Cache me if you Can: an Online Cost-aware Teacher-Student framework to Reduce the Calls to Large Language Models

Ilias Stogiannidis[†,‡] **Stavros Vassos**[‡] **Prodromos Malakasiotis**[†] **Ion Androutsopoulos**[†]
[†]Department of Informatics, Athens University of Economics and Business
[‡]Helvia.ai
{stoyianel, rulller, ion}@aueb.gr, stavros@helvia.ai

Abstract

Prompting Large Language Models (LLMs) performs impressively in zero- and few-shot settings. Hence, small and medium-sized enterprises (SMEs) that cannot afford the cost of creating large task-specific training datasets, but also the cost of pretraining their own LLMs, are increasingly turning to third-party services that allow them to prompt LLMs. However, such services currently require a payment per call, which becomes a significant operating expense (OpEx). Furthermore, customer inputs are often very similar over time, hence SMEs end-up prompting LLMs with very similar instances. We propose a framework that allows reducing the calls to LLMs by caching previous LLM responses and using them to train a local inexpensive model on the SME side. The framework includes criteria for deciding when to trust the local model or call the LLM, and a methodology to tune the criteria and measure the tradeoff between performance and cost. For experimental purposes, we instantiate our framework with a GPT-4 teacher and a k-NN student, using an intent recognition dataset. Experimental results indicate that significant OpEx savings can be obtained with slightly lower performance.

1 Introduction

Prompting pre-trained Large Language Models (LLMs) aligned to follow instructions (Ouyang et al., 2022; Köpf et al., 2023) performs impressively well in zero- and few-shot settings. Hence, small and medium-sized enterprises (SMEs) that cannot afford the cost of creating large task-specific training datasets for model fine-tuning, but also the cost of pretraining their own LLMs, are increasingly turning to third-party services that allow them to prompt LLMs. SMEs that provide customer support chatbots, for example, prompt LLMs like GPT-4 (OpenAI, 2023) to detect user intents and drive the chatbot-customer interaction (Ham et al., 2020). The best LLMs, however, currently require a payment per prompting call, and these payments become a significant operating expense (OpEx) for SMEs. Furthermore, customer inputs (e.g., dialog turns) are often very similar over time, hence SMEs end up calling LLMs to handle inputs that may be very similar to inputs already handled by the LLMs in previous (already paid) calls.

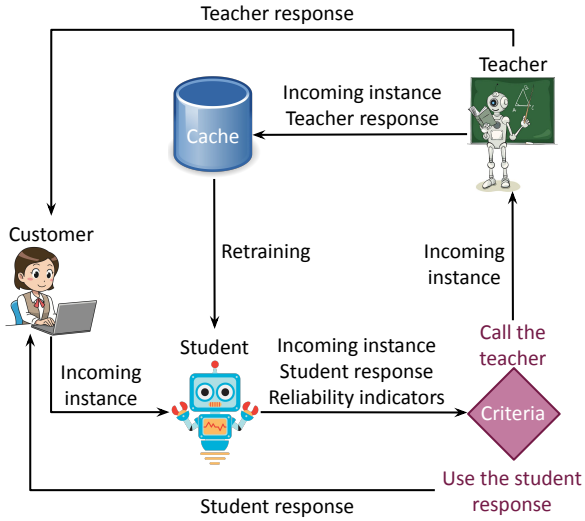

Figure 1: OCaTS architecture.

We introduce the *Online Cost-aware Teacher-Student* (OCaTS) framework that allows reducing the calls to a commercial LLM, treated as a teacher model, by caching its previous responses and using them to train a local inexpensive student model. OCaTS includes criteria for deciding when to trust the student or call the teacher, and a methodology to tune the criteria and measure the tradeoff between performance and cost. Unlike common teacher-student training for knowledge distillation (Hinton et al., 2015; Gou et al., 2021), here the teacher does not train the student on all the available instances (in our case, all the incoming customer inputs). Also, unlike teacher-student approaches to self-training (Mi et al., 2021; Li et al., 2021), the teacher is already reasonably effective (but expensive). In that sense, our work is closer to ac-

tive learning (Settles, 2012; Monarch, 2021), but OCaTS trains the student on labels provided by a teacher LLM, not humans, and there is initially no large pool of unlabeled instances (customer inputs) to select from, as instances arrive online.

OCaTS can be used with any service that allows prompting LLMs, and any kind of local student model. For experimental purposes, we instantiate OCaTS with a GPT-4 teacher and a k-NN student, using an intent recognition dataset from the banking domain. Experimental results indicate that significant OpEx savings can be obtained with only slightly lower performance. For example, our results show that the k-NN student can handle approximately two-thirds of the incoming instances (customer inputs) without calling GPT-4 (Fig. 2, left, red line) for a decrease of less than 0.5 percentage points in accuracy (Fig. 2, middle, red and black lines). OCaTS introduces discounted versions of common evaluation measures (e.g., accuracy) that allow an SME to quantify how much it prefers to lean towards fewer calls or less user frustration (different λ values in Fig. 2).

