RESEARCH REPORT
PREDII
CUSTOM ENTITY RECOGNITION

INTRODUCTION

In the automotive data analytics realm, extracting insights from vast data sets is crucial. With constant industry evolution driven by technology and consumer trends, precise data processing is essential. Custom entity recognition emerges as a key tool, enabling the identification and classification of specific entities within textual data.

This research project focuses on developing and evaluating a custom entity recognition model tailored for the automotive domain. Using advanced natural language processing techniques, the model aims to accurately identify vehicle models, features, components, and automotive terminology within textual data.

Accurate entity recognition is vital for downstream tasks like sentiment analysis, named entity recognition, and information extraction, providing deeper insights into consumer preferences, market trends, and product feedback.

By proficiently identifying and categorising automotive entities, this research aims to enable better data analysis, decision-making, and innovation within the automotive landscape.

Literature Review

- **SPACY**:spaCy is a popular open-source library for Natural Language Processing (NLP) in Python. It offers efficient tokenization, parsing, named entity recognition (NER), and various other linguistic features. Recent literature has praised spaCy for its speed, accuracy, and user-friendly interface. For instance, studies by Honnibal and Montani (2017) highlighted spaCy's robustness in NER tasks and its ease of integration into NLP pipelines. Additionally, Bhatt and Jain (2020) emphasized spaCy's role in text classification and sentiment analysis due to its advanced tokenization capabilities.

- **Neural Networks:** Neural networks have become the backbone of modern NLP systems due to their ability to learn complex patterns from data. Recent literature has focused on enhancing neural network architectures for tasks such as text classification, machine translation, and sentiment analysis. For example, Vaswani et al. (2017) introduced the Transformer architecture, which revolutionised NLP by enabling parallel processing of input sequences. Moreover, studies by Devlin et al. (2018) and Radford et al. (2018) demonstrated the effectiveness of large-scale pre-trained language models, such as BERT and GPT, in various NLP tasks.
- **NLP:** Natural Language Processing (NLP) research has seen significant advancements in recent years, driven by the availability of large-scale datasets and powerful computing resources. Researchers have explored various aspects of NLP, including syntax parsing, semantic understanding, and discourse analysis. For instance, Manning et al. (2014) provided a comprehensive overview of NLP techniques and applications, covering topics such as part-of-speech tagging, parsing, and machine translation. Additionally, Goldberg (2016) discussed the principles of deep learning for NLP, emphasising the importance of neural network architectures and optimization techniques.
- **Transformers:** Transformers have emerged as a dominant architecture in NLP, surpassing traditional recurrent and convolutional models in performance. Recent literature has extensively explored transformer-based models for tasks such as text generation, question answering, and language understanding. For example, studies by Brown et al. (2020) introduced GPT-3, a massively scaled transformer model capable of generating coherent and contextually relevant text. Furthermore, Lewis et al. (2020) proposed the T5

model, which achieved state-of-the-art performance across a wide range of NLP tasks by fine-tuning a single transformer architecture.

- **Qlora**: QLoRA tackles the challenge of fine-tuning massive Large Language Models by shrinking them down. It achieves this through a combination of parameter quantization (reducing precision) and Low-Rank Adapters (focusing on a smaller set of parameters for adaptation), enabling efficient fine-tuning on a single GPU with minimal performance loss.

Methodology

❖ PREPROCESSING OF DATA

USED IN SPACY MODEL

1. Dataset Description:

- Source: The dataset used for training and evaluation was custom-built using a NER (Named Entity Recognition) annotator tool.
- Size: Provide information on the size of the dataset in terms of the number of documents, sentences, and tokens/entities.
- Characteristics: Describe the nature of the data, including the types of entities annotated, the distribution of entity types, and any challenges or biases present in the data.

