Assessing User Perception and Utility of Few-Shot Learning for Model Completion: A User Study Protocol

Meriem ben Chaaben[†], Istvan David[†], Lola Burgueño[§], Houari Sahraoui[†]

[†]DIRO – Université de Montréal, Canada
[§]University of Málaga, Spain

Software Engineering Research Group

Version: 0.1.
March 14, 2023.

Abstract

This document defines the protocol for an empirical study on model completion by few-shot prompt learning.

Keywords

empirical research, user study, recommendation systems, model-driven engineering, software engineering.

Contents

1 Introduction

Model-driven engineering (MDE) is a frequently used development technique that helps managing the complexity of the problem at hand by the power of abstraction. Models, through the right domain-specific views close the gap between the problem at hand and level of human reasoning. The time to correctly model the problem at hand is proportional to the complexity and size of the problem. Similar to text editing and programming, modeling activities can benefit from high-quality completion mechanisms that might shorten the time required to carry out a task, and improve its quality.
Generating high-quality recommendations can be achieved by analyzing a vast number of models and learn concepts and associations from this training set. Unfortunately, as modeling languages tend to be highly specialized to the domain at hand, training data is scarce and does not allow for training a high-quality recommender logic. Previous work [1] has outlined the benefits of employing few-shot learning in generating recommendations for model completion.

Aim and scope

This study aims to identify the user-facing effects of few-shot learning in model completion generation.
In order to mitigate any threats to validity, first, we investigate whether few-shot learning changes the quality of recommendations, as measured by an objective quantitative metric, such as precision which quantifies the number of correct positive predictions made. (ratio of correctly predicted positive examples divided by the total number of positive examples that were predicted)
Then, we investigate the utility of the recommendations through the viewpoint of the end-user. We investigate how few-shot learning changes user behavior, and how users perceive recommenders with few-shot learning in the background.

2 Study design

The goal of this study is to understand how few-shot learning can improve the measured and perceived utility of model completion mechanisms. That is, in terms of the Goal-Question-Metric perspectives [2]:

Purpose	Observe and analyze
Issue	the perceived utility of
Object	few-shot learning
Context	on model completion recommendations
Viewpoint	from the end-user's point of view.
Question	Quality of recommendation
Metrics	Objective and quantitative: correctness (recall), etc
Question	User behavior
Metrics	Objective and quantitative: time to complete, creativity (divergence from mean editing profiles), etc
Question	User-perceived utility
Metrics	Subjective and qualitative: usefulness of recommender mechanisms, confidence in editing, etc

Table 1: The goal of this study.

2.1 Research questions

Based on the goals, we define the following research questions.

RQ1. *How does few-shot learning affect the* ***quality*** *of recommendations?*
Although not the main focus of this work, we briefly assess the quality of recommendations to mitigate threats to the validity of RQ1 and RQ2, and to better contextualize our results.

RQ2-A. *How does few-shot learning affect the* ***performance*** *of users of modeling tools?*
By answering this research question, we aim to identify objectively measurable changes in the users' behavior.

RQ2-B. *How does few-shot learning affect the* ***modeling experience****, as perceived by the users?*
By answering this research question, we aim to understand what attitude does few-shot learning augmented model completion elicit from users.

2.2 Variables

Here, we define the variables by which we answer the research questions.

2.2.1 RQ1 – Objective and quantitative

Measuring recall in this context would require a definition of relevant items and a comparison between the recommended items and a set of ground truth items. However, this is not feasible since we are dealing with large and diverse sets of items and users.
We rely on the usual definition of **precision**: `Precision = TruePositives / (TruePositives + FalsePositives)`, and apply it for the following elements.
We calculate precision for the following elements.

- Concept-level precision
  - Are we able to spot concepts already defined in the description of the domain to model? TODO: this has to be discussed again
  - Number of related concepts (exact matching or similar matching)
  - PS: For similar matching; We plan to compare manually the concepts in the final model.
- Attributes
  - Are the obtained results relative/ correct.
  - Note: We can evaluate both name and type.
- Associations. (per type)
  - Note: We can evaluate name, type, target, source...
  - Some patterns are there in the obtained design?
  - Some basic rules are respected?

Temporal precision: we calculate precision upon generating recommendations and we observe if there is an improvement as the model is being gradually built up.
Eventual precision: we calculate precision for the overall model.
TODO: Do these make sense? If yes, we need to formalize them. Let's discuss.

