Optimising Calls to Large Language Models with Uncertainty-Based Two-Tier Selection

Guillem Ramírez
University of Edinburgh
gramirez@ed.ac.uk

Alexandra Birch
University of Edinburgh
A.Birch@ed.ac.uk

Ivan Titov
University of Edinburgh
University of Amsterdam
ititov@inf.ed.ac.uk

Abstract

Researchers and practitioners operating on a limited budget face the cost-performance trade-off dilemma. The challenging decision often centers on whether to use a large LLM with better performance or a smaller one with reduced costs. This has motivated recent research in the optimisation of LLM calls. Either a *cascading* strategy is used, where a smaller LLM or both are called sequentially, or a *routing* strategy is used, where only one model is ever called. Both scenarios are dependent on a decision criterion which is typically implemented by an extra neural model. In this work, we propose a simpler solution; we use only the uncertainty of the generations of the small LLM as the decision criterion. We compare our approach with both cascading and routing strategies using three different pairs of pre-trained small and large LLMs, on nine different tasks and against approaches that require an additional neural model. Our experiments reveal this simple solution optimally balances cost and performance, outperforming existing methods on 25 out of 27 experimental setups.

1 Introduction

Large Language Models (LLMs) offer a high performance for a wide range of text tasks. Their widespread popularity both in research and industrial applications necessitates an understanding on how to optimally use them. Bigger models tend to have better performance, while smaller models are faster and cheaper to run. Deciding which model to use is a common dilemma for many researchers and practitioners with limited budgets, time-constraints or environmental concerns.

Recent works attempt to optimise calls to a set of LLMs. In this work we consider the set-up with two LLMs, where one is more expensive with greater performance than the other. In this scenario, there are two main strategies (Figure 1): *routing*, where a query from a user is directed to only one model based on a decision criterion; *cascading*, where the query always goes to the cheaper model and may subsequently go to the more expensive model depending on the cheaper model's output. These previous studies use one of these calling strategies, and involve either using an auxiliary model to score an LLM output (Chen et al., 2023; Sakota et al., 2023; Ding et al., 2024; Madaan et al., 2023) or using repeated calls to the small cheaper LLM (Yue et al., 2024; Madaan et al., 2023).

Studies that use an auxiliary model introduce further complexity in the optimisation approach, and it remains unclear when practitioners should rely on these auxiliary models. Not only do they require additional training, but they also usually require specific training data and the auxiliary models may not generalise to other tasks. For studies that rely on repeated calls to the small LLM, this approach can become expensive, undermining the original practical motivation for its use. It is perhaps for these reasons, that neither approach has gained traction among practitioners.

However, we question whether these additional models or the repeated calls are required to optimise LLM calls. We hypothesise that they may be unnecessary in many use cases, as we can extract confidence measures from the generations of the small model. To investigate this,

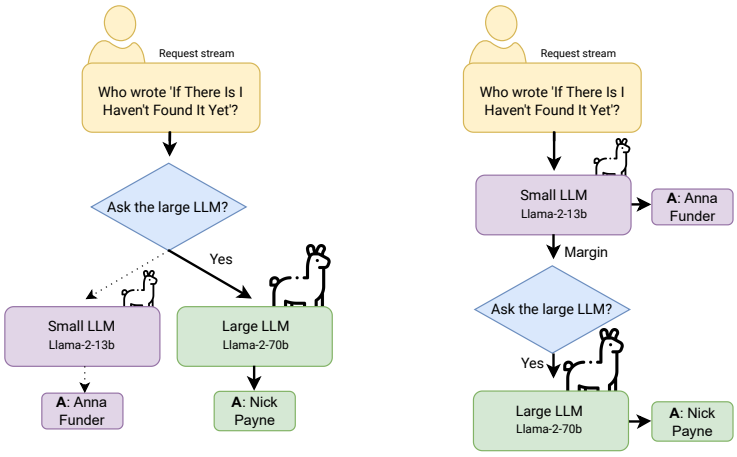

Figure 1: Routing (left) attempts to select the LLM with the best cost-accuracy trade-off given an incoming query. In cascading (right), all queries are passed through the small model, and depending on its output, the large LLM is consulted. We propose using a cascading approach that uses the margin of the generations to score outputs from the small LLM.

we propose a *cascading* policy that uses a simple measure of confidence of the small LLM to decide whether the large LLM needs to be called. We evaluate our policy on nine tasks, and for three different pairs of pre-trained small and large LLMs. We focus specifically on short-generation tasks, which are widely used and versatile, and do not include long-generation tasks in this study due to the additional complexity they introduce for evaluation. In our experiments on classification, multiple-choice and Question-Answering (QA) tasks, this policy outperforms methods that require additional training.

