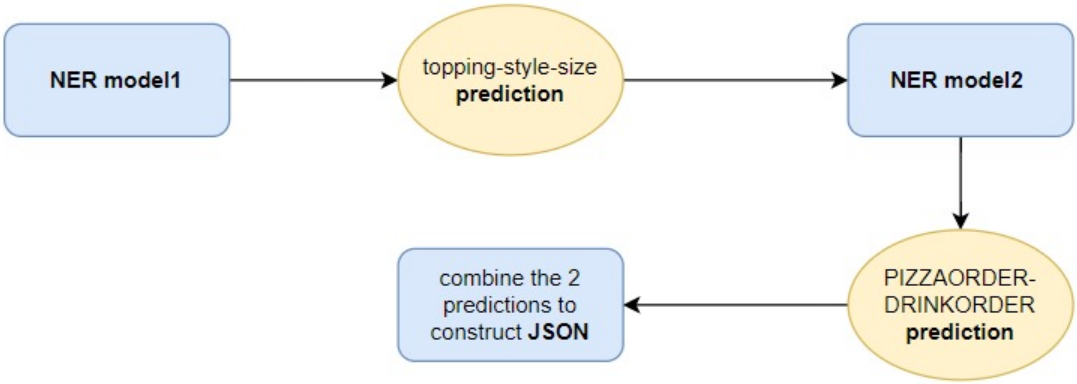
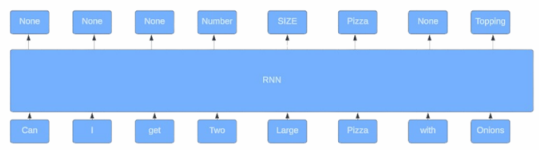
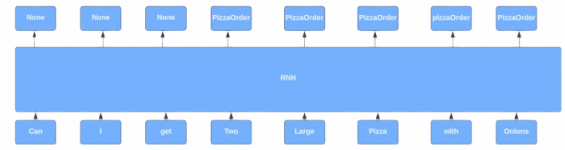
NLP project

Name	sec	BN
Nesma Abd-ElKader	2	28
Sara Gamal	1	20
Eman Ibrahim	1	14
Yousef Osama	2	32

Pipeline







Preprocessing and Feature Extraction

1. **Sampling**: Extract 1% of the data as a representative sample for training.
2. **Cosine Similarity with Sentence Embeddings**:
    - Compute sentence embeddings for each sentence.
    - Add a sentence to the training set only if its cosine similarity with all previously added sentences is below a defined threshold, ensuring

uniqueness.

3. **UNK Augmentation**:
    - Randomly introduce "UNK" tokens at various positions in sentences.
    - This encourages the model to better handle unseen words during training.
4. **Synonym Replacement**:
    - With a certain probability, replace words with their synonyms to help the model generalize and understand various word forms.
5. **Unique Words and Tags Selection**:
    - Identify and select distinct words and associated tags for the training data.
6. **BIO Tagging Format**:
    - Use the BIO tagging scheme, labeling entities with 'B' (beginning) and 'I' (inside) to mark entity boundaries.

Architecture

1. **Embedding Layer**: Each word in the sentence is mapped to a dense vector.
2. **Bidirectional LSTM**: This layer reads the input sequence in both forward and reverse directions, helping the model capture dependencies.

Models

- **Sequence-to-Sequence (Seq2Seq) Model**:
    - The output of this model is the EXR.
    - in preprocessing, we remove stop words
- **Two Named Entity Recognition (NER) Models**:
    1. The first model predicts the **topping-style** entities.
    2. The second model predicts the **PIZZAORDER** and **DRINKORDER** entities.
    3. The outputs of both NER models are then combined and converted into a structured **JSON** format for further processing.

Features we tried:

1. **Word Embedding**: Dense vector representations of words that capture semantic relationships, like synonyms and antonyms.
2. **Contextual Words**: Words around a target word that affect its meaning. Context helps models understand different meanings of the same word.
3. **POS Tagging**: Identifying a word's grammatical role (noun, verb, etc.) to understand sentence structure.
4. **One-Hot Encoding**: Represents words as binary vectors, but doesn't capture semantic relationships.

Best Feature: Word embeddings

Results and Evaluation:

model + (features)	accuracy on DEV dataset (exact match for the whole order)
seq2seq + word embedding	15%
Pipelined(NER) + word embedding	17%
Pipelined(NER) + word embedding + argumentation	55%
Pipelined(NER) + pos + argumentation	3%
Pipelined(NER) + one hot vector encoding +argumentation	35 %
Pipelined(NER) +Contextual Words +argumentation	6%

In the final test set submission, we used pipelined named entity recognition with word embeddings because it gave the best results when tested on dev.