2.2.2 RQ2-A – Objective and quantitative measures

- Time to complete task
  - We measure the time elapsed between starting and ending the experiment for each Participant. We infer this information from the log files we generate during the experiment. See Section 2.3.3 for details about the logging mechanism and format.
- Diversity and Creativity
  - Our objective is to assess how the recommendation tool affects the creativity of participants. We plan to do this by comparing the sequences of operations performed by participants and evaluating the degree to which these sequences differ. By analyzing these differences, we can determine the level of creativity displayed by the participants and determine if the tool is limiting or enhancing their creative and diverse output.

2.2.3 RQ2-B – Subjective and qualitative measures

TODO: find a comprehensive taxonomy for this

- User properties. *I felt...*
  - Confident (+)
  - Confused (-)
  - Creative (+)
  - Guided towards a good solution (+)
  - Productive (+)
- Tool properties. *I found the tool...*
  - Easy-to-use (+)
  - Intuitive (+)
  - Good fit for the purpose (+)
  - Slow (-)
  - Visually appealing (+)
- Recommendation properties. *I found the recommendations...*
  - Complete (+)
  - Confusing (-)
  - Correct (+)
  - Credible (+)
  - Inconsistent (-)
  - Intuitive (+)
  - Stimulating (gave me ideas about new concepts) (+)
  - Useful (+)
  - Well-timed (appeared right when I expected them) (+)

2.3 Hypotheses

A collection of hypotheses is assessed in order to provide answers to the research questions.

RQ1. MBC ▶*in ICSE paper we started the evaluation of the correctness of our approach, the use of gpt3 to recomend concepts; associations names and types, and attributes.)*◀

RQ2. We expect that

Participants are divided into groups and will try out all three modes, allowing for a comprehensive assessment of each mode's capabilities. Each group will complete three tasks that involve modeling different domains while using one of the modes. This provides valuable insights into the strengths and weaknesses of each mode and helps identify areas for improvement. The results of the user study will be analyzed and used to enhance the product, making it more user-friendly and effective for all users.
Participants are divided into groups randomly to ensure a fair and unbiased assessment of the prediction modes.

2.3.1 Recommender mechanisms

		How?	
		Manual	Automated
When?	During exercise	MD	AD
	End of exercise		AE

Table 2: Recommender mechanisms

1. Recommendation On Request. (**RR**) The designer asks for a specific type of prediction (Looks for potential attributes, concepts or associations)
2. Recommendation On Triggers (**GR**); While modeling, and as things progress, concepts and operations are suggested. Adding elements to the canvas triggers the suggestion of elements that the implemented algorithm finds relevant.
3. Recommendation at the End (**FR**): Evaluate and suggest what is missing at the end.

2.3.2 Domains

The plan is to carefully select three diverse domain models of varying sizes from the provided list, encompassing multiple disciplines. The selected models will be presented to the participants with comprehensive descriptions (Meta stories). Participants will then be tasked with designing class diagrams utilizing the implemented proof-of-concept.

One Domain model will be used as an explanatory example in the demo.

- Class Diagram for Online Shopping
- Class Diagram for Hospital Management System
- Class Diagram for Banking System
- Class Diagram for Library Management System
- Class Diagram for Hotel Management System MBC ▶*Check HotelManagementDescription.tex*◀

		Domain			
		D1	D2	D3	*Sample size*
Recommender mechanism	MD	G1 (n = 5)	G2 (n = 5)	G3 (n = 5)	15
	AD	G2 (n = 5)	G3 (n = 5)	G1 (n = 5)	15
	AE	G3 (n = 5)	G1 (n = 5)	G2 (n = 5)	15
	Sample size	15	15	15	

Table 3: Experimental design

2.3.3 Logging

To achieve this, we employ a logging system that tracks all user operations and decisions, along with the corresponding timestamps of when they occurred. We highlight the difference in time between the modes.

2.4 Experimental objects

Describe editor.

