Code Completion Context Collection Optimization Challenge: Technical Details

I. Competition Overview

The goal of our competition is to find the best possible strategy for collecting the context from the entire code base for the fill-in-the-middle code completion task. We provide a set of open source projects with predefined caret positions (*aka* completion points) and ground-truth completions. We invite the participants to implement their own context collector that yields the best results on average with three strong code completion models: a state-of-the-art proprietary model for code, such as Codestral by Mistral AI [1], a popular open-source model like Qwen2.5-Coder [2] or Code Llama [3], and our own model called Mellum [4].

We plan to run the competition in an open and transparent manner and publish the intermediate and final results so that the research community can benefit from the ideas gathered during this competition. In addition, we may apply the ideas from the submitted solutions to improve the quality of code completion in our IDEs, thus making sure that the results are also used in practice.

The competition consists of two tracks with the same problem definition, but with different target programming languages and the corresponding datasets. The first track is focused on Python, which is a popular target for many novel AI-based programming assistance techniques due to its very wide user base. The second track is focused on the Kotlin programming language.[1] Kotlin has historically had good support in JetBrains products but has received less interest in the research community. We invite the participants to submit to both tracks. We are particularly interested in universal solutions that can accommodate both a dynamically-typed (Python) and a statically-typed (Kotlin) programming languages.

II. Evaluation

Since the task in this competition is to implement only the context collector (Figure 1), we will use the following evaluation protocol:

1) the participant submits the collected context for each completion point (not the actual neural completion);
2) our competition platform accepts the submission and transforms each context into a model-specific prompt for each of the three models;
3) the platform requests completions, receives the results, returns the evaluation scores for each of them, and computes the average;
4) the scores are shown in the leaderboard on the platform (public test);

[1]https://jetbrains.com/kotlin

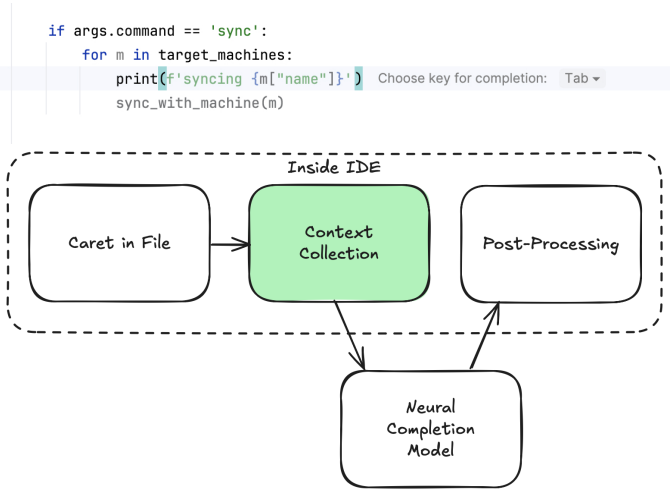

Fig. 1. When the completion is requested, the IDE collects the context in the caret position and creates the corresponding prompt for the neural completion model. The model's output is then post-processed and shown as the suggested completion. In our competition, we want to find the best possible approach for context collection (colored block) while assuming that all the other moving parts remain unchanged.

5) by the end of the competition, we review and run the best solutions from the leaderboard on a separate held-out dataset to determine winners (private test).

Our research by Evtikhiev et al. [5] shows that the chrF evaluation criterion [6], which is popular in machine translation, is currently one of the best indicators of code completion quality thanks to its interpretability and flexibility. chrF is defined similarly to the F-score criterion from the information retrieval field:

$$\mathrm{chrF} = 2\frac{\mathrm{chrP} \cdot \mathrm{chrR}}{\mathrm{chrP} + \mathrm{chrR}},$$

where chrP is the percentage of n-grams in the suggestion that have a counterpart in the ground-truth completion, and chrR is the percentage of character n-grams in the ground-truth completion that are also present in the suggestion. We are planning to use chrF as the evaluation criterion in our competition everywhere, including the leaderboard (public test).

Soon before the end of the competition, we will invite the authors of the top 10 solutions to submit to us a working container image implementing their approach (we call it the *reproduction stage*). We will run these containers on our machines on the *private test* dataset with exactly the same protocol as the public part. The solutions maximizing chrF on

the private test subset of our competition dataset are considered the winners of our challenge.

As a follow-up after the proposed competition, we plan to implement the ideas from the submitted solutions in one of our IDEs and run an A/B test to study its impact on the real users after the competition. If we observe a statistically significant improvement in the percentage of total characters accepted [7] without a degradation of computational performance and user experience, it will become the main context collection strategy in our IDEs. This gives the participants a chance to contribute to improving the experience of millions of users.

III. Awards

We will offer three kinds of awards besides the certificates of winning or attendance: monetary prizes and JetBrains product licenses.