Our main contributions are: (i) We introduce a general teacher-student framework that helps SMEs reduce the prompting calls to commercial LLMs and the corresponding OpEx costs by caching the responses of the LLMs and training local student models. (ii) We introduce discounted versions of common evaluation measures that allow the SMEs to quantify how much they prefer fewer LLM calls vs. increased user frustration and tune the criteria of the framework that decide when to trust the local student model or call the LLM teacher. (iii) We instantiate the framework with a GPT-4 teacher and a k-NN local student. (iv) We perform experiments on a well-known intent recognition dataset from the banking domain and show that significant cost savings can be obtained with only slightly lower performance. This is a first step towards exploring the benefits of the proposed framework with more datasets, models, and business scenarios.

2 Framework

Architecture: The proposed framework consists of three main components (Fig. 1): a *teacher*, typically a resource-intensive model offering premium results; a *student*, a cost-effective model that is typically much smaller and simpler than the teacher; a *cache*, a repository of incoming instances (e.g., customer requests) that have already been processed by the teacher. We assume that the framework is employed to handle a task for which there is no available large dataset for supervised training, apart from a few incoming instances (possibly a handful per class) annotated with the ground truth (e.g., correct labels). This is a very common case for SMEs that cannot afford the cost of creating large task-specific training datasets, but can easily construct small numbers of demonstration instances. The teacher-student setting is *online*, as every incoming instance is handled at inference time as follows. First, the student is called to handle the instance. Then some student- and task-specific *criteria*, which assess the reliability of the student's output, indicate if the student's output (e.g., label) should be used or if the teacher should be consulted. If the student's output is selected, it is returned as the response to the incoming instance. Otherwise, the teacher is called to handle the instance. In the latter case, the instance along with the teacher's result are stored in the cache. Depending on the type of student, periodic re-training takes place, to update the student with the cached instances.

Instantiations: Section 3 below instantiates the framework with a GPT-4 teacher and a distance-weighted k-NN classifier with MPNet-based (Song et al., 2020) vector representations of incoming instances, for a single-label intent recognition task. The instantiation includes criteria that consider (i) the entropy of the probability distribution (over the label set) produced by the k-NN classifier for the incoming instance, and (ii) the distance of the vector representation of the incoming instance from the centroid of the vector representations of the k nearest neighbors. Consult Nguyen et al. (2022) for other possible criteria. We leave other instantiations of the architecture (with other teachers, students, tasks, representations) for future work.

Discounted evaluation measures: The main goal of the proposed architecture is to reduce the number of calls to the expensive teacher model by caching previous teacher responses and using them to train a local inexpensive student model on the SME side. This introduces a tradeoff between the OpEx cost of calling the teacher and the frustration of the end-users when the less accurate student model is used instead. To quantify this tradeoff, we introduce a *discounted* variant $\hat{\phi}$ of any common evaluation measure ϕ (e.g., accuracy, F1), as follows:

$$\hat{\phi} = \phi - \lambda \cdot \frac{M}{N} = \phi - \lambda \cdot \rho, \qquad (1)$$

where N is the number of incoming instances that

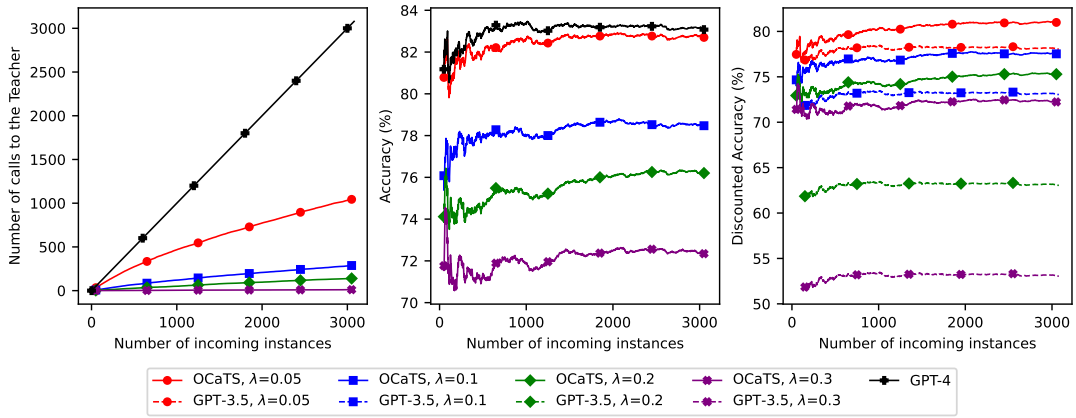

Figure 2: Number of calls to the teacher (left), accuracy (middle), and discounted accuracy (right), using a GPT-4 teacher and a k-NN student, for various λ values. Dashed lines show the discounted accuracy when calling GPT-4 for all incoming instances. The larger the λ the more the SME prefers fewer calls at the expense of increased user frustration. As expected, OCaTS has a better discounted accuracy than always calling the GPT-4 teacher.

