“NER with BERT”

"Named Entity Recognition with BERT for the Automotive Industry"

Adithya Harsha

DePaul university , Chicago, IL, USA

What is Named Entity Recognition (NER) in BERT?

- **Named Entity Recognition (NER)** is a technique in Natural Language Processing (NLP) used to locate and classify entities in text into predefined categories, such as:
- **Names of people** (e.g., "Elon Musk"),
- **Organizations** (e.g., "Tesla"),
- **Locations** (e.g., "Detroit"),
- **Dates** (e.g., "2024"),
- **Products** (e.g., "Model S").

BERT (Bidirectional Encoder Representations from Transformers)

- BERT is an advanced deep learning model developed by Google that significantly enhances NER by using its **transformer architecture**.
- BERT captures the context of words by processing text bidirectionally—looking at both the words that come before and after a given word.
- This capability allows it to understand and differentiate between similar entities with greater accuracy.

How NER Works in BERT

•**Tokenization**: BERT breaks down text into smaller components called tokens, which represent words or sub-words.

•**Bidirectional Encoding**: BERT processes text in both directions (left-to-right and right-to-left) to understand the context in which entities appear. For example, it can distinguish between "Apple" (the company) and "apple" (the fruit) based on surrounding words.

•**NER Prediction Layer**: After encoding the text, BERT’s pre-trained knowledge is used to classify the tokens (words or sub-words) as entities (like "person", "organization", "location") or non-entities (ordinary words). When fine-tuned for NER, BERT learns to recognize specific types of entities based on training data.

What BERT Does in NER Tasks

- BERT applies its deep understanding of context to identify entities in text more accurately than traditional NER models. Here's how it works step by step:

1. **Input Text**: A sentence or document is input into the BERT model.
2. **Tokenization**: The text is split into smaller units (tokens), including full words and sub-words (for handling out-of-vocabulary words).
3. **Contextual Understanding**: BERT processes these tokens in both directions, understanding the context by considering both previous and following words.
4. **Entity Prediction**: Each token is assigned a label such as "Person", "Location", "Organization", "Date", or "Product".

- For example, the sentence **"Tesla launched the new Model S in California."** would result in:
- "Tesla" → **[Organization]**
- "Model S" → **[Product]**
- "California" → **[Location]**

