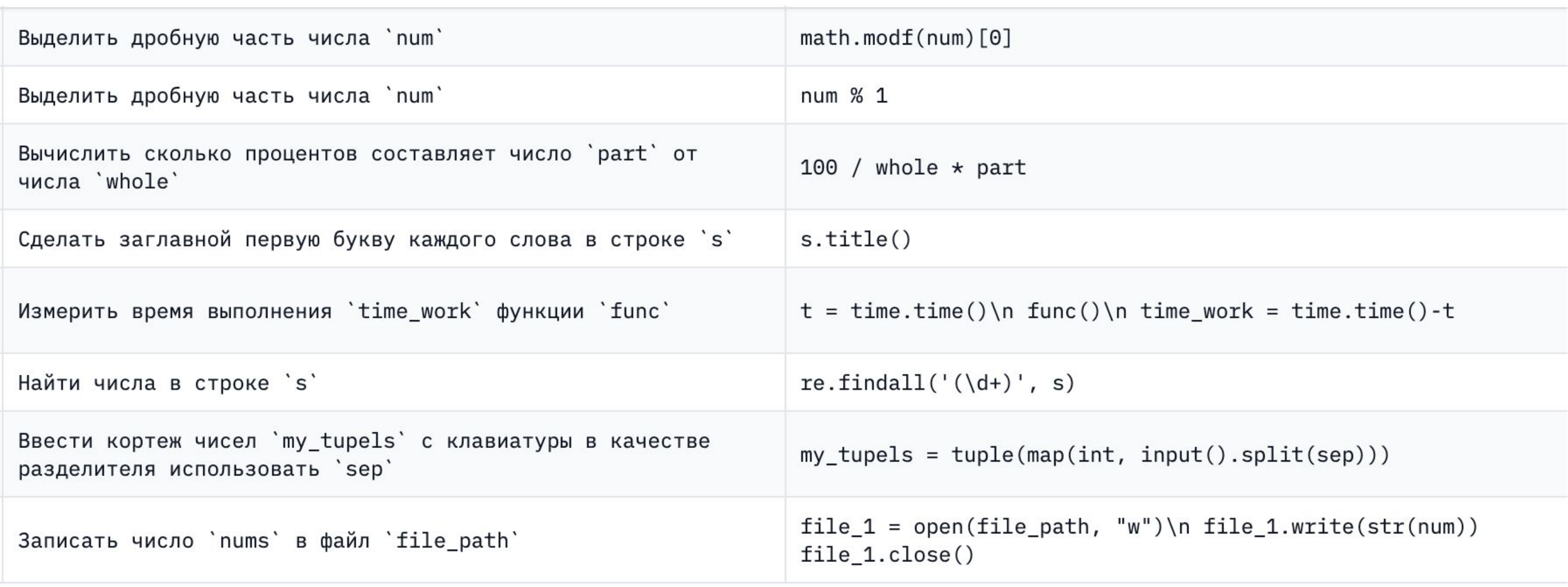
Code Generation For Low-Resource Natural Languages

Team 13: Shaurya Singh (shauryas), Geerisha Jain (geerishj), Riya Singhal (riyapras)
School of Computer Science, Carnegie Mellon University

Research Problem

In the field of code generation, the primary objective is to produce code in response to natural language prompts. Currently, the term "multilingual capabilities" in the context of code generation models refers to the ability of these models to generate output in multiple coding languages. However, a notable challenge arises from the fact that the prompts provided to these models are primarily in English. This discrepancy poses significant accessibility barriers, hindering the utilization of these models across linguistic boundaries. To address this issue, our work aims to explore strategies for leveraging existing code generation models and expanding their functionality to encompass Low-Resource natural languages for prompting.

Dataset Overview



Prompt	Code Snippet
Выделить дробную часть числа `num`	math.modf(num)[0]
Выделить дробную часть числа `num`	num % 1
Вычислить сколько процентов составляет число `part` от числа `whole`	100 / whole * part
Сделать заглавной первую букву каждого слова в строке `s`	s.title()
Измерить время выполнения `time_work` функции `func`	t = time.time()\n func()\n time_work = time.time()-t
Найти числа в строке `s`	re.findall('(\d+)', s)
Ввести кортеж чисел `my_tupels` с клавиатуры в качестве разделителя использовать `sep`	my_tupels = tuple(map(int, input().split(sep)))