have been processed (on which ϕ is measured), M is the number of calls made to the teacher while processing the N instances, $\rho = \frac{M}{N}$ shows for what percentage of the incoming instances we call the teacher, and λ is a scalar specifying how intensively the measure should be discounted. Assume, for example, that the accuracy of the teacher-student combination is $\phi = 0.8$, but that this accuracy is achieved with $\rho = \frac{1}{3}$. If the SME considers this ρ value (which would translate, e.g., to a monthly cost) as costly as a loss of one percentage point of accuracy, then $\hat{\phi} = 0.7$, and Eq. 1 becomes $0.7 = 0.8 - \lambda \cdot \frac{1}{3}$, from which we obtain $\lambda = 0.3$. Larger (or smaller) λ values correspond to cases where the SME considers the same ρ value more (or less) costly in terms of loss of accuracy points. We can also reformulate Eq. 1 as $\delta = \lambda \cdot \rho$, where $\delta = \phi - \hat{\phi}$ shows how much ϕ gets discounted to account for the cost of ρ. Then λ can intuitively be thought of as a currency exchange rate, showing how expensive ρ is in terms of δ (e.g., loss of accuracy in percentage points).[1]

3 Experiments

Dataset: For our experimental analysis, we employ Banking77 (Casanueva et al., 2020), an intent recognition dataset from the banking customer service domain. It includes 13,083 customer messages. The ground truth assigns to each message a single label (intent) from the 77 available. The dataset is divided into training (10,003 instances) and test (3,080) subsets. More statistics in Appendix A.

Few-shot training and development sets: Assuming that an SME can only afford to construct a small number of training instances per class, we use only $3 \times 77 = 231$ instances from the original training set, three per class, as a few-shot version of the training set. These instances and their ground truth labels are included in each prompt call of the teacher as demonstrators (in-context learning); see Appx. B for more details. The 231 instances were manually selected to avoid unclear cases, e.g., similar instances with different ground truth labels. Similarly, we created a few-shot development set of $13 \times 77 = 1,001$ instances from the original training set, for hyperparameter tuning (see below).

Incoming instances and evaluation measure: We use the original test set of Banking 77 as the incoming instances. We repeat each experiment with five random shufflings of the test set and report average scores over the shufflings. We set ϕ to accuracy, since the test set is balanced (Appendix A).

Teacher: We used GPT-4 (OpenAI, 2023) as the teacher, the most capable LLM for few-shot learning tasks at the time. Each prompt included instructions, demonstrators, and the incoming instance to be classified; see Appendix B for details.

Student: A distance-weighted k-NN classifier was used as the student. Vector representations of the incoming instances are generated with a Sentence-Transformer (Reimers and Gurevych, 2019) variation of MPNet (Song et al., 2020).[2] Appendix C provides more information on the distance weighting used. It also shows (Fig. 3) that in a more conventional setting, where a large manually la-

[1] We implicitly assume that the exchange rate λ is constant for all the values of δ and ρ. In practice, it may be different for different ranges of δ and ρ, but we leave this for future work.

[2] We used `gpt-4-0314` and `all-mpnet-base-v2`, in particular, for the teacher and student, respectively.

beled training set is available, the k-NN classifier clearly outperforms GPT-4 in accuracy (92% vs. 82%). Note that for the k-NN student, no retraining (Fig. 1) is necessary, since the cache coincides with the memory of the k-NN classifier. The cache (memory) is initialized with the 3-shot training examples of the classes (231 instances in total).

Criteria: We instantiate the criteria of Fig. 1 with two conditions. Both have to be satisfied for the student's response to be used; otherwise we call the teacher. The first condition is that the cosine distance between the (MPNet-based) vector representation of the incoming message and the *weighted centroid vector* $\mathbf{c}$ of the k nearest neighbors should be less than a threshold $t_{\mathbf{c}}$. Here $\mathbf{c} = \sum_i \hat{w}_i \cdot \mathbf{v}_i$, and $\hat{w}_i = w_i / \sum_{j=1}^{k} w_j$, where w_i is the weight assigned by distance weighting (Appendix C) to the i-th neighbour, and $\mathbf{v}_i$ is the (MPNet-based) vector representation of the neighbour. Intuitively, this condition ensures that the incoming instance is sufficiently close to cached instances.

To define the second condition, let C be the set of the labels (classes) of the k nearest neighbors (hereafter simply neighbors). Let $w_{i,c}$ be the weight (assigned by distance weighting) to the i-th neighbour belonging in class c, and let W_c be the sum of all weights of neighbors of class c, i.e., $W_c = \sum_i w_{i,c}$. We define the probability p_c of each $c \in C$ as $p_c = \frac{\exp(W_c)}{\sum_{c' \in C} \exp(W_{c'})}$. The *entropy* $\mathcal{H}$ of the probabilities p_c of the labels of the neighbors is:

$$\mathcal{H} = -\sum_{c \in C} p_c \log p_c.$$

The second criterion requires $\mathcal{H}_w$ to be less than a threshold $t_{\mathcal{H}}$. Intuitively, it requires the neighbors to agree on the label of the incoming instance.

