Midhuneshwar Kandasamy Kanivalvan
midhuneshwar01@gmail.com

Deep Learning and Artificial Intelligence - Project Proposal

Abstract

This research project aims to develop a machine-learning model for answering Python-related questions. The model will be trained to generate accurate responses and categorize queries into relevant topics by leveraging a dataset of questions and answers. The study will contribute to automated support systems, online learning platforms, and AI-driven coding assistants. The proposed methodology encompasses data preprocessing, feature extraction, model selection, and performance evaluation using standard metrics.

1. Introduction

Natural Language Processing (NLP) has gained significant traction in recent years, particularly in automated question-answering systems[1]. The increasing reliance on AI-driven solutions in education and programming communities necessitates the development of models that can provide precise and contextual answers to technical queries. This study focuses on leveraging machine learning techniques to improve the efficiency of answering Python-related questions.

2. Dataset Description

2.1 Dataset Source

The study will utilize a specialized Chinese single-turn question-and-answer (Q&A) dataset for Python learners, containing 5,960 questions with corresponding structured, specialized answers[4].

2.2 Dataset Structure

The dataset comprises:

- Question: The Python-related inquiry
- Answer: A corresponding response to the query
- Category: Classifications such as syntax, debugging, libraries, or algorithms

2.3 Data Format

The dataset is structured in CSV or JSON format, facilitating machine learning preprocessing and analysis[4].

3. Methodology

The research will follow a systematic approach, divided into the following stages:

3.1 Data Preprocessing

- Cleaning and formatting textual data
- Tokenization and vectorization using methods such as TF-IDF, Word2Vec, or BERT embeddings[1]
- Addressing missing data and ensuring dataset balance

Midhuneshwar Kandasamy Kanivalvan
midhuneshwar01@gmail.com

3.2 Model Selection

- Baseline Models: Logistic Regression, Naive Bayes with TF-IDF
- Advanced Approaches: Deep learning techniques including Long Short-Term Memory (LSTM) networks and Transformer-based models (BERT, GPT) to enhance prediction accuracy[3]
- Fine-Tuning: Utilizing pre-trained NLP models for improved contextual understanding

3.3 Evaluation Metrics

To assess the model's performance, the following metrics will be used:

- Classification Models: Accuracy, Precision, Recall, and F1-score
- Answer Quality Assessment: BLEU and ROUGE scores[1]

4. Challenges and Limitations

4.1 Data Quality

Some questions in the dataset may lack clarity or context, potentially leading to misclassification[4].

4.2 Computational Resources

Deep learning models, particularly Transformers, require substantial computational power for effective training and fine-tuning[3].

4.3 Generalization and Performance

The model should accurately process new, unseen Python-related queries while maintaining performance across different question types[1].

4.4 Labeling and Categorization

Additional techniques such as unsupervised learning or topic modeling will be explored if the dataset lacks predefined categories.

5. Expected Results and Applications

5.1 Predicted Outcomes

- A trained classification model capable of categorizing Python questions with high accuracy
- A question-answering system that generates meaningful and contextually relevant answers[1]

5.2 Practical Applications

- Deployment as an interactive tool, API, or chatbot for programming communities
- Integration into online learning platforms for enhanced user support
- Contribution to AI-driven knowledge retrieval systems in the field of programming assistance[4]

Midhuneshwar Kandasamy Kanivalvan
midhuneshwar01@gmail.com

6. Conclusion

This research aims to develop a robust machine-learning model for answering Python-related questions efficiently. By leveraging NLP techniques and deep learning architectures, the project seeks to improve the accuracy and reliability of AI-powered programming assistants. Future work may involve expanding the dataset, refining the model, and integrating it into real-world applications.

Citations:

[1]https://wandb.ai/mostafaibrahim17/ml-articles/reports/The-Answer-Key-Unlocking-the-Potential-of-Question-Answering-With-NLP--VmlldzozNTcxMDE3
[2]https://www.datacamp.com/blog/machine-learning-projects-for-all-levels
[3]https://www.digitalocean.com/community/tutorials/how-to-train-question-answering-machine-learning-models
[4]https://arxiv.org/html/2412.18093
[5]https://asperbrothers.com/blog/question-answering-python/
[6]https://community.openai.com/t/training-openai-on-a-private-dataset/38601
[7]https://www.thetalkingmachines.com/article/simple-question-answering-qa-systems-use-text-similarity-detection-python
[8]https://towardsdatascience.com/building-a-question-answering-system-part-1-9388aadff507?gi=1c8beeb8a75a