How NER with BERT Helps the Automotive Industry

- Lets dive into Car Dataset for example - how NER helps

Step 1 : Firstly Load the Car dataset

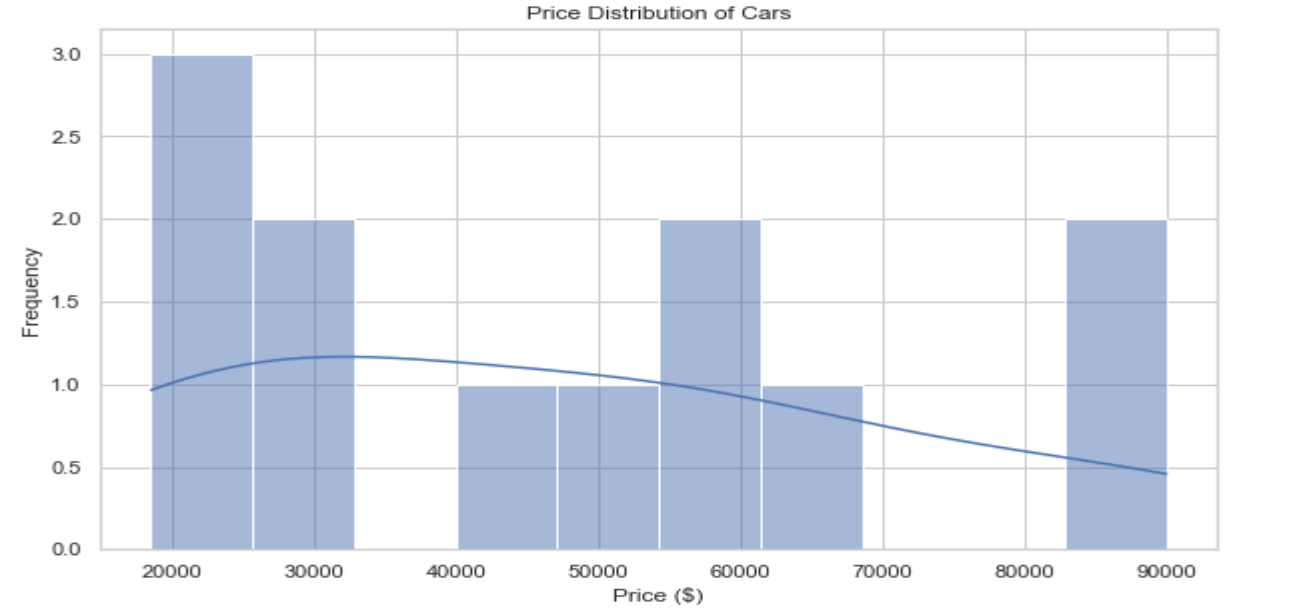
```
print(cars_df.head())
```

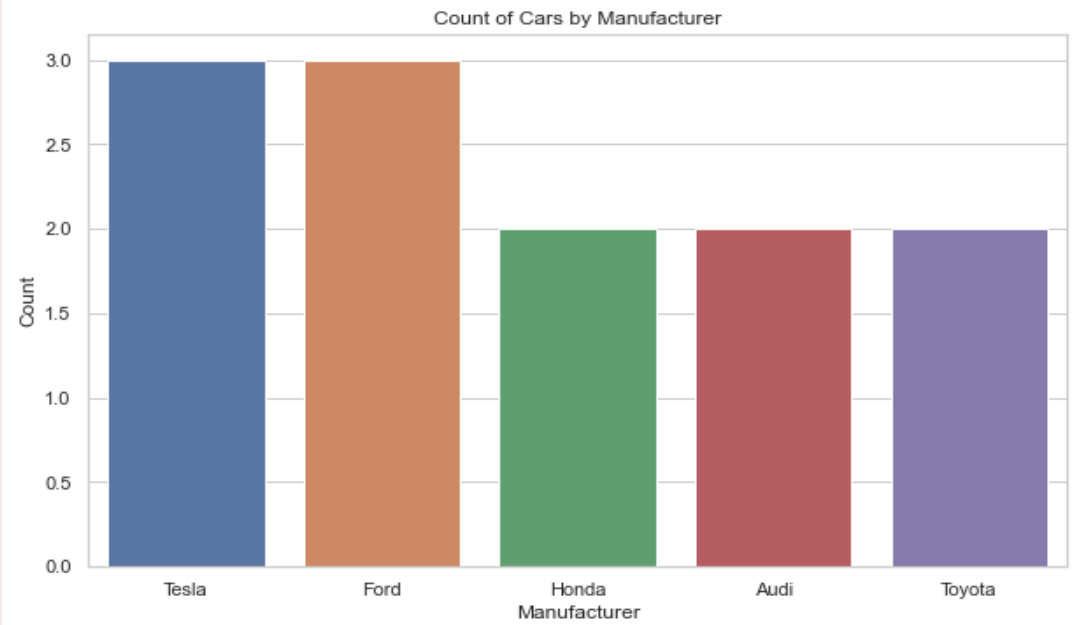
```
Initial DataFrame:
  Car_Model Manufacturer  Year  Engine_Size Fuel_Type  Price     Location
0   Model S        Tesla  2022          2.0  Electric  85000    Palo Alto
1     Civic        Honda  2020          1.5    Petrol  25000        Tokyo
2   Mustang         Ford  2021          3.5    Petrol  58000      Detroit
3        A4         Audi  2022          2.0    Diesel  42000       Berlin
4   Corolla       Toyota  2019          1.8    Petrol  20000  Los Angeles
```

Step 2: Missing Values Check - Found no Missing values

Step 3 : Summary of Visualization Code Functionality:

- **Price Distribution Visualization:**
  - Plots a histogram with a kernel density estimate (KDE) overlay to visualize the distribution of car prices, providing insights into price ranges and frequency of occurrences





- **Count of Cars by Manufacturer:**
  - Creates a count plot to display the number of cars for each manufacturer, allowing for easy comparison of vehicle counts across different brands in the dataset.

