Lab 3: Regex and Arabic NLP with Embeddings

This lab combines rule-based Natural Language Processing (NLP) using Regex with word embedding techniques for text representation and analysis. Below is a detailed explanation of its main components.

Student Name	ID	Section
Lujain Bukassim Almarri	S20106753	1

Objective

The lab aims to:

1. Use Regex for text extraction and processing.
2. Explore different word embedding techniques for text representation, focusing on Arabic text.

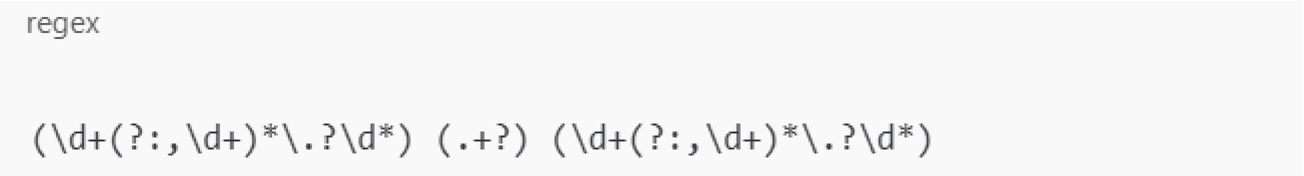
Part 1: Rule-Based NLP and Regex

Task: Generating a Bill

- Objective: Extract product names, quantities, and prices from user-provided text to calculate a total bill.
- Example Input: "I bought three Samsung smartphones 150 $ each, four kilos of fresh banana for 1.2 dollar per kilogram, and one Hamburger for $4.5."

Steps:

1. **Tokenization:**
   - **Split the text into smaller chunks using Regex patterns.**
   - **Example: Split phrases by commas or conjunctions like "and."**
2. **Preprocessing:**
   - **Replace word-based numbers (e.g., "three") with numeric values (e.g., 3) using a predefined dictionary.**
   - **Remove stopwords (common but non-essential words) to focus on meaningful data.**
3. **Regex Matching:**
   - **Use patterns to extract:**
     - **Quantity**
     - **Product name**
     - **Unit price**
   - **Example Regex Pattern:**

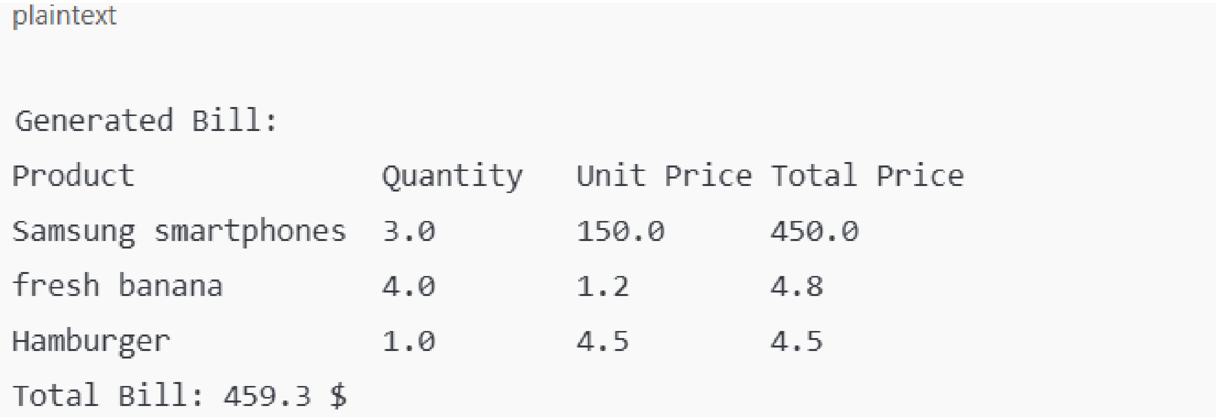


```regex
(\d+(?:,\d+)*\.?\d*) (.+?) (\d+(?:,\d+)*\.?\d*)
```

This captures numbers (for quantity and price) and text (for product name).

4. **Output:**

   - Generate a structured table showing product details and calculate total costs.



plaintext

```
Generated Bill:
Product               Quantity   Unit Price Total Price
Samsung smartphones   3.0        150.0      450.0
fresh banana          4.0        1.2        4.8
Hamburger             1.0        4.5        4.5
Total Bill: 459.3 $
```

Part 2: Word Embedding Techniques

Word Embedding Approaches

1. One-Hot Encoding:

   - Represents words as binary vectors.

   - Limitations: Sparse representation and lack of semantic meaning.

2. Bag of Words (BoW):

   - Represents text as word frequency counts.

   - Ignores word order but works well for simple tasks.

3. TF-IDF (Term Frequency-Inverse Document Frequency):
    - Assigns weight to words based on their frequency in a document relative to all documents.
    - Highlights important words while downweighting common ones.

Advanced Embeddings

1. **Word2Vec:**
    - **Two models:**
        - CBOW (Continuous Bag of Words): Predicts a word based on its context.
        - Skip-gram: Predicts surrounding words for a given word.
    - Captures semantic relationships between words.
2. **FastText:**
    - Extends Word2Vec by considering subword information.
    - Useful for morphologically rich languages like Arabic.
3. **t-SNE Visualization:**
    - Reduces high-dimensional word vectors into 2D space for visualization.
    - Example: Arabic words with similar meanings cluster together.

Implementation Highlights:

- Train embeddings on Arabic text.
- Visualize embeddings using tools like PCA and t-SNE.
- Example Output:
    - t-SNE clusters show semantically similar words grouped together (e.g., synonyms or related concepts).

Takeaways

- **Regex Applications: Efficient for rule-based extraction in structured data like bills or invoices.**
- **Embedding Techniques: Enable nuanced understanding of text, particularly in languages like Arabic with complex morphology.**
- **Visualization: Techniques like t-SNE provide insights into word relationships and model effectiveness.**