HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

SCHOOL OF ELECTRICAL AND ELECTRONIC ENGINEERING

PROJECT REPORT

AC3110E - Natural Language Processing

Topic: Question Classification

Instructor: Prof. Do Thi Ngoc Diep

Member: Nguyen Minh Huyen – 20224272

Nguyen Mai Huong – 20224314

Class: ET – E16 02 – K67

Ha Noi, January 2025

ABSTRACTS

This project focuses on the task of question classification for Vietnamese text, aiming to address a fundamental challenge in natural language processing (NLP). Question classification is a crucial component in the development of intelligent systems such as chatbots, virtual assistants, and automated customer support solutions. The primary goal is to categorize questions into predefined types: "Where," "When," "What," "Why," "How," "Who," "Yes/No," and "Quantity."

Vietnamese poses unique challenges due to its complex linguistic features, including tones, compound words, and a lack of spacing between certain phrases. Furthermore, the limited availability of labeled datasets for Vietnamese text adds to the difficulty. To overcome these challenges, we created a custom dataset of 1,200 questions, evenly distributed across the eight classes, ensuring a balanced and comprehensive representation of each question type.

Three classification methods were investigated: Support Vector Machines (SVM), Large Language Models (LLMs), and Logistic Regression. After a thorough analysis of their advantages and trade-offs, Logistic Regression was chosen for its simplicity, interpretability, and strong performance on the dataset. The model was trained and validated using a carefully preprocessed dataset, with a 70-15-15 split for training, validation, and testing, respectively. The final model achieved an accuracy of 90% on the test set, demonstrating its effectiveness in handling the complexities of Vietnamese text.

This report details the entire process, including dataset preparation, feature engineering, model evaluation, and challenges encountered. The findings highlight the feasibility of using lightweight models like Logistic Regression for Vietnamese question classification while addressing resource constraints. Additionally, the project opens pathways for further exploration, including incorporating more advanced models like neural networks or fine-tuning large language models specifically for Vietnamese. The insights gained can serve as a foundation for future research and applications in Vietnamese NLP.

TABLE OF CONTENTS

I. Introduction

Question classification is a fundamental task in Natural Language Processing (NLP) that aims to categorize a given question into predefined classes based on its intent or semantic structure. This task plays a pivotal role in applications such as question-answering systems, chatbots, and virtual assistants, as it enables the system to understand the type of question being asked and select the most appropriate response.

In this project, we focus on classifying Vietnamese questions into eight categories: "Where", "When", "What", "Why", "How", "Who", "Yes/No", and "Quantity". These categories represent common types of questions that can be asked in everyday conversations or during information retrieval tasks. Given the complexity of the Vietnamese language and its rich structure, the task requires careful preprocessing and feature extraction to enable accurate classification.

We use Logistic Regression, a well-known linear model, to train on a dataset of Vietnamese questions. The report details the process of data preparation, model training, evaluation, and prediction to assess the model's performance and its potential application in question-answering systems.

II. Project Objectives

2.1. Problem Statement

The goal of this project is to develop a system that can automatically classify Vietnamese questions into eight predefined categories: "What", "Why", "How", "Who", "When", "Where", "Quantity", and "Yes/No". These categories represent common types of questions in both everyday conversations and information retrieval tasks.

Classifying questions in Vietnamese presents unique challenges due to the complexity of the language. Vietnamese is a tonal language with a flexible word order, meaning that word placement in a sentence can significantly affect its meaning. This can make it difficult for automated systems to accurately classify questions without considering both the syntactic structure and the contextual clues. Furthermore, Vietnamese uses specific linguistic elements such as classifiers, particles, and question words that are key to understanding the question type. These elements add another layer of complexity to the task, as the system must effectively identify and interpret these features to determine the correct category.

To address these challenges, the project focuses on developing robust preprocessing techniques that can clean and prepare the data for classification, as well as feature extraction methods that capture the necessary linguistic patterns. By accurately classifying questions, the system can be applied in a variety of practical scenarios such as virtual assistants, automated question-answering systems, and chatbots, where understanding user intent is crucial for providing meaningful and accurate responses.

2.2. Literature Review: Methods Studied

2.2.1. Logistic Regression

Logistic Regression is a linear model that has become a cornerstone of classification tasks in machine learning. Its simplicity, interpretability, and computational efficiency make it a practical choice for text classification problems, including question classification.

Overview of Logistic Regression

Logistic Regression predicts the probability of a given input belonging to a specific class. It does this by applying the **sigmoid function** to a linear combination of input features, producing output values between 0 and 1. These probabilities are then used to classify the input into one or more categories based on a threshold.

Logistic Regression is particularly effective in scenarios where the features are well-structured and linearly separable. For multi-class classification, extensions like **One-vs-Rest (OvR)** or **Softmax Regression** allow Logistic Regression to handle multiple categories simultaneously.

Logistic Regression in Question Classification

Question classification involves assigning a question to one of several predefined categories based on its content or intent. In this project, we classify Vietnamese questions into eight categories, such as "What," "Why," "How," etc. Logistic Regression can effectively handle this task using the following steps:

- Text Preprocessing:

○ **Tokenization:** Splitting text into individual words or tokens.

○ **Normalization:** Lowercasing and removing accents from Vietnamese text.

○ **Stopword Removal:** Eliminating uninformative words like "the" or "is".

- Feature Extraction:

○ **TF-IDF (Term Frequency-Inverse Document Frequency):** Converts text into numerical vectors by weighing terms based on their frequency and importance in the corpus.

○ **Word Embeddings:** While Logistic Regression primarily relies on simpler feature representations like TF-IDF, embeddings can also be used when paired with dimensionality reduction techniques.

- Model Training: The Logistic Regression model learns weights for each feature to maximize the likelihood of correctly classifying the training examples.

○ Cross-entropy loss is often used as the objective function.

- Classification and Evaluation:

○ The trained model predicts the category of new questions based on the learned weights.

○ Performance is evaluated using metrics like accuracy, F1-score, precision, and recall.

Strengths of Logistic Regression in Question Classification

- Simplicity: Logistic Regression is easy to implement and interpret, making it a good baseline model.

- Efficiency: The algorithm is computationally efficient, especially for smaller datasets or feature spaces.

- Robustness with TF-IDF: Logistic Regression works well with sparse feature spaces created by techniques like TF-IDF, which are common in text classification.

- Scalability to Multi-Class Problems: With extensions like OvR or Softmax, Logistic Regression handles multi-class classification tasks effectively.

Challenges of Using Logistic Regression

- Linear Assumptions: Logistic Regression assumes a linear relationship between input features and output probabilities, which may not hold in complex datasets.

