# Entity Extraction Pipeline

## Preprocessing

The text preprocessing pipeline underwent multiple iterations to achieve optimal results. Below are the details of the tried and final approaches.

### Tried Approaches

- Spell checker  
- Contractions  
- Strip  
- Lowercasing  
- Punctuation removal  
- Lemmatization

### Final Approaches

- Spell checker  
- Strip  
- Lowercasing  
- Punctuation removal

## Label Preparation

Preparing labels involved multiple approaches before settling on a custom solution.

### Tried Approaches

- Regular expressions (Regex)

### Final Approaches

- Custom function tailored for the dataset

## Tokenization

For tokenization, the final approach used BertTokenizer for consistency with the model architecture.

## Model Architectures

The model design explored various architectures, focusing on different encoder-decoder setups and transformers.

### Architecture Variants

1. Encoder-decoder model for pizza or drink entities:  
 - Trainable embeddings  
 - Bidirectional LSTM for the encoder with 128 units  
 - LSTM decoder with 256 units  
 - Attention layer  
  
2. Shared encoder with two decoders:  
 - Trainable embeddings  
 - Dropout layer with probability 0.2  
 - Bidirectional LSTM with dropout for the encoder (128 units)  
 - LSTM decoder with 256 units and dropout  
 - Attention layer  
 - Two dense output layers for pizza and drink entities  
  
3. Encoder-decoder for PIZZAORDER and DRINKORDER:  
 - Trainable embeddings  
 - Dropout layer with probability 0.2  
 - Bidirectional LSTM with dropout for the encoder (128 units)  
 - LSTM decoder with 256 units and dropout  
 - Attention layer  
 - Dense output layers  
  
4. Transformer model with contextual embeddings:  
 - Contextual embeddings using TFBertModel (untrainable)  
 - Two Bidirectional LSTM layers  
 - Dense output layer

## Model Training

The training process involved systematic preprocessing, label preparation, and tokenization, followed by training the models with various callbacks for optimization and monitoring.

### Steps in Training

- Preprocess text using the final preprocessing pipeline.  
- Prepare labels and add padding to ensure uniform input lengths (maximum 30 words).  
- Tokenize sentences using BertTokenizer, converting them to token IDs and attention masks.  
- Train the model using token IDs, attention masks, and padded labels.  
- Utilize callbacks such as Checkpoint, TensorBoard, ReduceLR, and EarlyStopping.

## Evaluation

The evaluation focused on validation accuracy and exact matching to identify overfitting issues. Efforts were concentrated on addressing these issues before expanding to other metrics.

## Final Testing Model

The final model generates three predictions:  
- Two from the shared encoder with two decoders.  
- One from the encoder-decoder for PIZZAORDER or DRINKORDER.  
The predictions are combined to produce the final output in the TOP format.

## Other Approaches

To augment the dataset, all words within each entity and unique sentence sequences were extracted from the training set. These were used to generate diverse combinations for training augmentation.