Записать число `nums` в файл `file_path`	file_1 = open(file_path, "w")\n file_1.write(str(num))
	file_1.close()

Methodology

Proposed Methods

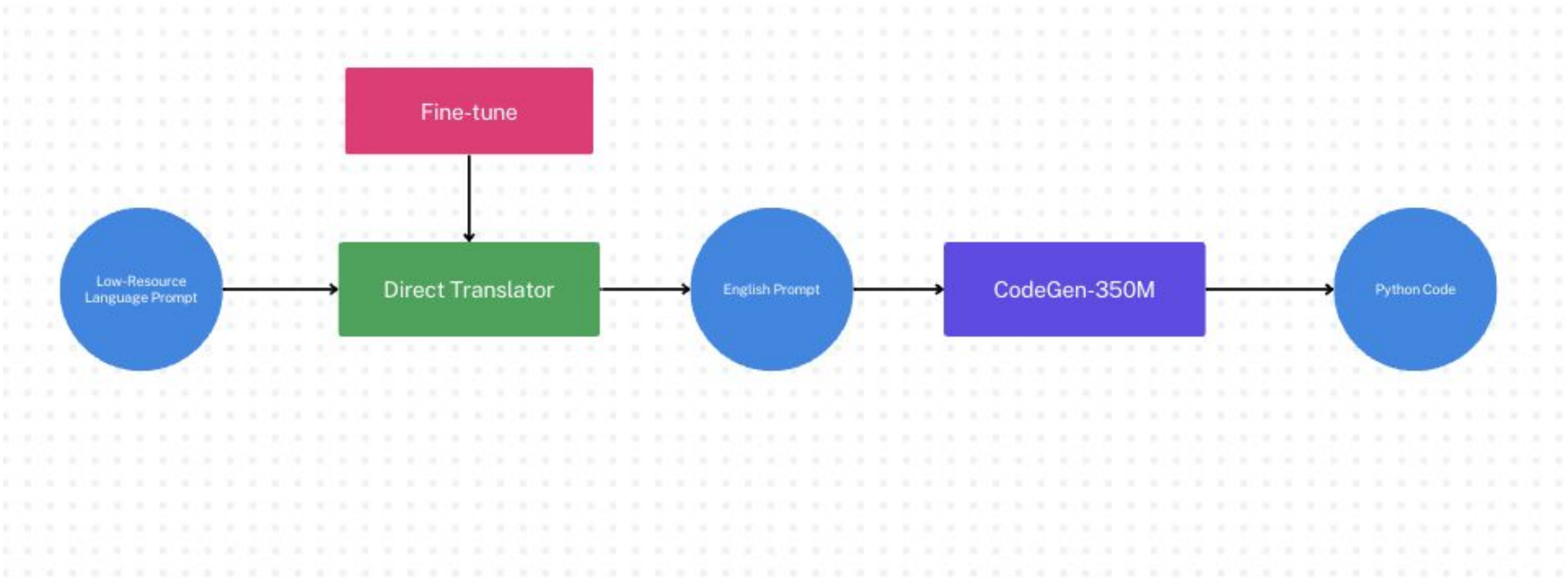
- **Direct Translation -** Use a Seq2Seq or Large Language Model for converting the Low-Resource Language prompt to English, and pass it through the code generation model.
- **Indirect Translation -** Use a “Pivot” Language. Convert the Low-Resource Language prompt to the “Pivot” Language and then convert the “Pivot” Language prompt to English. Finally, pass the English prompt through the code generation model.
- **Fine-tuned Translator -** Use the same translator used previously for **Direct Translation**, but fine-tune it using LoRA on synthetic, model-based Low-Resource - English coding prompt pairs.

Evaluation Metrics

- **Perplexity -** A metric measuring the uncertainty of a language model's predictions, where lower values indicate better performance.
- **Pass @ k -** A metric evaluating code generation models by considering whether the correct code is present within the top-K generated candidates, enhancing evaluation beyond single-best metrics.

Conclusion

Proposed Architecture



References

- CodeGen: An Open Large Language Model for Code with Multi-Turn Program Synthesis
- Evaluating Large Language Models Trained on Code
- MCoNaLa: A Benchmark for Code Generation from Multiple Natural Languages
- CS11-711 Advanced NLP, Lecture - Multilingual NLP
- Improving LLM Code Generation with Grammar Augmentation