- Feature Engineering Dependency: The quality of the features, such as TF-IDF vectors, significantly affects model performance.

- Limited Non-Linearity Handling: Logistic Regression cannot naturally handle non-linear relationships without manual feature transformations.

Relevance to Question Classification

Logistic Regression's ability to model probabilities and its reliance on high-quality feature extraction techniques make it suitable for question classification. By leveraging techniques like TF-IDF and effective preprocessing, Logistic Regression provides a fast, interpretable, and competitive solution for categorizing questions into predefined classes. This makes it a valuable model for building systems like chatbots, virtual assistants, and question-answering applications.

Reference: Seidakhmetov, T. (2020). *Question Type Classification Methods Comparison*. Stanford University

Link: https://arxiv.org/pdf/2001.00571

2.2.2. Support Vector Machines (SVM)

Support Vector Machines (SVM) is a robust supervised learning algorithm widely recognized for its effectiveness in text classification tasks, including question classification. The method is particularly suited for high-dimensional feature spaces, which are common in text data.

Overview of SVM

SVM works by identifying an optimal hyperplane that separates data points of different classes. This hyperplane is chosen to maximize the margin between the classes, ensuring that the classifier generalizes well to unseen data. SVM can handle both linearly separable and non-linear data by leveraging kernel functions to map the data into higher-dimensional spaces.

SVM in Question Classification

In question classification, the goal is to assign a question to one of several predefined categories based on its content and intent. For instance, this project categorizes Vietnamese questions into types such as **"What", "Why", "How",** etc. The implementation in the referenced paper incorporates the following steps:

Text Preprocessing

- **Tokenization:** Breaking sentences into individual words or meaningful phrases.

- **Stopword Removal:** Eliminating frequent but uninformative words (e.g., "là", "của", "và").

- **Normalization:** Lowercasing text and remove special characters to standardize the input.

Feature Extraction

- **Bag-of-Words (BoW):** Representing text as sparse vectors based on word occurrence in documents.

- TF-IDF (Term Frequency-Inverse Document Frequency): Assigning weights to words based on their frequency and significance across the dataset. This approach helps reduce the impact of common but non-informative words.

- Syntactic Features: Leveraging additional features like part-of-speech tags to capture grammatical relationships.

Model Training

- Training Process: SVM is trained on labeled data to find the hyperplane separating different question categories.

- Linear Kernel: A linear kernel is often used for high-dimensional text data, providing computational efficiency and simplicity.

- Parameter Optimization: The model's performance is fine-tuned by selecting the best values for parameters such as regularization (C).

Classification and Evaluation

- Classification: The trained SVM model assigns new questions to the most likely category.

- Evaluation: The model's performance is assessed using metrics such as accuracy, precision, and recall. Confusion matrices are also used to visualize classification errors.

Strengths of SVM in Question Classification

- High Dimensionality: SVM effectively handles the high-dimensional nature of text data, such as TF-IDF features.

- Margin Maximization: The focus on maximizing the margin between classes enhances generalization.

- Robustness: SVM is less prone to overfitting, especially with proper regularization.

- Efficient for Small Datasets: The method performs well even with smaller labeled datasets, provided the feature engineering is effective.

Challenges of Using SVM

- Dependence on Feature Engineering: SVM's performance relies heavily on the quality of the extracted features.

- Complexity with Non-Linear Kernels: While the linear kernel is efficient, using non-linear kernels like RBF can increase computational overhead.

- **Hyperparameter Sensitivity:** Parameters such as the regularization constant (C) and kernel settings require careful tuning.

Relevance to Question Classification

In the referenced study, SVM demonstrated strong performance in question classification tasks by leveraging robust feature engineering techniques like TF-IDF and syntactic analysis. By optimizing hyperparameters and selecting appropriate feature representations, SVM achieved significant improvements over other traditional methods like Naive Bayes. These results validate SVM as a reliable and effective tool for categorizing questions based on their intent and structure, laying the foundation for building intelligent question-answering systems.

Reference: Zhang, D., & Lee, W. S. (2003). Question classification using support vector machines. Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 26–32

Link: https://www.comp.nus.edu.sg/~leews/publications/p31189-zhang.pdf

2.2.3. Large Language Models (LLMs)

Large Language Models (LLMs) represent the forefront of natural language processing (NLP) technology, leveraging architectures like **Transformer** to process and generate human-like text. These models, including **GPT-3**, **ChatGPT**, and **BERT-based variants**, are pre-trained on vast datasets and fine-tuned for specific tasks. Their ability to understand and generate text across multiple languages makes them highly versatile for question classification tasks, even in languages like Vietnamese.

Overview of LLMs

LLMs are built on the Transformer architecture, which uses self-attention mechanisms to capture long-range dependencies in text. These models are pre-trained on massive corpora using unsupervised objectives like:

● **Causal Language Modeling (CLM):** Predicting the next token in a sequence (used by GPT-based models).

● **Masked Language Modeling (MLM):** Predicting masked tokens in sentences (used by BERT-based models).

After pre-training, LLMs can be fine-tuned on specific datasets to adapt them to tasks such as classification, summarization, or question answering.

LLMs in Question Classification

The task of question classification involves categorizing questions into predefined intent-based categories. LLMs excel in this area due to their ability to:

- **Comprehend Context:** LLMs process questions holistically, understanding their intent even when phrased in complex or ambiguous ways.

- **Handle Multilingual Text:** Many Large Language Models are trained on multilingual datasets, making them effective for languages like Vietnamese.
- **Adapt via Fine-Tuning:** Fine-tuning LLMs on a labeled dataset of Vietnamese questions allows them to specialize in the nuances of the language and the specifics of the classification task.

- **Zero-Shot and Few-Shot Learning:** Even without fine-tuning, LLMs can perform reasonably well on question classification tasks by leveraging their pre-trained knowledge. In few-shot learning, the model can improve by seeing a few examples of the task.

Advantages of LLMs in Question Classification

- **Deep Understanding of Language:** LLMs capture semantic and syntactic relationships effectively, enabling robust classification even for complex questions.

- **Versatility:** They are capable of handling various languages, including Vietnamese, and can be applied to diverse NLP tasks beyond classification.

- **Handling Unseen Data:** With their extensive pre-training, LLMs can generalize well to unseen or rare question types.

- **Ease of Use:** Pre-trained LLMs can be used directly through APIs or fine-tuned for better performance, simplifying their integration into applications.

Challenges of Using LLMs

- **Resource-Intensive:** LLMs require significant computational resources for fine-tuning and inference, making them less accessible for smaller projects.