Hyperparameter tuning: There are three hyperparameters, the number of neighbors k, and the thresholds $t_{\mathbf{c}} \in [0, 2]$, $t_{\mathcal{H}} \in [0, 6.26]$. We fix $k = 5$ as a practical choice considering that there are 3 examples per class initially. For each one of four indicative λ values (0.05, 0.1, 0.2, 0.3), we employ Bayesian optimization on the few-shot development set (Section 3) to determine the optimal combination of the two thresholds that maximize $\hat{\phi}$. More precisely, we use Optuna's (Akiba et al., 2019) implementation of the Tree-Structured Parzen Estimator (TSPE) algorithm (Bergstra et al., 2011) after first performing a 10×10 grid search on the range of values of the two thresholds as a head start. The resulting contour maps and the optimal values of the two thresholds per λ value can be found in Appendix D.

Results: We evaluate OCaTS for each of the four indicative λ values, using the same incoming instances (original test set of Banking 77), and the λ-specific tuned thresholds $t_{\mathbf{c}}$, $t_{\mathcal{H}}$. As illustrated in Fig. 2, OCaTS succeeds in managing the tradeoff between calls to the teacher vs. accuracy: on the left, we see that as the discount factor λ increases, fewer calls to the teacher are made; on the middle, we see how much accuracy is sacrificed for this OpEx relief. In particular, for $\lambda = 0.05$ the accuracy of OCaTS is very close to the accuracy of the GPT-4 teacher, within a margin of 0.37 percentage points (83.05% vs. 82.68% for the entire test set), while calling the teacher for only 1/3 of the incoming instances (1050 out of 3080). For larger values of λ, we see the intended drop in accuracy to achieve an increasingly smaller number of calls to the teacher. Figure 2 (right) also shows that the discounted accuracy $\hat{\phi}$ of OCaTS (solid lines, one per λ value) is always clearly higher than the corresponding discounted accuracy of always calling the GPT-4 teacher (dashed lines), as intended.

4 Conclusions

We introduced OCaTS, an Online Cost-ware Teacher-Student framework that helps SMEs to reduce OpEx costs, by caching the responses of commercial LLMs and training local students. We also introduced discounted versions of common evaluation measures allowing SMEs to quantify the trade-off between LLM calls and user frustration. By instantiating OCaTS with a GPT-4 teacher and a k-NN student and experimenting with Banking77, we showed that the calls to the teacher can be significantly reduced (1/3) with only a slight performance drop (0.37 percentage points).

In the future we plan to experiment with even more realistic datasets and tasks (e.g., question answering), and suggest adaptive policies for λ to allow more OpEx costs at the beginning when the cache is cold and be more selective in the long run. We also plan to enhance OCaTS with indicators that suggest how much we can trust the teacher responses. Finally, we intend to incorporate more financial metrics (e.g., student costs) in the discounted versions of the evaluation measures and study more complex strategies (e.g., game-theoretic, reinforcement learning) to select the thresholds that support the corresponding versions.

5 Limitations

The main scope of this work was to propose a flexible framework (OCaTS) that will allow SMEs to reduce the costs when incorporating commercial LLMs in their solutions. Thus, we considered only one instantiation of OCaTS where the teacher is GPT-4 and the student is a distance-weighted k-NN. Although GPT-4 can in principle be used for zero-shot inference, we considered in-context learning with 231 examples (three examples per class). These examples where manually selected to be diverse and indicative of the corresponding classes. This is realistic to some extent; SMEs often request a small number of examples from their customers but the quality of these examples is not always guaranteed. In addition, the test set of Banking77 is balanced and thus not so realistic; the incoming instances will not always arrive in a uniform way with the respect to their classes. Note also, that to tune t_c and $t_{\mathcal{H}}$ we used a development set, extracted from the training data. Such a development set is not always available in practice, but we used it for the sake of the analysis. Interested SMEs can use our analysis, as a starting point for their applications and reduce the number of trials needed to find suitable values for t_c and $t_{\mathcal{H}}$. Another limitation is that $\hat{\phi}$ takes into consideration only the cost to call the teacher (ρ), and indirectly the frustration of the user, as it can be implied by the performance drop. A more detailed analysis would also incorporate the student cost and other financial metrics possibly with different weights; OCaTS can be easily extended in that direction. Finally, we did not compare against already existing caching libraries, e.g., GPTCache.[3] These libraries are quite simplistic and less flexible than OCaTS which can be used with a variety of teacher-student settings.

6 Ethics Statement

Constantly querying LLMs to solve everyday tasks is not only costly but also has a large energy footprint as well. Our framework aims to alleviate both phenomena. Nonetheless, our study required a significant amount of resources but we believe by making the framework and the analysis publicly available, we will pave the way towards reducing the resources required by SMEs to handle their day-to-day tasks in the long run.

