the target learner. The explanation of the processes of transfer learning and the use cases in modern day has been outlined in comprehensive study (Dilmegani, 2023). A comparative study between traditional ML and Transfer learning is as follows:

Table 1.1: Traditional ML v/s Transfer Learning, Data source Dilmegani, 2023

Traditional ML	Transfer Learning
Isolated, learns a single task at a time.	New tasks are learned based on the previously learned tasks.
Model learns each task from scratch thereby increasing the time taken to learn each cost,	Learning process is faster and more accurate and less training data is required.

requires considerably more training data and computational power.	

Transfer learning, therefore, helps in transferring knowledge and getting a new model when there is not enough training data or the time to train an independent model is extremely high, sometimes even executing for days. An example of transfer learning is using pe-trained models to create a target model trained in a specific task. The broad steps as outlined in the article (Dilmegani,2023) of transfer learning are specified as follows:

- Selecting a source model: Choosing a pre-trained model which would transfer its knowledge to the target model.
- Model Adaptation: The next task is to adapt the source model and create a target model. The source model may have different features which do not match the target model. Therefore, the source model needs to be adapted to align with the target model before knowledge transfer.
- Training or fine tuning the source model to achieve the target model. This is done by using a new set of data and starting the training from a pre-trained source model.

A new model can also be developed depending on the business requirements. Creating a new model to transfer its knowledge to the main task is also a situation where researchers create new model. An example would be that a road transport authority wants to distinguish between trucks and buses but has very limited data to operate with. There might be quite a substantial amount of cars data. In such a scenario the model can be first trained on the cars data where the model extracts feature and learns. The model can then be used on the limited bus and trucks data to perform the classification.

Some of the applications using transfer learning are as follows:

- Image recognition
- Natural language processing
- Speech recognition

These applications are then used in several industries. Some common usages outlined in the article (Dilmegani ,2023) can be found in the below industries: -

- Autonomous Driving – A model trained in driving any autonomous vehicle on the road can be used to drive other vehicles, for example trucks and buses. Transfer learning can help in transferring the knowledge thereby reducing the turnaround time and improving accuracy. This can also be used in case of object detection on the road. A model trained in detecting cars can be used for detecting other vehicles like buses and trucks on the road.
- Gaming – A model trained to play a certain game can be used to get trained on another game of the same domain. Chess and Go being a classic example. A model trained in Go can be trained there by leveraging the knowledge already learned to get trained in chess.
- Health care – In the field of health care a model trained on identifying Electromyographic (EMG) signals can be used to detect Electroencephalographic brainwaves which are used for gesture recognition.

According to Wikipedia ("Hugging Face",2016), Hugging face, a company founded in 2016 offers the transformers library where open-source implementation of transformers is present. The implementation of the transformer's models ranging from text, image, and audio tasks. Apart from the models there are datasets, documentation, tutorials present. Each model has a model card which can be used to download and use the model. The hugging face library is one of the most widely used libraries across deep learning enthusiasts.

1.3 Objectives of the thesis

The objective of the project is as follows:

1. Create a unique dataset related to cars (hatchback / small cars) present in the Indian subcontinent.
2. Fine tune a language model to cater to the end user with information related to specification, expert review, and user review of each car.

2. Literature Review

The literature review for this research work can be divided into 2 broader divisions. This division is done based on the study performed with relevance to the research. The first significant research work before the introduction of Seq2Seq (Sutskever et al., 2014) architecture is Word2Vec (Mikolov et al., 2013). Furthermore, with the introduction of Seq2Seq (Sutskever et al., 2014) architecture improved performance and made language tasks more capable. Finally, the introduction of transformers (Vaswani et al., 2017) paved the way to more powerful models which ultimately led to the large language models. This study hence can be divided into the important work done prior to the introduction of the transformers and post introduction of transformers. Natural language processing gained significant momentum with the release of the paper Word2Vec (Mikolov et al., 2013). The study documented in this paper was focused on finding techniques which would help in learning high quality word vectors from big datasets containing billions of words. Before the experiments performed in this paper none of the previous research had successfully trained models with more than a few million. The experiments performed in this paper led to two novel model architectures which are Continuous Bag-of-Words model and Continuous Skip-gram model. The paper provides a technique to measure the quality of the vector representation with an understanding that not only do similar words tend to be close to each other, but words can have multiple degrees of similarity. This can be further explained with the example of nouns which can have multiple word endings. In this paper it was observed that similarity of words is not confined to the boundaries of syntactic rules. As for example an algebraic operation on vector(“King”) – vector(“Man”) + vector(“Woman”) would result in a vector which can be closely associated to the word “Queen”. This paper creates a model which would maximize accuracy of these operations which would outline the relations between words yet preserve the meanings. The semantic and syntactic regularities were measured using a comprehensive test set. Prior to the models proposed by the authors there had been many models which were proposed to estimate continuous of words. The authors evaluate the accuracy of their work with the previously proposed models. The models proposed previously are as follows:

- Feed Forward Natural Net Language Model - This probabilistic neural network model consists of input, projection, and output layers. The model takes N previous words in an encoded format using I-of-V coding where V is the vocabulary size. P acts as the projection layer where the input layer is projected. P has a dimensionality of N x D using a shared projection matrix. This is a complex architecture as the projection layers are dense in nature adding to the computational complexity. The computational complexity of such model can be termed as -

  $$Q = N \times D + N \times D \times H + H \times V \qquad (2)$$

  where N represents the previous words whose value is most taken as 10. The size of P might range from 500 to 2000. The hidden layers range between 500 to 1000. Computation hence becomes complex due to the presence of $N \times D \times H$.
- Recurrent Neural Net Language Model – This model has been known to overcome some of the drawbacks of the previous Feed Forward Natural Net Language Model. The RNN model consists of only the input and the hidden and the output layer. It is traditionally a more powerful model as it remembers the context. The model contains the recurrent matrix which has a hidden matrix which has a time delayed connection. This provides the functionality of short-term memory. The complexity of Recurrent Neural Net Language Model is as follows:

  $$Q = H \times H + H \times V \qquad (3)$$

  Where the word representation D , the word representation has the same dimensionality as that of the hidden layer H. The term $H \times V$ can be reduced to $H \times \log_2(v)$, with the help of hierarchical SoftMax however, $H \times H$ is what increases the complexity.
- Continuous Bag-of-Words Model – The Continuous Bag-of-Words model pr CBOW has an architecture which is like the architecture of the Feed Forward Natural Net Language Model. The only modification made to the neural network model is that the hidden layers are removed, leading to all words projected in the same position. This architecture is named as the Bag-of-Words as the words as the projection is never influenced by the center of the word in history. Four future and four history words are

taken as input and the target is to correctly classify the middle word. The training complexity can then be defined as follows:

$$Q = N \times D + D \times \log_2(v) \qquad (4)$$

The weight matrix between the input and the projection layer is shared for all positions.

- Continuous Skip-gram Model – The architecture of this model is like the Continuous Bag-of-Words Model. However, instead of predicting the current word based on the given context. The task here is to maximize the classification of words based on other words in the sentence. The current word is used as input to the long linear classifier with a continuous projection layer and finally predict a word within a certain range of the current and the future word. If the range is increased the quality of the word generated is better but the complexity of the model increases. The training complexity of the architecture is then defined as follows:

  $$Q = C \times (D + D \times \log_2(v)) \qquad (5)$$

  C is the maximum distance of the words, therefore if C is set to 5, for each training word a random word R in range $< 1; C >$ and R words are used from history and future of the current word as correct labels. Hence, a $R \times 2$ word classification is done with current word as input and $R + R$ are done. The value of C is set to 10.

Google News Corpus contains 6 billion tokens. The vocabulary size is restricted to 1 million most frequent words. The result of COBW is shown below:

Table 2.1: COBW results, Data source (Mikolov et al., 2013)

Training words	24M	49M	98M	196M	391M	783M
50	13.4	15.7	18.6	19.1	22.5	23.2
100	19.4	23.1	27.8	28.7	33.4	32.2
300	23.2	29.2	35.3	38.6	43.7	45.9
600	24.0	30.1	36.5	40.8	46.6	50.4

The accuracy on subset of the semantic – syntactic word relationship test set. Questions containing 30K most frequently used words are used. The over – all comparison between the proposed models and all the other state of the art models are as follows –

Table 2.2: Performance of models, Data source (Mikolov et al., 2013)

Model	Vector Dim.	Training words	Semantic accuracy	Syntactic accuracy	Total
CBOW	300	783M	15.5	53.1	36.1
Skip-gram	300	783M	50	55.9	53.3

These novel approaches helped in making new roads to the Natural Language processing tasks. The introduction of Seq2Seq and its success opened new areas and introduction of reinforcement learning gave rise to more sophisticated language models generating more coherent and meaningful answers.

2.1. Introduction of Seq2Seq models and reinforcement learning

Deep neural networks were extremely successful since it has succeeded in recognition-based tasks like speech and visual object recognition. The DNNs can perform parallel computation within a small number of steps. DNNs can also be trained with supervised back propagation if there are enough labeled training set containing information related to the network parameter. However, despite such power and flexibility DNNs can only be applied to problems whose input and targets can be encoded with vectors of fixed dimensionality. This constraint with the length of the input proved important as most information cannot always be expressed with inputs of fixed length. As speech recognition and machine translation are considered as sequential problems, question answering can also be considered as sequence-to-sequence mapping task. The problems with not knowing the length of the input and the output are a major problem for the DNNs. The goal of the paper proposed on Seq2Seq (Sutskever et al., 2014) solves the problem with the help of a LSTM (Long Short-Term Memory). The architecture is designed around LSTM so that 1 LSTM reads the input sequence one timestep at a time. This generates a vector representation of the input of fixed length. There is another LSTM which essentially is a Recurrent Neural Network language model conditioned to the input. The Recurrent Neural Network is a variant of the standard feed forward neural network to sequences. If we consider the input sequences of $(x_1, \ldots \ldots, x_n)$, the output of $(y_1, \ldots \ldots, y_n)$ is computed by the RNN with the help of the following equation:

$$h_t = sigm(W^{hx} x_t + W^{hh} h_{t-1}) \quad (6)$$

$$y_t = W^{yh} h_t \quad (7)$$

The issue is when the input and the output lengths are not known ahead of time. The input and out lengths can be different lengths. This can be avoided by using a fixed size vector to hold the output of 1 RNN and map the vector to another fixed sized input to the next RNN. This solves the problem but creates problems in training long-term dependencies. LSTM solves this problem as it learns the features of long-range temporal dependencies. The target of the LSTM is to estimate the conditional probability. The conditional probability being $p(y_1, \ldots., y_{T'} | x_1, \ldots., x_T)$ where $(x_1, \ldots., x_T)$ denotes the input sequence length and $(y_1, \ldots., y_{T'})$ denotes the output sequence length. The length T in the input sequence may be different from the length T' of the output sequence. The conditional probability is determined by the LSTM by first determining the fixed length context vector representation v of the input sequence $(x_1, \ldots., x_T)$. This information is provided by the last hidden state of the LSTM. The next step is computing the probability of $(y_1, \ldots., y_{T'})$ using the hidden state representation v. The final equation is as show below:

$$p(y_1, \ldots., y_{T'} | x_1, \ldots., x_T) = \prod_{t=1}^{T'} p\,(y_t | v, y_1, \ldots, y_{t-1}) \quad (8)$$

In the above equation for each $p(y_1, \ldots., y_{T'} | x_1, \ldots., x_T)$ is represented with a SoftMax over all words in the vocabulary. For this model each sentence needs to finish with an end-of-sentence symbol which helps the model to determine the distribution over all possible length sequences. The models presented in this paper differ from the standard LSTM implementation in 3 different ways which are the usage of 2 different LSTMs, namely one for the input and another for the output. This is done as it increases the trainable parameters without increasing the computational cost exponentially. The deep LSTMs were more powerful than the shallow ones hence an LSTM with 4 layers was chosen. The orders of the words were reversed while processing so instead of asking the model to map a, b, c to α, β, γ the LSTM had to map c, b, a to γ, β, α . This is done to have a proximity relation of a with α , b with β and c with γ . This helps SGD to form a connection between the input and the output. The dataset used for the experiments were the WMT'14 English to French dataset. The model was trained on 12 million sentences consisting of 348 M French words and 304 million English words. There was a fixed vocabulary used for both the languages. A total of 160,000 words from the source language and 80,000 of the most frequently used words from the target language were used. The out of vocabulary word was replaced by the "UNK" token. The evaluation was performed on a cased BLEU score to evaluate the quality of the translation. The BLEU scores were computed using multi-bleu.pl which is one of the variations of the BLEU score. The BLEU score recorded by the authors stands at 37.0 which has improved from the previous BLEU score of 35.8. The best results recorded were from an ensemble of LSTMs which had random initializations and order of batches. The ensemble of 5 reversed LSTMs and beam size of 12 renders a BLEU score of 34.81 which was the highest among all the state-of-the-art models. The performance of LSTMS on translation of big sentences from English to French had been extremely impressive.

The Seq2Seq model based on LSTMs has generated promising outcomes for conversational models. The sentences generated by Seq2Seq models, although good quality but have issues with long contexts. The models tend to predict one token at a time but not having much far sightedness for the impact on future outcomes. This poses a serious problem in terms of creating coherent and interesting dialogues. A proposed solution to this problem was given with the using reinforcement learning with pretrained Seq2Se2, Li et al. (201). The paper outlines the process of integrating deep reinforcement learning to curate the responses in dialog generation. The model created and experimented on by the authors has simulated dialogues between two virtual agents. Policy gradient method has been used to reward the generated sentences. This has resulted in creating dialogues which showcase informative, coherence and ease of answering. This paper addresses the problem of the Seq2Seq models and proposes solutions to overcome the problem. The problems addressed in this paper are as follows:

- Generation of high amount of "I don't know." responses regardless of the input. This can be due to the high frequency of generic responses found in the training which fit in as diverse range of conversational contexts.

- There is another problem observed is the issue of the model stuck in an infinity loop. The primary cause of this is due to the Maximum Likelihood based Seq2Seq models does not taking into consideration the repetitive responses. It has been observed that while in conversation between 2 machines fall into the repetitive response generation problems. It has been observed that both the agents started generating null and generic responses in an infinite loop, typically in the lines of "I don't know what you are talking about" or "you don't know what you are talking about".

The purpose of the paper then is to create a model which can do the following –

- Integrate reward system defined by the developer which can make the responses feel authentic and coherent.
- Understanding the context in the current dialogue to model the future impact.

The neural reinforcement learning method is used which will optimize the long-term rewards. The model used is an encoder-decoder model which simulates the dialogue between two virtual agents. The dialogue simulation provides ample opportunity to explore actions which may result in better dialogue generation or in terms of reinforcement learning a maximum reward. The reward for a good conversation is done by defining simple heuristic approximation where the rewards are more for forward looking, interactive and informative. The encoder decoder RNN is designed in such a way that a policy is defined over all possible utterance in the action space. The agent then learns the policy by optimizing the long-term rewards. This is done by using the policy gradient method. This contrasts with the MLE driven system where the MLE objective governs the Seq2Seq. Seq2Seq are known for generating good quality sentences and combined with it the capabilities of reinforcement learning thereby using good qualities of both the technologies. The proposed model in this paper is the simulation of conversation between 2 agents who take turns in conversing with each other thereby generating by executing dialogs generated one after the other as a sequence. As interaction between two agents can result in a dialogue generation which can be expressed as $p_1, q_{1,} p_2, q_2, p_3, q_3 \ldots p_i, q_i$. These generated sentences can be considered as actions which the agent will take according to the policy defined by the model. Policy search optimizes the parameters of the network model by maximizing the future reward. In this experiment the policy gradient method is used instead of the Q-learning method. This is because the model can be first initialized with the maximum likelihood parameters and then optimized using the policy gradient method to generate more coherent answers. Initializing the model with the maximum likelihood estimate helps to start with a model which already generates good quality responses but suffers from repetitive and non - coherent responses in long conversations. The policy optimization part includes the action, state, and reward respectively. The action is defined as 'a' which is a dialogue in a conversation. The action space can be infinite since any number and length of dialogues can be generated. The previous two dialogue turn can represent the state which is (p_i, q_i). The dialogue history is then converted into a vector representation by concatenating all the history and using it as input in the LSTM. The policy can be defined as $p_{RL}(p_{i+1}|p_i, q_i)$ which is the LSTM encoder-decoder. A stochastic representation of the policy is taken as the deterministic policy would result in discontinuous objective and optimizing such policy using gradient based models are extremely difficult. r is the reward which is calculated based on the following criteria:

- Ease of answering – The dialogue generated by the system or the machine in its turn should be easy to answer and should help in continuation of the conversation. The dull responses are penalized with the negative log likelihood. This helps in avoiding utterances generated like "I don't know what you are talking about." The reward function is defined as follows:

  $$r_1 = -\frac{1}{N_{\mathbb{S}}}\sum_{s \in S}\frac{1}{N_s}\log P_{seq2seq}\,(s|a) \qquad (9)$$

  In this equation $N_{\mathbb{S}}$ is the cardinality and N_s represent the number of the tokens present in the dull response of s. $P_{seq2seq}$ represents the likelihood output generated by the Seq2Seq models.