- **Data Dependency:** Fine-tuning requires high-quality labeled data, and poor data quality can degrade performance.

- **Latency and Cost:** In real-time applications like chatbots, the inference time and cost of running large models can be prohibitive.

- **Overfitting Risks:** Fine-tuning on small datasets poses the risk of overfitting, which can severely limit a model's ability to generalize and perform well on unseen data, thus impacting its overall effectiveness and adaptability.

Relevance to Question Classification

For Vietnamese question classification, LLMs bring substantial benefits:

● **Pre-trained Knowledge:** LLMs understand general linguistic and contextual patterns, which can be fine-tuned to classify questions accurately.

● **Rich Representations:** Their embeddings encapsulate deep semantic relationships, enabling nuanced distinctions between categories like "Why" and "How."

● **Flexibility:** They handle the diverse structures of Vietnamese questions, including tonal and contextual nuances, effectively.

Reason for Not Choosing:

Despite their impressive performance, the resource-intensive nature of LLMs, along with the requirement for large-scale computational infrastructure, makes them impractical for this project. Additionally, the dataset size (1200 samples) may not fully utilize the potential of such advanced models.

Reference: Said Al Faraby, Ade Romadhony, Adiwijaya. *Analysis of LLMs for Educational Question Classification and Generation*. School of Computing, Telkom University, Bandung, Indonesia. 2022.

Link: https://doi.org/10.1016/j.caeai.2024.100298

2.3. Selected Method

Logistic Regression was chosen for this project due to its simplicity, efficiency, and effectiveness in text classification. It is a linear model that works well with high-dimensional, sparse data, which is typical in text classification tasks. When combined with feature extraction techniques like **TF-IDF**, Logistic Regression can achieve competitive performance.

Despite its simplicity, Logistic Regression offers good interpretability and speed, making it suitable for this task. It serves as a strong baseline for classifying Vietnamese questions into predefined categories, offering a balance between accuracy and computational efficiency.

III. Dataset Preparation

3.1. Dataset Collection

A custom dataset consisting of 1200 questions was created for this project, categorized into 8 classes, specifically designed for Vietnamese text processing. Each class contains 150 questions, making a total of 1200 samples. The classes are as follows:

- **What** (Câu hỏi về cái gì)
- **Why** (Câu hỏi tại sao)
- **How** (Câu hỏi như thế nào)
- **Who** (Câu hỏi về ai)
- **When** (Câu hỏi về thời gian)
- **Where** (Câu hỏi về địa điểm)
- **Quantity** (Câu hỏi về số lượng)
- **Yes/No** (Câu hỏi có/không)

This dataset was manually created to cover a wide range of question types in Vietnamese, ensuring that each class was well-represented with a balanced number of examples. This balance is crucial for training machine learning models, as it prevents the model from being biased toward certain classes.

The dataset was collected and structured in a way that each question is tagged with its appropriate class label, making it suitable for supervised learning tasks such as text classification. The language used is Vietnamese, with a focus on commonly asked questions in various contexts.

3.2. Data Preprocessing

Data preprocessing was a critical step to ensure that the text data was in a format suitable for modeling. The following steps were implemented:

1. **Text Conversion to Lowercase**

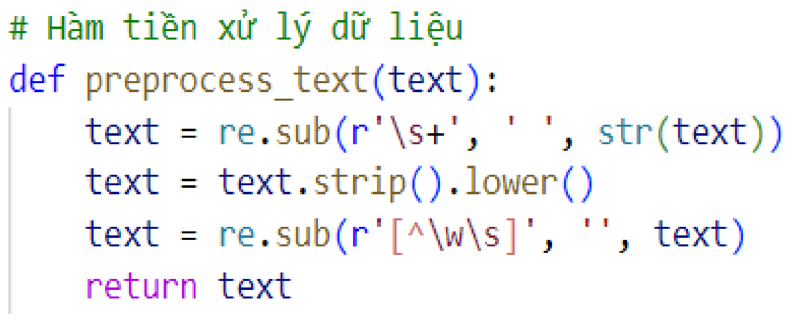
All text data was converted to lowercase to standardize the input, ensuring that the model wouldn't treat the same word in different cases as separate entities.

2. **Removing Punctuation and Special Characters**

Punctuation marks, symbols, and other non-alphanumeric characters were removed using regular expressions. This ensures that the model focuses on meaningful words, ignoring any extra noise from special characters.

3. **Tokenization**

The text was tokenized into words using regular expressions and custom preprocessing steps. This method is crucial for Vietnamese text because it handles compound words and phrases that may not have clear spaces between them, which is common in the Vietnamese language. Tokenization ensures that each word is treated as a separate unit, facilitating more accurate analysis and model training.



```
# Hàm tiền xử lý dữ liệu
def preprocess_text(text):
    text = re.sub(r'\s+', ' ', str(text))
    text = text.strip().lower()
    text = re.sub(r'[^\w\s]', '', text)
    return text
```

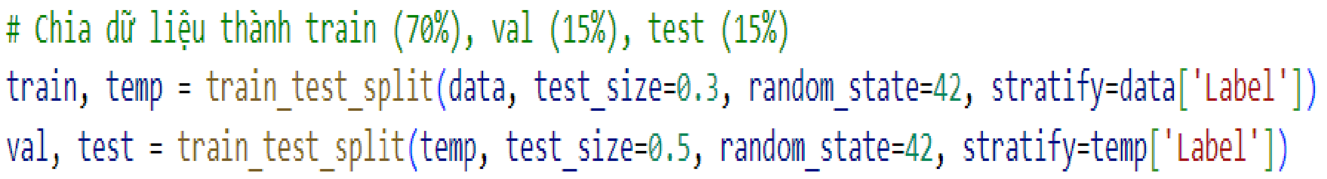
3.3. Data Split

After preprocessing, the dataset was divided into three subsets for training, validation, and testing. The split is as follows:

● **Training Set (70% of the data):** The largest portion of the dataset (840 samples) was used for training the model. This subset is used to teach the model the patterns in the data.

● **Validation Set (15% of the data):** A validation set (180 samples) was created to tune the model's hyperparameters and evaluate its performance during training.