[3] https://github.com/zilliztech/GPTCache

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A Next-generation Hyperparameter Optimization Framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. 2011. Algorithms for Hyper-Parameter Optimization. In *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45, Online. Association for Computational Linguistics.

Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *Int. J. Comput. Vision*, 129(6):1789–1819.

Donghoon Ham, Jeong-Gwan Lee, Youngsoo Jang, and Kee-Eung Kim. 2020. End-to-end neural pipeline for goal-oriented dialogue systems using GPT-2. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 583–592, Online. Association for Computational Linguistics.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the Knowledge in a Neural Network. *ArXiv*, abs/1503.02531.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, Shahul ES, Sameer Suri, David Glushkov, Arnav Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Mattick. 2023. OpenAssistant Conversations – Democratizing Large Language Model Alignment.

Shiyang Li, Semih Yavuz, Wenhu Chen, and Xifeng Yan. 2021. Task-adaptive Pre-training and Self-training are Complementary for Natural Language Understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1006–1015, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Fei Mi, Wanhao Zhou, Lingjing Kong, Fengyu Cai, Minlie Huang, and Boi Faltings. 2021. Self-training

Improves Pre-training for Few-shot Learning in Task-oriented Dialog Systems. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1887–1898, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Robert Monarch. 2021. *Human-in-the-Loop Machine Learning*. Manning Publications.

Vu-Linh Nguyen, Mohammad Hossein Shaker, and Eyke Hüllermeier. 2022. How to Measure Uncertainty in Uncertainty Sampling for Active Learning. *Mach. Learn.*, 111(1):89–122.

OpenAI. 2023. GPT-4 technical report. *ArXiv*, abs/2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Burr Settles. 2012. *Active Learning*. Morgan & Claypool Publishers.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. In *NeurIPS 2020*. ACM.

A Dataset statistics

We provide more details about the Banking77 dataset. Figure 4 shows the distribution of the labels of the original training and test subsets. The training subset exhibits a significant class imbalance, whereas the test subset is balanced. In Table 1, we provide further statistics which, along with the label distribution, support the selection of the dataset as a realistic case for performing our experiments.

B More details about the GPT-4 teacher

To prompt GPT-4, we use the OpenAI chat completion API, which takes as input a *system* message and history of *user* and *assistant* messages and generates a (assistant) message as output. The system message specifies the model's behavior as a chat assistant; we use it to provide instructions for classifying user messages and providing the class in the output-generated assistant message (Fig. 5). For our few shot setting, we provide pairs of user-assistant messages as examples (Fig. 6). The incoming instance to be classified is added as the last user message in the history.

Banking77 Statistics	**Train**	**Test**
Number of examples	10,003	3,080
Minimum length in characters	13	13
Average length in characters	59.5	54.2
Maximum length in characters	433	368
Minimum length in words	2	2
Average length in words	11.9	10.9
Maximum length in words	79	69
Number of intents	77	77

Table 1: Statistics for Banking77.

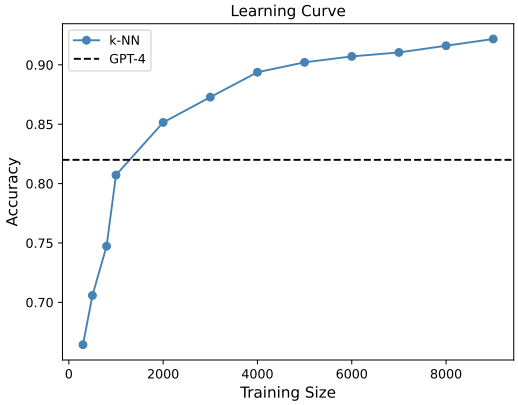

Figure 3: The learning curve of the distance-weighted k-NN student (solid line) when trained on the original training set and evaluated (accuracy) on the few-shot development set. The accuracy of GPT-4 on the few-shot development set is also shown (dashed line).

C More details about the k-NN student

The distance weighting of the k-NN classifier is performed as follows. Let d_i be the cosine distance between the incoming instance and the nearest neighbor i. The weight of the neighbor is $w_i = \frac{1}{{d_i}^2}$. For each class, we sum the weights of the neighbors with this class. We then classify the incoming instance into the class with the largest sum.