- Information Flow: The agents involved in the dialogue exchange should contribute in each turn some new information so that the flow of dialogue moves ahead. A repetitive response by the same agent is penalized due to semantic similarity. Negative log of the cosine similarity between the generated utterances is the reward here.

  $$r_2 = -\log\cos\left(h_{p_i}, h_{p_{i+1}}\right) = -\log cos\frac{h_{p_i}.h_{p_{i+1}}}{\left\|h_{p_i}\right\|\left\|h_{p_{i+1}}\right\|} \qquad (10)$$

- Semantic Coherence – The quality of responses is another important aspect which needs to be controlled. It is possible that different responses are generated with high rewards, but they are not grammatically correct. The reward function is defined as follows:

$$r_3 = \frac{1}{N_a}\log p_{seq2seq}(a|q_i, p_i) + \frac{1}{N_{q_i}}\log p_{seq2seq}^{backward}(q_i|a) \quad (11)$$

$p_{seq2seq}(a|q_i, p_i)$ is the probability of generating response a from the dialog utterance of (p_i, q_i). On the other hand, $p_{seq2seq}^{backward}$ represent the probability of generating the previous dialog utterance based on the current response a.

The final reward is for an action a is a weighted sum of all the rewards functions which are outlined above. The final reward is as follows:

$$r(a, [p_i, q_i]) = \lambda_1 r_1 + \lambda_2 r_2 + \lambda_3 r_3 \quad (12)$$

In the above equation $\lambda_1 + \lambda_2 + \lambda_3 = 1$. For the experiments $\lambda_1 = 0.25$, $\lambda_2 = 0.25$, $\lambda_3 = 0.5$ have been considered. The experiments were performed on the Open Subtitles dataset. The evaluation was performed on human judgements and automatic evaluation of conversation length and diversity. The scores are as follows:

Table 2.3 Conversation length, Data source (Li et al.,2016)

Model	# of simulated turns
Seq2Seq	2.68
Mutual information	3.40
RL	4.48

Table 2.4 Diversity scores, Data source Li et al. (2016)

Model	Unigram	Bigram
Seq2Seq	0.0062	0.015
Mutual information	0.011	0.031
RL	0.017	0.041

As explained in the above table of both conversation length and diversity scores the reinforcement learning model outperforms both the mutual information model and the sequence-to-sequence models.

The research of Li et al. (2016) showed the usage of Deep Reinforcement learning in the field of dialogue generation where the responses generated by the Seq2Seq model were optimized to create better coherent and fluent results. The work of Sankar et al. (2019) further takes the work to the next step by using the dialog attribute prediction there by reducing the size of the action space. Encoder - Decoder models have been used to model open dialogue conversations. This is done by framing the next utterance generation as a machine translation problem by treating the dialogue history as the source sequence and the next utterance as the target sequence. Maximum likelihood (MLE) is used to train the model end to end. This is a data driven approach which does not contain hand crafted structures like slot-value pairs or dialogue managers. The problem with the encoder-decoder approach is it lacks specificity in non-goal-oriented dialogs. The generation of generic or dull responses like ‘I don’t know’ are one of the primary problems faced by the encoder-decoder based models. The primary cause of such generic responses can be attributed to implicit imbalance datasets. Imbalances in dialogue datasets can be many-to-one and one-to-many. Many-to-one happens when there are very similar responses to different contexts in the datasets. The decoder learns to ignore the context and then cannot generalize a new context thereby predicting generic responses for all contexts. One-to-many, the dataset may exhibit certain types of generic responses that may be present in abundance compared to other plausible interesting responses. These types of datasets when trained using the Maximum Likelihood, the probability of the most observed responses, thereby little variance is observed in generating responses. This paper attempts to remove the repetitive and generic response problem found in the Open domain dialogue system. This is done by conditioning the response generation on interpretable discrete dialogue attributes and composed attributes. The dialogue attribute prediction is formulated as a reinforcement learning problem where in policy gradients were used to optimize the utterance generation using long term rewards. The action space was limited to dialogue attributes which made the policy optimization more practical and sample efficient. The authors therefore propose the new conditional dialog generation model. This new model will generate the next utterance which is conditioned on the dialogue attribute which is conditioned to the next utterance. The high-level dialogue attributes are predicted corresponding to the next response. The next utterance generated is conditioned on the dialogue context and predicted attributes. The dialogue attributes of an utterance refer to discrete features or aspects associated with the utterance such as dialogue acts, sentiments, emotions, speaker-id, speaker personality or other user-

defined discrete features associated with the utterance. Compared to the previous methods which only focused on the attributes of the next utterance as control variable this model predicts the attributes in an end-to-end manner. Reinforcement learning is used to predict the dialogue attribute selection. This is done by optimizing the policy initialized by the supervised training using REINFORCE (Williams,
1992). The supervised pre-training helps in generating utterance coherent with the dialogue history, the RL formulation encourages the model to generate utterances optimized for long term rewards. Thus, the optimization of the policy happens over a discrete dialogue attribute space thereby making it more efficient since it is not performed over the entire vocabulary. Using Reinforce to further optimize the dialogue attribute selection process the authors show improvement in specificity of the generated responses both quantitatively and qualitatively. The diversity scores of distinct-1 and distinct-2 are computed as fractions of unigrams and bi-grams in the generated responses. For this experiment the dialogue datasets are annotated using dialogue attribute classifiers. These classifiers are trained with tagged datasets like Switchboard and Frames. The quantitative and qualitative generating interesting responses are showcased. The results are showcased using the 2 attributes sentiment and dialogue acts. In this paper the focus is not to train the classifier with very high accuracy, but instead to showcase that with low accuracy are able to increase the token perplexity. The presence of different dialogue attributes results in increasing the regularization effect and generating interesting results.

The paper explores the impact of jointly modelling extra discrete dialogue attributes along with dialogue history for the next utterance generation and their contribution to addressing generic response problems. The model used for the experiments in this paper is the extension of the HRED model proposed by (Serban et al.,2016). The HRED model consists of a token level RNN encoder and an utterance level RNN encoder. This is to summarize the dialog context into a vector. This is followed by a decoder at token level to generate the next utterance. The mechanism of this joint probability can be put together as a sequence of dialog attribute prediction followed by generation of the next dialogue which are conditioned on the dialog attributes. The predicted dialog attribute is shown below: -

$$P(U_m, DA_{1:k}|U_{1:m-1}) = \prod_{i=1}^{k} P(DA_i|U_{1:m-1}) * P(U_m|\, U_{1:m-1}, DA_{1:k}) \qquad 13$$

In the above equation $DA_{1:k}$ represents k dialog attributes with respect to the utterance U_m. U_m is the mth utterance while $U_{1:m-1}$ is the past utterance. The dialog attributes correspond to the value of k. The value of k is set to 3 if the dialog attributes of sentiment, dialog acts and emotion are considered. This has been assumed that the dialog attributes are different from the dialog context. The attributes of the next utterances were predicted and then the previous context and predicted attributes are conditioned to generate the next sentence. The context vector containing the combined input is used to predict the dialog attribute of the previous utterance. The attributes of all previous utterances are passed through the RNN first. The last hidden state of the RNN along with the context vector is combined to predict the dialog attributes of the next utterance. In case if the dialogues present in the dialog dataset does not contain the dialog attribute, then in that case a classification MLP is used to classify the dialog attributes. This is done using a manually annotated dataset. The next step is the conditional response generation in which once the dialog attribute is predicted the next utterance is generated using the dialog context and the predicted attributes. The hidden states of the dialog attribute and the dialog context are used to combine the dialog. The below equation explains this combination:

$$h_{dec_{m,n}} = f_{dec}(h_{dec_{m,n-1}}, w_{m,n-1}, c_m) \qquad (14)$$

In the above equation the hidden state is represented by $h_{dec_{m,n}}$. This hidden state is after $n-1$ words in the mth utterance. c_m is represented by the following equations-

$$c_m = [s_{m-1}; da_m^1; da_m^2; da_m^3; \ldots\ldots; da_m^k] \qquad (15)$$

The summary of the previous $m-1$ utterances are represented by s_{m-1}. The dialog attribute embeddings with respect to the mth utterance is represented as $da_m^1; da_m^2; da_m^3; \ldots\ldots; da_m^k$. The dialog attribute prediction is done using reinforcement learning. The feature for optimizing policy to maximize long term reward is used for dialog attribute prediction. The models operating on maximum likelihood model focuses on increasing the utterances probabilities by influencing the model to place higher probabilities on the words which appear the most. The reinforcement learning methodology avoids this by maximizing long term rewards thereby promoting coherence and diversity. The issue observed in the reinforcement learning methodology where policy gradient is used is the large action space which at any given time consists of all possible worlds which the decoder can generate. The other issues observed with the previous approaches to working with vocabulary space. While policy optimization might create utterances with the highest reward, it can always be grammatically incorrect. The issues are addressed by the authors by avoiding the issues by limiting the action space to the dialog attribute. The action space for the model is therefore the dialog attribute and the state space is the dialog context. The dialog attributes are therefore viewed by the system as control variables which in turn help in creating more engaging conversations. The model comprises of encoders and attribute prediction networks. The previous utterances are encoded by encoders. This is then sent to the attribute prediction network. The attribute prediction network then predicts the action. The reward function used is ease of answering which computes the negative log likelihood of a set of dull utterances. The policy optimization is done by implementing the REINFORCE algorithm. The policy is represented by the following:

$$P_{RL}(DA_{1:k}|U_{1-m-1}) \tag{16}$$

The equation for the expected reward is as follows:

$$J(\theta) = \mathbb{E}[R(U_{1-m-1}, DA_{1:k})] \tag{17}$$

The following equation is used to estimate the gradient:

$$\nabla J(\theta_{RL}) = (R - b)\nabla \log P_{RL}\,(DA_{1:k}|U_{1-m-1}) \tag{18}$$

The reward baseline is represented by b. The model is trained on the Reddit, Switchboard and Frames dataset. The performance of the dialog act classifier on the Reddit dataset has given an accuracy of 68%. The dialog act prediction accuracy has achieved 71% for Frames dataset, 68.1% for Reddit dataset and 67.9% for switch board dataset. Please find the below diagram depicting the percentage of dull responses after finetuning with Reinforcement learning.

Generic Responses	RL(%)	Seq2Seq + Attr(%)
thank you so much	7.56	**7.32**
i dont understand why	**0.0**	15.64
i would love to see	**0.66**	5.65
i dont know how	**0.0**	13.97
i dont want to	**1.66**	3.99
i dont know why	**0.0**	3.66
i would love to be	**0.99**	2.21
i have no idea	4.31	**3.33**

Figure 2.1: Performance against generic responses. Data source Sankar et al. (2019)

A gradual progression is seen from the introduction of Word2Vec to Seq2Seq and then the use of reinforcement learning to create more accurate and coherent output. This method of using reinforcement learning to the natural language processing tasks was further enhanced by the work on Hierarchical reinforcement learning (Saleh et al., 2020).

The paper (Saleh et al., 2020) outlines the use of reinforcement learning with the action space limited to the dialog attribute unlike previous work where the action space comprises of the entire vocabulary. The approach proves good as it reduces complexity and improves quality of dialog generation. However, the main issues were that reinforcement learning was applied to open domain dialog generation at a word level which stopped the model from learning credit assignments for long term rewards. To minimize this issue a novel hierarchical reinforcement learning (HRL), VHRL. This uses policy gradient which is used to tune the utterance level embedding. This helps in providing flexibility and learning long-term rewards. In this paper the reinforcement learning is learning from self-play in which the model talks to a fixed self-copy. The reward functions are computed on generated conversations. The rewards are human centered which will minimize toxicity of the conversation to inappropriate responses. The rewards are designed based on the psychology of good conversation; therefore, the conversation is more positive and engaging and avoids repetitions. The approach outlined in this paper focuses on Hierarchical Reinforcement learning where architecture is divided into top and bottom level or worker policy and manager policy. At the bottom level of hierarchy, a set of options which are policies over actions interacts with the environment until they are terminated by the agent. At the top level a policy over action selects options to be executed until they are terminated. Once they are terminated a new option is selected and the process is repeated all over again. The hierarchy introduces different levels of abstraction and gives better long-term planning as compared to traditional reinforcement learning approaches.

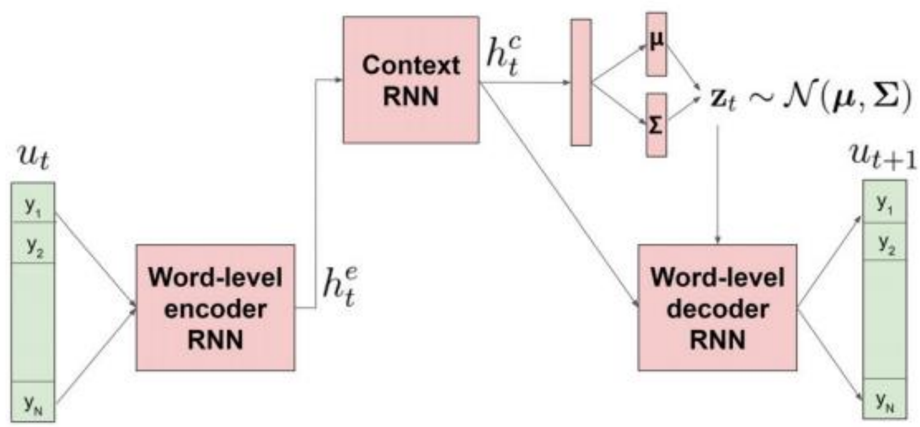


Figure 2.2: VHRED model architecture, Data source (Saleh et al., 2020)

The above diagram is that of a VHRED or Variational Hierarchical Recurrent Encode and Decoder. As shown in the diagram the architecture uses 3 neural networks to generate the next utterance. The Word-Level encoder RNN takes as input the utterance $u_t = [y_1, y_2, \dots, y_n]$ and encodes the utterance to create a vector $h_t^e = f^e(u_t)$. This is then passed on the context RNN which is the upper level of the hierarchy. This context RNN is updated post each utterance instead of each token. These less frequent updates help the context RNN to keep track of the long-term conversations. The output of the context RNN is $h_t^e = f^e(h_t^e)$. This produces the embedding z_t . This vector representation is fed to the word level decoder RNN f^d. This produces the output utterance of u_{t+1}, one token at a time. The standard reinforcement learning framework is adopted. The task of dialog generation is now considered as a reinforcement learning problem. The state is denoted as s_t, which is the model's knowledge of the dialog turn till utterance t. The dialog history and the dialogues generated are used to calculate the reward. All the previous work where reinforcement learning has been used in language generation has done it at a word level where the distribution modeled by the policy π is over generating the next word in the sequence. In this novel approach the context RNN is considered as the manager in the VHRED model responsible for generating sentences. The VHRED learns the probability distribution over the latent variable z_t which is formed from the manager.

REINFORCE methodology is implemented on a pretrained Variational Hierarchical Recurrent Encoder Decoder. This is to tune the variational component where the latent vector z_t serves as the continuous action. The distribution of the prior latent variables $p_\theta(z_t|s_t)$ serves as the manager policy and the distribution of the output words serves as the workers policy $\pi_{\theta|}(\widehat{y_1} \dots .. \hat{y}_t|z_t, s_t)$. This workers policy gets the parameters based on the managers decision. The generated utterance conditioned on the managers decisions z_t serves as the joint probability to determine the workers action a_t. The new approach proposed allows both the worker and the manager to jointly optimize the total expected future return by minimizing the loss as given below:

$$\mathcal{L}_\theta = -(\alpha R_t^m \ln p_\theta (z_t|s_t) + \beta R_t^w \ln \pi_\theta (a_t |z_t, s_t)) \tag{19}$$

In the above loss equation R_t^m is the managers future reward and it is defined as below:

$$R_t^m = \sum_{k=t+1}^{T} \gamma^{k-t-1} r_k^m \tag{20}$$

The works future reward is defined as below:

$$R_t^w = \sum_{k=t+1}^{T} \gamma^{k-t-1} r_k^w \tag{21}$$

This is like the approach defined in the REINFORCE algorithm. This shifts the model's focus and encourages the positive reward by the model and discourages the negative reward. α and β are hyper parameters which regulate the effects of the reward at each level of hierarchy. This architecture is in big contrast to the Hierarchical reinforcement learning where the manager and the workers are trained separately in a decoupled manner. This novel approach is termed Variational Hierarchical Reinforcement Learning. In such a closely coupled architecture the workers gradient flows through the manager and both flow through the encoder. The VHRL performs better than the rest of the models' state of the art models of VHRED, REINFORCE on a Likert scale. The results of human evaluation are shown below:

Table 2.5: Human evaluation results. Data Source (Saleh et al., 2020)

Model	Quality	Fluency	Diversity	Contingency	Total
VHRL	2.91	4.65	4.26	2.67	14.49

The below graph shows the toxicity level analysis where responses generated by VHRL are less toxic in nature.