STEP 4

```
# Generate descriptions for each car
car_descriptions = []
for index, row in cars_df.iterrows():
    description = f"The {row['Manufacturer']} {row['Car_Model']} is a {row['Fuel_Type'].lower()} car priced at ${row['Price']:,.2
    car_descriptions.append(description)

# Display a sample description
print("\nSample Description:")
print(car_descriptions[0])
```

•**Car Description Generation:** Creates descriptive strings for each car in the dataset with details like manufacturer, model, fuel type, price, location, engine size, and year.

•**Sample Output:** Displays a sample description of the Tesla Model S, summarizing its key features.

Step 5: Perform Named Entity Recognition (NER) on Car Descriptions

1. Import the Pipeline

```
from transformers import pipeline
```

The pipeline function from Hugging Face simplifies the usage of complex models. You can easily create pipelines for different tasks like NER, text classification, translation, etc.

2. Load the Pre-trained NER Model

```
ner_pipeline = pipeline("ner", model="dbmdz/bert-large-cased-finetuned-conll03-english")
```

-Loading the NER pipeline using a pre-trained BERT model: dbmdz/bert-large-cased-finetuned-conll03-english. This model is trained specifically for identifying entities like persons, organizations, and locations based on the CoNLL-2003 dataset.

- "**ner**": This indicates creating a pipeline for Named Entity Recognition.
- dbmdz/bert-large-cased-finetuned-conll03-english: This is ***the model fine-tuned on NER tasks, and it has been trained to recognize named entities such as names of companies, products, people, and places***.

3. Perform NER on each car description

```
# Perform NER on each car description
for description in car_descriptions:
    ner_results = ner_pipeline(description)
    print(f"Description: {description}")
    for entity in ner_results:
        print(f"  Entity: {entity['word']}, Type: {entity['entity']}, Confidence: {entity['score']:.4f}")
    print("\n")
```

4 . This applies the pre-trained NER model to the car_description text.
The **pipeline** function automatically tokenizes the input, runs it through the BERT model, and predicts the named entities.

The **ner_pipeline** processes the text and returns a list of entities, including the tokens (words or sub-words) and the entity type. Each entity is represented by:

- **entity['word']**: The word or token that was identified as part of an entity.
- **entity['entity']**: The label or class of the identified entity (e.g., B-ORG for organization, B-LOC for location, etc.).

OUTPUT OF NER (Named Entity Recognition) results :

```
Description: The Tesla Model S is a electric car priced at $85,000.00, available in Palo Alto. It has a 2.0-liter engine and
was manufactured in 2022.
  Entity: Te, Type: I-MISC, Confidence: 0.9982
  Entity: ##sla, Type: I-MISC, Confidence: 0.9971
  Entity: Model, Type: I-MISC, Confidence: 0.9934
  Entity: S, Type: I-MISC, Confidence: 0.9969
  Entity: Pa, Type: I-LOC, Confidence: 0.9981
  Entity: ##lo, Type: I-LOC, Confidence: 0.9864
  Entity: Alto, Type: I-LOC, Confidence: 0.9980


Description: The Honda Civic is a petrol car priced at $25,000.00, available in Tokyo. It has a 1.5-liter engine and was manu
factured in 2020.
  Entity: Honda, Type: I-MISC, Confidence: 0.9966
  Entity: Civic, Type: I-MISC, Confidence: 0.9976
  Entity: Tokyo, Type: I-LOC, Confidence: 0.9997


Description: The Ford Mustang is a petrol car priced at $58,000.00, available in Detroit. It has a 3.5-liter engine and was m
```

