PyLinguist: Automated Translation of Python for Hindi Programmers

First Author
Antara Tewary
G01413546
atewary@gmu.edu

Second Author
Ankit Kumar
G01436204
akumar37@gmu.edu

Third Author
Ankit Kumar
G01436204
akumar37@gmu.edu

1 Introduction

The paper addresses a fundamental challenge in natural language processing: how to effectively train text classification models with very limited labeled data and whether the knowledge contained in pre-trained language models can be leveraged more effectively for few-shot learning by reformulating input examples as cloze-style phrases.

1.1 Task / Research Question Description

Core research questions are-
- Can natural language task descriptions be effectively combined with supervised learning to improve few-shot text classification?
- How can we exploit the implicit knowledge in pre-trained language models through carefully designed patterns?
- Is it possible to achieve better performance than standard supervised learning with as few as 10 labeled examples per class?

1.2 Motivation & Limitations of existing work

The need for this research is driven by practical limitations in current NLP applications-

- *Standard supervised learning*: Requires large amounts of labeled training data, performs poorly in few-shot scenarios, and fails to leverage the implicit knowledge in pre-trained language models.

1.3 Proposed Approach

Briefly describe the core contribution of the paper's proposed approach.

1.4 Likely challenges and mitigations

What is hard about this task / research question? What are your contingency plans if the reproduction turns out to be harder than expected or experiments do not go as planned?

2 Related Work

Include 3-4 sentence descriptions of no less than 4 relevant papers (as applicable). Also mention how your work differs from these. Note that prior work should be properly cited in References, e.g., when you use the BERT model (**?**) you could cite it in this way; if you want to refer to the authors of a certain paper, you should use `citet`, e.g., "**?** proposed the BERT model." See `https://acl-org.github.io/ACLPUB/formatting.html` for instructions.

3 Experiments

3.1 Datasets

Please list which datasets you used, whether or not you have access them, and whether or not they are publicly available with the same preprocessing and train / dev / tests as the previous work you will be comparing to (if applicable). If you plan to collect your own dataset for evaluating robustness, please describe clearly the data plan (the data source, how you plan to collect it, how you would preprocess it for the task, etc.).

3.2 Implementation

Please provide a link to a repo of your reimplementation (if applicable) and appropriately cite any resources you have used.

4 Evaluation

4.1 Dataset and Test Configuration

- **Total Dataset**: 500k Python code samples from hugging face (jtatman, 2024)
- **Test Configuration**:
  - Translation set: 10 samples selected for translation
  - Example sets: Varying sizes (5,10,20,30)

that are given to GPT model for reference
- Human evaluation set: 20 samples evaluated by human evaluators

- **Keyword Dictionary**: 234 pre-mapped English-Hindi keyword pairs translated by Joshua Otten.

4.2 Human Evaluation Framework

Two bilingual evaluators (Hindi-English) with 4 years of experience in Python programming evaluated 20 code samples generated by our model. They followed the following rating scale-
Rating Scale(1-5):

- **1**: Unusable and incorrect translation
- **2**: Partially correct translation, major revisions needed
- **3**: Mostly correct translation, minor revisions needed
- **4**: Good translations with minimal revisions needed
- **5**: Perfect translation, no revisions needed

5 Results

Provide a table comparing your results to the published results.

5.1 Discussion

Discuss any issues you faced. Do your results differ from the published ones? If yes, why do you think that is? Did you do a sensitivity analysis (e.g. multiple runs with different random seeds)?

5.2 Resources

Discuss the cost of your reproduction in terms of resources: computation, time, people, development effort, communication with the authors (if applicable).

5.3 Error Analysis

Perform an error analysis on the model. Include at least 2-3 instances where the model fails. Discuss the error analysis in the paper – what other analyses could the authors have ran? If you were able to perform additional error analyses, report it here.

6 Robustness Study

Explain your approach for Evaluating the Model Robustness. Describe what robustness analysis you have performed. Provide sufficient details about your perturbation data, how you created it, how you used it as a robustness benchmark to evaluate the model, in what metrics, etc.

6.1 Results of Robustness Evaluation

Describe the evaluation results of your reproduced model on the robustness benchmark that you created. Include at least 2 examples where the model performs well and 2 examples where it fails (i.e., being not robust). Provide sufficient analysis and your thoughts on the observations.

6.2 Discussion

Provide any further discussion here, e.g., what challenges did you face when performing the analysis, and what could have been done if you will have more time on this project? Imagine you are writing this report to future researchers; be sure to include ”generalizable insights” (e.g., broadly speaking, any tips or advice you’d like to share for researchers trying to analyze the robustness of an NLP model).

7 Workload Clarification

Describe how the team divides the workload in this checkpoint. Note that each team member should contribute roughly the same amount of work to this assignment.

8 Conclusion

Is the paper reproducible?

References

jtatman. 2024. Python code dataset 500k. https://github.com/jtatman/python-code-dataset-500k. Accessed: 05-12-2024.