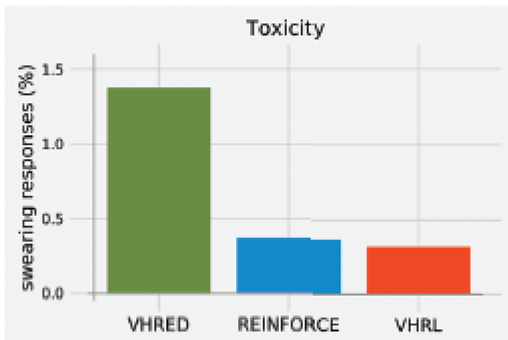


Figure 2.3: Toxicity Level analysis of different models, Image source Saleh et al.,2020

The above work (Saleh et al., 2020) by outlines the novel solution of reinforcement learning using variational hierarchical reinforcement learning where manager and the worker are jointly trained which provides greater flexibility for long term conversational reward. Furthermore, from here another diverse research on text generation with the help of reinforcement learning had been outlined in the works of Zhang et al. (Hierarchy Response Learning for Neural Conversation Generation). Traditional methods have explored the idea of diversifying the output at a word level or at a discourse level using a flat model. The paper proposes the idea of a hierarchical model to effectively capture different levels of diversity. This is done by using a conditional auto encoder. The framework is essentially two modules which consists of an expression reconstruction module which combines the hierarchical correlation between expression and intention. Another is the expression attention module to combine the expression and intention. This proposed new method can capture the conversion intent in a much more coherent and natural way. A new reconstruction loss is introduced to improve the training of hierarchical response learning for neural conversation generation. The paper investigates and tries to answer the following questions of automatically learning the hierarchical model to naturally capture the response generation process. The technique to learn and adjust the influence ratio between expression and content. As a proposed solution a 2-level probability structure is proposed to understand the process of response generation. The first level structure of the model randomly generates the expression and the second level. The second level, which is the expression attention fills out the expression with the content. The paper successfully recreates human response model is generated using a hierarchical model, create an end-to-end framework incorporating expression and content into dialog generation. The paper empirically proves that the new model can generate better responses than traditional state of the art models. Given a context c and dialog act a , the goal is to generate responses $(y_1, y_2 \dots y_n)$ such that they are coherent with the dialog act a . The model estimates the probability as explained below: -

$$P(y|c,a) = \Pi_t P(yt|y < t, c, a) \qquad (22)$$

Previous studies have shown that embedding the same in a Seq2Seq tends to generate generic responses or safe responses which are not coherent in nature and tend to discontinue the conversation. In the proposed model an expression e is modeled such that it can be thought off as a conditional distribution on the dialog act a which can be expressed as $p_\theta(e|a)$. The response is generated by feeding into the model the expression e based on $p_\theta(e|a)$. The simplified training objective can be seen as below:

$$P(y|c,a) = \int P(y|C,e)p_\theta(e|a)de \qquad (23)$$

If is it to be considered that the meaning of the expression is independent of the context, then the model can be trained by maximizing the variational lower bound which can be expressed as follows:

$$L(\theta, \phi; y, C, a) = -KL(q_\phi(e|y)||p_\theta(e|a) + \mathbb{E}_{e \sim q_\emptyset}[\log p_\theta\,(y|C,e,a)] \leq \log p(y|C,a) \qquad (24)$$

In the above equation KL is the Kullback Liebler divergence which measures the distance between the two distributions. The Daily Dialog Corpus was used to evaluate the proposed model. The evaluation metrics used are the greedy matching, embedding average and vector extrema. The proposed model gave a greedy matching score of 28.1, embedding average of 84.6, vector extrema of 37.9 and a total accuracy of 80.4 which is the most efficient than most of the contemporary models.

2.2. Introduction to Transformers

The transformers architecture was proposed in the process of solving a machine translation problem. Once of the reasons why the transformers architecture was introduced was the need to memorize longer sentences. For a long time, the recurrent neural network-based encoder – decoder models were widely used, and a lot of research was done using them as shown in the previous section. The use of reinforcement learning to create more coherent responses. More variations of the encoder – decoder models like Conditional Variational Encoder-Decoder, Hierarchical Encoder and Decoder as well we Variational Encoder and Decoder models. The primary objective of using these models was to generate high quality texts. However, there are some issues with the existing architecture of the RNNs. The primary problem is the sequential nature of the RNN. Every hidden state depends on the output of the previous hidden state. This poses a significant problem for the GPU. These models require a good amount of computational power. The GPUs must wait for the data to be available for processing. This does not help RNNs to be paired with technologies like CuDNN since it slows down the process. The second significant issue with the RNN is the long-range dependency. Memorizing things over an extended period poses a significant problem. For short sentences the memorization technique works fine but for longer sentences falters. Another issue is the dependencies which arise due to the contexts present in input tokens and output tokens. The dependencies of the words within a sentence are not very clearly demarcated thereby generating sentences which are not coherent in nature. The paper by Vaswani et al., 2017, proposed the idea of transformers. The blogpost “Why do Transformers yield Superior Sequence to Sequence (Seq2Seq) Results?” (Singh, 2019) explains in detail the transformer architecture and why is it so successful as compared to Seq2Seq architecture. Transformers solved the primary problem faced by the RNN encoder-decoder modules. The transformer architecture allows the encoder and decoder to have a view of the entire input sequence at the same time. These dependencies are directly modeled dependencies using attention. The transformers can be interpreted as the first transduction model which relies entirely on self-attention to generate sentences. The idea is implemented by the component called Multi Head Attention block.

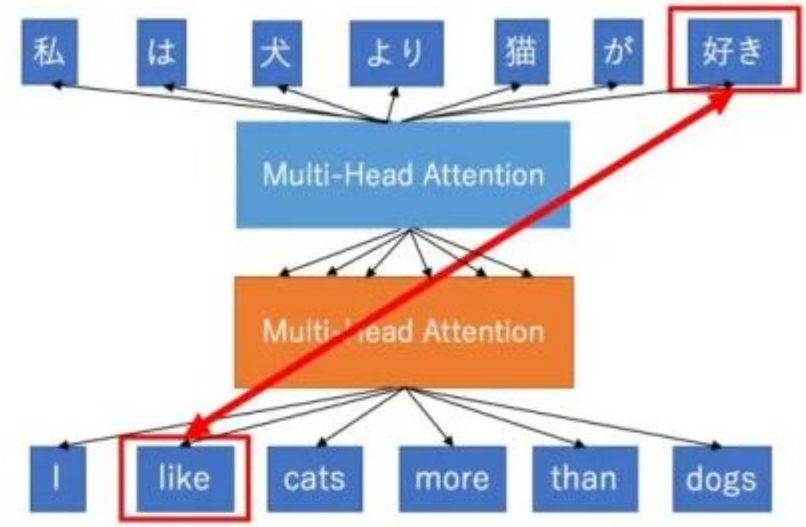


Figure 2.4: Multi Head Attention, image source Singh, 2019

As shown in the above diagram of the multi head attention it can be viewed that the length of the path (shown in red in the above diagram) is independent of the length of the source and target sentences. To the transformer the encoded input can be viewed as a set of key value pairs (K, V). The hidden state of encoder consists of the key value pairs. A query (QQ) is formed with the previous compressed output in the decoder. The next set of output is created by mapping the query and the key value pair set.

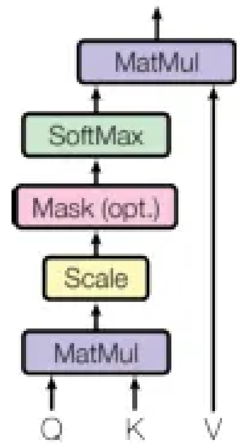


Figure 2.5: Scalar dot product attention (Image source: Singh, 2019)

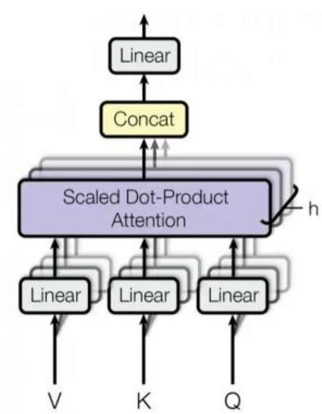


Figure 2.6: Scalar dot product attention with linear transformation. (Image source Vaswani, et, al.2017)

The scalar dot product attention is used in the transformer. The output of the same is the sum of the weighted values. The weight allocated to every value is determined by the scalar product of the query with all the keys. The equation of the same is as follows:

$$Attention(Q, K, V) = softmax\left(\frac{QK}{\sqrt{n}}\right)V \qquad (24)$$

The attention can be viewed as a dot product of the query and the key. A rescaling operation is done as the dot product between query and key increases in size and this poses a risk of exploding due to huge values. The attention mechanism ascertains the permissible information which can be called values and is dependent on keys and queries. The source sentences are embedded, and these are used by the encoder which uses the keys, values and queries and the decoder uses the output of the encoder for its keys, values and embedding of the target sequence which is then used in the query. Using multiheaded attention enables the transformer to retain the different aspects of the input which otherwise would have been lost if a single headed attention was used. Multiple attention weighted sum is performed instead of single ones capturing all the context information. Varieties of representations are learned by different linear transformations.

The encoder in the architecture is responsible for generating attention-based representation which can identify a piece of information from a large context. A stack of N=6 uniform layer. There are point wise fully connected feed forward neural networks which consists of multi head self-attention layer. Layer normalization and residual connection are present in each sub layer. The dimensions of all sub layer output data are off the same length which is 512.

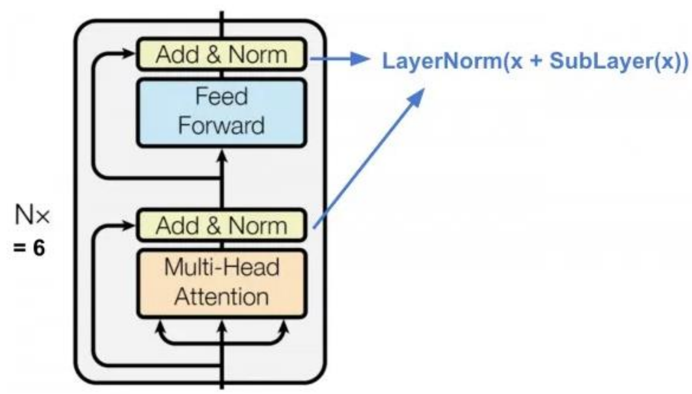


Figure 2.7: Transformer Architecture (Image source Vaswani, et, al.2017)

The residual connection as shown in the diagram above is taking the input and adding it to the output of the sub layer. The layer normalization can be expressed by the following equations.

$$y = LayerNorm(x + Sublayer(x)) \qquad (25)$$

According to the above equation the sublayer is the feed forward neural network with multi head attention. The primary mechanism of the encoder layer is to perform parallel matrix multiplication and then element wise transformation. The parallelism achieved makes the transformer architecture faster than all the state-of-the-art architectures. The fast architecture is thus achieved by stacking one layer over the other. The encoder is followed by the decoder.

The decoder has the same number of stacked layers as the encoder (N=6). The layer consists of a sub-layer of fully connected feed forward neural network as well as two sub layers consisting of multi headed attention mechanism. Residual connection and layer normalization is present in each sub layer. A masked multiheaded attention is created by modifying the first multiheaded attention layer. Masked attention is used as the inputs are kept hidden from the decoder and will be revealed in future time steps. The entire architecture containing the encoder and the decoder is as follows:

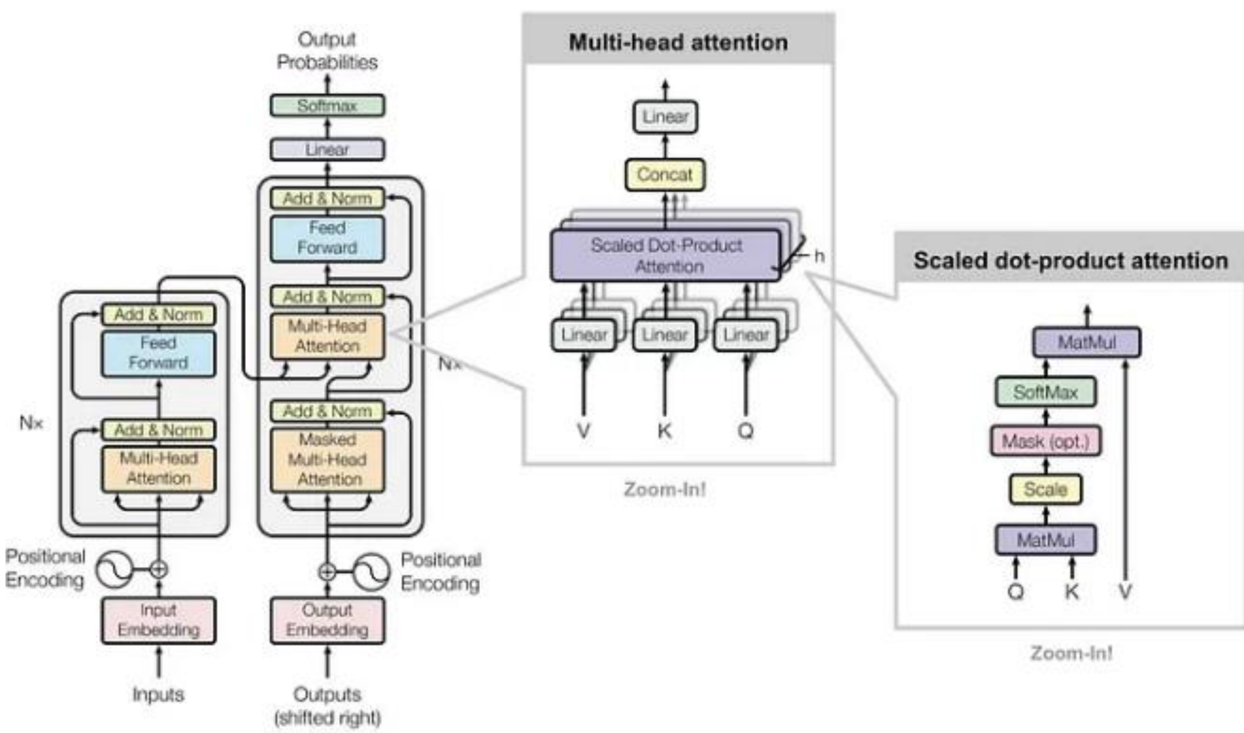


Figure 2.8: Complete Transformer Architecture (Image source Vaswani, et, al.2017)

The source and target sequence are passed through the embedding layers. This produces data of the dimension of 512. A sinusoidal wave based positional encoding is applied and summed with embedding output. A SoftMax layer and a linear layer are added to the final decoder output. In the above diagram positional encodings which encode the position of the input words as vectors and pass the same to add them to the input embeddings. The positional encodings used are as follows:

$$PE[pos, 2i] = \sin\left(pos/10000^{2i/d_{model}}\right) \quad (26)$$

$$PE[pos, 2i+1] = \cos\left(pos/10000^{2i/d_{model}}\right) \quad (27)$$

In the above equation, i is the dimension and pos represents the position. The model uses this to learn the relative positions. The model architecture was tested for machine translation of English-to-German and English-to-French WMT dataset translation tasks. The architecture records state of the art BLEU scores. The below diagram shows the performance of the transformer architecture against state-of-the-art models.