- Breakdown of the Output

1. **Description and Context**:
    1. Each description summarizes key features of a car, including the model, type, price, availability, engine size, and manufacturing year. This contextual information is crucial for understanding the broader automotive market.
2. **Identified Entities**:
    1. Each entity is extracted from the description, classified into various categories, and accompanied by a confidence score:
        1. **Entity Type**: Indicates the nature of the entity:
            1. **B-MISC/I-MISC**: Represents miscellaneous items, often product names or specifications.
            2. **B-LOC/I-LOC**: Indicates locations (like cities).
        2. **Confidence Score**: A numerical value (between 0 and 1) that reflects the model's certainty in its prediction, with higher values indicating greater confidence.

```
Description: The Tesla Model S is a electric car priced at $85,000.00, available in Palo Alto. It ha
s a 2.0-liter engine and was manufactured in 2022.
  Entity: Te, Type: I-MISC, Confidence: 0.9982
  Entity: ##sla, Type: I-MISC, Confidence: 0.9971
  Entity: Model, Type: I-MISC, Confidence: 0.9934
  Entity: S, Type: I-MISC, Confidence: 0.9969
  Entity: Pa, Type: I-LOC, Confidence: 0.9981
  Entity: ##lo, Type: I-LOC, Confidence: 0.9864
  Entity: Alto, Type: I-LOC, Confidence: 0.9980


Description: The Honda Civic is a petrol car priced at $25,000.00, available in Tokyo. It has a 1.5-
liter engine and was manufactured in 2020.
  Entity: Honda, Type: I-MISC, Confidence: 0.9966
  Entity: Civic, Type: I-MISC, Confidence: 0.9976
  Entity: Tokyo, Type: I-LOC, Confidence: 0.9997


Description: The Ford Mustang is a petrol car priced at $58,000.00, available in Detroit. It has a
```

Example Outputs:

- Tesla Model S:
  - Recognized entities include "Tesla" (company name), "Model S" (car model), "Palo Alto" (location), and parts of the sentence (e.g., “Te” and “Pa”) that are subwords as a result of tokenization.
- Honda Civic:
  - Key entities such as "Honda" (manufacturer), "Civic" (model), and "Tokyo" (location) are identified, showcasing its geographical relevance.
- Ford Mustang:
  - Similar entity recognition occurs, highlighting the brand and model alongside its price and location.

How This Helps

1.Entity Recognition Performance:
•The NER model effectively identified key entities related to car descriptions, including manufacturers (e.g., Tesla, Honda, Ford), car models (e.g., Model S, Civic, Mustang), and locations (e.g., Palo Alto, Tokyo, Detroit). The high confidence scores (mostly above 0.99) indicate that the model performed reliably in recognizing these entities.

2.Manufacturer Insights:
•**Diversity of Manufacturers**: Multiple well-known car manufacturers were recognized, such as Tesla, Honda,
•Ford, Audi, and Toyota, indicating a diverse range of car brands represented in the dataset.
•**Brand Recognition**: The recognition of complete brand names (e.g., “Honda,” “Toyota”) alongside their models (e.g., “Civic,” “Corolla”) highlights the model's ability to capture both general and specific brand information.

3.Location Identification:
•**Global Presence**: The model accurately identified various locations associated with the cars, including major cities like Tokyo, Detroit, Berlin, and Los Angeles, which suggests a global distribution of car offerings.
•**Market Analysis**: These location entities can be instrumental for market analysis, helping manufacturers and dealers understand regional preferences and trends in car purchases.

•**Data Completeness**:

•The NER results show that car descriptions are rich with information, including price, fuel type, engine size, and year of manufacture. This completeness suggests that the dataset could be leveraged for further analysis, such as price trends based on manufacturers or geographical locations.

5. Recommendations for Future Work:

•**Model Improvement**: While the NER model performed well, there are instances where entities were split (e.g., “Te,” “##sla” for "Tesla"). Fine-tuning the model on a domain-specific dataset could enhance its ability to recognize complete entities accurately.