Figure 3 shows the learning curve of a distance-weighted k-NN classifier that is trained on the original banking77 training set and evaluated on the few shot development set (Section 3). We observe that with sufficient training data (approx. 1000 instances) the k-NN classifier clearly outperforms GPT-4 in accuracy (92% vs. 82%), which justifies

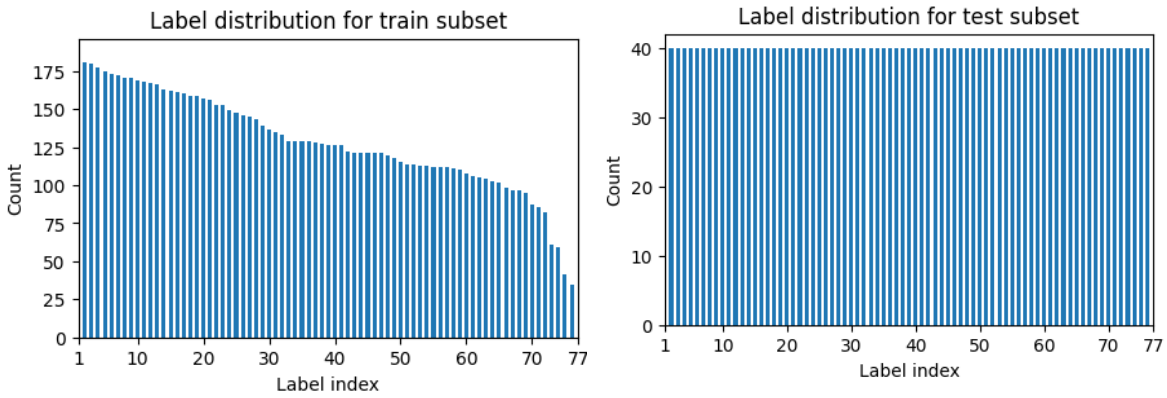

Figure 4: Label distribution of the original train (left) and test(right) subset of Banking77.

```
You are an expert assistant in the field of
customer service. Your task is to help workers
in the customer service department of a company.
Your task is to classify the customer's question
in order to help the customer service worker to
answer the question. In order to help the worker
you MUST respond with the number and the name of
one of the following classes you know.
In case you reply with something else, you will
be penalized.
The classes are:
- activate_my_card
- age_limit
...
```

Figure 5: System message used in the GPT-4 teacher.

our choice to use it as a student in our experiments.

D Hyperparameter Tuning

We tuned the thresholds t_c and $t_{\mathcal{H}}$ anew for each λ value. For each λ value, we first performed a 10×10 grid search in the ranges of values of the two thresholds, to evaluate indicative combinations for the thresholds. These are considered as starting points for Bayesian optimization (Section 3). Figure 7 illustrates the presence of several good points adjacent to the best point, all of which maximize the discounted metric to a significant extent. This shows that several threshold combinations may be considered optimal. In addition, there are large areas, often with a wide range of values for a threshold that are comparable in terms of maximizing the discounted accuracy measure.

λ	$t_{\mathcal{H}}$	t_c
0.05	0.8359	0.2269
0.1	1.324	0.5656
0.2	1.558	0.76
0.3	2.336	0.4993

Table 2: Tuned thresholds for each λ.

```
User: My new card is here, what's the process for
activating it?
Assistant: activate_my_card
User: I am unable to activate my card, it won't
let me.
Assistant: activate_my_card
User: Can you help me activate my card
Assistant: activate_my_card

User: What is the youngest age for an account?
Assistant: age_limit
User: What is the appropriate age for my child to
be able to open an account?
Assistant: age_limit
User: How do I set up an account for my children?
Assistant: age_limit
...
```

Figure 6: Few-shot examples (demonstrators) used in the GPT-4 teacher as conversation history.

Table 2 provides the optimal threshold combination as selected by the optimization process for each of the four λ values. We can observe that the tuned value for t_c decreases from $\lambda = 0.2$ to $\lambda = 0.3$, instead of increasing, which can be accounted for by the aforementioned observation that multiple points maximize $\hat{\phi}$. However, we also notice the general increasing trend for both thresholds, which leads to fewer calls to the GPT-4 teacher, as one would expect, since higher values of λ represent that calls to the teacher are more costly.

E Human Teacher

We repeated the same experiments with a human teacher (simulated using ground truth labels). The experimental results of this setting (Fig. 8) demonstrate trends similar to the experiments with GPT-4. As λ increases, fewer calls to the teacher are made, and accuracy drops. Again, for $\lambda = 0.05$, we get a