2. Preprocessing Steps:

- Tokenization: Utilised a tokenizer to split the text into individual tokens, ensuring that each word or subword is represented as a separate token.
- BILOU Tagging: Implemented BILOU tagging scheme for multi-word entity

```
BILOU Method/Schema

| ------|---------------------|
| BEGIN | The first token     |
| ------|---------------------|
| IN    | An inner token      |
| ------|---------------------|
| LAST  | The final token     |
| ------|---------------------|
| Unit  | A single-token      |
| ------|---------------------|
| Out   | A non-entity token  |
| ------|---------------------|
```

BILOU TAGGING REPRESENTATION

USED IN LLM MODEL

1. Dataset Description:

- Source: The dataset used for training the Large Language Model (LLM) was provided in JSON format, containing text samples (instances) along with annotated entities (answers).
- Size: Detail the size of the dataset, including the number of instances and the distribution of entity types.
- Characteristics: Describe any notable characteristics of the dataset, such as the variety of entity types, potential biases, or imbalances in the annotations.

2. Preprocessing Steps:

- Tokenization: Employed a tokenizer to segment the text into individual tokens or subwords, ensuring compatibility with the LLM's input format.
- Formatting: Prepared the dataset to adhere to the required input format of the LLM, which typically involves encoding the text and entity annotations in a specific format suitable for model training.
- Special Tokens: Introduced special tokens, such as [CLS] and [SEP], to denote the beginning and end of each instance, respectively, as per the requirements of the LLM architecture.

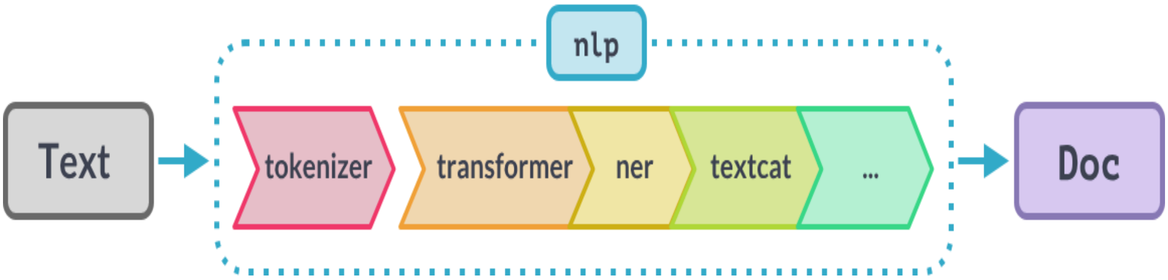
❖ MODEL ARCHITECTURE

SPACY MODEL

Architecture:

- Transformer-Based Architecture: The spaCy-Transformers model utilizes transformer-based architectures such as BERT (Bidirectional Encoder Representations from Transformers) or RoBERTa (Robustly optimized BERT approach) for named entity recognition (NER) on your automotive dataset.
- Tokenization: Input text from your automotive dataset is tokenized into subwords or word pieces using a pre-trained tokenizer specific to the transformer architecture, ensuring compatibility with the underlying model.

- Pre-Trained Transformer Layers: The model leverages pre-trained transformer layers to encode contextual information from the input tokens, capturing complex linguistic patterns and dependencies relevant to the automotive domain.
- Custom NER Layers: Additional custom layers may be added on top of the pre-trained transformer layers to adapt the model to the specific entities and nuances present in your automotive dataset, allowing the model to learn domain-specific entity recognition patterns.
- Output Layer: The final layer of the model predicts the most likely entity type for each token in the input text sequence from your automotive dataset, typically employing a softmax activation function to produce probability distributions over the set of entity types relevant to the automotive domain.



MODEL ARCHITECTURE SCHEMATIC DIAGRAM

LARGE LANGUAGE MODEL

Architecture:

While the exact details of Llama-7b's architecture haven't been publicly released by Meta.Here's a breakdown of the base architecture and how QLoRA integrates:

Base Architecture:

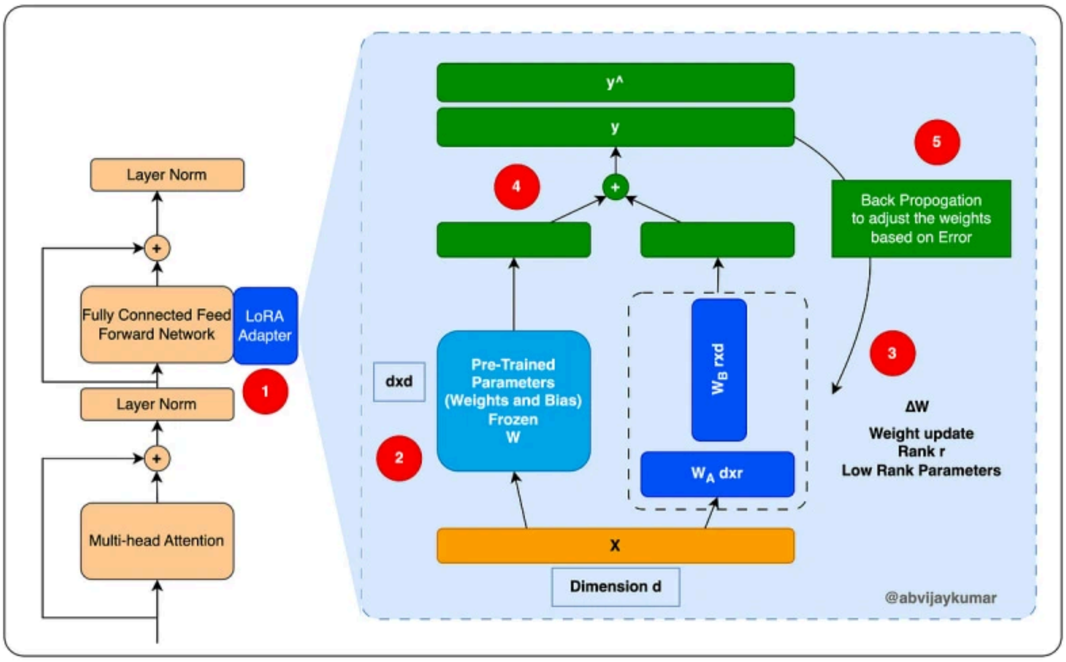
- **Encoder-Decoder Structure**: The model consists of two main parts:
    - Encoder: Processes the input text, capturing its meaning and relationships between words. It likely uses multiple encoder layers, each containing:
        - **Self-Attention layers**: Analyse relationships between words within the input sequence.
        - **Feed-forward layers**: Process the information further for better representation.

- **Decoder**: Generates the output text, one word at a time. It likely uses multiple decoder layers, each containing:
    - **Self-Attention layers**: Focus on the generated text so far.
    - **Attention layer**: Attends to the encoded representation from the encoder, incorporating context.
    - **Feed-forward layers**: Process the information for generating the next word.
- **Positional Encoding**: Since Transformers lack inherent understanding of word order, positional encoding is added to the input embedding, conveying the position of each word in the sequence.

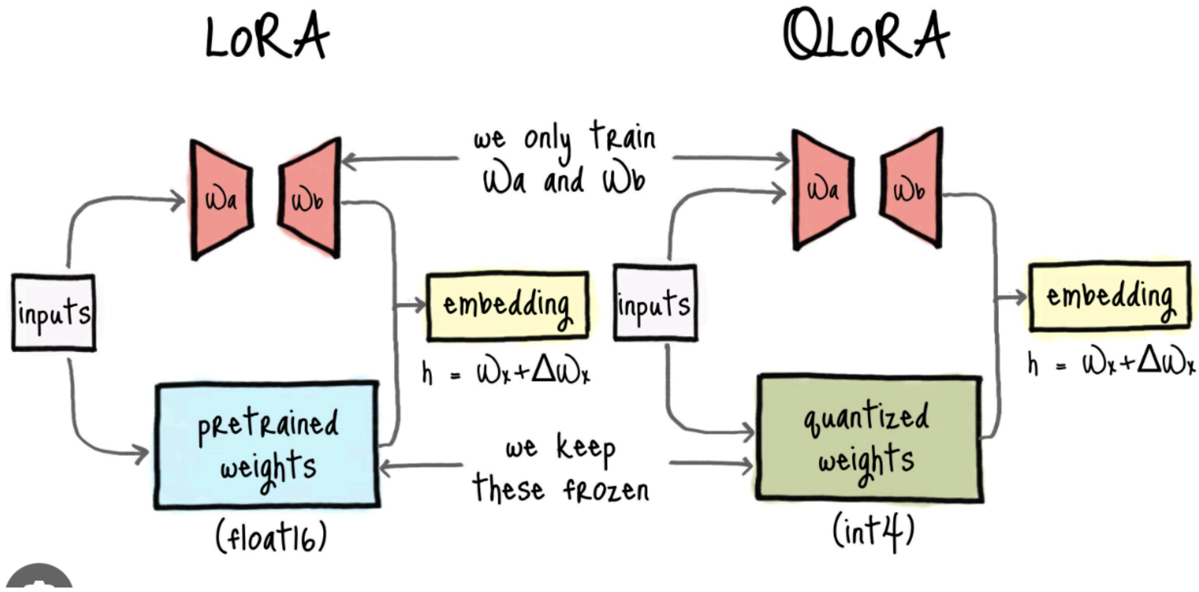
QLoRA Integration:

During fine-tuning with QLoRA:

- **Weight Quantization**: The model's weights (values in the encoder and decoder layers) are converted from high precision (e.g., 32-bit floats) to a lower precision format (e.g., 4-bit NF4) for memory efficiency.
- **LoRA Modules**: A small subset of parameters within the model are wrapped in LoRA modules. These modules learn to adapt the pre-trained Llama-7b to the specific fine-tuning task, reducing the amount of fine-tuning required for the entire model.
- Full fine-tuning is not possible here: we need parameter-efficient fine-tuning (PEFT) techniques like **LoRA or QLoRA**.
- To drastically reduce the VRAM usage, we must fine-tune the model in **4-bit precision**, which is why we'll use QLoRA.



QLORA TRAINING WITH HELP OF LoRA Adapter



QLoRa AND LoRa Configuration of in Large language model

TRAINING:

SPACY MODEL

Training and Fine-Tuning:

- **Transfer Learning:** The spaCy NER component supports transfer learning, allowing pre-trained models (e.g., on general domain data) to be fine-tuned on domain-specific datasets.
- **Custom Entity Types**: Users can define custom entity types relevant to their domain (e.g., automotive entities like CAR_MODEL, MANUFACTURER, PART), enabling the model to recognize and classify domain-specific entities accurately.
- **Data Augmentation:** Techniques such as data augmentation and synthetic data generation can be employed to enhance the model's performance, especially when training data is limited.

LLAMA MODEL TRAINING:

Fine-tuning with QLoRA:

- This stage assumes you have a pre-trained **Llama-7b model**.
- Here's how **QLoRA** is applied for fine-tuning:
- **Data Preparation:** Your specific dataset for the fine-tuning task is prepared and formatted for the model.

- **Quantization**: The pre-trained Llama-7b model's weights are converted from high precision (e.g., 32-bit) to a lower precision format (e.g., 4-bit NF4) using a specific quantization algorithm. This significantly reduces the model size.

- **LoRA Module Integration**: A small set of trainable parameters within the model are wrapped in LoRA modules. These modules become the focus for adaptation during fine-tuning.