•**Integration of Additional Data**: Incorporating more features, such as car performance specifications or consumer ratings, could provide a more comprehensive analysis and aid in predicting market trends.

•**Visualization**: Future presentations of these findings could benefit from visualizations depicting the frequency of manufacturers and locations, enhancing interpretability and engagement.

Aggregated results of NER Outputs

```python
# Filter the DataFrame to show only location entities
loc_entities = ner_df[ner_df['Type'] == 'I-LOC']
print(loc_entities)

# Similarly, you can filter for miscellaneous entities
misc_entities = ner_df[ner_df['Type'] == 'I-MISC']
print(misc_entities)
```

```
                                          Description     Entity   Type
4   The Tesla Model S is a electric car priced at ...         Pa  I-LOC
5   The Tesla Model S is a electric car priced at ...       ##lo  I-LOC
6   The Tesla Model S is a electric car priced at ...       Alto  I-LOC
9   The Honda Civic is a petrol car priced at $25,...      Tokyo  I-LOC
12  The Ford Mustang is a petrol car priced at $58...    Detroit  I-LOC
16  The Audi A4 is a diesel car priced at $42,000....     Berlin  I-LOC
21  The Toyota Corolla is a petrol car priced at $...        Los  I-LOC
```

```
   Confidence
0       0.998
1       0.997
2       0.993
3       0.997
4       0.998
5       0.996
6       0.997
7       0.999
8       0.996
```

```
Entity counts by type:
Type
I-LOC     19
I-MISC    40
```

```
                                          Description    Entity    Type  \
riced at ...       Te  I-MISC
riced at ...    ##sla  I-MISC
riced at ...    Model  I-MISC
riced at ...        S  I-MISC
d at $25,...    Honda  I-MISC
d at $25,...    Civic  I-MISC
ed at $58...     Ford  I-MISC
ed at $58...  Mustang  I-MISC
```

The output shows results from Named Entity Recognition (NER) identifying parts of a car description:

- **Entity**: The identified tokens (e.g., "Te," "Model," "Pa").

- **Type**: Categories of the entities (e.g., **I-MISC** for miscellaneous items like brands and **I-LOC** for locations).

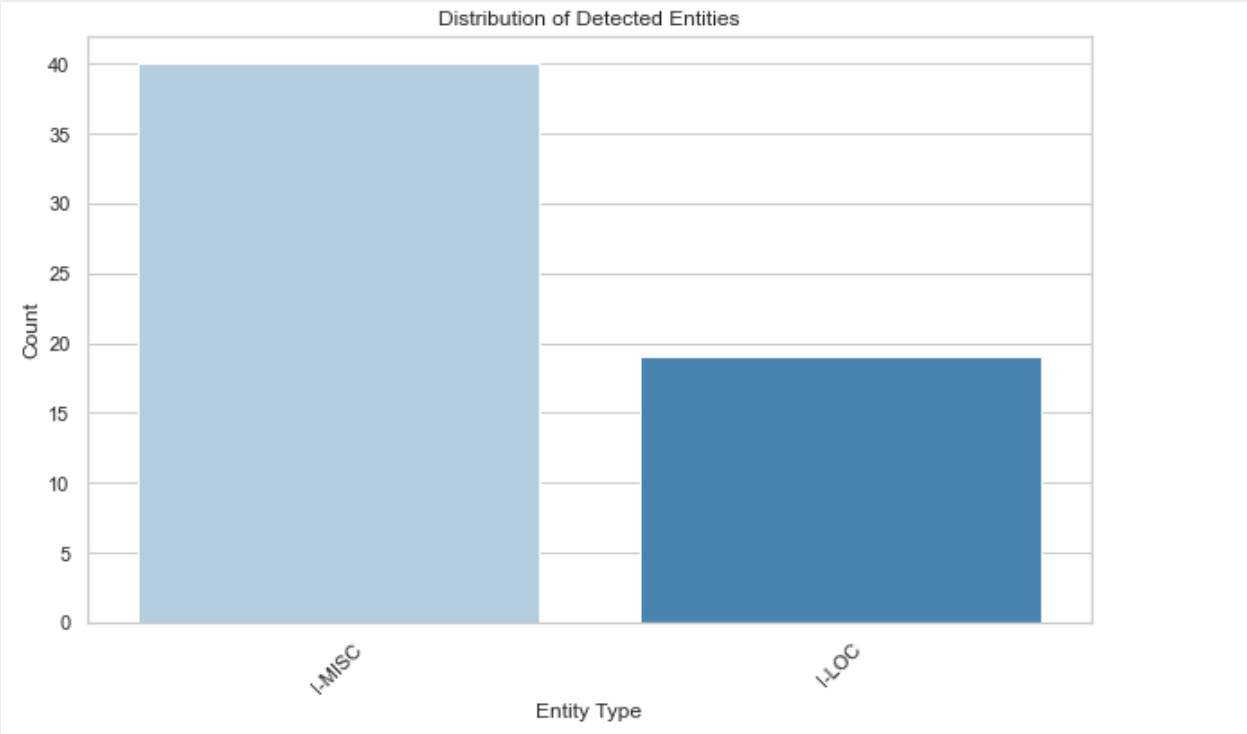
- **Confidence**: The model's certainty in its predictions (scores close to 1 indicate high confidence).