● **Test Set (15% of the data):** The test set (180 samples) was used to assess the final performance of the model after training.



```
# Chia dữ liệu thành train (70%), val (15%), test (15%)
train, temp = train_test_split(data, test_size=0.3, random_state=42, stratify=data['Label'])
val, test = train_test_split(temp, test_size=0.5, random_state=42, stratify=temp['Label'])
```

3.4. Saving and Storing the Preprocessed Data

After preprocessing and splitting the data, the cleaned and split datasets were saved into separate Excel files for further use:

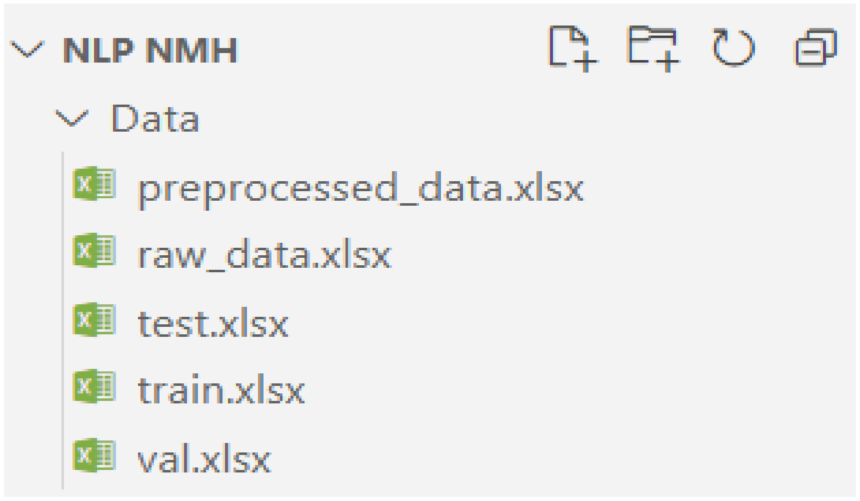
Preprocessed Data: The processed data was saved into preprocessed_data.xlsx.

```
preprocessed_data = pd.DataFrame({
    'Processed_Question': data['Question'].apply(preprocess_text),
    'Label': data['Label']
})

preprocessed_file_path = os.path.join(data_dir, "preprocessed_data.xlsx")

# Lưu dữ liệu đã xử lý vào tệp Excel
preprocessed_data.to_excel(preprocessed_file_path, index=False)
print(f"Dữ liệu đã xử lý được lưu vào: {preprocessed_file_path}")
```

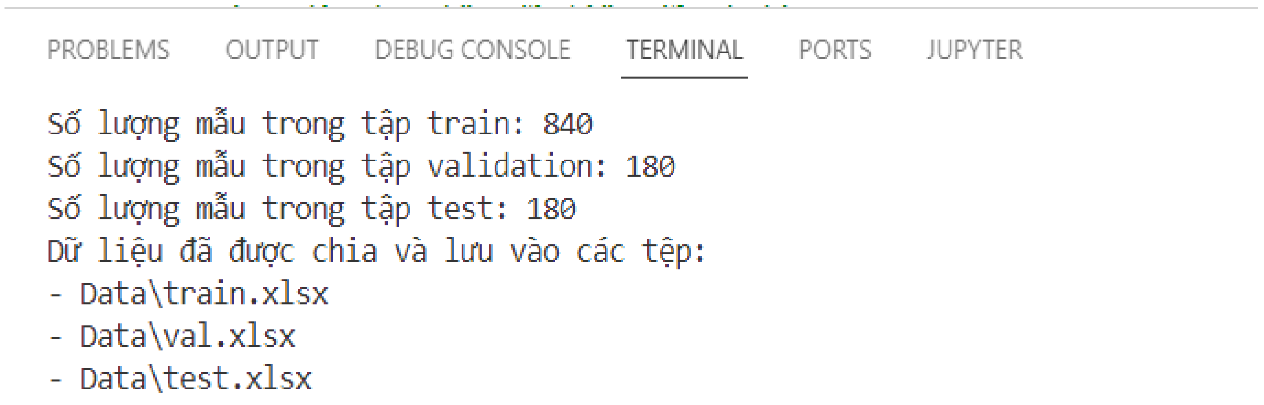
Training, Validation, and Test Sets: These subsets were saved as separate files: train.xlsx, val.xlsx, and test.xlsx.

```
# Đường dẫn lưu các tệp mới
train_path = os.path.join(data_dir, "train.xlsx")
val_path = os.path.join(data_dir, "val.xlsx")
test_path = os.path.join(data_dir, "test.xlsx")

# Lưu dữ liệu ra các tệp Excel
train.to_excel(train_path, index=False)
val.to_excel(val_path, index=False)
test.to_excel(test_path, index=False)
```

The results after saving:



```
NLP NMH
  Data
    preprocessed_data.xlsx
    raw_data.xlsx
    test.xlsx
    train.xlsx
    val.xlsx
```



```
Số lượng mẫu trong tập train: 840
Số lượng mẫu trong tập validation: 180
Số lượng mẫu trong tập test: 180
Dữ liệu đã được chia và lưu vào các tệp:
- Data\train.xlsx
- Data\val.xlsx
- Data\test.xlsx
```

IV. Methodology

4.1. Feature Extraction

In this project, **TF-IDF (Term Frequency-Inverse Document Frequency)** was used as the feature extraction method. This method is widely recognized in text classification tasks due to its simplicity and effectiveness in capturing the importance of words in a document relative to the entire dataset

- Term Frequency (TF): The frequency of a term in a document. It measures how often a word appears relative to the total number of words in the document.

$$\text{TF(t)} = \frac{Total\ number\ of\ words\ in\ document}{Number\ of\ occurrences\ of\ t\ in\ document}$$

- Inverse Document Frequency (IDF): The uniqueness of a term across all documents in the dataset. Rare terms are weighted higher than frequent terms common across documents.

$$\text{IDF(t)} = \log \frac{Number\ of\ documents\ containing}{Total\ number\ of\ documents}$$

By combining **TF** and **IDF**, the method emphasizes significant terms while downplaying common but uninformative words.

$$\text{TF-IDF(t)} = \text{TF(t)} \times \text{IDF(t)}$$

- N-gram Usage: Setting *ngram_range=(1, 2)* allowed the model to capture both individual words (unigrams) and adjacent word pairs (bigrams), enhancing context comprehension.