Table 2.6: Evaluation of Transformers (Data source Vaswani, et, al.2017)

Model	BLEU (EN – DE)	BLEU (EN – FR)
Transformer (Base Model)	27.3	38.1
Transformer (Big)	28.4	41.8

The transformers have thus helped revolutionize the way in creating language models and further large language models which can house somewhere around 40 billion parameters and therefore can perform an array of tasks. Traditionally tasks related to natural language processing which can be summarization, question – answering, translation have adopted the method of supervised learning. In 2018-2019, based on the transformer architecture a new architecture called Bidirectional Encoder Representations from Transformers (Devlin et al., 2018). The paper proposes an approach of pre-training language models which proved to be a major success and opened new areas of NLP tasks. The major limitation before the introduction of BERT was the use of the unidirectional model. The new language model BERT removes the unidirectional approach restricting the architecture options for pre-training. The previous models use the left to right architecture where in every token can see the token to their left. It has been observed that these kinds of restrictions were not optimal for fine tune-based approaches. This issue related to the fine tune-based approach was solved with the introduction of the BERT model. The issue of unidirectionality was resolved by the introduction of the masked language model. Some of the input tokens are chosen and they are masked randomly. The task then is to predict the original vocabulary id of the masked word based on its context alone. The MLM model allows the view of both the left and right context. This helps in training the deep bi-directional transformer. BERT has introduced a 2-step architecture which can be categorized as pre-training and fine-tuning. This unsupervised pre-training is followed by fine-tuning on specific tasks with labeled data, which adapts BERT to various downstream applications such as question answering and sentiment analysis. BERT has a unified architecture across different tasks. The difference between the pre-trained architecture and the final downstream architecture is minimal. BERT has a multi-layer encoder-based architecture. There are 2 types of BERT which are $BERT_{BASE}$ and $BERT_{LARGE}$. The $BERT_{BASE}$ architecture has 12 layers, 768 hidden layers and 12 self-attention head. The total parameters present are 110 million. The $BERT_{LARGE}$ has 24 layers, 1024 hidden layers, 16 self-attention head and a total parameter of 340 million. The pre-training of BERT is performed on 2 unsupervised tasks which are language modeling and next sentence prediction. WordPiece embeddings (Wu et al.,2016) with 30000 token vocabulary are used for the experiments. For the task of language modeling 15 % of the tokens are masked at random. Only the masked words are predicted instead of the entire sentence. The next sentence prediction can be thought of as a task of questioning and answering where the relationship between sentences is important. The GLUE benchmark (Wang et al., 2018) was used to for the model to finetune on. The F1 scores for QQP and MRPC are 71.2 for $BERT_{BASE}$ and 72.1 for $BERT_{LARGE}$. Spearman correlations are reported to be 94.9. BERT was thus the state-of-the-art model developed by google. In the year 2019, the paper "Language models are unsupervised multitask learners" (Radford et al., 2019) brought in the decoder only architecture. This work proposed the idea of a new model GPT-2. The largest GPT-2 model had 1.5 B parameters. The GPT-2 achieved state of the art results in 7 out of 8 tasks. The GPT-2 architecture, unlike the BERT, is a decoder-based architecture which uses the mechanism of masked self-attention along with positional encoding. The purpose of the new GPT-2 model is to predict the next token in the sequence. The model showed excellent results in machine translation and text generation tasks. XL-Net was another language model introduced in the year 2019 (Yang et al., 2019). The architecture was an alternative to BERT designed by google. It had 340 million parameters and was trained on 33 billion words. Subsequently OpenAI released GPT-3 in 2020 which is a proprietary product and can be accessed through API using the OpenAI end point. GPT-3 was trained on 300 billion tokens and has 175 billion parameters. Open AI released GPT-4 in the year March 2023 which is also a proprietary product.

3. Dataset Creation

3.1. Data Gathering

The aim of the project is to create a chatbot circled around cars. The car market in the subcontinent of India is a huge one. According to the official figures given by the Society of Indian Automobile Manufacturers a total of 38,90,114 passenger vehicles have been sold so far in the financial year of 2022-23. There is a total of 23 cars in the hatchback or small car segments hence the data on the hatchback was only considered. The automobile manufacturers and their hatch back cars currently being sold in the Indian market are as follows:

Table 3.1: Manufacturers and Cars

Manufacturers	Cars
Maruti Suzuki	Alto, WagonR, Celerio, S.Presso,Swift,Baleno,Ignis
Hyundai	Grand I10 Nios, I20, I20 N Line
Tata Motors	Tiago, Altroz, Punch
Renault	Kwid
Mahindra	KUV
Nissan	Magnite
MG	MG Comet EV
BMW	Mini Cooper
Strom	R3
PMV	EaSe
Mercedes	Mercedes Benz AMG A 45S
Toyota	Glanza
Citroen	C3

The initial search and prior knowledge gave a total of 26 cars. On searching in various online forums, it was revealed that out of the 3 cars from the manufacturers like Ford which had the Figo, Nissan which had the Micra and Toyota with its entry level hatchback Etios were discontinued and the list was finalized at the 23 shown above. This selection is quite broad as most of the cars have variants which differ from one another in terms of features. Once the data was identified the next step was to ascertain the type of data required for each car. The 23 cars in the hatchback space have some elements common and some different which creates their unique selling points. This information is at a basic level of specifications, colours, features and a very important feature which differentiates each car would be their fuel efficiency. However, in most hatchbacks it has been observed that the fuel efficiency is similar hence it can be paired with the interior and exterior features helps consumers in the decision-making process. The most important data which were decided to be used for each variant of car is as follows –

Table 3.2: Features and Details of cars.

Features	Details
Specification	Variants
	Seating Capacity
	Fuel Tank Capacity
	Engine Displacement
	Power
	Torque
	Transmission
	Fuel
	Fuel Efficiency
	Ex-showroom price
	Boot Space
	Colour
Expert Review	The reviews given by various car experts.

User Reviews	The reviews recorded by the car owners.

Once the features based on each car and its variants are finalized then the source of the data needs to be finalized. This is because there are various sources providing information on the details required. On careful consideration 10 websites were considered for gathering data for the project. The websites are as follows:

Table 3.3: 3rd party data source

3rd party data source
https://www.zigwheels.com/
https://www.cardekho.com/
https://www.carwale.com/
https://www.bsmotoring.com/
https://autoportal.com/
https://www.drivespark.com/
https://www.team-bhp.com/
https://www.autocarindia.com/
https://www.overdrive.com/
https://www.cartrade.com/

Apart from this the data of each car was also looked from the cars official website which are as follows:

Table 3.4: Official website of car manufacturers.

Official Manufacturers Website
https://www.marutisuzuki.com/
https://www.hyundai.com/in/en
https://www.tatamotors.com/
https://www.renault.co.in/

https://www.mahindra.com/
https://www.nissan.in/
https://www.mgmotor.co.in/
https://www.citroen.no/
https://www.bmw.in/en/
https://www.strommotors.com/gev
https://pmvelectric.com/product/ease/
https://www.mercedes-benz.co.in/?group=all&subgroup=see-all&view=BODYTYPE
https://www.toyotabharat.com/showroom/glanza/

The specifications of the car are all taken from the official website of the vehicle for accuracy of the data. The expert reviews and the user reviews are not present on the official website of the vehicle. These were curated from over 90 articles. Every data source listed has at least 4 expert reviews for the consumers. These reviews are read and curated so that there are no repetitions in any of the reviews gathered. Some of the cars have more than one review. For example, cars which were launched 5 years ago have been through certain facelifts. These cars typically have reviews pertaining to long term feedback, short term feedback and feedback given before a launch. Feedback ranging from the time the car was first launched till the time a new facelift was launched in the market. This provides the consumer with constant update on the performance of the car over a period. In the experiments performed in this paper the data has been restricted to the most current review of the car. A total of max 4 reviews have been taken into consideration. The User reviews on the other hand are more complicated to curate from. This is because the expressions of different people are different. An example of such a case would be smileys representing emotions regarding the car. There can be issues like incomplete sentences, sentences written in complete or partial vernacular language which a language model will not interpret. In some cases, there are also inappropriate vernacular languages used which had to be filtered out. All the sources pertaining to expert reviews and user reviews were read and selected as input to the language model. Another set of data containing the frequently asked questions and answers were also collated to add variety to the data.

3.2. Data Scraping

The next phase after selection of the data is the extraction of the data. The amount of data to be extracted is huge. An estimate on the amount of data to be extracted can be given by a total of 21 cars having 4 expert reviews and at max 100 user reviews curated to avoid repetition, toxic content, vernacular content. To extract all these data from the web, a web scrapping tool "Octoparse" was used. Octoparse is a free web scrapping tool which creates workflow which can be used to extract necessary information. The various type of data was extracted using the workflow as follows:

Figure 3.2: Octoparse Workflow Pipeline

The workflow as shown above can be created in the tool 'Octoparse' itself. The workflow is automatic, however in many cases the data extraction is difficult owing to numerous information in the web page which leads to inconsistent data selection. The workflow pipeline has options to extract more data in web pages which have "Next" button or "Load more data" button in the page. In such cases once the automatic pipeline is created it needs to be manually corrected to extract data. The data selection appends in the ‘loop Item’ as shown in the picture above. The example of the same is given below:

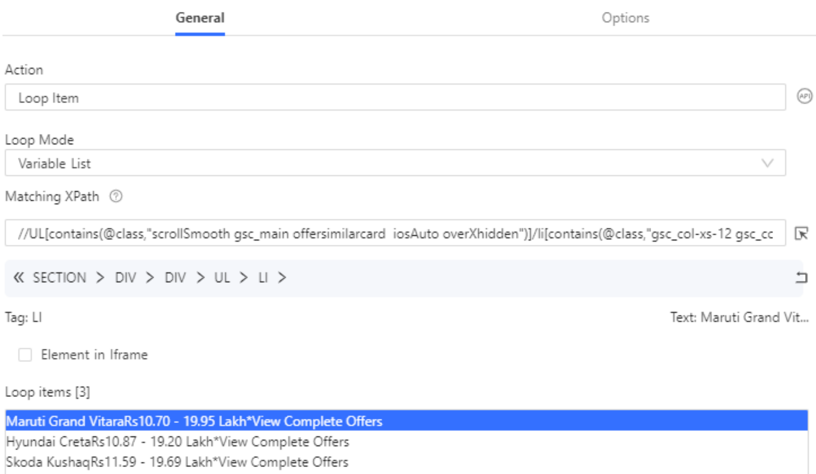


Figure 3.3: Octoparse dynamic X-ref window

The ‘Matching XPath’ can be updated to retrieve different data combinations from the table. The loop can be used to then check if the intended data is being retrieved. The data retrieved can be seen in the “Extract Data” tab. The “Extract Data” section is shown below:

Figure 3.4: Octoparse selected data

The data selected in the “Extract Data” section can be manipulated to suit the requirements. For example, the data in the columns can be added / deleted or modified. Once the data is confirmed the ‘Run’ button starts a backend job which can then extract the data in any format (excel, csv, json.xml) or export it to the database with options such as Google Sheets, SqlServer, MySql. This process is used to collect data from the sources which comprise both the official site as well as the various data sources website. The data for each car is extracted in multiple excel files from different websites. A total of 7 to 8 excel files are present for each car. Files containing the specifications, expert review, and user review.

3.3 Data Pre-processing

The data in the excel files were all unstructured. Unstructured raw data can very well be used in training, but this will only hold true if the training process is performed multiple times. First on unstructured raw data and the second time for a specific downstream task which can range from summarization, question answering or conversational. As the data collected is primarily in chunks and due to time constraints, the format of the data cannot be modified, therefore, to have coherent and consistent data it needs to be pre-processed before sending it to the model. The pre-processing is required to remove data in vernacular language primarily. As these websites were pan India hence there could have been any language. This was performed manually. Once the data was cleaned the next step was to collate all the files to have 1 file per car. The files for each car were kept in a separate folder in the google drive and using Python (open-source programming language Python (www.python.org)) the files were collated together. The files were renamed in a format such that the specification file is the first to come followed by the details of the expert review and finally the user review. A view of the files in the google drive is as follows:

```
file_name /content/drive/MyDrive/Final - Tata Altroz/tata_altroz_1.xlsx
file_name /content/drive/MyDrive/Final - Tata Altroz/tata_altroz_2.xlsx
file_name /content/drive/MyDrive/Final - Tata Altroz/tata_altroz_3.xlsx
file_name /content/drive/MyDrive/Final - Tata Altroz/tata_altroz_4.xlsx
file_name /content/drive/MyDrive/Final - Tata Altroz/tata_altroz_5.xlsx
file_name /content/drive/MyDrive/Final - Tata Altroz/tata_altroz_6.xlsx
file_name /content/drive/MyDrive/Final - Tata Altroz/tata_altroz_7.xlsx
```

A function was created which would read through all the files with the extension .xlsx and collate them to create a single file. The result was 23 such single files were created which were then collated to form 1 file containing all the data of all the cars.

3.4. Data creation for parameter efficient tuning

The dataset created to perform the experiments are chunks of data. The Falcon 7B needs data in the format of question and answers. This data was extracted from the website QnA section. There was a total of 650+ pairs of questions and answers obtained from the various sites. These question-and-answer pair were curated, and a Json file was created. This file was fed to the Falcon 7B model to create a QnA model. A sample from the Json file is given below.

```
{
  "questions": [
    {
      "question": "question",
      "answer": "answer"
    },
    {
      "question": " Is Hyundai Grand i10 Nios available in automatic
transmission?",
      "answer": "Yes. Hyundai Grand i10 Nios available in automatic
variants."
    },
    {
      "question": " Is Hyundai Grand i10 Nios available in petrol
version?",
      "answer": "Yes. Hyundai Grand i10 Nios available in petrol engine
option."
    },
```

4. Methodology

Data selection and gathering is an important phase in this project as it deals with the creation of an entirely new data set. This section deals with methodology employed to use the data and achieve the objective by performing experiments on the models. The choice of a generative model plays a vital role in creating the generative conversational automotive model. There is a list of conversational models available. The introduction of transformers which brought in the stacking mechanism brings in more flexibility from an architectural perspective. Language models have emerged since 2018 with the onset of BERT. Currently the models which can be chosen from are as follows: BERT, RoBERTa, GPT-2, BART, T5, LaMDA, XLNet, DistilBert, BLOOM.

4.1. Model Selection

Selection of language models depends largely on the objective of the task. A key factor in model selection is the considering the size of the model. A detailed analysis is outlined in the blogpost" Choosing the right language model for your NLP use case" (Lipenkova , 2022). Therefore, it's a twofold approach where in a balance needs to be achieved when selecting the model for the experiments. Size of the model is playing a vital role since a large model with close to a billion parameters cannot easily be trained or finetuned. Fine-tuning is an obvious choice as it would mean utilizing a pre-trained model which would require less data to be fine-tuned. Training language models requires GPU and although there are some platforms like Google Colab, offering free GPU for a limited period it is normally unstable while performing experiments for a longer time. The pro and pro+ versions of Colab are paid services like Amazon Sagemaker and Google Cloud Platform. Apart from size the most important feature is the kind of tasks that a model can perform with respect to the objective of the experiments. The data collated is in the form of chunks. So, for 23 cars there are 23 text files with each file containing a chunk of structured text containing specification, expert review, and user review. A comprehensive view of the available models and their performance in various tasks are shown below:

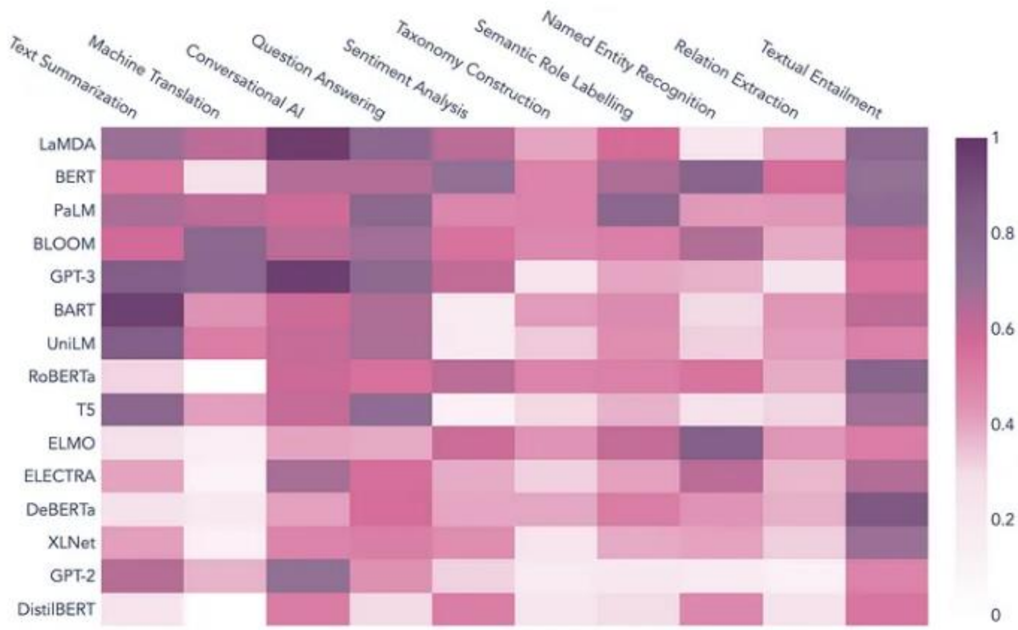
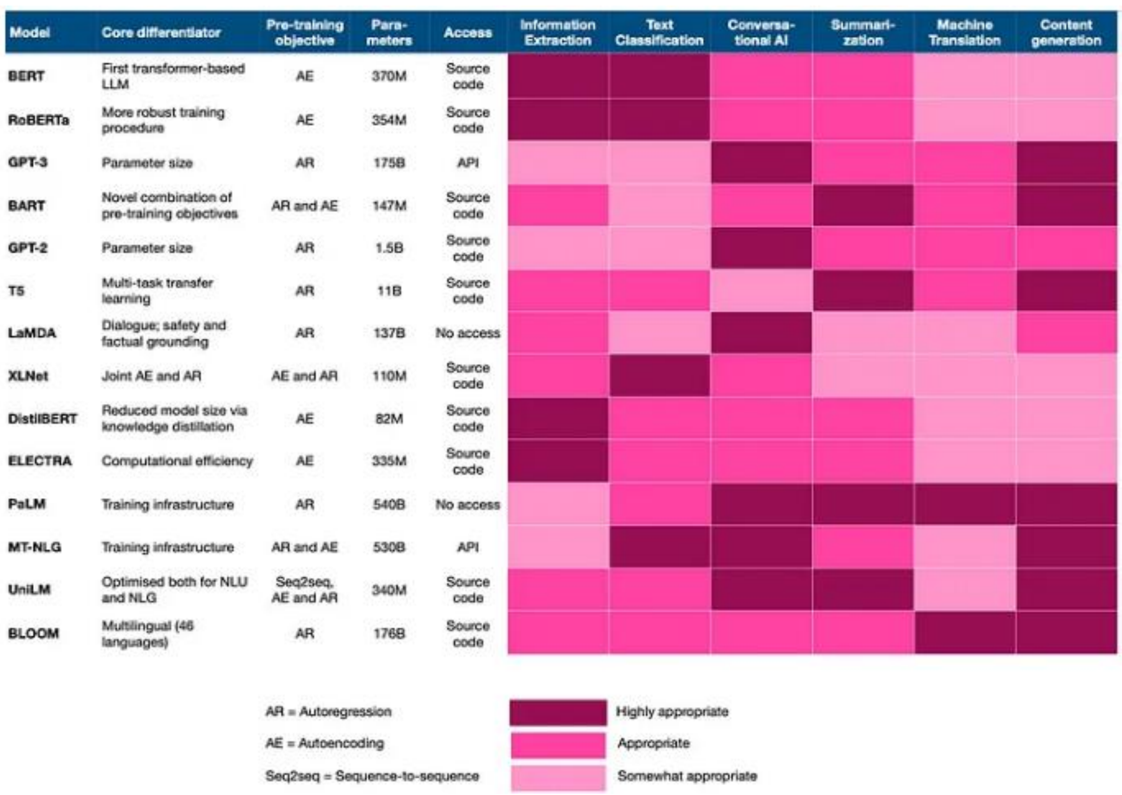