This helps in recognizing and categorizing key information in text related to cars.

Visualtions of outputs and saving data for further use

```
# Count occurrences of each entity type
entity_counts = ner_results_df['Type'].value_counts()

# Visualize the distribution of entity types
plt.figure(figsize=(10, 6))
sns.barplot(x=entity_counts.index, y=entity_counts.values, palette='Blues')
plt.title('Distribution of Detected Entities')
plt.xlabel('Entity Type')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```



- **I-MISC** has a higher count, close to 40.
- **I-LOC** has a lower count, just below 20

```
# Save the NER results to a CSV file
ner_results_df.to_csv('ner_results.csv', index=False)
print("NER results saved to ner_results.csv")
```

```
NER results saved to ner_results.csv
```

Conclusions

•**Understanding Customer Preferences**:
•By identifying and analyzing common entities across car descriptions, automotive companies can gain insights into popular brands, models, and geographical preferences.
For instance, if multiple reviews mention "Tesla Model S" in various contexts, it can signal its popularity or emerging trends in electric vehicles.

•**Market Analysis**:
•The NER results can help automotive firms track how often certain models are mentioned in reviews, discussions, or social media. This data can feed into market analysis, helping manufacturers understand where to focus their marketing efforts or product improvements.

Inventory and Product Management:
•With accurate entity extraction, businesses can categorize their inventory based on recognized models and manufacturers, streamlining processes such as product listing, inventory management, and supply chain logistics.

•**Competitor Benchmarking**:
•By extracting and analyzing mentions of competitors' products and features, companies can benchmark their offerings against market trends. For example, recognizing frequent mentions of "Honda Civic" in discussions about affordability can help a manufacturer position their product more effectively.

7. Improving Customer Service:
•Automating the extraction of entities from customer feedback allows automotive companies to quickly address common issues or questions raised by customers.

For instance, if many customers mention "battery problems" or specific engine types, the company can prioritize those areas for customer service or
product development.

6. Risk Management:
•By analyzing data from various sources (e.g., social media, reviews, news articles), companies can identify potential risks associated with certain models or geographical areas, allowing them to respond proactively.

Overall Summary

Using the NER output helps automotive companies better understand market dynamics, enhance customer experience, optimize inventory management, and maintain a competitive edge. By leveraging machine learning techniques like BERT for NER, the automotive industry can automate data analysis processes that were previously manual and time-consuming, allowing for more efficient operations and informed decision-making

Thank you

Adithya Harsha

Aharsha@depaul.edu

Data Science Capstone - Software Presentation

NER with BERT