<u>Example:</u> For the Vietnamese question:

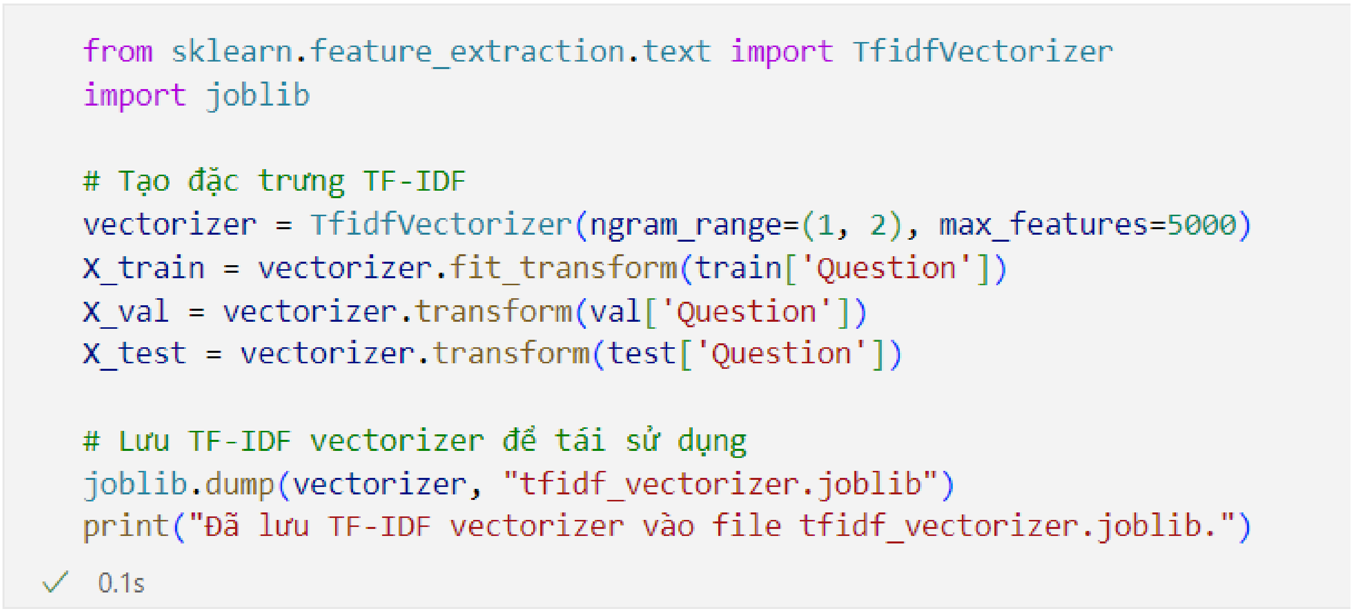
"Thời gian học cần thiết để chuẩn bị kỳ thi là bao lâu? ":

● Unigram features: ["thời", "gian", "học", "cần"]

● Bigram features: ["thời gian", "học cần", "cần thiết"]

These features provide richer context, helping the classifier distinguish subtle differences between questions.

- Implementation:



```
from sklearn.feature_extraction.text import TfidfVectorizer
import joblib

# Tạo đặc trưng TF-IDF
vectorizer = TfidfVectorizer(ngram_range=(1, 2), max_features=5000)
X_train = vectorizer.fit_transform(train['Question'])
X_val = vectorizer.transform(val['Question'])
X_test = vectorizer.transform(test['Question'])

# Lưu TF-IDF vectorizer để tái sử dụng
joblib.dump(vectorizer, "tfidf_vectorizer.joblib")
print("Đã lưu TF-IDF vectorizer vào file tfidf_vectorizer.joblib.")
```

✓ 0.1s

The project employed TfidfVectorizer with ngram_range=(1, 2) and max_features=5000 to limit the vocabulary size while capturing both unigrams and bigrams. This enhanced contextual understanding of Vietnamese questions.

4.2. Model Training

- Logistic Regression: models the relationship between features X and the probability P(y = c |X) of the target class Y using the **sigmoid function**:

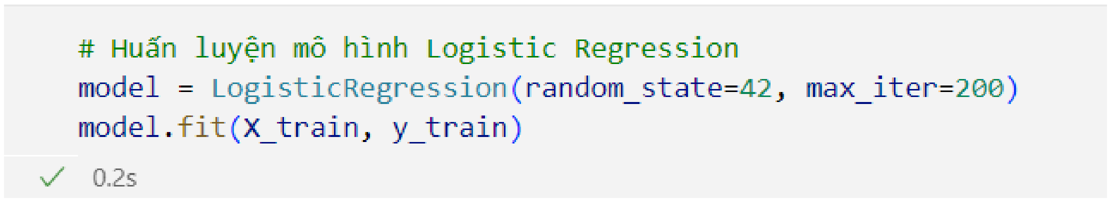
$$P(\,y = c \mid x\,) = \frac{1}{1 + e^{-(w \cdot x + b)}}$$

where w represents the feature weights, x is the feature vector, and b is the bias term.

- Decision Rule: For multi-class classification, the model employs the **softmax function**

$$P(\,y = c \mid x\,) = \frac{e^{w_k \cdot x}}{\sum_{j=1}^{K} e^{w_j \cdot x}}$$; where K is the number of classes.

- Implementation:



```python
# Huấn luyện mô hình Logistic Regression
model = LogisticRegression(random_state=42, max_iter=200)
model.fit(X_train, y_train)
```

```
✓ 0.2s
```

The Logistic Regression model in this project was implemented using the **scikit-learn** library. Key configurations include:

- **max_iter=200:** Ensures sufficient iterations for model convergence.
- **random_state=42:** Guarantees reproducibility of results.

V. Results

5.1 Evaluation Metrics

5.1.1 Accuracy

Definition: Accuracy measures the overall percentage of correctly classified instances among all predictions. It is calculated as:

$$\text{Accuracy} = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions}$$

Accuracy is a straightforward metric that provides an overall sense of how well the model performs. However, it can be misleading in the case of imbalanced datasets, where the model could achieve high accuracy by focusing on the majority class.

5.1.2 Precision, Recall, and F1-Score

These metrics evaluate the model's performance for each individual class and are particularly useful for imbalanced datasets.

- **Precision** measures how many of the predicted positive instances are actually correct. It is important when false positives carry significant consequences.

$$\text{Precision} = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

- **Recall** evaluates how well the model identifies all relevant instances. It is critical in cases where missing a positive instance is costly.

$$\text{Recall} = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

- **F1-Score** is the harmonic mean of Precision and Recall, balancing the trade-off between these two metrics. It is particularly useful when the dataset is imbalanced.

F1-Score = $2 . \frac{Precision . Recall}{Precision + Recall}$

5.1.3 Confusion Matrix

Definition: The confusion matrix is a table that summarizes the performance of a classification model by comparing the true labels with the predicted labels.

Structure:

- **Diagonal Cells:** Represent the correctly classified instances for each class.

- **Off-Diagonal Cells:** Represent misclassifications (e.g., instances of one class incorrectly predicted as another).

The confusion matrix provides granular insights into specific areas of strength and weakness:

- It helps identify which classes are being confused, offering actionable insights for improvement.