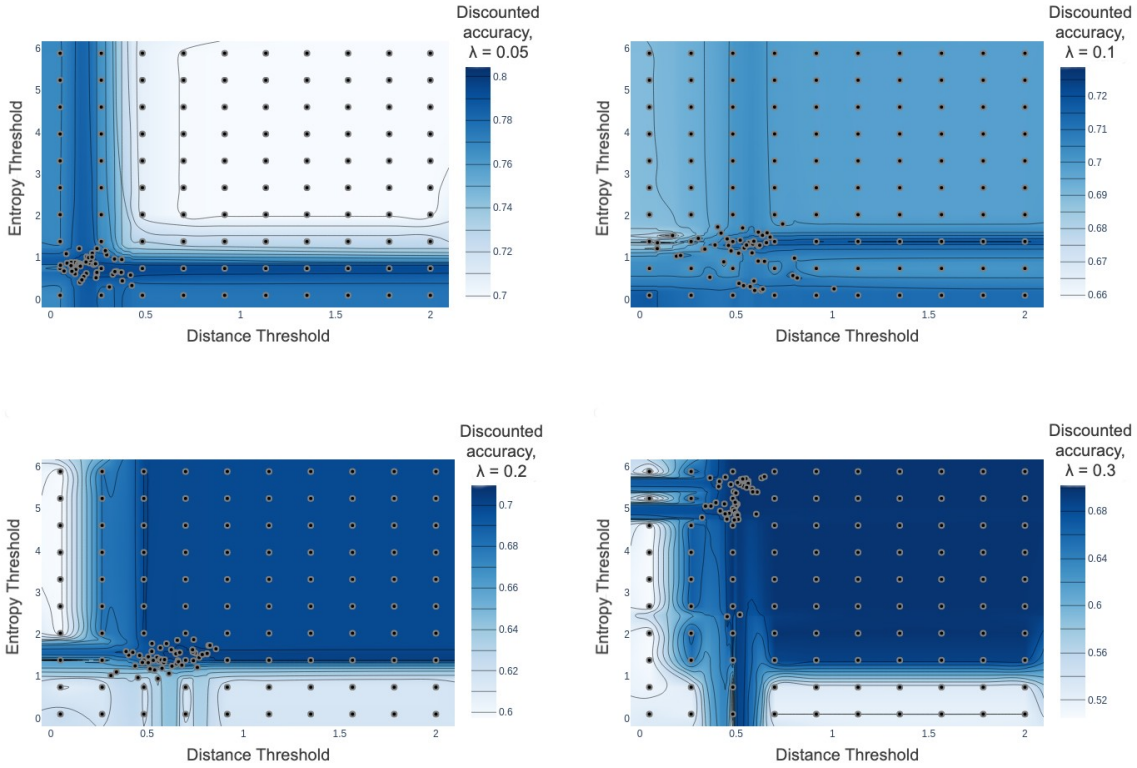

Figure 7: Contour plots (discounted accuracy) obtained during threshold tuning, for various λ values.

substantial reduction in calls to the teacher, at the expense of a small accuracy drop. However, it is now very difficult for OCaTS (solid lines) to outperform the (perfect) human teacher (dashed lines) in terms of discounted accuracy (left); this is only achieved for $\lambda = 0.3$ where the discount of calling the teacher is larger.

F MLP over MPNet embeddings Student

We also conducted these experiments with a Neural Network as a student. The new student is initially trained on the MPNet embeddings of the 231 instances, which are also stored in the cache. We performed hyperparameter tuning to identify the best hyperparameters for the network architecture. When new data points come in, we compute the entropy of the MLP and the distance of the new instance from the weighted centroid of the k most similar cached instances. If either of these metrics exceeds its equivalent threshold, we employ the teacher to perform the classification. As shown in Figure 9, the experiments with the new student have a performance similar to that of the k-NN student, further improving our claims and demonstrating the flexibility and adaptability of OCaTS.

G Sentiment Analysis Task

We also performed experiments with an additional task and dataset to strengthen our findings. We chose the binary sentiment analysis task and the IMDB reviews dataset (Maas et al., 2011). To simulate the limited data setting, we created the training, development and test sets by selecting 10 instances, 5 for each sentiment, based on the median token size of the training set. For the development set, we randomly chose 1000 instances with the same technique. Finally, due to limited resources, we used only 5,000 instances of the original test set. As a student, we used the original distance-weighted k-NN classifier which, in the restricted data setting, yielded an accuracy score of 0.63 on the test set. For the teacher, we chose GPT-3.5 (**?**), which achieved an accuracy score of 0.94. We use the same algorithm as previously outlined to choose whether the student or the teacher should perform the task. Figure 10 demonstrates that with OCaTS, we can achieve a similar performance while reducing the total number of calls to the teacher by half.