Figure 4.1: Comprehensive view of the models and the tasks, image source (Lipenkova, 2022)



Model	Core differentiator	Pre-training objective	Para-meters	Access
BERT	First transformer-based LLM	AE	370M	Source code
RoBERTa	More robust training procedure	AE	354M	Source code
GPT-3	Parameter size	AR	175B	API
BART	Novel combination of pre-training objectives	AR and AE	147M	Source code
GPT-2	Parameter size	AR	1.5B	Source code
T5	Multi-task transfer learning	AR	11B	Source code
LaMDA	Dialogue; safety and factual grounding	AR	137B	No access
XLNet	Joint AE and AR	AE and AR	110M	Source code
DistilBERT	Reduced model size via knowledge distillation	AE	82M	Source code
ELECTRA	Computational efficiency	AE	335M	Source code
PaLM	Training infrastructure	AR	540B	No access
MT-NLG	Training infrastructure	AR and AE	530B	API
UniLM	Optimised both for NLU and NLG	Seq2seq, AE and AR	340M	Source code
BLOOM	Multilingual (46 languages)	AR	176B	Source code

Figure 4.2: Language models and their preference based on task. image source (Lipenkova, 2022)

The task to be performed by the model is to help the consumer with information based on which the consumer can take appropriate action. This can be achieved in several ways, one of them is creating a question-and-answer model which will answer relevant questions on the topic, it can also be a conversational model which can create a conversation around cars with any interested consumer, another model is a prompt completion model which can receive a prompt from the user and then finish the prompt. In the above list when it comes to achieving a balance between size and task, the GPT-2 model is chosen. The GPT-2 model is extremely potent for solving auto-regression tasks. GPT-2 is based on the transformers architecture and generates token or the next word in a sequential manner. During text generation the model GPT-2 predicts the next sequence of tokens. The autoregressive approach helps the GPT-2 to generates coherent and meaningful sentences. The objective is to create a generative model which will provide information on cars and GPT-2 is expected to receive the input prompt from the user and complete the context with the relevant information. The context can range from the specification of the car to its internal review and the user review of the car. As per the above chart GPT-2 has 1.5 billion parameters which is the largest variant of GPT-2. There are other variants of GPT-2 available which are GPT-2 small which has 124 million parameters, followed by GPT-2 medium which has 355 million parameters. GPT-2 Large which has 774 million parameters, and GPT-2 XL which has 1.5 billion parameters. The pre-trained versions of these models are available in the Hugging face library. This helps anyone to download the model or use the model card to train the model. This is because the Falcon family of models were launched for commercial use much later (a month back, June – July 2023). This time constraint of creating data exclusively to test the utility of the model for this specific use case could not be done therefore this model was tested to see the capability with the limited data. A total of 650+ pairs of questions and answers were scrapped and changed to a JSON format to feed it to the model. The decision to use a Falcon 7B model has future benefit. If the model fairs well with a limited amount of data then it can be considered for future use where maintenance and execution of the model would be cost effective.

4.2. Model Architecture

The model architecture consists of 2 models which were used for the experiments. The GPT-2 was the primary model and the Falcon 7B was the secondary model.

4.2.1. GPT-2 architecture

The GPT-2 model released in the year 2019 by OpenAI can be termed runs on the mechanism of next word prediction. The paper "Language models are unsupervised multitask learners." (Radford et al., 2019) was centered around systems which had the capability perform multiple tasks rather than being experts on a single task. The language model thus proposed was GPT-2. The blogpost "The Illustrated GPT-2" (Alammar, n.d.) gives a detailed description of all the architectural details of GPT-2. It was trained on 40 GB of data. GPT-2 small has 117 million parameters, GPT-2 medium has 345 million parameters, GPT-2 large has 762 million parameters and GPT-2 XL has 1.5 billion parameters. The GPT-2 small has a dimension of 768, the GPT-2 medium has a dimension of 1024, GPT-2 large has a dimension of 1280 and GPT-2 XL has a dimension of 1600. These dimensions can be considered the number of decoder layers stacked in the model. A traditional transformer architecture was built on the stacks of encoder – decoder blocks. Since the inception of transformer, the architecture has transformed giving rise to encoder blocks or decoder blocks models. These are either stacks of encoder or decoder. Bert (Bidirectional Encoder Representations From Transformers), a model created by Google in 2018. This is an encoder only architecture. GPT-2 on the other hand is a model developed by Open AI this is a decoder only model. On the other hand, Transformer XL is a model which is built by stacking recurrent decoders. GPT-2 generates one token at a time. The GPT-2 is an auto regression model which works in such a way that when one token is generated it is then added to the sequence of input. This sequence serves as input in the next step. The model XLNET on the other hand has both autoregresion used in GPT-2 and the power of incorporating the context on both sides like BERT. The GPT-2 architecture can be viewed as a stack of decoders as shown below:

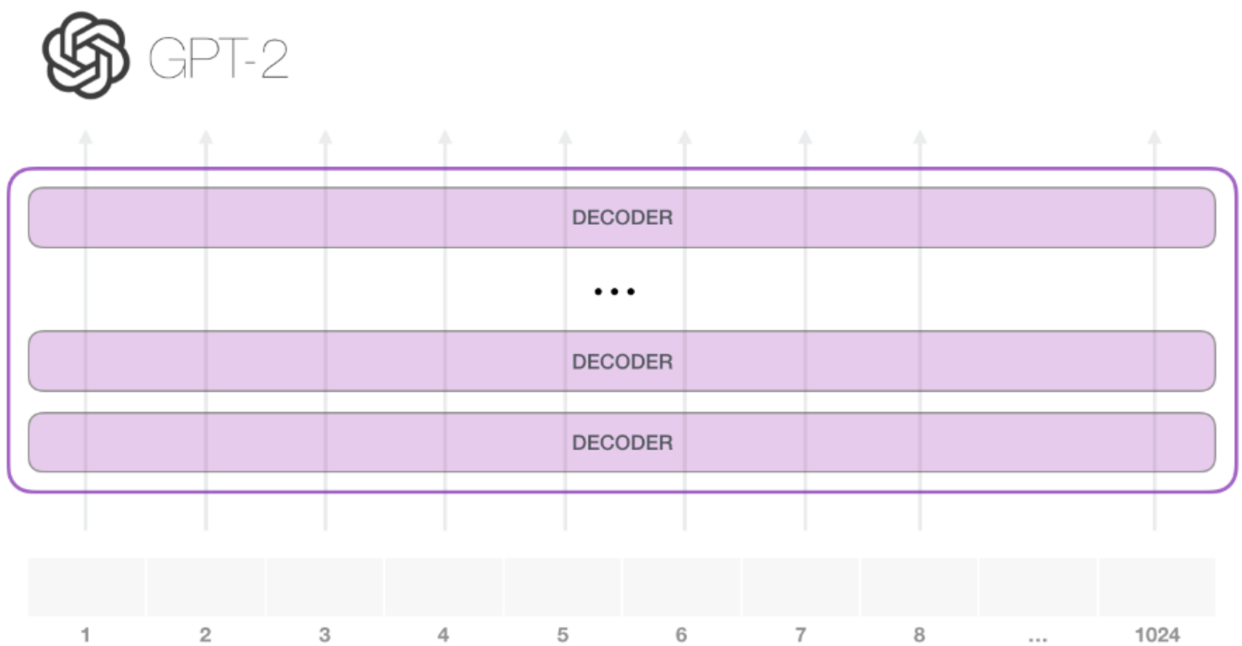


Figure 4.3: GPT-2 internal architecture. Image source (Alammar, n.d.)

The original architecture of transformer describes 2 types of transformers blocks the encoder and the decoder block. The GPT-2 is a decoder only architecture which masks the future tokens. This is called masked self-attention. A normal self-attention block allows any position of the view of the tokens to its right and a mask self-attention does not allow this to happen. The masking is implemented as a matrix called attention mask. If there is a sequence of 5 words to be processed by the model, then in this scenario of language modeling the sequence is absorbed in five steps one per word. This is shown in the below image.

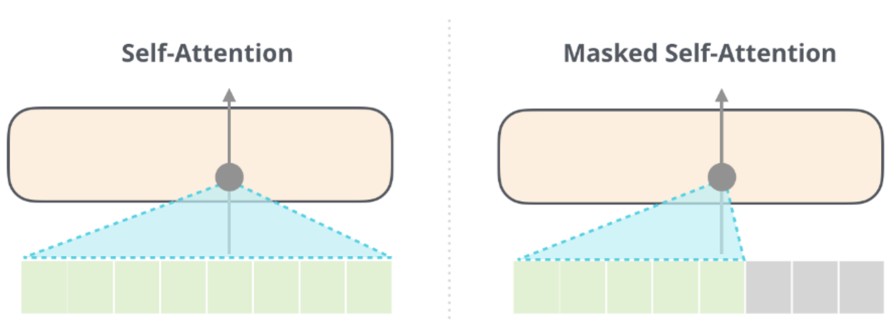


Figure 4.4: Self- Attention and Masked Self Attention. Image source (Alammar, n.d.)

The stack of decoder takes in a single token as input. The token is processed through the subsequent layers of the decoder and that creates a vector. The vector is then scored against the vocabulary of the model which is 50,257 words. This scoring mechanism picks up the token with the highest probability. In the subsequent step the model's output from the previous step is taken and fed to the model as the input to generate the next token prediction. The process of feeding the input encoding starts with the embedding of the input word searching in the embedding matrix. This is the component which is received as a part of the trained model. The below diagram shows the token embedding matrix.

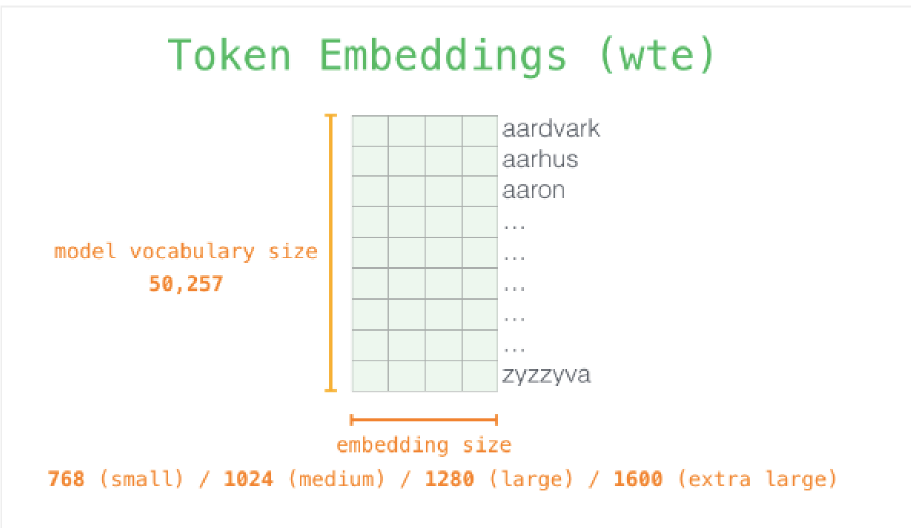


Figure 4.5: Token Embeddings. Image soure (Alammar, n.d.)

Each row as represented in the above diagram is a word embedding which is essentially a list of numbers representing a word and some context to it. Positional encoding which is a process to define the order of the words is incorporated into the input. A matrix consisting of positional encoding of each of 1024 positions in the input. This is how the input words are processed before sending them to the first block of the transformers. The below picture describes the sending of the input to the first block of the transformer.

Figure 4.6: Data processing in transformer. Image source (Alammar, n.d.)

As shown in the diagram sending a word in the first block of the transformer would mean looking up at the embedding and adding the positional encoding vector for the position. In each block the input is sent through the self-attention process and then through the neural network. Every stack has its own weight, self-attention layer and neural network sublayers. The first block of the transformer takes the input processes the token and transfers the resulting vector to be processed by the next stack. This goes on for all the stacks present in the model. The self-attention is done by the following steps: -

- Creation of queries, keys, and values - The first step is the creation of the queries, keys and values typically addressed to as Q, K, V. The very input for the token of the first block would be the embedding of the word and the positional encoding of the place where the word is present. Each block has its individual weights. The weight matrix is used to create the queries, keys, and values. The multiplication operation which is performed is basically concatenating the query, key, and value vector for the given word. Self-attention is performed multiple times on different parts of the vector Q, K, V. Splitting the attention head is reshaping the long vector into matrix.
- Scoring – The next phase is the token to get scored against all the keys of the other tokens. These keys were calculated in the attention head in the previous iterations.

- Sum – Each value is multiplied to its score and then summation is performed on them. This produces the result of self-attention heads. The attention heads are concatenated into a single vector. This vector is then turned into a homogeneous form to be passed on to the next level.
- Projecting – The model then learns the best mapping of the self-attention results into a vector which can be fed to the feed forward neural network. This creates the large weight matrix which projects the result of the attention heads into the output vector of the self-attention sub-layer. The result is then ready to be sent to the next layer.

The process of the input token which has the context associated with it by the self-attention mechanism is done in the feed forward neural network section of the block. This consists of 2 layers. The first layer is 4 times the size of the model. The GPT-2 small has a size of 768 so the network would have 3072 units. Factor 4 is taken as it has been observed that multiplying it with 4 gives the model a good representational capacity to handle the tasks. The second layer has the purpose of projecting the result from the first layer to the model dimensions which is basically a multiplication operation. The result of the transformer block for this token is this multiplication operation. The details of the architecture present within a single decoder layer is shown below.

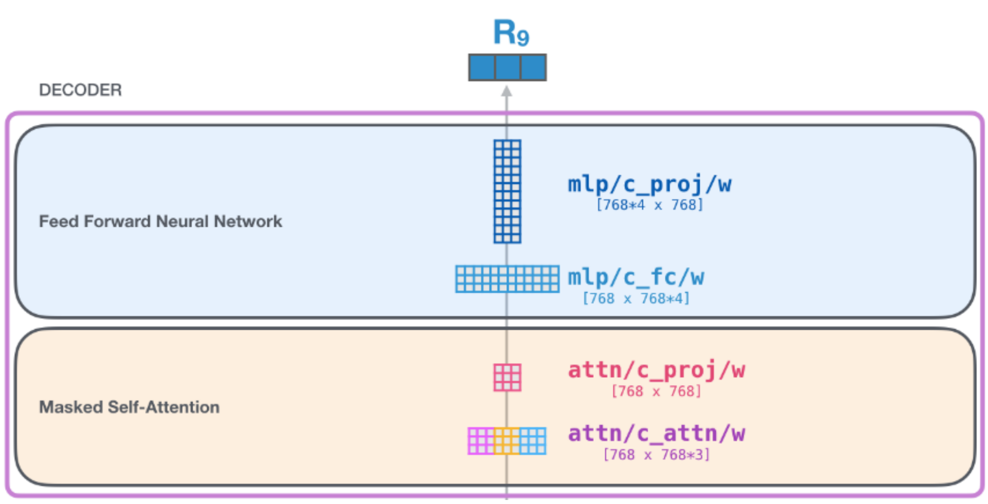


Figure 4.7: Inside the decoder stack. Image source (Alammar, n.d.)

The above is a single block containing its own set of weights and a single token embedding matrix and one positional encoding matrix is present.

