Title: Maintaining Tax Prep Software with Large Language Models
Authors: Varsha Dewangan (CU Boulder)*, Sina Gogani-Khiabani (UT El Paso)*, Nina Olson (Center for Taxpayer Rights), Ashutosh Trivedi (CU Boulder), and Saeid Tizpaz-Niari (UT El Paso).
Contact: Saeid Tizpaz-Niari (saeid@utep.edu)

Abstract

As the US tax law evolves to adapt to ever-changing politico-economic realities, tax preparation software plays a significant role in helping taxpayers navigate these complexities. The dynamic nature of tax regulations poses a significant challenge to accurately and timely maintaining tax software artifacts. The state-of-the-art is time-consuming and error-prone as it involves manual code analysis combined with an expert interpretation of tax law amendments. We posit that the rigor and formality of tax amendment language, as expressed in IRS publications, makes it amenable to automatic translation to executable specifications (code). This research leverages recent advances in large language models (ChatGPT, Llama) to extract code differentials from IRS publications and automatically integrate them with the prior version of the code to automate tax prep software maintenance.

Summary

Background. The ever-growing complexity of tax law and policies has significantly increased the role of tax preparation software in navigating the intricacies of legal accountability and compliance. The use of tax software is increasing, and in 2020, over 72 million people prepared their taxes without the help of tax professionals, a 24 percent increase from 2019 [3]. For example, the IRS provides a free tax filing service (IRS Free File) for low-income taxpayers. However, as the tax law gets updated, maintaining the compliance and trustworthiness of tax prep software is challenging. The authors, in their prior works [2], showed that open-source tax prep software often fails to correctly update their code for new tax legislation.

Motivation. Following the new tax legislation and the IRS publications of new regulations; the tax software needs to be updated to reflect the changes in the software artifacts. However, as the tax law has evolved over different years, updating the corresponding software manually is error-prone and tedious. We study the following research question: *can we leverage recent breakthroughs in AI, in particular with pre-trained large language models, to assist software developers in automatically updating the implementation of tax law in software artifacts such as tax preparation software?* If feasible, our approach not only expedites the update process but also enhances the accuracy of new software in compliance with the evolving tax laws.

Objectives. The core objective of this project is to evaluate the **efficacy** and **trustworthiness** of pre-trained large language models, such as ChatGPT, in updating tax software to comply with new tax legislation as outlined by IRS publications. This investigation seeks to determine the capability of AI to autonomously reformulate software code, ensuring that it is following the amended tax regulations, thereby providing a measure of the practical applicability of AI in the domain of tax software administration.

Methodology. Figure 1 shows the proposed framework for updating tax software after the IRS tax policies are amended for a new tax season. The framework involves a cyclical process beginning with the LLM analyzing the previous year's tax software code and the latest tax policy updates from IRS publications. Then, it generates the updated tax software via various prompting that provides guidance and contexts to the LLM model (e.g., ChatGPT 3.5) in the process. Since the AI-generated code might be faulty, our framework adapts metamorphic testing [2] to validate the correctness of updated software. Metamorphic testing [1] is a software testing paradigm that tackles the oracle problem, that is the problem of unavailability of ground truth (e.g., the ground truth of tax returns is often unknown). In the metamorphic testing paradigm, the correctness of software on a tax profile does not require knowing the "ground truth" (i.e., tax return); rather, the correctness can be validated by comparing the outputs of software for that tax profile with the output of a slightly *metamorphosed* one. As an example, we can test software to make sure that the following metamorphic relation holds true: *For any two individuals that differ only in their disability status (e.g., blindness), the federal tax benefits of the disabled individual must be greater than or equal to that of the individual without a disability due to the higher standard deductions for disabled taxpayers.*

*First two authors contribute to this proposal equally.