First, we will offer **monetary prizes** for the first three places in each of the two competition tracks: \$3000 for the 1st place, \$2000 for the 2nd place, and \$1000 for the 3rd place. We will also additionally cover the registration fee for one representative of each team from the top 3 who will present their solution at the workshop session. The entire prize pool is $2 \times \$6000 = \$12{,}000$ plus conference registration fees for the winning team representatives.

Second, our collaborators from **Mistral AI will provide the winning teams with API keys** for all the models available on La Plateforme,[2] which they can use for any purpose they choose.

Last but not least, we will gift a **yearly license of JetBrains All Products Pack** to each member of the three winning teams.[3] This pack contains 12 IDEs, 3 extensions, and 2 profilers; its retail price is \$289 for individual use.

IV. Implementation Details

We are hosting our competition on the EvalAI platform [8].[4] We have prepared the Python track for internal testing (Figure 2; we are currently setting up the one for Kotlin. Below we provide details on the dataset, infrastructure, and privacy concerns.

A. Dataset

We build our competition on top of our existing benchmark called Long Code Arena (LCA) [9] that includes the single-line repository-level code completion task. This dataset mimics the actual way developers write code by using git commit histories to separate the file being completed and the repository snapshot used for context collection. The dataset prepared for this competition is built from scratch and has several important differences from LCA. Namely, it features multi-line completion with fill-in-the-middle instead of prefix-based completion, support for Python and Kotlin programming languages, and a separation between training, public test set, and private test

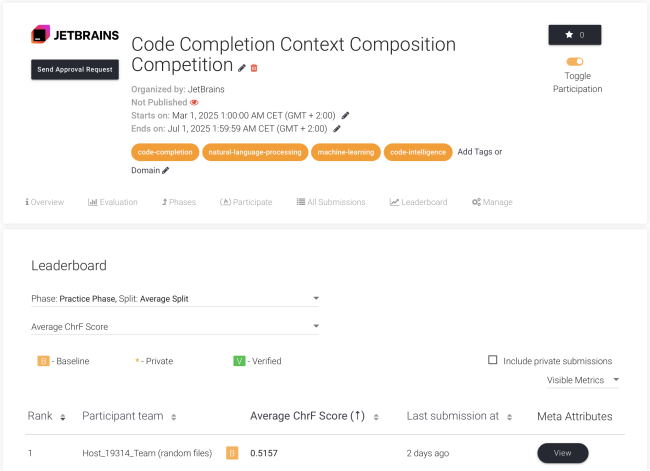

Fig. 2. An example of a baseline submission for the Python track in the EvalAI platform. We have not published the competition yet to keep it private, but we can provide the ASE workshop chairs with access to the competition instance if needed.

set. We produced these multi-line completions using the in-house code analysis tools that are used in our IDEs, but we believe the results of this processing are not specific to our products and can be useful for a broad range of applications.

Since we use open-source code, we need to prevent data leaks by separating the intermediate and final results by following the public/private test approach from our competition at WSDM Cup 2023 [10]. Our training dataset for Python was built on the same collection of repositories as in our LCA benchmark [9] and contains 4,057 completion points from 84 repositories in 3,714 revisions. Our test dataset for Python has 3,268 completion points from 73 repositories in 3,038 revisions; the test dataset for Kotlin has 3,911 completion points from 94 repositories in 3,468 revisions. We use only large repositories of code like IPython[5] and dukat[6] in our datasets to be able to employ sophisticated context collectors on realistic code bases. We separate the test dataset for each language into two parts: *public test* and *private test*. The *public test* dataset will be available immediately at the start of the competition. We will release the private part after the competition ends. The dataset separation will be performed once before the start of the competition, but after we finish the internal testing. The datasets will not change during the course of the competition.

As an example for the review, we uploaded a sample of the aforementioned training dataset for Python to a temporary folder on Google Drive.[7]

B. Competition Infrastructure

For this competition, we decided to use the EvalAI platform. Since our competition requires collecting contexts for the

[2]https://docs.mistral.ai/deployment/laplateforme/overview/
[3]https://www.jetbrains.com/store/?section=personal&billing=yearly
[4]https://eval.ai/

[5]https://github.com/ipython/ipython
[6]https://github.com/Kotlin/dukat
[7]https://drive.google.com/drive/folders/1kv3tANJRxQ8ltuQNfZW5snQWdYyIDJ8C?usp=sharing

given set of completion points, and our evaluation protocol implies calling the neural completion models, we implemented calling these models on the platform side with API keys at our expense. We will release a convenient starter pack for the participants to allow rapid start and simplify prototyping.

C. Privacy Concerns

Our dataset only contains permissively licensed code. We have requested a thorough legal review of the dataset release by our legal counsel and got their permission to release the dataset. We will publish the final submissions of the participants under the CC BY 4.0 license.[8] During the public test stage of the competition, the participants need to submit data files without their code. However, for the reproduction (private test) stage, we will ask the participants to share container images and their source code to run on our machines. The participants might use any content in the provided dataset. We will offer means in the competition public forums for the participants to ask for the permission to use external tooling, such as Web search, in their solution.