4.2.2. Large Language Model (Falcon 7B)

The Falcon family of large language models have been introduced into the hugging face family last month (June-July 2023). The largest available model from the Falcon family is Falcon 40B which has 40 billion parameters. One of the first blog posts on the Falcon family of models was written by the hugging face team "The Falcon has landed in the Hugging Face ecosystem" (Leandro von Werra et al., 2023). The Falcon 7B is used in this experiment as it only takes approx. 15 GB of GPU memory there by making it possible to execute even with a single GPU. The models are trained on RefinedWeb which is a massive dataset. The Falcon 7B model can be fine-tuned using PEFT (Lester, B,2022). The mechanism of PEFT along with QLoRA can be used to finetune adapters which are placed on top of a 4-bit model. This makes it possible to train a very small fraction of parameters instead of a huge model. Once the large language model has already been trained then it need not be saved again since the base model is always kept frozen.

Use of QloRA technique – The hugging face blog post "Making LLMs even more accessible with bitsandbytes, 4-bit quantization and QLoRA " (Belkada et al., 2023) explains the mechanics behind QloRA. The QLoRA technique is an efficient approach to fine-tuning which significantly reduces memory usage. A 65B parameter model can be fine-tuned on a single 48GB GPU, while preserving full 16-bit fine-tuning task performance. The gradients are backpropagated through a frozen, 4-bit quantized pretrained language model into Low Rank Adapters (LoRA) as part of the QLoRA process. QLoRA utilized a 4-bit NormalFloat (NF4) data type optimized for normally distributed weights. The datatype used by QLoRA is This means it compresses the pre-trained model. The strategy of double quantization reduces the average memory footprint by quantizing quantization constants, while the memory spikes are managed by the Paged Optimizers. This helps in finetuning more efficiently.

4.3. Experiments

The experiments were conducted with the objective in mind that the model should be good enough to give valuable information to the consumer. The data created is in chunks of pre-processed texts hence the appropriate choice would be to have a language model which could process the text , learn the information provided by the new dataset and complete the task of text generation. The section contains both the experiments conducted on the primary model GPT-2 and the new model Falcon 7B.

4.3.1. Execution of GPT-2

The model selected is a pre-trained GPT-2 medium. This is the model which has a token limit of 1024. The model is available in the hugging face library. The objective is then fine tuning a pre-trained GPT-2 medium model on the created dataset. Hardware plays an important role in fine-tuning. GPT-2 medium can be termed as an advanced language model which works efficiently in creating coherent texts. Now this is achieved by complex computaton which also uses parallelism. The size of the model and depth of the same plays an important role in deciding the hardware. GPT-2 medium has a total to 24 layers. A GPU is therefore necessary to fine-tune the model optimally. Google Colab free version was inintially used to train the model but the free version is unstable and the avaialability of GPU is not consistent . A Colab pro version was tried however it only gives 100 computational units hence performing extensive training and multiple experiments were not sufficient to train. Thereforre Google Colab Pro+ subscription was used to train the model. The A100 GPU was used with a high ram. The Colab Pro+ also gives the benefit of back ground execution. The 23 txt files containing data of 23 cars are collated to form a single file. 15% of the total data are separated out for validation. The tokenizer used is the GPT-2 tokenizer is from the pre-trained tokenizer from hugging face library. The train and test datasets are then written out as .txt files and stored in the google drive which is accessible from google colab. The path of the test and the train files are then sent to the load_dataset function present in the dataset library provided by hugging face. The load_dataset loads the dataset directly from the directory which in this case is the google drive accessed from google colab. The TextDataset is the class provided by the hugging face library. This takes as input the tokenizer , the path to the training file or testing file, and the block size which is set to 128. The block size refers to the sequence of tokens that will be processed by he language model in a single forward or backward pass. It is a hyperparameter which determines the length of the text sequence which will be used for training and evaluation. The pytorch framework is used for execution. The hugging face library provides the Trainer class. The training argments are used to access the customization duing training. The TrainingArguments help in overriding the hyperparameters of the Trainer class. The hyperparameters used for training are as follows:

- output_dir – The directory where the output model checkpoint will be saved.
- overwrite_output_dir – This is set to True which overwrites the directory with successive new checkpoints.
- num_train_epochs – The number of epochs used for training is 400.
- per_device_train_batch_size – This is the bach size required for training which is set to 32.
- per_device_eval_batch_size – This is the evaluation batch size which is set to 64.
- eval_steps – This is the hyperparameter which is sets the interval at which the models performance is evaluated. In this experiment it is set to 400.
- save_steps – This defines the steps after which the model is saved.
- warmup_steps – This hyperparameter is set to 500. This will keep the learnig rate low for the 500 steps and then increase it to an initial assigned value of `5e-5`.
- Learning_rate = The learning rate is `5e-5`.

The model was trained for 400 epochs and the details of the training parameter after the training is as follows:

Table 4.1: Training arguments

global_step	30400
training_loss	0.2886754341815647
train_runtime	7425.0562
train_samples_per_second	130.262
train_steps_per_second	4.094
total_flos	6.31804133376e+16
epoch	400

The loss at the end of 400 epochs was recorded at 0.2886754341815647.

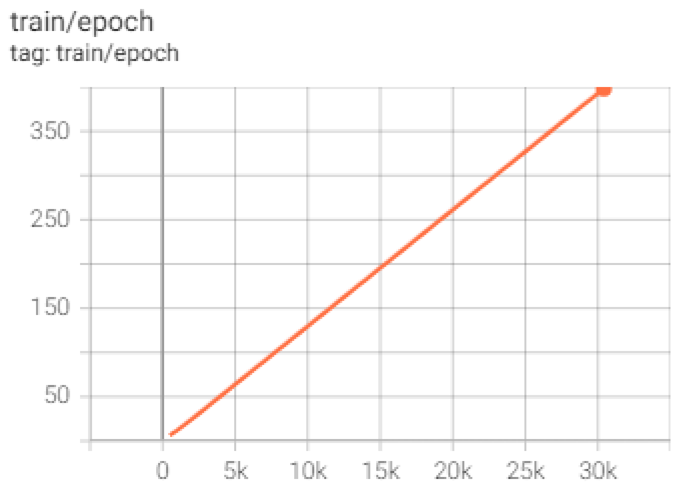


Figure 4.8: training per epoch

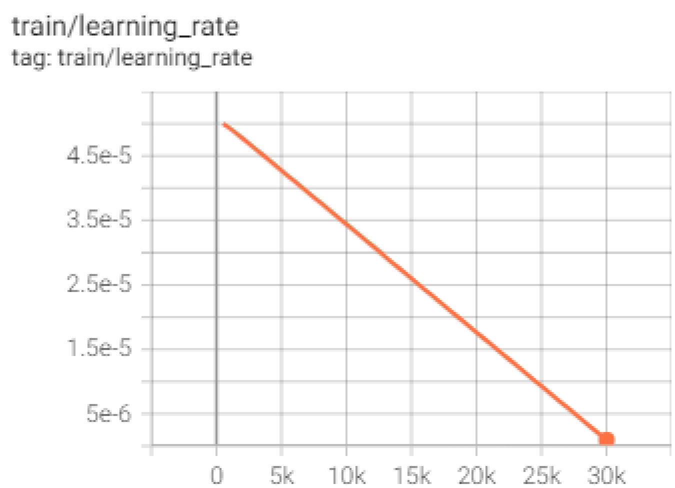


Figure 4.9: learning rate throughout the training steps

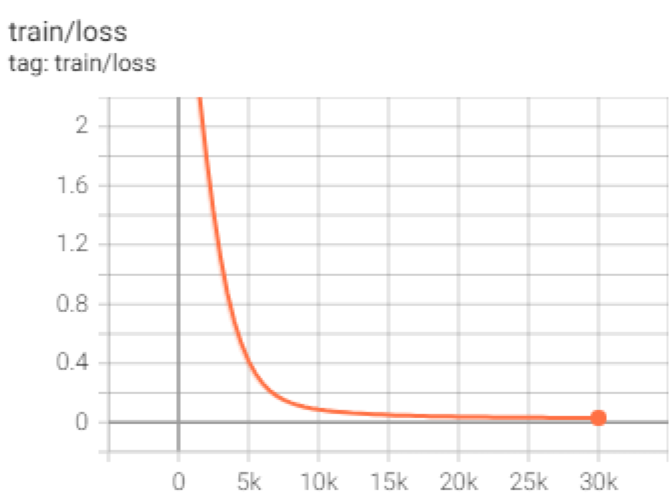


Figure 4.10: loss reducing over training steps.

4.3.2. Execution of Falcon 7B

The model name used is a pre-trained version of Falcon 7B from the hugging face library "tiiuae/falcon-7b". The dependencies used are as follows :

```
!pip install -Uqqq pip --progress-bar off
!pip install -qqq bitsandbytes==0.39.0 --progress-bar off
!pip install -qqq torch==2.0.1 --progress-bar off
!pip install -qqq -U
git+https://github.com/huggingface/transformers.git@e03a9cc --progress-bar
off
!pip install -qqq -U git+https://github.com/huggingface/peft.git@42a184f -
-progress-bar off
!pip install -qqq -U
git+https://github.com/huggingface/accelerate.git@c9fbb71 --progress-bar
off
!pip install datasets==2.12.0 --progress-bar off
!pip install loralib==0.1.1 --progress-bar off
!pip install einops==0.6.1 --progress-bar off
```

The library bitsandbytes is used to get LoraConfig, PeftConfig,PeftModel, get_peft_model, prepare_model_for_kbit_training. The 4 bit model is loaded. These libraries help in squeezing the model and executing a large model with the most basic hardware. The hyperparameters for the model are as follows :

Table 4.1.1: Training arguments for Falcon 7B

Training Parameters	Values
per_device_train_batch_size	1
gradient_accumulation_steps	4
num_train_epochs	5
learning_rate	2e-4
fp16	True
save_total_limit	3
max_steps	800
optic	paged_adamw_8bit

warmup_ratio	0.05

A total of 5 epochs / 800 steps were executed on a dataset which has 650+ pairs of questions and answers. A total off 4718592 parameters out of 3613463424 which is 13% of the total parameters were trained. This process of parameter effcient training has proven that training large language models can be relatively easier with a small infrastructure.

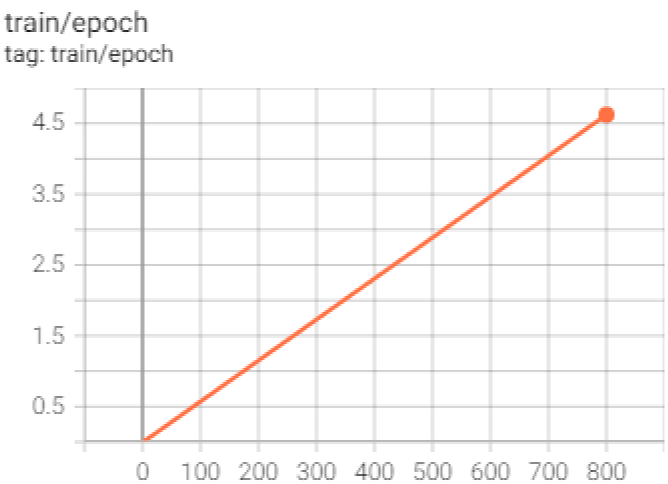


Figure 4.11: training per epoch

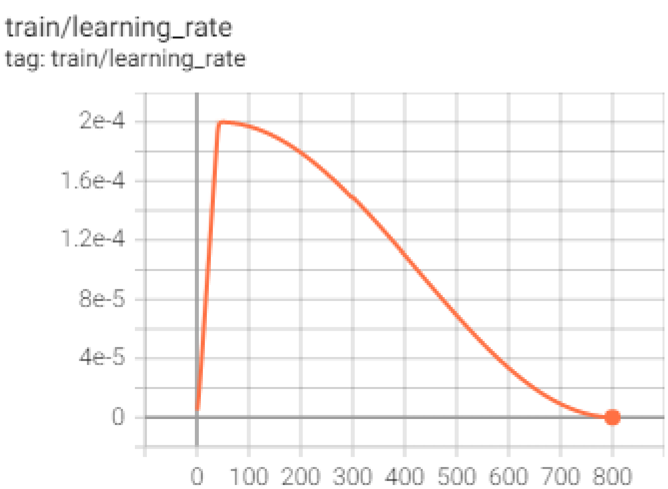


Figure 4.12: Learning rate through out training steps

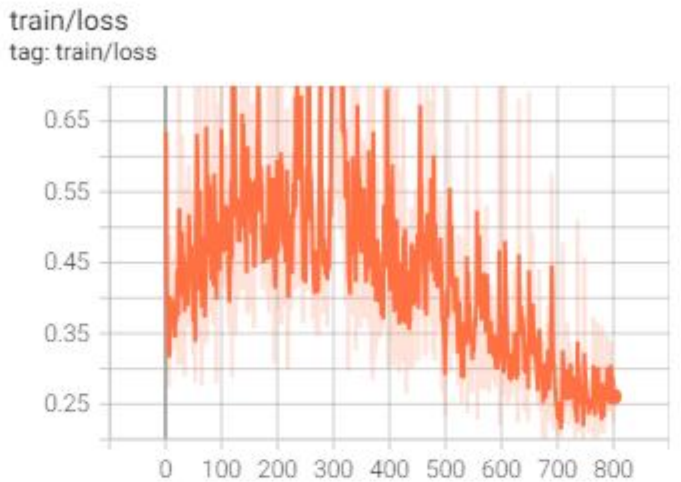


Figure 4.13: Loss reducing over training steps.

4.4. Results

The results of both GPT-2 and Falcon 7B has been listed below. The testing of the model was perfomed by a using a dataset consisting of prompts based on the car details. The objective of the project is to have a model which generates information about cars thereby helping the end user to have access to quick information related to hatch backs in the Indian subcontinent.
The metrics used are BLEU, ROUGE, TER. Bilingual Evaluation Understudy or BLEU measures the overlaps of n-grams which are representing the consequetive sequence of words. This is measured between the generated and the reference sentence.This is mostly used for machine translation however, it has been widely used as evaluation metrics for other applications as well. Rouge is commonly used to measure the accuracy of summaries. It can also be used to measure the performance of short texts by comparing both the texts. TER is also a metric used to measure the accuracy of machine generated translations. Ter tracks the numbe of insertions, deletions and substitutions required by the texts to align functionally. Human evaluation is pehaps the best form of evaluation to be done on text generation. Unfortunately, for this project this could not be possible hence this will be considered for future work. However, some snippets of execution have been given in the result sections.

4.4.1 Results of GPT-2

In order to test the capability of the model a dataset of 30 prompts were taken and fed to the model. The output of the same is described in the below table.

Table 4.2: Ground truth vs Generated text

Ground Truth	Generated Text
The seating capacity of Maruti Suzuki Ignis Sigma is 5 people.	The seating capacity of Maruti Suzuki Ignis Sigma is 5 people.
The fuel tank capacity of Maruti Suzuki Ignis Zeta AMT is 32 litres.	The fuel tank capacity of Maruti Suzuki Ignis Zeta AMT is 37litres.
The max torque of Maruti Suzuki Ignis Alpha AMT is 113Nm@4200rpm.	The max torque of Maruti Suzuki Ignis Alpha AMT is 113Nm@4400rpm.
The transmission of Maruti Suzuki Ignis Alpha AMT is Automatic.	The transmission of Maruti Suzuki Ignis Alpha AMT is Automatic.
The boot space of Maruti Suzuki Swift VXI AMT is 268L (Litres).	The boot space of Maruti Suzuki Swift VXI AMT is 268L (Litres).
The fuel efficiency of Tata Altroz XM Plus S is 19.33 kmpl.	The fuel efficiency of Tata Altroz XM Plus S is 19.8 kmpl.
The ex-showroom price of Maruti Suzuki Ignis Zeta AMT is INR 751000.	The ex-showroom price of Maruti Suzuki Ignis Zeta AMT is INR 693000.
The fuel efficiency of Maruti Suzuki Ignis Zeta AMT is 20.89 kmpl.	The fuel efficiency of Maruti Suzuki Ignis Zeta AMT is 18.8 kmpl.