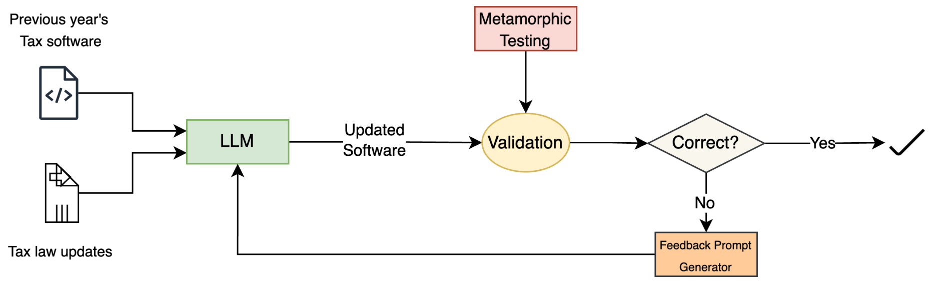

Figure 1: AI-assisted framework to update tax software following the updated tax policies.

If we find that updated software is passing the validation stage without any failed test cases; we deem the update correct and return it to the user. On the other hand, if it fails on one or more test cases, we use the Feedback Prompt Generator (FPG) to guide the LLM model to refine the generated tax software such that the updated software in the next iteration can pass these test cases. We iterate our process multiple times until either the tax software passes all the test cases, or a time-out occurs.

Results. Table 1 shows the preliminary results. The accuracy of updated software is shown in two domains: 1) calculating tax brackets for various filing statuses and tax rates and 2) computing earned income tax credits (EITC) based on income, age, filing status, number of children, and disabilities. Our results show that while the LLMs fail to generate a correct code without providing the tax law documents and prior software; it starts generating the correct software once such information is provided. However, we also notice that as the scenarios get complicated (e.g., updating code for both tax bracket computations and EITC), LLMs fail to generate correct code even after providing the tax law documents and prior tax software. However, our preliminary results do not provide feedback on failed test cases to the LLMs (we generate the code once and test its correctness). Our next goal is to provide the failed test cases as feedback to the LLMs and enable them to generate the correct code in multiple steps. In addition, we plan to explore other domains such as child tax credits, education benefits, etc in our full paper submissions.

Table 1: Preliminary Results. **TBC**: Tax Bracket Calculations, **EITC**: Earned Income Tax Credit, **SD**: Standard deductions of the elderly and blind, **I**: Income, **A**: Age, **FS**: Filing Status, **#C**: Number of children, **D**: Disability, **S**: Single, **MFJ**: Married Filing Jointly.

Domain	**AI Prompts**	**Tax Law**	**Prior Software**	**Outcome**
Tax Brackets (2020)	Generate Code for TBC (S)	Not Given	Not Given	Correct except for tax rates 35%+37%
Tax Brackets (2020)	Generate Code for TBC (S+MFJ)	Not Given	Not Given	Correct except for tax rates 35%+37%
Tax Brackets (2021)	Generate Code for TBC (S)	Not Given	Given (2020)	Correct
Tax Brackets (2021)	Generate Code for TBC (S+MFJ)	Not Given	Given (2020)	Correct
Tax Brackets (2021)	Generate Code for TBC (S+MFJ+SD)	Given	Not Given	Correct
EITC (2020)	Generate Code for EITC based on (I, A, FS, #C, D)	Not Given	Not Given	Failed on cut-offs and for single (S)
EITC (2020)	Generate Code for EITC based on (I, A, FS, #C, D)	Given	Not Given	Correct
EITC (2021)	Generate Code for EITC based on (I, A, FS, #C, D)	Not Given	Given	Failed on cut-offs and credit amounts

References

[1] S. Segura, G. Fraser, A. B. Sanchez, and A. Ruiz-Cortés. A survey on metamorphic testing. *IEEE Transactions on Software Engineering*, 42(9):805–824, Sept 2016.

[2] Saeid Tizpaz-Niari, Verya Monjezi, Morgan Wagner, Shiva Darian, Krystia Reed, and Ashutosh Trivedi. Metamorphic testing and debugging of tax preparation software. In *2023 IEEE/ACM 45th International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*, pages 138–149, 2023.

[3] US.IRS. Filing statistics for week ending december 11 2020. filing-season-statistics-for-week-ending-december-11-2020. online.