We believe our findings suggest that the optimisation of LLM calls should gravitate towards understanding their existing available signals, as opposed to training and running further auxiliary models. Moreover, simple and cheap policies have a lower entry cost for researchers and practitioners, which has an impact on their adoption.[1]

The key contributions of this work are as follows:

- We propose using the margin of the generations of LLMs to optimise LLM calls. The approach is simple, non-parametric, and does not require data or running multiple calls to the small model.
- We test our policy on nine tasks with three different pairs of small and large LLMs, and against relevant neural methods that use auxiliary models. We find that Margin Sampling outperforms the other methods, despite using fewer resources. In addition, we further test our policy in a multi-task set-up and obtain similar results.
- Our results underscore the importance of understanding signals from LLMs in the context of optimising calls to them, in contrast with previous work, which require additional training of auxiliary models.

2 Related work

LLM uncertainty The margin between the two most likely classes has been widely regarded as an uncertainty measure adopted from early Active Learning literature (Scheffer et al., 2001; Luo et al., 2004) and considered more recently in the context of Knowledge Distillation from LLMs (Baykal et al., 2023; Ramírez et al., 2023). Other measures of uncertainty exist for LLMs (Baan et al., 2023; Huang et al., 2023). The primary challenge in

[1]We make our code publicly available: https://github.com/guillemram97/margin_llms

text generation lies in the difficulty of distinguishing between various forms of uncertainty. Specifically, when evaluating the confidence level of a text generator, we would ideally want to concentrate on uncertainties related to the change in meaning. This entails distinguishing uncertainties that affect the intended meaning from meaning-preserving variations in the generated text. In this study, we address this challenge by concentrating on tasks that require generating very short sequences of tokens. We find that even a basic approach to uncertainty can be advantageous. Additionally, while it might seem logical to assess uncertainty through multiple text outputs from a model, this would lead to significant computational costs, which are impractical for our purposes. Therefore, we employ a straightforward method that does not necessitate generating multiple text samples.

Optimisation of inference costs Several methods have been proposed to improve the latency of LLMs, such as speculative decoding (Leviathan et al., 2023), knowledge distillation (Bucila et al., 2006; Hinton et al., 2015) and model quantisation (Jacob et al., 2018). However, operational costs rather than latency are the focus of this work. Our method only requires access to the logits of a small pre-trained LLM with no need for its parameters.

Optimisation of LLM API Calls Recent work deals with the problem of optimising calls to a pool of LLMs (Wang et al., 2024). Sakota et al. (2023) and Lu et al. (2023) propose to train an auxiliary model that predicts the success of calling each LLM. Similarly, Ding et al. (2024) developed a model that predicts the quantitative benefit of utilizing a smaller language model over a larger one.

Chen et al. (2023) proposed *cascading*: using an auxiliary model to predict the accuracy of the small LLM's output. Madaan et al. (2023) and Zhang et al. (2023) used cascading in conjunction with multiple calls to the small model. Finally, Yue et al. (2024) propose a cascading approach that requires repeated calls of the small LLM for reasoning tasks; we show that this can be simplified to just looking at the most likely tokens for short-generation tasks.

Ramírez et al. (2023) showed that the margin on a knowledge-distilled model could optimise calls to the larger LLM. This work was limited to one pre-trained large LLM and a smaller fine-tuned local model. However, many practitioners and researchers may not be able to fine-tune their own models due to budget constraints. We extend these findings to pairs of both small and large pre-trained LLMs, as well as to other generation tasks including a multi-task setup.

Concurrent work (Gupta et al., 2024) showed that the uncertainty at the token level can be used to effectively leverage a smaller LLM in a cascade setup. Their focus is on the combination of the multiple tokens involved in longer generation, and propose a supervised method; we propose using the margin, a simple rule that optimises short-generation tasks without needing additional data. In addition, we show its applicability in the multi-task setup.

Our proposed method differs from previous studies as it does not require previously annotated data for the task and does not require repeated calls to the small LLM. Finally, it does not require training and deploying an auxiliary model.