The boot space of Maruti Suzuki Ignis Zeta is 260L (Litres).	The boot space of Maruti Suzuki Ignis Zeta AMT is 268L (Litres).
The fuel tank capacity of Tata Tiago XE is 60.	The fuel tank capacity of Tata Tiago XE CNG is 37litres.
The engine displacement of Tata Tiago XZ Plus is 1199 cc.	The engine displacement of Tata Tiago XZ Plus CNG is 1197 cc.
The max torque of Tata Tiago XZA Plus Dual Tone Roof AMT is 95Nm@3500rpm.	The max torque of Tata Tiago XZA Plus Dual Tone Roof AMT is 115Nm@3250+/-100rpm.
The fuel tank capacity of Tata Altroz XE CNG is 37.	The fuel tank capacity of Tata Altroz XE CNG is 37litres.
The ex-showroom price of Tata Altroz XM Plus is INR 745000.	The ex-showroom price of Tata Altroz XM Plus S is INR 840000.
The boot space of Tata Altroz XM Plus is 345L (Litres).	The boot space of Tata Altroz XM Plus is 336L (Litres).
The seating capacity of Maruti Suzuki Ignis Alpha AMT is 5 people.	The seating capacity of Maruti Suzuki Ignis Alpha AMT is 5 people.
The max torque of Tata Altroz XM Plus S is 200Nm@1250-3000rpm.	The max torque of Tata Altroz XM Plus S is 89Nm@3500rpm.
The ex-showroom price of Tata Altroz XE Plus is INR 680000.	The ex-showroom price of Tata Altroz XE Plus is INR 660000.
The colors available for Tata Altroz XM Plus S are Arcade	The colors available for Tata Altroz XM Plus S are solid fire

Grey, High Street Gold, Opera Blue, Downtown Red, Avenue White, Harbour Blue and Cosmo Dark.	red, pearl arctic white, solid fire red with pearl midnight black, pearl metallic lucent orange, metallic silky silver, pearl midnight black, pearl metallic midnight blue.
The AMG A 45 S 4MATIC Plus has a toque of 500Nm5000-5250rpm.	The AMG A 45 S 4MATIC Plus has a toque that is 22.35 kmpl.
The AMG A 45 S 4MATIC Plus has a seating capacity of 5 people.	The AMG A 45 S 4MATIC Plus has a toque that is 22.35 kmpl.
The seating capacity of Maruti Suzuki Alto K10 VXI Plus AT is 4 people.	The seating capacity of Maruti Suzuki Alto K10 VXI Plus AT is 5 people.
The fuel efficiency of Maruti Suzuki Alto K10 VXI S-CNG is 33.85 km/kg.	The fuel efficiency of Maruti Suzuki Alto K10 VXI S-CNG is 26.8 km/kg.
The torque of Maruti Suzuki Alto K10 VXI Plus is 82.1Nm@3400rpm.	The torque of Maruti Suzuki Alto K10 VXI Plus is 113Nm@3500rpm.
The fuel efficiency of Hyundai i20 N Line N8 DCT is 20.25 kmpl.	The fuel efficiency of Hyundai i20 N Line N8 DCT is 18.75 kmpl.
The ex-showroom price of Hyundai i20 N Line N8 iMT Dual tone is INR 1136000.	The ex-showroom price of Hyundai i20 N Line N8 iMT Dual tone is INR 775000 only.
The boot space of Hyundai i20 N Line N8 DCT is 300L (Litres).	The boot space of Hyundai i20 N Line N8 DCT is 268L (Litres).

The fuel efficiency of Hyundai i20 N Line N6 iMT Dual Tone is 20.0 kmpl.	The fuel efficiency of Hyundai i20 N Line N6 iMT Dual Tone is 19.8 kmpl.
The fuel type of Maruti Suzuki Ignis Alpha is Petrol.	The fuel type of Maruti Suzuki Ignis Alpha AMT is Petrol.
The engine displacement of Tata Tiago XT is 1199 cc.	The engine displacement of Tata Tiago XT Rhythm is 1199 cc.

The generated sentences or utterances were used to evaluated on the metrices of rough, bleu and tar. The results are as follows:

Table 4.3: Evaluation score

BLEU Score	0.7793296089385475
Rough Score	0.6141304347826086
Ter Score	0.14467254967254967

The metrices were executed on texts which were quick and easy to access. One of the objectives of the model is to generate coherent and meaningful text aligned to the context. Informaton like reviews of interior , exterior as well as pros and cons of a particular car are valuable and aid in the decision making process of the consumer when it comes to buying or simply searching for a car. The response to these texts should not be a single sentence but a comprehensive take on the details asked. Length of the generated context plays a vital role here. For information which are inherently elaborate in nature the maximum length is set to 200. There are 10 prompts and their generated

Table 4.4: Propmt generating longer text.

Prompt	‘Tata Altroz Engine and performance'
Text Generated	‘Tata Altroz Engine and performance.Tata has decided to only offer this automatic with the naturally aspirated petrol engine. It makes

	the same power and torque as the manual, which is 86PS and 113 Nm. With the new transmission, the key responsibility of this drivetrain is to be a smooth and lag-free commuter. And that it does very well. As soon as you get off the brakes, the crawl acceleration is gentle and smooth. The DCT has quick gear changes and because the engine performance is linear and not particularly quick, they remain jerk free. The quick gear changes are good enough to haul four people in city traffic with minimal luggage. And because the engine performance is linear and not particularly quick, they remain jerk free.As soon as you get off the brakes, the crawl acceleration is gentle and smooth. The DCT has quick gear changes and because the engine performance is linear and not particularly quick, they remain jerk free. The crawl acceleration is also non-‘
Prompt	'Tata Altroz driving experience'
Text Generated	‘Tata Altroz driving experience.Tata Altroz The Ultimate Comes With Great Features.Tata Altroz has become a renowned and popular premium hatchback with a price tag of INR 653000 - INR 991000 ex-showroom. It has a sleek and aerodynamic exterior with sharp lines, a bold grille, and elegant taillights. The

	Altroz borrows Tata's Impact philosophy, which emphasizes efficiency and fun-to-drive rather than competition.Unleashing Dynamic Performance In A Compact Package.The Tata Altroz is a dynamic performer. With its spirited engine and precise handling, it effortlessly outshines the competition. The Altroz delivers a balance between elegance and performance, making it an ideal choice for those seeking a spirited driving experience.Best Services.The Tata Altroz offers a well-designed and stable interior. It provides comfortable seating for five passengers with ample legroom and headroom, especially in both the front’
Prompt	'MG Comet EV First Drive Review'
Text generated	‘MG Comet EV First Drive Review.MG Comet The Technical Details.It is based on the pure electric GSEV platform and houses a 17.3kWh Li-ion battery pack in the middle, under the front seats. The Comet EV has a rear-mounted electric motor (yes, it’s RWD!) rated at 42PS of power and 110Nm. Unlike most other EVs in the market, there is only one battery and powertrain on offer. The Comet EV claims a range of up to 230km but based on our experiences with EVs so far, we'd subtract that by around 35 per cent and you can expect 150km in the real world with

	ease. It comes with three drive modes - Eco, Normal and Sport; and three regenerationlevels - Light, Normal and Heavy.The Comet EV is also offered with a range of up to 230km but based on our experiences with EVs so far, we'd subtract that by around 35 per cent and you can expect’
Prompt	‘Maruti Suzuki Ignis Interior’
Text Generated	‘Maruti Suzuki Ignis Interior.The interior of the Maruti Suzuki Ignis is quite enjoyable to drive. Due to the wide opening and flat floor, there is a good amount of legroom and headroom for both front and rear passengers. The interior really feels like a city car and is fairly spacious. The dashboard though is quite spacious due to the plastic quality used.Maruti Suzuki Ignis Verdict.The Maruti Suzuki Ignis petrol AMT ticks the boxes for those who want a car that's fresh and unique as well as packs in the convenience of an automated transmission. At INR 759000 (on-road Mumbai), the Ignis AMT is decently priced, with a claimed fuel efficiency of 19.8kmpl. At INR 919000 (on-road Mumbai), it's more than enough to justify the premium hatchback’s premium interiors. But, as convincing as the Ignis is

	economically, it's more than adequate for those’
Prompt	range of Comet EV
Text Generated	range of Comet EV makes it feel quite confident.Nice Car.It is a nice car with the best comfort, design, safety, and stylish look. The best thing is its battery size is also small so it doesn't take up most of the space in the car. The performance is great and it is suitable for city driving as well.Comfortful Car.It is a comfortable car with spacious interior and good legroom for both front and rear passengers. The best thing is the battery's capacity which is about 20 BHP. The pickup is also good, which is a boon in the urban environment.Handling Is Good.It is a very good car with comfortable and stylish design and a powerful engine. The gear shifting is smooth and the steering movement is also effortless. Overall it is a very good car. AC is very good in this car.Best Car.It is a very good car in terms of AC and pickup, styling and drive quality. Best features with good mileage
Prompt	expert reivew of PMV
Text Generated	expert reivew of PMV-E is great. The car has a great design, loaded with features, and appealed to the masses. However, the OVERALL EXPERIENCE is the opposite. When it comes to appreciating the Mini’s design, individuality, and rabid driveability, you really

	need to become a fan.This Mini Cooper SE is fresh, modern, stylish, and non-conventional. Is it really that special a Mini car and how much of that translates into everyday usability for you? We got a short spin in Jaisalmer to find out just that.Mini Cooper Design:Mini Cooper Theories:Mini Cooper was first launched in India in 2019 as a sub 4-metre hatchback that was claimed to be propelled by an SUV body style. The car was later christened the PMV EaS-E, and now known for its eco-friendly design language. In its time with the small segment, the range has seen it
Prompt	Interior review of Mercedes AMG
Text Generated	Interior review of Mercedes AMG A45S 4MATIC Plus.A lot has been said about the AMG A 45S being a subpar electric car and false. However, there is one thing which we like and that's important - it's the cabin. The A 45 S is a very practical cabin to drive around in and around town. Withcester ground clearance, a couple of stories and a couple of test drives gave it a serious thought. So we have come to the conclusion that a good amount of space in this car keyboards and other basic bits are essential to a good city runabout and that's exactly what we will be driving here.As we said, the cabin is quite practical with over ten storage and a decent amount of gadgets. New seat upholstery and old-fashioned seat upholstery add a sense of serenity to the cabin. It's also

	a good set of storage space for your knick knacks
Prompt	Exterior review of Toyota Glanza
Text Generated	Exterior review of Toyota Glanza,Things have been upgraded under the hood and for good mind you! The 2022 Toyota Glanza comes with a number of segment-first features, including a head-up display, cruise control, automatic headlights, rear AC vents, 6 airbags, a colour MID between the dials and a Harman developed eight speaker (pure) sound system which makes for a good studio experience.A Feature-Rich Package.The cabin of the 2022 Toyota Glanza is packed with features to the brim; including a touchscreen infotainment system with Android Auto and Apple CarPlay, eight-speakers Harman sound system, digital instrument cluster, and premium upholstery. Apart from your usual staples, there’s also a host of features to grab your attention; including a 7-inch infotainment touchscreen infotainment system, Android Auto and Apple CarPlay, steering-mounted controls, automatic climate control, keyless entry, rear AC vents, and key
Prompt	What's missing from the MG Comet EV?
Text Generated	What's missing from the MG Comet EV?'We expect the fully loaded variant that we drove

	to cost around INR 1100000, which is a pretty penny for something with just around 150km of real-world range and no boot.The Comet EV is not as practical as something like the Tiago EV, and caters to an entirely different buyer. This is nobody's only car and unlikely to be anyone's first family car. The Comet EV is a city runabout which can act as a green and compact alternative to your household's other premium car, likely an SUV.Is the cabin of the MG Comet EV any good?According to a report from MG, the cabin of the Comet EV is a combination of city and highway cruiser. There will be no DC fast charging or even AC, only a tiny 12V outlet. This will be a microsimulator and not a traditional electric display. What will be available is a 10.25 kW AC charger
Prompt	Hyundai Grand i10 Nios review
Text Generation	Hyundai Grand i10 Nios review.It's raining crossovers all over CarWale's HQ. Having big farewell to our long term Kia Sonet after six months and testing over a dozen different new models during the same period, here we have our new long term crossover and it's come from Hyundai. I have actually long mused over the Nios's value proposition and now that I am about to spend four months with it, I am looking forward to assessing its

	positives and negatives in the 'real world'.So why would you pick this car over the other cars in this price range? As it's a brand that has cut a lot of corners and is looking for customers who want a compact sub-four metre SUV that is easy and practical on the road. As for the design, it's quite obvious that Hyundai has cut corners and that's what makes this car stand out in this

4.4.2 – Results of Falcon 7B

The training was peformed for 5 epochs with 800 steps max. The training took 8-10 minutes and the performance the model has shown is phenomenal. There are certain issues with the length f the sentences generated therefore the output were trimmed to fit in the conversation window. Over a period of time with more question and answer data this model can perform better.

A conversation based on Question and Answer was performed on the fine tuned Falcon 7B. The results are as follows:

```
Chatbot: Hello! How can I assist you?
You: What are the petrol variants of Hyundai i20?
Chatbot: : Hyundai i20 is available in 11 Petrol variants, out of which 3
are automatic and 8 are manual
You: What are the specs of Hyundai i20?
Chatbot: : Hyundai i20 is currently available in Petrol engine
You: How many gears available in Hyundai i20?
Chatbot: : There are 5 Speed gears available in Hyundai i20
You: What is the top speed of Hyundai i20 N Line?
Chatbot: : Hyundai i20 N Line has a top speed of 150 km/h
You:  What type of airbags available in Hyundai i20 N Line?
Chatbot: : There are Driver and Passenger airbags available in the top
model of Hyundai i20 N Line
You: What is the kerb weight of Hyundai i20?
Chatbot: : The kerb weight of Hyundai i20 is 1170 Kg
You: What are the top competitors of Mahindra KUV 100 NXT?
Chatbot: : The top competitors of Mahindra KUV 100 NXT are Maruti Suzuki
Alto K10,Maruti Suzuki Wagon R,Maruti Suzuki Swift,Hyundai Exter
You: What are the petrol variants of Mahindra KUV 100 NXT?
Chatbot: : Mahindra KUV 100 NXT is available in 5 Petrol variants, out of
which 2 are automatic and 3 are manual
You: What is the width of Maruti Suzuki Alto K10?
```

Chatbot: : The width of Maruti Alto K10 is 1495 mm
You: What is the mileage of Maruti Suzuki Alto K10 Petrol Version?
Chatbot: : The mileage of Maruti Alto K10 Petrol is 24
You: What is the mileage of Maruti Suzuki Alto K10 Diesel Version?
Chatbot: : Currently there is no diesel engine option available on Maruti Alto K10
You: What type of airbags available in Maruti Suzuki Alto K10?
Chatbot: : There are Driver and Passenger airbags available in the top model of Maruti Alto K10
You: What are the automatic variants of Maruti Suzuki Ignis?
Chatbot: : Maruti Ignis is available in 1 Automatic variant with the price of Rs
You: What is the engine power of Maruti Suzuki Baleno petrol version?
Chatbot: : The engine power of Maruti Suzuki Baleno petrol version is 88
You: How many cylinders does Maruti Suzuki Baleno has?
Chatbot: : Maruti Suzuki Baleno comes with 4 cylinders
You: What is the ground clearance of Maruti Suzuki Baleno?
Chatbot: : The ground clearance of Maruti Suzuki Baleno is 167mm
You: What is the length of Maruti Suzuki Baleno?
Chatbot: : The length of Maruti Suzuki Baleno is 3990 mm
You: What is the tyre size of Maruti Suzuki Celerio?
Chatbot: : The tyre size of Maruti Celerio is 165/70 R14
You: What is the mileage of Maruti Suzuki Celerio?
Chatbot: : The mileage of Maruti Celerio ranges from 24
You: What are the specs and features of Maruti Suzuki Celerio?
Chatbot: : Maruti Suzuki Celerio is currently available in Petrol and CNG engines
You: What are the top competitors of Maruti Suzuki Celerio?
Chatbot: : The top competitors of Maruti Suzuki Celerio are Maruti Ignis,Hyundai Exter,Maruti S-Presso,Maruti Suzuki Wagon R
You: What type of airbags available in Maruti Suzuki S-Presso?
Chatbot: : There are Driver and Passenger airbags available in the top model of Maruti S-Presso
You: What is the width of MG Comet EV?
Chatbot: : The width of MG Comet EV is 1470 mm
You: What is the wheel size of MG Comet EV?
Chatbot: : The wheels size of MG Comet EV is 10
You: What is the top speed of MG Comet EV?
Chatbot: : The top speed of MG Comet EV is 120 km/h
You: Is MINI Cooper 3 DOOR available in automatic transmission?
Chatbot: : Yes
You: What is the ground clearance of Nissan Magnite?
Chatbot: : The ground clearance (Unladen) of Nissan Magnite is 205 in mm, 20
You: hat is the length of Nissan Magnite?
Chatbot: : The length of Nissan Magnite is 3995 mm
You: What is the mileage of Nissan Magnite Diesel Version?
Chatbot: : Currently there is no diesel engine option available on Nissan Magnite
You: What type of airbags available in Nissan Magnite?
Chatbot: : There are Driver and Passenger airbags available in the top model of Nissan Magnite
You: What are the automatic variants of Renault KWID?

```
Chatbot: : Renault KWID is available in 1 Automatic variant with the price
of Rs
You: What is the price of Renault KWID?
Chatbot: : Renault KWID price in Delhi starts at Rs
You: Is Strom Motors R3 available in automatic transmission?
Chatbot: : Yes
You: What type of airbags available in Tata Tiago?
Chatbot: :  There are Driver and Passenger airbags available in the top
model of Tata Tiago
You: quit
Chatbot: Goodbye!
```

The BLEU score generated for the Falcon 7B is 0.29183500067308654. This is due to the fact that the sentences were getting created more than tha maxmum length speficified hence the sentences had to be trimmed to make the user experience more acceptable. However, the quality of sentences generated were impressive considering the training was conducted on `4718592` parameters out a total of `3613463424` which is `13%` of the total parameters.