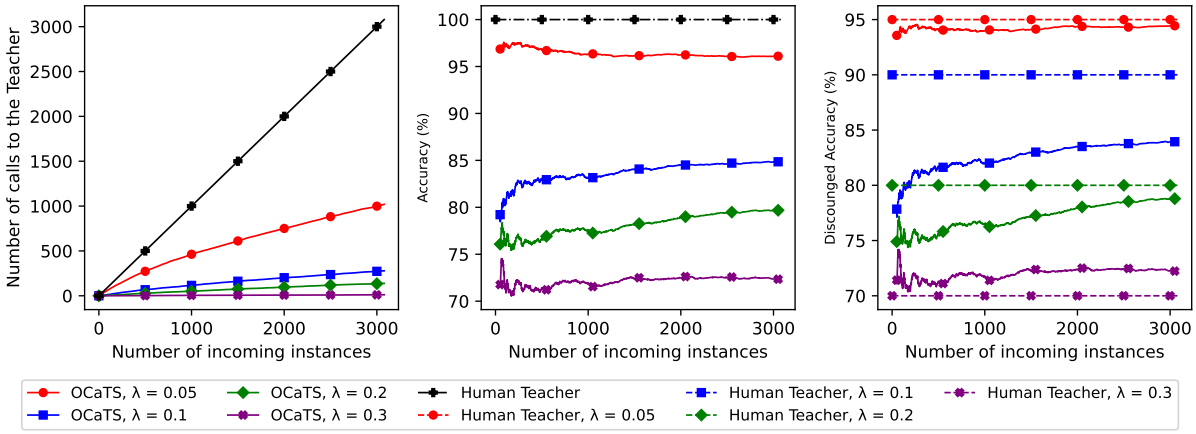

Figure 8: Number of calls to the teacher (left), accuracy (middle), and discounted accuracy (right), using a **human teacher** and the same k-NN student as before, for various λ values. Dashed lines show the discounted accuracy when we always call the human teacher. The larger λ, the more the SME prefers fewer calls at the expense of increased user frustration. As expected, OCaTS has a better discounted accuracy than always calling the teacher.

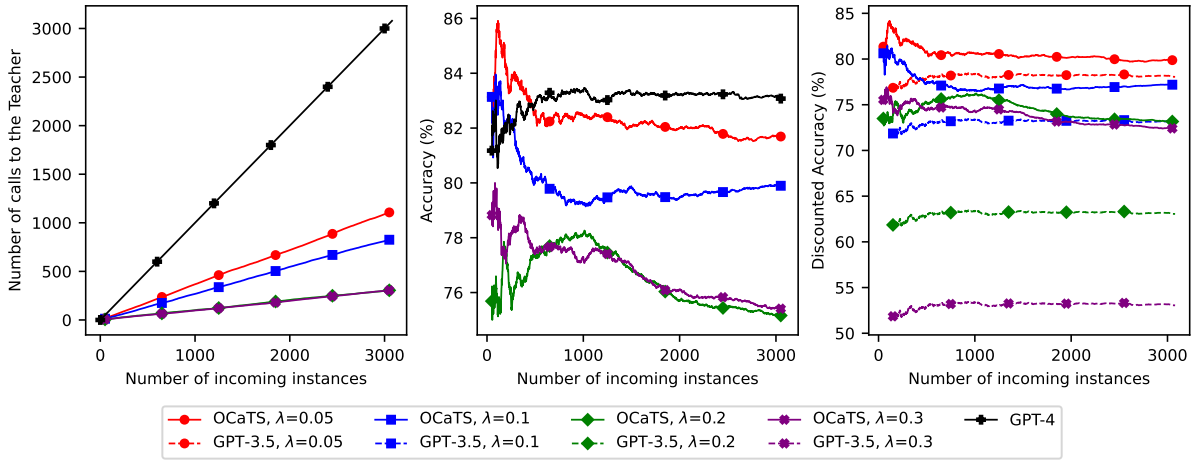

Figure 9: Number of calls to the teacher (left), accuracy (middle), and discounted accuracy (right), using a **NeuralNet student** and the same GPT-4 teacher as before, for various λ values.

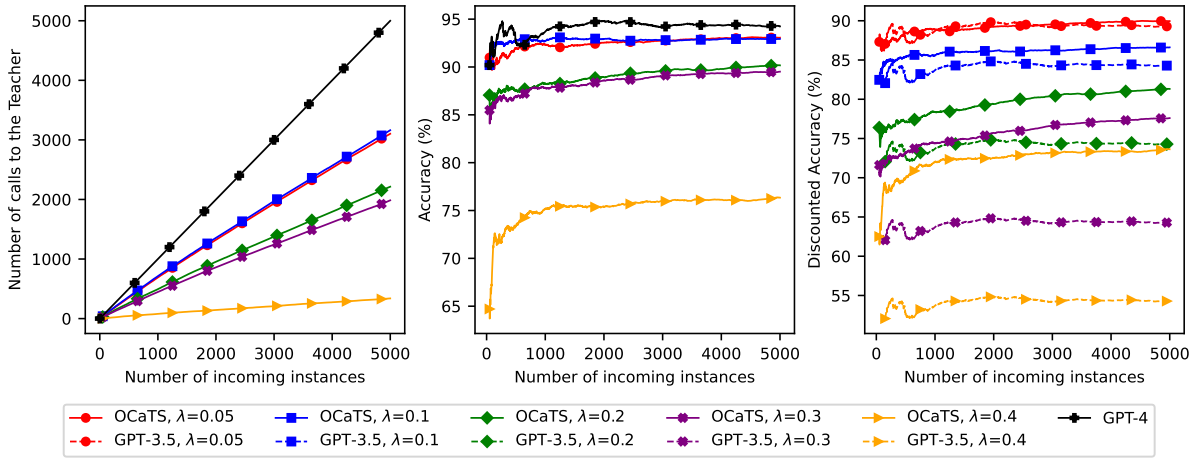

Figure 10: Number of calls to the teacher (left), accuracy (middle), and discounted accuracy (right), using the k-NN and the GPT-3.5 teacher, for various λ values.