2.5 Participants

We hire fifteen participants, each group of 5 will perform three different modeling tasks. We make sure that the three groups test the three different modes and report the results.
The participants are selected based on their familiarity with object-oriented modeling and the use of similar tools in the past. To ensure a diverse range of perspectives, the participants are carefully selected to have a range of technical abilities, from beginner to advanced, to ensure that the editor is evaluated in a manner that accurately reflects its potential impact on a diverse user base. It is important to note that gender does not play a role in the selection criteria for participants in this user study. The goal is to gather diverse perspectives to ensure that the editor is effective and usable for a wide range of users.

3 Conducting

The experiment is conducted by the following steps.

1. The Participant fills in the input survey. (Section 3.1)
2. If the Participant qualifies for the experiment, they are invited to the experiment. The experiment is conducted in a computer laboratory where the setup to complete the modeling task is prepared. The setup includes a computer and the description of tasks.
3. The Participant watches an on-boarding video explaining essential information about the experiment and what is expected of the Participant. At least one researcher will be present to provide guidance and further help to the Participant. (Section 3.2)
4. The Participant performs the experimental tasks. (Section 3.3)
5. The Participant fills in the exit questionnaire. (Section 3.4)

3.1 Input survey

Before the experiment, potential participants fill in a questionnaire to allow us to

- understand their background;
- collect relevant (but anonymized) demographic data; and
- assess whether their possess the minimal knowledge required to conduct the experiments.

The questionnaire is available here: and in Appendix B, and will be published in the replication package.

Minimum requirements of participation

The minimum requirements to participate in the experiment are the following:

- Mandatory: Fundamental knowledge of UML Class Diagrams. The participant must be able to understand and interpret basic UML Class Diagrams.
- Preferred: Experience with modeling tools.

Questions

Background

- Primary background
  - Technical (STEM)
  - other

Demographics

- Age
- Gender
- Location (Country)
- Primary interest within computer science (free text)
- Current position
  - Undergrad student
  - Grad student
  - PhD student
  - Postdoc
  - Faculty

Familiarity with UML Class Diagrams

- How experienced are you with UML Class Diagrams?
  - Never used CD before
  - Learned about them
  - Classroom exercise
  - My own project (school or personal)
  - Occasional usage in work
  - Frequent usage in work
- For what problems did you use UML Class Diagrams before?
  - Software modeling
  - Domain modeling
  - Domain-specific languages (through profiles)
- Please, describe in your own words, what do you see in the class diagram in Figure 1. Would suggest any improvements?
  - Minimal threshold is finding: *Car* has only 3 wheels, should be 4.
  - Minimal is finding: *Car.transimission* is a String, Enum would be more appropriate.
  - TODO: Redraw Figure 1 (without the error markers, of course).
- What tools did you use before to design class diagrams?
  - None
  - Analog tools: whiteboard, pen and paper, etc.
  - Digital tools:
    * Eclipse
    * StarUML
    * Enterprise Architect
    * Visio
    * LucidCharts
    * MagicDraw
    * Umple
    * yEd
    * Other: (elaborate / free text)
  - Minimal threshold is: at least one digital tool.

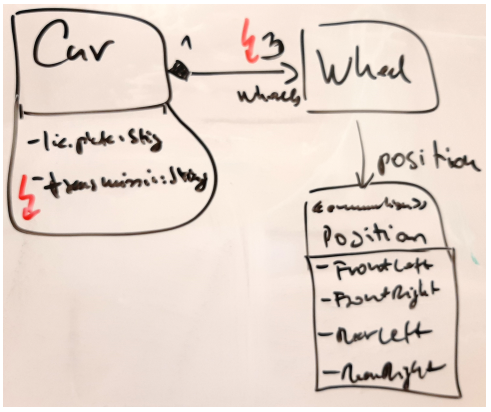

Figure 1: Test model.

3.2 On-boarding

Video

The on-boarding of the Participant starts with a video that explains the experiment and briefly introduces the tool. The video is available here: TODO and will be published with the replication package.

Cheat sheet

The Participant is provided with a printed one-page cheat sheet displaying the most important features of the tool. The cheat sheet is available in Appendix **??**.

Discussion

After the video presentation and handing over the cheat sheet, the Participant can ask for clarification and explanation about the experiment, concepts, tool, etc.
Mandatory readings before the experiment?
To ensure that all the participants have the necessary software installed...
Document explaining how to use the tool.
Reading of these documents is a mandatory task in order to participate in the experiment. Participants who do not read them will be excluded.

3.3 Experiment

We shuffle the order in which the participants perform the different tasks
permissions to record their screens?
upload the final file somewhere?