4.5. Model evaluation

There is no existing state of the art models which are dedicated to the automobile industry. Most of the exiting 3rd party services have bots which do not provide this kind of information to the consumer. Instead, they focus on connecting the consumer to the buyer or seller or an agent who would help in buying or selling their vehicle. Hence, the evaluation on other models have been resorted to the ChatGPT. ChatGPT is not entirely dedicated to a conversational task for the automobile industry, yet it can be used to pass some contexts and retrieve the relevant information. Some data from the previous list of prompts on specification are selected which are before 2021.

Table 4.5: ChatGPT response vs GPT-2(our model) response

ChatGPT	GPT-2 (Our model)
The engine displacement of the Tata Tiago XZ Plus CNG is 1199 cc.	The engine displacement of Tata Tiago XZ Plus CNG is 1197 cc.
The fuel tank capacity of the Tata Tiago XE CNG is 35 liters.	The fuel tank capacity of Tata Tiago XE CNG is 37litres.

The max torque of the Tata Tiago XZA Plus Dual Tone Roof AMT is 113Nm@3500-5500rpm.	The max torque of Tata Tiago XZA Plus Dual Tone Roof AMT is 115Nm@3250+/-100rpm.
The seating capacity of the Maruti Suzuki Alto K10 VXI Plus AT is 4 people.	The seating capacity of Maruti Suzuki Alto K10 VXI Plus AT is 5 people.
The fuel efficiency of the Maruti Suzuki Alto K10 VXI S-CNG is 33.85 km/kg.	The fuel efficiency of Maruti Suzuki Alto K10 VXI S-CNG is 26.8 km/kg.
The ex-showroom price of the Hyundai i20 N Line N8 iMT Dual tone is INR 1136000.	The ex-showroom price of Hyundai i20 N Line N8 iMT Dual tone is INR 775000 only.
The boot space of the Hyundai i20 N Line N8 DCT is 300L (Litres).	The boot space of Hyundai i20 N Line N8 DCT is 268L (Litres).

The above information furnished by ChatGPT is mostly correct in terms of both context as well as sentence generation. Our model too generates good quality text however the authenticity of the information cannot be guaranteed.

5 Discussions

There were 2 types of results that were generated by the pre-trained GPT-2 on the new dataset created on cars. The analysis of the results can be done in 2 phases. There were approximately 30 sentences used to analyze the results. One of the objectives is to create meaningful sentences. This has been achieved and that can be established by the following sentences listed below:

Table 4.5: Sample Generated Text

Generated Text
The seating capacity of Maruti Suzuki Ignis Sigma is 5 people.
The fuel tank capacity of Maruti Suzuki Ignis Zeta AMT is 37litres.
The max torque of Maruti Suzuki Ignis Alpha AMT is 113Nm@4400rpm.
The transmission of Maruti Suzuki Ignis Alpha AMT is Automatic.
The ex-showroom price of Maruti Suzuki Ignis Zeta AMT is INR 693000.
The engine displacement of Tata Tiago XT Rhythm is 1199 cc.
The fuel efficiency of Maruti Suzuki Ignis Zeta AMT is 18.8 kmpl.
The fuel tank capacity of Tata Tiago XE CNG is 37litres.
The colors available for Tata Altroz XM Plus S are solid fire red, pearl arctic white, solid fire red with pearl midnight black, pearl metallic lucent orange, metallic silky silver, pearl midnight black, pearl metallic midnight blue.
The fuel efficiency of Maruti Suzuki Alto K10 VXI S-CNG is 26.8 km/kg.

The quality of generated texts has shown coherence. The important aspect is the numerical values being generated to match the context. The seating capacity should be numerical and that is outlined correctly. The transmission can be manual and automatic and since the name of the car has AMT at the end hence the model has predicted the next word correctly a 'Automatic'. The ex-showroom price, engine displacement, fuel efficiency and fuel tank capacity should all have numeric values as answer to the prompt. The model correctly identifies the same. The max torque should be represented as 'Nm' as well as with 'rpm'. The fuel efficiency should be kmpl for all the petrol engine cars and that has been identified correctly. For a CNG engine it should be represented as km/kg. This is properly shown in Maruti Suzuki Alto K10 VXI S-CNG. These showcase that the text generated are according to the context.

There were large texts generated by the model as well. The large texts give a more detailed analysis. An example of the large text is present below:

Table 4.6: Sample Generated long text.

Prompt	'Maruti Suzuki Ignis Interior'
Text Generated	'Maruti Suzuki Ignis Interior.The interior of the Maruti Suzuki Ignis is quite enjoyable to drive. Due to the wide opening and flat floor, there is a good amount of legroom and headroom for both front and rear passengers. The interior really feels like a city car and is fairly spacious. The dashboard though is quite spacious due to the plastic quality used.Maruti Suzuki Ignis Verdict.The Maruti Suzuki Ignis petrol AMT ticks the boxes for those who want a car that's fresh and unique as well as packs in the convenience of an automated transmission. At INR 759000 (on-road Mumbai), the Ignis AMT is decently priced, with a claimed fuel efficiency of 19.8kmpl. At INR 919000 (on-road Mumbai), it's more than enough to justify the premium hatchback's premium interiors. But, as convincing as the Ignis is economically, it's more than adequate for those'

The above example outlines the interior review of the Maruti Suzuki Ignis. The text generated us aligned to the context where it explains the space, dashboard and the legroom which are essentially interior features of the car. The inclusion of the ex-showroom price and the fuel efficiency are unnecessary details which have been included. Apart from this the text is more aligned to the context.

The model generated coherent contexts and, in some cases, generated texts which had issues as well. Some of the texts are as follows:

Table 4.7: Sample generated text with issues.

The fuel tank capacity of Maruti Suzuki Ignis Zeta AMT is 37litres.
The AMG A 45 S 4MATIC Plus has a toque that is 22.35 kmpl.
The AMG A 45 S 4MATIC Plus uses petrol as fuel.

The above texts are related to the specification of the cars. Now, a general observation is that the model tends to put 37 litres as fuel tank capacity irrespective of any variant of the car. The sentence, although correct, provides the wrong information to the consumer. The other issue is the model could not predict torque correctly. It predicts the torque to be of 22.35 kmpl which is incorrect both as a unit of measurement as well as in terms of the ground truth. This may be because the spelling of the word torque is incorrect, and it is expected for the model to still come up with the word correctly. This is because the user may type words incorrectly while conversing with the model. In another instance the seating capacity was asked, and the model replied with the fuel used. This is also an example of both incorrect context as well as far from the ground truth. There were issues observed in the generated large text as well. An example is given below:

Table 4.8: Sample generated long text with issues.

Prompt	Exterior review of Toyota Glanza
Text Generated	Exterior review of Toyota Glanza,Things have been upgraded under the hood and for good mind you! The 2022 Toyota Glanza comes with a number of segment-first features,

	including a head-up display, cruise control, automatic headlights, rear AC vents, 6 airbags, a colour MID between the dials and a Harman developed eight speaker (pure) sound system which makes for a good studio experience.A Feature-Rich Package.The cabin of the 2022 Toyota Glanza is packed with features to the brim; including a touchscreen infotainment system with Android Auto and Apple CarPlay, eight-speakers Harman sound system, digital instrument cluster, and premium upholstery. Apart from your usual staples, there’s also a host of features to grab your attention; including a 7-inch infotainment touchscreen infotainment system, Android Auto and Apple CarPlay, steering-mounted controls, automatic climate control, keyless entry, rear AC vents, and key

In the above example the sentences created are coherent, but the text generated is far from the context. The model here generates output without adhering to the functional requirement of ‘Exterior’ review. Rather it creates a paragraph outlining the interior review of the car Toyota Glanza. Another example is shown below:

Table 4.9: Sample generated long text with issues.

Prompt	expert reivew of PMV
Text Generated	expert reivew of PMV-E is great. The car has a great design, loaded with features, and appealed to the masses. However, the OVERALL EXPERIENCE is the opposite. When it comes to appreciating the Mini’s design,

	individuality, and rabid driveability, you really need to become a fan.This Mini Cooper SE is fresh, modern, stylish, and non-conventional. Is it really that special a Mini car and how much of that translates into everyday usability for you? We got a short spin in Jaisalmer to find out just that.Mini Cooper Design:Mini Cooper Theories:Mini Cooper was first launched in India in 2019 as a sub 4-metre hatchback that was claimed to be propelled by an SUV body style. The car was later christened the PMV EaS-E, and now known for its eco-friendly design language. In its time with the small segment, the range has seen it

In the above review the prompt was to give an expert review of the PMV. The model starts the answering well however it starts describing the Mini Cooper in the middle and then ends with PMV Eas-E. This is a consistency problem. In this section examples were put forward from the tests conducted. As per the details explained above the model adheres to coherent and meaningful text generation in most cases. The important aspect of putting the right numerical where it is appropriate is correct almost 90% of the time. It also learns and understands the context correctly for non-numerical values. The texts pertaining to features like interior. Exterior, pros and cons demand a more comprehensive answer and they have been outlined correctly. However, there are a few drawbacks to the generated output, one of which is consistency as in some cases it has been observed that the model loses context. There is also a problem of consistency in which the model generates texts which are not as per to the contexts. The responses generated by ChatGPT vs Our model showcases that even though the pre-trained GPT-2 is old compared to ChatGPT (GPT-3.5) it still can generate results which are coherent in nature. This information our model generates is not correct functionally but the quality of sentence with respect to the context generated is good. The Falcon 7B model fine-tuned using Qlora on a small dataset of 650+ QnA has proved to be extremely successful. There are a few drawbacks which have been observed within the short span of time that it has been made available for commercial use. This drawback is primarily targeted towards the length of the generated sequences. The model creates meaningful and coherent answers but tends to over-shoot the maximum defined length. To make it more coherent a filter has been set to only send back the first sentence. A 30+ QnA conversation was carried out and even after filtering the sentences the text produced can be declared as good quality. The Falcon 40B and 7B has been made available most recently therefore a lot of research in terms of how it can be used have not been done. However, with limited knowledge and resources the performance of the model is quite impressive.

6. Conclusion and Future work

The creation of a unique dataset containing all details of a hatchback was successfully created. A total of 23 text files were created which contains the details of all the hatchbacks which are present in the Indian automobile market. These chunks of data are then collated, and a pre-trained GPT-2 language model was finetuned with that data. The text generated achieved a BLEU score of 0.779. The text passed on to the reader as information in almost 80% of cases was coherent and were contextually correct. However, there were issues observed in the authenticity of the data and on some occasions the consistency of the texts generated. The project named "GenarativeAutoAI" has been thought off as a one stop solution for all the queries related to a car. Initially it has been developed for a particular segment, but the future work would include more segments like Sedans, Convertibles, SUV, MUV, Pickup trucks as well as super and hyper cars. Data plays a key role in this project. The data created currently needs to be scaled up to fulfill such demands. The format of the data is also crucial. In this project due to time constraints the data is used as a chunk of text. For future work the data format needs to have more variations like Question and Answering, Conversational. This variety of data will help in making a more robust model. Another important factor is the choice of model. Due to time constraint and infrastructure issues a pre-trained GPT-2 model had to be chosen. However, many new models have now been included in the Hugging Face library such as Falcon 40B and LLAMA. These are large language models which have over 40 billion parameters and they are capable of capturing minute details in terms of the relationships of the words and contexts. A twin training regime can be approached which is like the one done for training ChatGPT. The first training will be done on the raw chunk of text. This captures the information present in the data and in this case the cars. This is performed in this project. The second set of training, or in this case fine tuning, should be done on specific downstream tasks like question and answering, conversational. The authenticity of the data plays the most important role in this project. The user should not only get a response from the model, but that response must be correct. This can be achieved by Reinforcement Learning from Human Feedback. This is the concept behind the accuracy of popular generative models like ChatGPT. The blogpost "Illustrating

Reinforcement Learning from Human Feedback (RLHF)" (Lambert et al., 2022) provides a deeper insight into the world of RLHF. In RLHF, is a technique which trains a reward model based on the human feedback. This technique uses the model as a reward function which optimizes the agent's policy using reinforcement learning using the Proximal Policy Optimization. The policy is optimized to predict is the output is good or bad. A good output is accompanied by a high reward and a bad with a low reward. The idea is to have a commercially viable generative model hence these above-mentioned steps are a few ones which need to be done as the next step on this existing model.

References

GoMechanic. (2020, December 24). Why Are Hatchbacks So Popular in India? Mystery Solved. *Why Are Hatchbacks So Popular in India? Mystery Solved*. https://gomechanic.in/blog/hatchbacks-popular-in-india/

Wikipedia contributors. (2023, August 14). Automotive industry in India. *In Wikipedia.* https://en.wikipedia.org/w/index.php?title=Automotive_industry_in_India&oldid=1170253620

Natural Language Processing. (2023, Jan 11). In DeepLearing.AI https://www.deeplearning.ai/resources/natural-language-processing/

Torrey, L.A., & Shavlik, J.W. (2009). *Chapter 11 Transfer Learning.*

Dilmegani, C. (2023, January 20). *Transfer Learning in 2023: What It Is & How It Works.* AI Multiple. https://research.aimultiple.com/transfer-learning/

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., ... & He, Q. (2020). A comprehensive survey on transfer learning. *Proceedings of the IEEE*, *109*(1), 43-76. https://doi:10.1109/JPROC.2020.3004555.

Wikipedia contributors. (2023, July 14). Hugging Face. In Wikipedia. https://en.wikipedia.org/w/index.php?title=Hugging_Face&oldid=1165377079

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, *27*. https://doi.org/10.48550/arXiv.1409.3215

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. Attention is all you need. 2017. doi: 10.48550. *arXiv preprint ARXIV.1706.03762*.

Li, J., Monroe, W., Ritter, A., Galley, M., Gao, J., & Jurafsky, D. (2016). Deep reinforcement learning for dialogue generation. *arXiv preprint arXiv:1606.01541*. https://doi.org/10.48550/arXiv.1606.01541

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, *8*, 229-256. https://doi.org/10.1007/BF00992696

Sankar, C., & Ravi, S. (2019). Deep reinforcement learning for modeling chit-chat dialog with discrete attributes. *arXiv preprint arXiv:1907.02848*. https://doi.org/10.48550/arXiv.1907.02848

Serban, I. V., Lowe, R., Charlin, L., & Pineau, J. (2016). Generative deep neural networks for dialogue: A short review. *arXiv preprint arXiv:1611.06216*. https://doi.org/10.48550/arXiv.1611.06216

Saleh, A., Jaques, N., Ghandeharioun, A., Shen, J., & Picard, R. (2020, April). Hierarchical reinforcement learning for open-domain dialog. In Proceedings of the AAAI conference on artificial intelligence (Vol. 34, No. 05, pp. 8741-8748). https://doi.org/10.1609/aaai.v34i05.6400

Alammar, J. (n.d.). The Illustrated GPT-2 (Visualizing Transformer Language Models). https://jalammar.github.io/illustrated-gpt2/#part-1-got-and-language-modeling

Singh, A. (2019, May 28). Why do Transformers yield Superior Sequence to Sequence (Seq2Seq) Results? Saarthi AI. https://medium.com/saarthi-ai/transformers-attention-based-seq2seq-machine-translation-a28940aaa4fe

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*. https://doi.org/10.48550/arXiv.1810.04805

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., ... & Dean, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*. https://doi.org/10.48550/arXiv.1609.08144

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. R. (2018). GLUE: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*. https://doi.org/10.48550/arXiv.1804.07461

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI blog, 1(8), 9.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., & Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. Advances in neural information processing systems, 32.

Lipenkova, J. (2022, September 26). Choosing the right language model for your NLP use case. https://towardsdatascience.com/choosing-the-right-language-model-for-your-nlp-use-case-1288ef3c4929

Lester, B., Al-Rfou, R., & Constant, N. (2021). The power of scale for parameter-efficient prompt tuning. arXiv preprint arXiv:2104.08691.

von Werra, L., Belkada, Y., Mangrulkar, S., Tunstall, L., Dehaene, O., Cuenca, P., Schmid, P., & Sanseviero, O. (2023, June 5). Falcon: Hugging Face's Next-Generation Training Pipeline for Transformers. Hugging Face Blog. https://huggingface.co/blog/falcon

Belkada, Y., Dettmers, T., Pagnoni, A., Gugger, S., & Mangrulkar, S. (2023, May 24). Making LLMs even more accessible with bitsandbytes, 4-bit quantization and QLoRA. Hugging Face. https://huggingface.co/blog/4bit-transformers-bitsandbytes

List of Attachments

- Hatchback-Cars-Input.txt – Input file containing all details of cars.
- QnA-Cars-File.json – Input file for the Falcon Model
- GPT-2 model - GenerativeCarAI-GPT2.ipynb
- Falcon model - GenerativeCarAI-Falcon.ipynb