- It complements metrics like accuracy by showing the distribution of errors

5.2 Performance Summary

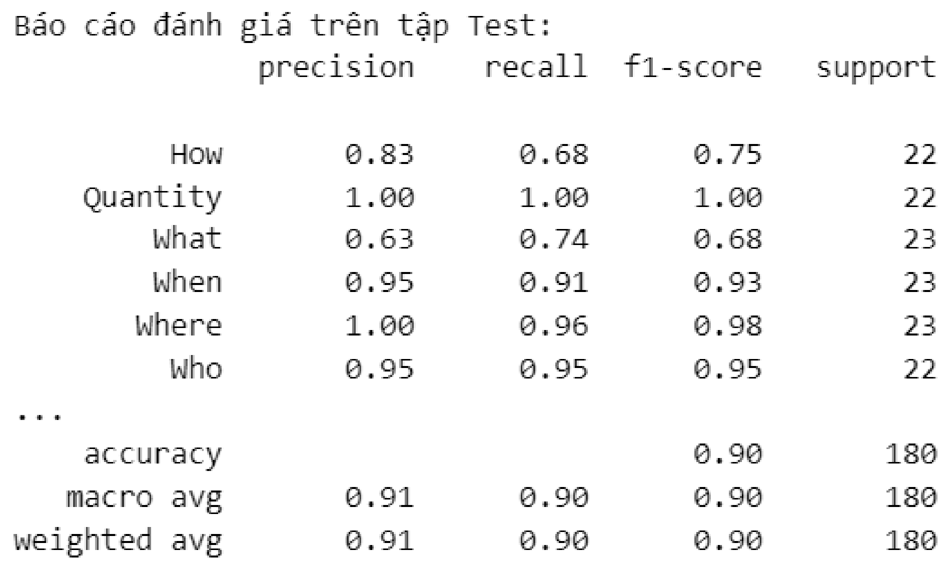
5.2.1. Accuracy

- **Validation Accuracy:** The Logistic Regression model achieved a validation accuracy of **88.88%**, indicating strong generalization on the unseen validation dataset. This suggests that the model effectively captures the patterns in the training data without overfitting.

- **Test Accuracy:** On the test dataset, the model recorded an accuracy of **90%**, further demonstrating its robustness and reliability in classifying Vietnamese questions into predefined categories.

5.2.2. Precision, Recall, F1-Score

The evaluation metrics of Precision, Recall, and F1-Score provide a deeper insight into the performance of the model across individual categories:



```
Báo cáo đánh giá trên tập Test:
              precision    recall  f1-score   support

         How       0.83      0.68      0.75        22
    Quantity       1.00      1.00      1.00        22
        What       0.63      0.74      0.68        23
        When       0.95      0.91      0.93        23
       Where       1.00      0.96      0.98        23
         Who       0.95      0.95      0.95        22
...
    accuracy                           0.90       180
   macro avg       0.91      0.90      0.90       180
weighted avg       0.91      0.90      0.90       180
```

Precision:

- Indicates the proportion of correctly predicted samples in each category out of all predicted samples.

- Categories such as **Quantity**, **Who**, and **Yes/No** achieved exceptionally high precision scores (close to or at 1.00), reflecting the model's accuracy in these distinct categories.

- Lower precision for categories like **What** and **How** (e.g., 0.63 for **What** in the test set) highlights some misclassifications in these ambiguous categories.

Recall:

- Measures the proportion of correctly predicted samples in each category out of the actual samples.

- Categories like **Quantity**, **Who**, and **Yes/No** maintained perfect or near-perfect recall, indicating the model's ability to identify most instances of these categories.

- For **What** and **How**, recall scores were slightly lower (e.g., 0.68 for **How** in the test set), reflecting the difficulty in capturing certain patterns in these categories.

F1-Score:

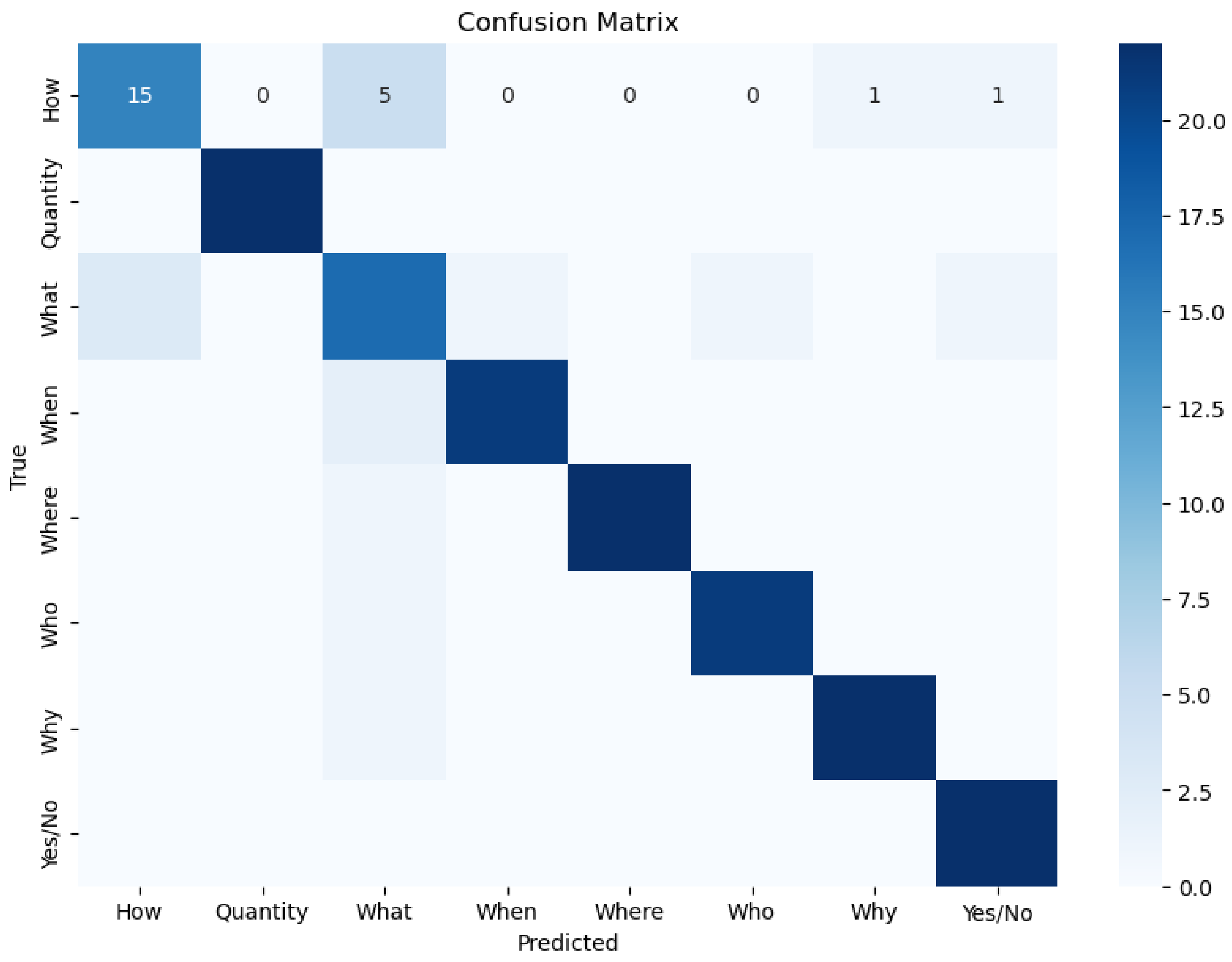
- Provides a balance between precision and recall, serving as a single metric to evaluate the model's overall performance in each category.