V. Host information

Dr. Dmitry Ustalov. Dmitry leads the JetBrains AI Evaluation team that is responsible for offline and online evaluation of machine learning models, including the ones for code completion. His research interests include natural language processing, datasets, and benchmarks. His studies are published at NeurIPS, COLI, ACL, COLING, SIGIR, and WSDM; he organized machine learning competitions at CLEF 2024, WSDM Cup 2023, and at ACL-sponsored series of TextGraphs workshops in 2024, 2022, and 2019–2021.

Egor Bogomolov. Egor leads the Machine Learning Division at JetBrains Research. His work focuses on applying machine learning to software engineering tasks, with a particular emphasis on analyzing, modeling, and understanding source code. His research contributions center on benchmarking and evaluating ML4SE models, as well as adapting language models to effectively process programming languages and software projects.

Alex Bezzubov. Alex is Research Engineer at JetBrains Research. He co-organized several scientific and industrial events, including tracks at conferences, tutorials, and workshops.

Yaroslav Golubev. Yaroslav is the Research Administrator at JetBrains Research. As a part of his duties, he helps organize events, conferences, gatherings, etc. Yaroslav served as the Proceedings chair at the International Workshop on Refactorings in 2021, launched the Workshop on Integrated Development Environments in 2024 and 2025, and also helped prepare a tutorial at the Technical Symposium on Computer Science Education in 2025. In his own research, Yaroslav studied the application of machine learning to refactorings and the ways programmers employ AI assistants.

Evgeniy Glukhov. Evgeniy is an ML Researcher at JetBrains Research. He is the main contributor to the project-level code completion task of the LCA benchmark [9]. His work focuses on long context utilization and in-context learning for software projects.

Georgii Levtsov. Georgii is an intern at the AI Evaluation team at JetBrains, who is responsible for revision-aware source code acquisition and analysis. He is also a student at the Neapolis University Pafos. He has extensive experience in participating in and conducting olympiads in mathematics and programming.

Dr. Vladimir Kovalenko. Vladimir is Head of External Relations at JetBrains Research. His responsibilities include overseeing community outreach and external research collaborations. Vladimir is a prominent member of the Software Engineering research community with a solid academic track record.

[8] https://creativecommons.org/licenses/by/4.0/

References

[1] Mistral AI, "Codestral," 2025, Mistral AI. [Online]. Available: https://mistral.ai/news/codestral

[2] B. Hui *et al.*, "Qwen2.5-Coder Technical Report," 2024. [Online]. Available: https://arxiv.org/abs/2409.12186

[3] B. Rozière *et al.*, "Code llama: Open foundation models for code," 2024. [Online]. Available: https://arxiv.org/abs/2308.12950

[4] JetBrains, "Introducing Mellum: JetBrains' New LLM Built for Developers," 2024, JetBrains Blog. [Online]. Available: https://blog.jetbrains.com/blog/2024/10/22/introducing-mellum-jetbrains-new-llm-built-for-developers/

[5] M. Evtikhiev, E. Bogomolov, Y. Sokolov, and T. Bryksin, "Out of the BLEU: How should we assess quality of the Code Generation models?" *Journal of Systems and Software*, vol. 203, p. 111741, 2023.

[6] M. Popović, "chrF: character n-gram F-score for automatic MT evaluation," in *Proceedings of the Tenth Workshop on Statistical Machine Translation*. Lisbon, Portugal: Association for Computational Linguistics, 2015, pp. 392–395.

[7] O. Dunay *et al.*, "Multi-line AI-Assisted Code Authoring," in *Companion Proceedings of the 32nd ACM International Conference on the Foundations of Software Engineering*, ser. FSE 2024. Porto de Galinhas, Brazil: Association for Computing Machinery, 2024, pp. 150–160.

[8] D. Yadav *et al.*, "EvalAI: Towards Better Evaluation Systems for AI Agents," 2019. [Online]. Available: https://arxiv.org/abs/1902.03570

[9] E. Bogomolov, A. Eliseeva, T. Galimzyanov, E. Glukhov, A. Shapkin, M. Tigina, Y. Golubev, A. Kovrigin, A. van Deursen, M. Izadi, and T. Bryksin, "Long Code Arena: a Set of Benchmarks for Long-Context Code Models," 2024. [Online]. Available: https://arxiv.org/abs/2406.11612

[10] D. Ustalov, N. Pavlichenko, S. Koshelev, D. Likhobaba, and A. Smirnova, "Toloka Visual Question Answering Benchmark," 2023. [Online]. Available: https://arxiv.org/abs/2309.16511