3.4 Exit survey

The Participant is provided with an exit survey immediately after finishing the experiment. The exit survey measures the user experience in a qualitative fashion, using Likert items covering the variables in Section 2.2.3.
The questionnaire is available here: TODO: online and in Appendix C, and will be published in the replication package.
Ref ; SUS The System Usability Scale: Past, Present, and Future [3]

4 Analysis

References

[1] M. Ben Chaaben, L. Burgueño, H. Sahraoui, Towards using few-shot prompt learning for automating model completion, CoRR abs/2212.03404 (2022). arXiv:2212.03404, doi:10.48550/arXiv.2212.03404.

[2] V. R. Basili, G. Caldiera, H. D. Rombach, The Goal Question Metric Approach, in: Encyclopedia of Software Engineering, Vol. 2, Wiley, 1994, pp. 528–532.

[3] J. R. Lewis, The system usability scale: Past, present, and future, International Journal of Human–Computer Interaction 34 (7) (2018) 577–590. arXiv:https://doi.org/10.1080/10447318.2018.1455307, doi:10.1080/10447318.2018.1455307.
URL https://doi.org/10.1080/10447318.2018.1455307

Appendices

A Cases

A.1 Online shopping system

The primary entity in the domain model for an online shopping system is typically the product being sold, which may have attributes such as name, description, price, and availability. The system also includes information about the customer, such as name, shipping address, and payment information. Another important entity is the shopping cart, which tracks the items selected by the customer for purchase. The system may also track orders, including the order number, order date, and shipping information. Relationships between entities may include a customer adding items to their shopping cart, submitting an order, and receiving confirmation of the purchase. Additional entities in the domain model may include promotions, discounts, and returns. The system may also include features such as reviews and ratings, which allow customers to provide feedback on products and help other customers make informed purchasing decisions.

A.2 Hospital management

The software platform typically includes various concepts, such as patients, medical records, appointments, and medical staff. Patients are a key concept in a hospital management system, with attributes such as name, date of birth, and contact information associated with each patient. Medical records are another important concept, with attributes such as diagnosis, treatment, medication, and medical history associated with each patient. Appointments are another key concept in a hospital management system, with attributes such as date, time. Medical staff are also an important concept, with attributes such as name, job title, and contact information associated with each staff member. Every person in the medical staff is associated with a given department.

A.3 Banking system

The system includes a core banking software that manages customer account information, processes transactions and communicates with various channels, such as ATMs. Accounts are a key concept in a bank system and can be further divided into types such as checking, savings, and money market accounts, among others. Transactions are another important concept in a bank system and can be further divided into types such as withdrawals, deposits, and transfers. Customer information is also an important concept in a bank system, with details such as name, address, and contact information associated with each customer. Bank employee information is another concept, with job titles and employee IDs associated with each employee.

A.4 Library management system

A domain model for a Library Management System includes several key objects and their relationships. It includes information about the library itself, such as its name, address, and hours of operation. The primary entity in the model is the book, which typically has attributes such as title, author, publication date, and ISBN. Another important entity is the borrower, who may have attributes such as name, address, and borrowing history. The system also tracks the library staff responsible for managing the library, with attributes such as name, position, and contact information. Relationships between entities may include a borrower checking out a book, a staff member checking in returned books, and the library system enforcing due dates and fines for late returns. Additional entities in the domain model may include reservations, holds, and interlibrary loan requests.

A.5 Hotel room reservation

One of the primary features of the software system will be managing room reservations. The system should be able to handle various types of reservations, such as single room, double room, suite, and so on. The software system should also be able to handle guest check-ins and check-outs. This involves collecting guest information, assigning rooms, and processing payments. The system should be able to handle

multiple payment methods, such as credit card, cash, and mobile payments. Restaurant management is another important aspect of the hotel domain. The software system should be able to manage restaurant reservations, track table availability, and process food and beverage orders. Housekeeping is another important aspect of the hotel domain. The software system should be able to schedule housekeeping tasks, track the status of cleaning and maintenance tasks, and manage housekeeping staff assignments. Staff management is also a key feature of the system. This involves managing staff schedules.The system should be able to handle various types of staff, such as front desk staff, housekeeping staff, restaurant staff..

B Input questionnaire

TODO

C Exit questionnaire

TODO