- Categories such as **Quantity**, **Who**, and **Yes/No** recorded the highest F1-scores (close to 1.00), confirming the model's strong performance for these question types.

- F1-scores for **What** (0.68 in the test set) and **How** (0.75 in the test set) indicate room for improvement, possibly due to semantic overlap with other categories.

5.2.3. Confusion Matrix:

The model demonstrates strong classification capabilities, as reflected by the high number of correct predictions across most categories. This is evident from the prominent diagonal elements in the confusion matrix, indicating accurate classification for a significant proportion of samples.



Strengths:

● **"Quantity":**

This category shows perfect predictions, with no misclassifications. The model effectively captures the unique patterns of questions like "Bao nhiêu" (How many) or "Mấy" (How much).

● **"Who" and "Where":**

The model achieves high accuracy in classifying questions starting with "Ai" (Who) and "Ở đâu" (Where), highlighting its ability to identify these distinct patterns.

● **“Yes/No”:**

The Yes/No category has shown significant improvement, with most questions accurately classified. This suggests that the model successfully learned the patterns specific to polar questions.

Misclassifications:

● **“How” and “What”:**

Some questions from the How category were misclassified as What. This can be attributed to semantic overlap between these categories in the Vietnamese language. For instance, "Làm thế nào" (How) and "Cái gì" (What) share structural and contextual similarities.

This issue highlights the challenge of distinguishing between intent when linguistic cues are less distinct.

● **“When” and “Why”:**

A few questions from the **When** category were classified as **Why**. Temporal and causal questions in Vietnamese often overlap in phrasing, contributing to these misclassifications.

Insights and Observations:

- **Distinct Patterns:** Categories like **Quantity**, **Who**, and **Where** benefit from distinct phrasing in Vietnamese, leading to high classification accuracy.

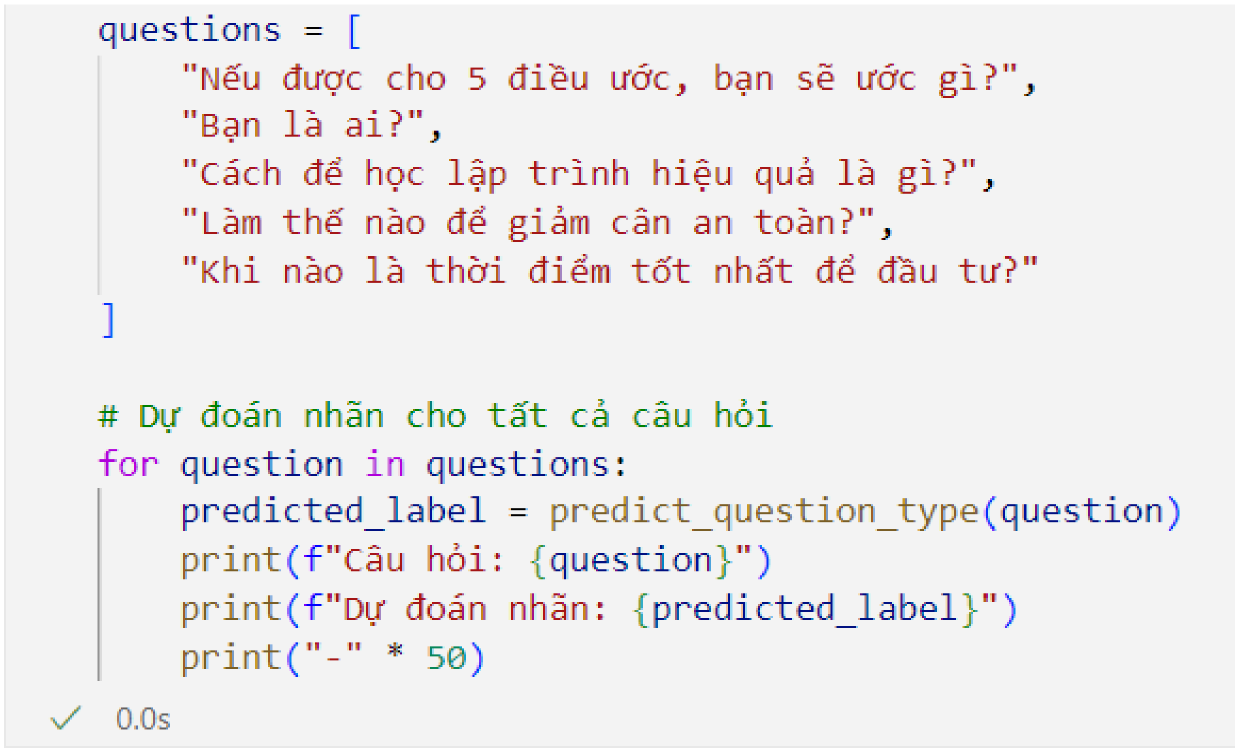
- **Semantic Ambiguity:** Overlap in the structure and intent of questions from categories such as **How** and **What** causes the majority of misclassifications.

- **Context-Dependence:** Some misclassifications in categories like **Yes/No** indicate that the model occasionally struggles with the nuanced context required to classify polar questions accurately.

Conclusion: Overall, the model is well-suited for Vietnamese question classification tasks, achieving reliable performance in distinguishing between diverse question types. Nevertheless, the results also highlight areas for improvement, such as enhancing classification for ambiguous categories through richer contextual features or advanced methods like contextual embeddings. This demonstrates the potential for further optimization and scalability of the approach.