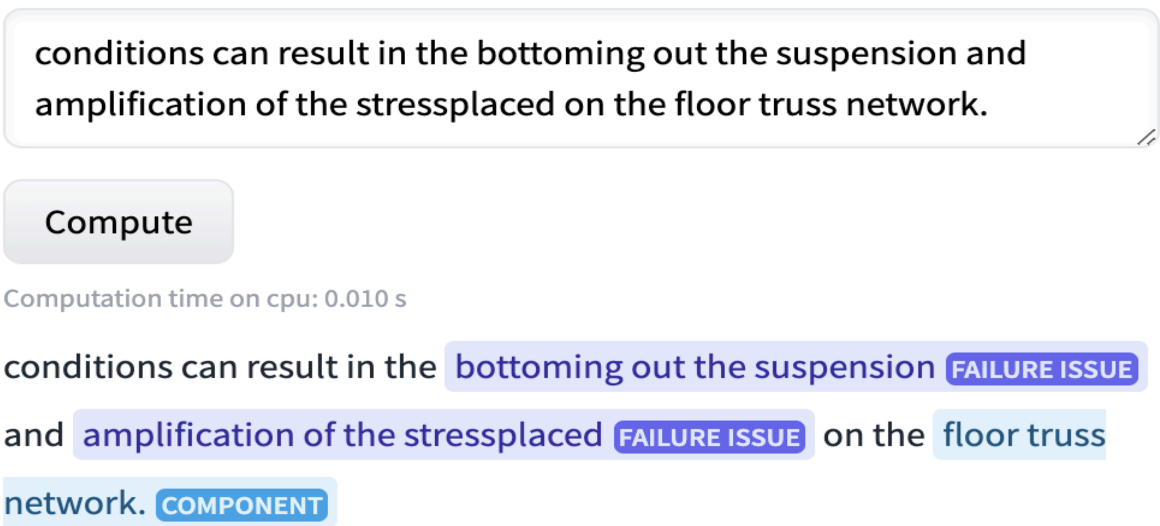
- **Fine-tuning**: The model is trained on your specific dataset. However, instead of fine-tuning all weights, only the LoRA modules and a small subset of the original weights are updated. This significantly reduces memory usage and training time compared to fine-tuning the entire model.

RESULTS AND EVALUATION

SPACY MODEL

SPACY MODEL HAS AN **ACCURACY OF 82%** of the test dataset

- SOME OF THE SNIPPETS OF MY INFERENCE MODEL OF MY SPACY MODEL
- CUSTOM NER MODEL HAS BEEN HOSTED ON HUGGING FACES https://huggingface.co/aryan10022001/en_predii_ner



the additional stress can result in the fracture of welds securing the floor truss network system to the chassis frame rail and/or fracture of the floor truss network support system.

Compute

Computation time on cpu: cached

the additional stress FAILURE ISSUE can result in the fracture of welds FAILURE ISSUE securing the floor truss network system COMPONENT to the chassis frame rail CHASSIS TYPE and/or fracture of the floor truss network support system. FAILURE ISSUE

the possibility exists that there could be damage to electrical wiring and/or fuel lines which could potentially lead to a fire.You could contact to the MONACO CORPORATION

Compute

Computation time on cpu: cached

the possibility exists that there could be damage to electrical wiring FAILURE ISSUE and/or fuel lines which could potentially lead to a fire. FAILURE ISSUE You could contact to the MONACO CORPORATION MANUFACTURER

THE MODEL INFERENCE LINK:

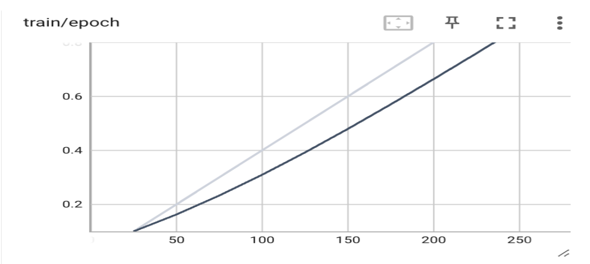
https://huggingface.co/spaces/aryan10022001/predii_ner

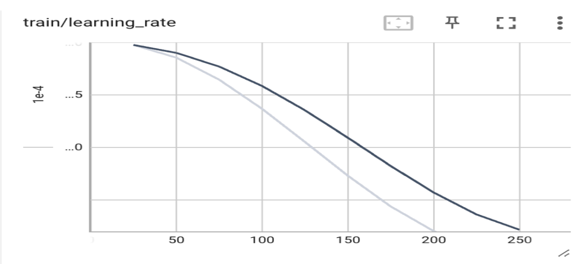
LLAMA MODEL RESULTS:

- Some of the snippets of the inference model
- These are the results and the prompt that i have given

```
<s>[INST]
You will act as a Named Entity Recognition Model which will be used to classify sub-segments of a given sentence into entities related to the automation field.

Following are the Named Entities which will be used as an entity for Named Entity Recognition and its description that will be used to classify the part of the sentence:-
"COMPONENT" : "Identifies a specific part or element within a system.",
"FAILURE_ISSUE" : "Describes a problem or malfunction encountered within the system.",
"VEHICLE_MODEL" : "Refers to the specific model or type of vehicle affected by the issue.",
"CORRECTIVE_ACTION" : "Specifies the steps or measures taken to resolve the problem or malfunction.",
"MANUFACTURER" : "Denotes the company or entity that produced the vehicle or component.",
"CHASSIS_TYPE" : "Specifies the type or configuration of the vehicle chassis involved.",
"PROCESS" : "Refers to a series of actions or steps followed to accomplish a task within the automation system.",
"PARTS" : "Identifies individual components or pieces used within the system."

Given the below sentance and the above mentioned instruction give the output.

Sentance = the additional stress can result in the fracture of welds securing the floor truss network system to the chassis frame rail and/or fracture of the floor truss network support system. [/INST]

Entities:

* COMPONENT: floor truss network system
* FAILURE_ISSUE: fracture
* VEHICLE_MODEL: [None]
* CORRECTIVE_ACTION: [None]
* MANUFACTURER: [None]
* CHASSIS_TYPE: [None]
* PROCESS: [None]
* PARTS: floor truss network support system

Explanation:
The given sentence highlights the failure of the floor truss network system, which is a component of the automation system. The failure is described as fracture, which is a specific type of failure issue
```

```
<s>[INST]
You will act as a Named Entity Recognition Model which will be used to classify sub-segments of a given sentence into entities related to the automation field.

Following are the Named Entities which will be used as an entity for Named Entity Recognition and its description that will be used to classify the part of the sentence:-
"COMPONENT" : "Identifies a specific part or element within a system.",
"FAILURE_ISSUE" : "Describes a problem or malfunction encountered within the system.",
"VEHICLE_MODEL" : "Refers to the specific model or type of vehicle affected by the issue.",
"CORRECTIVE_ACTION" : "Specifies the steps or measures taken to resolve the problem or malfunction.",
"MANUFACTURER" : "Denotes the company or entity that produced the vehicle or component.",
"CHASSIS_TYPE" : "Specifies the type or configuration of the vehicle chassis involved.",
"PROCESS" : "Refers to a series of actions or steps followed to accomplish a task within the automation system.",
"PARTS" : "Identifies individual components or pieces used within the system."
Given the below sentance and the above mentioned instruction give the output.

Sentance = conditions can result in the bottoming out the suspension and amplification of the stress placed on the floor truss network. the additional stress can result in the fracture of welds securing the floor truss network sys

Entities:

* COMPONENT: suspension
* FAILURE_ISSUE: bottoming out
* VEHICLE_MODEL: [None]
* CORRECTIVE_ACTION: [None]
* MANUFACTURER: [None]
* CHASSIS_TYPE: [None]
* PROCESS: amplification
* PARTS: floor truss network, welds, fuel lines

The output is based on the named entities provided, and the corresponding descriptions are used to classify the sub-segments of the sentence. In this case, the named entities are used to identify the components, failures, and syst
```





MODEL CAN BE ACCESS BY:

https://huggingface.co/aryan10022001/llama-7b-train_quantized

COMPARISON:

- **SPACY** model seems to be working much better and providing much better results because of the smaller size of spacy model,better BILOU tagging and Better encoding properties as it is based on BERT(BIDIRECTIONAL ENCODER REPRESENTATIONAL OF TRANSFORMERS)
- While llama-7B is very much capable of doing ner ,but since i have the constraints of GPU and and thus token size is very small for inference and hence the results are not that good
- Another Reason is that Due to constraints of GPU, i had to use quantization which may have affected the performance
- While the spacy model is very high in performance , it has one limitation of smaller context .

- But one advantage of spacy is its very high inference speed and very small size to be hosted anywhere easily with less resources

RESULTS:

- In my opinion the spacy model have worked very good in determining the custom entities of the dataset than llama-7b and hence the Spacy model
- Llama-7b would have provided better results given that if I had the enough resources.