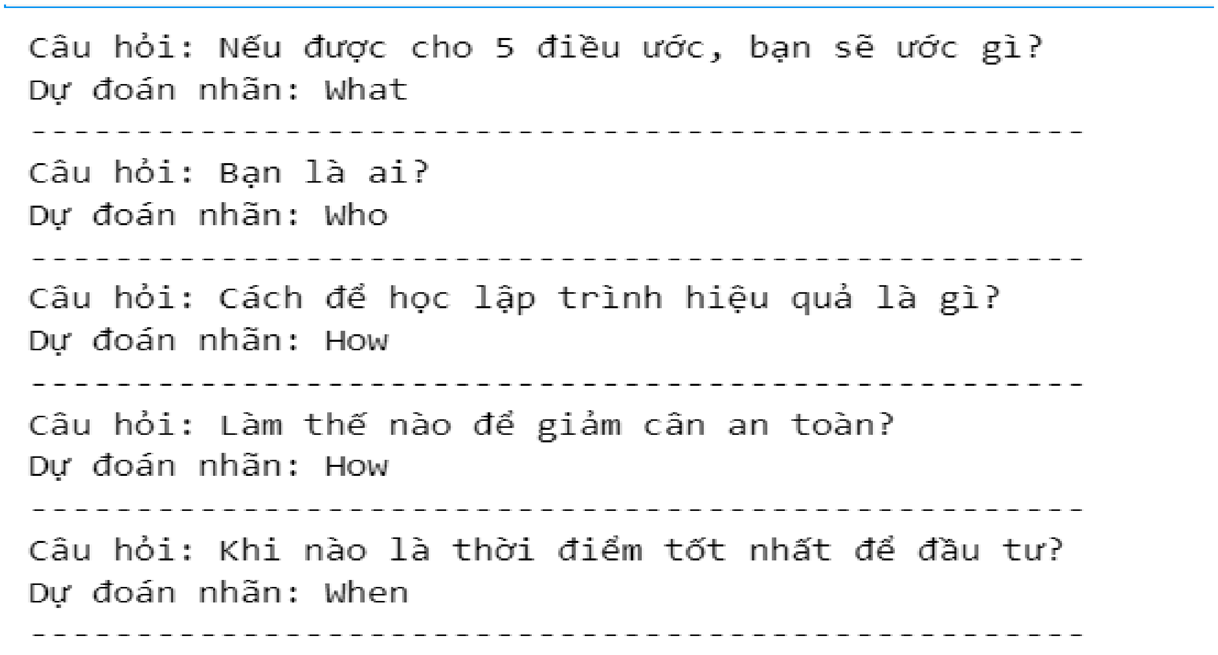
5.3. Testing on Real Examples

To evaluate the model's performance on real-world data, we conducted a test with five sample questions. These questions cover a variety of scenarios to assess the model's ability to classify different types of queries. Below are the results:



```
questions = [
    "Nếu được cho 5 điều ước, bạn sẽ ước gì?",
    "Bạn là ai?",
    "Cách để học lập trình hiệu quả là gì?",
    "Làm thế nào để giảm cân an toàn?",
    "Khi nào là thời điểm tốt nhất để đầu tư?"
]

# Dự đoán nhãn cho tất cả câu hỏi
for question in questions:
    predicted_label = predict_question_type(question)
    print(f"Câu hỏi: {question}")
    print(f"Dự đoán nhãn: {predicted_label}")
    print("-" * 50)
```

✓ 0.0s



```
Câu hỏi: Nếu được cho 5 điều ước, bạn sẽ ước gì?
Dự đoán nhãn: What
--------------------------------------------------
Câu hỏi: Bạn là ai?
Dự đoán nhãn: Who
--------------------------------------------------
Câu hỏi: Cách để học lập trình hiệu quả là gì?
Dự đoán nhãn: How
--------------------------------------------------
Câu hỏi: Làm thế nào để giảm cân an toàn?
Dự đoán nhãn: How
--------------------------------------------------
Câu hỏi: Khi nào là thời điểm tốt nhất để đầu tư?
Dự đoán nhãn: When
--------------------------------------------------
```

These results showcase the model's ability to handle diverse types of questions. The predictions provide valuable insights into areas where the model performs well and where it may require further refinement, such as resolving ambiguities or improving classification accuracy for overlapping categories.

VI. Future Work

To enhance the performance and applicability of the question classification system, future work should consider the following directions:

1. **Adopting Advanced Models:** Exploring Transformer-based architectures, such as PhoBERT or GPT-based models, to capture richer contextual information and improve classification accuracy.

2. **Dataset Expansion:** Collecting a more extensive and diverse dataset to address class imbalances and improve the model's generalizability.

3. **Multilingual Capabilities:** Extending the model to support multilingual question classification, increasing its usability across diverse user bases.

4. **Real-Time Integration:** Developing optimized inference pipelines for deploying the model in real-time applications, such as chatbots and virtual assistants.

VII. Conclusion

This project demonstrated the effectiveness of Logistic Regression for Vietnamese question classification. By leveraging robust preprocessing techniques and TF-IDF feature extraction, the model achieved a test accuracy of **90%**. The findings underscore the importance of feature engineering and model simplicity for practical NLP applications. While Logistic Regression provided a strong baseline, there is substantial room for improvement through advanced models and enriched datasets. The project lays the groundwork for future developments in Vietnamese question classification systems, with the potential to enhance user interaction in digital platforms.

References

1. Seidakhmetov, T. (2020). "*Question Type Classification Methods Comparison*." Stanford University. Retrieved from https://arxiv.org/pdf/2001.00571.

2. Zhang, D., & Lee, W. S. (2003). "*Question classification using support vector machines*." Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 26–32

3. Said Al Faraby, Ade Romadhony, Adiwijaya. "*Analysis of LLMs for Educational Question Classification and Generation*." School of Computing, Telkom University, Bandung, Indonesia. 2022.

4. Powers, David M.W. *"Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation."* Journal of Machine Learning Technologies 2, no. 1 (2011): 37-63.

5. Fawcett, Tom. *"An introduction to ROC analysis."* Pattern Recognition Letters 27, no. 8 (2006): 861-874.

6. Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. *"Logistic Regression."* The Elements of Statistical Learning. Springer, 2009.

7. Ramos, Juan. *"Using tf-idf to determine word relevance in document queries."* Proceedings of the First International Conference on Machine Learning (ICML). 2003.

8. Cavnar, William B., and John M. Trenkle. *"N-gram-based text categorization.*" Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval. 1994.

9. NLP Course Materials and Lectures.