3 Optimisation of LLM calls

3.1 Problem definition

In this work, we predict mappings between elements in the input space, $\mathcal{X}$, and the corresponding labels in the output space, $\mathcal{Y}$. We have access to the small and the large LLMs that we can prompt to become predictors $f_s, f_l : \mathcal{X} \to \Delta(\mathcal{Y})$, where $\Delta(\mathcal{Y})$ denotes the class of probability distributions over $\mathcal{Y}$. We simulate the online setting, where users send queries sequentially. We have q queries $(x_1, \ldots, x_q) \overset{\text{iid}}{\sim} \mathcal{X}$ and we predict the corresponding labels $(y_1, \ldots, y_q)$. For each incoming query x_i, we decide whether to call an LLM based on a calling strategy (see below), and incur a given cost $c_s(x_i)$ or $c_l(x_i)$ respectively. The average

cost of the queries by both models is then given by $\hat{c}_\mathrm{s} = \frac{1}{q}\sum_i c_\mathrm{s}(x_i)$; $\hat{c}_\mathrm{l} = \frac{1}{q}\sum_i c_\mathrm{l}(x_i)$. We assume that $\hat{c}_\mathrm{s} < \hat{c}_\mathrm{l}$.

3.2 LLM Calling Strategies

For strategies that require training an additional model, we use a train dataset $X_\mathrm{train}, Y_\mathrm{train}$, and use either the corresponding labels from the small LLM only, $f_\mathrm{s}(X_\mathrm{train})$, or from both LLMs, $f_\mathrm{s}(X_\mathrm{train})$ and $f_\mathrm{l}(X_\mathrm{train})$, depending on the strategy. For the training of the auxiliary models, we follow the original papers as much as possible and perform a hyperparameter search where values are omitted. See Appendix A for a detailed explanation of the training process.

3.2.1 Routing Strategies

Since only the small or the large LLM is called in routing strategies, then for a given target average cost per query c ($\hat{c}_\mathrm{s} \leq c \leq \hat{c}_\mathrm{l}$), we call the large LLM with a probability p_r calculated from the re-arranged form of Equation 1.

$$c = (1 - p_\mathrm{r})\hat{c}_\mathrm{s} + p_\mathrm{r}\hat{c}_\mathrm{l} \tag{1}$$

Random routing For every incoming query we call the large LLM with the probability p_r.

Routing (Sakota et al., 2023; Lu et al., 2023) We train a meta-model to predict the performance of the small LLM only, given an incoming query. If this prediction is below a threshold value related to the probability p_r, it indicates the small LLM's performance is insufficient and thus we must call the large LLM.

HybridLLM (Ding et al., 2024) The performance of both the small and large LLMs are modelled in this strategy. We train a meta-model to predict if an incoming query is likely to be better solved by the small LLM than by the large LLM. As in Routing above, if this prediction is below a threshold value related to the probability p_r, we call the large LLM, otherwise the small LLM.

3.2.2 Cascading

Since the small LLM is always called, then for a given target average cost per query, c, we call the large LLM with a probability p_c calculated from the re-arranged form of Equation 2.

$$c = \hat{c}_\mathrm{s} + p_\mathrm{c}\hat{c}_\mathrm{l} \tag{2}$$

FrugalGPT (Chen et al., 2023) We train a model that, given a query and a candidate answer, predicts if the latter is correct. If this prediction is below a threshold value related to the probability p_c, we call the large LLM.

Margin Sampling (ours) We suggest using the uncertainty of the output, namely the margin (Scheffer et al., 2001; Luo et al., 2004), defined by:

$$\mathrm{Margin}_{f_\mathrm{s}}(x_i) = P_{f_\mathrm{s}}(y_i = k_1^{t=1} \mid x_i) - P_{f_\mathrm{s}}(y_i = k_2^{t=1} \mid x_i) \tag{3}$$

where $k_1^{t=1}$ and $k_2^{t=1}$ are the first and second most likely tokens, respectively, according to the distribution of f_s for the first predicted token position, $t = 1$. One advantage of this approach is that it does not require generating a full sequence to be able to compute uncertainty. Moreover, for the tasks we consider there is generally more uncertainty in the first token. If the margin is below a threshold value related to the probability p_c, we call the large LLM.

3.3 Dynamic threshold

All the investigated strategies require setting a threshold for the decision criterion, and we select a dynamic threshold in this work. An initial threshold is calculated using the first 20 queries. We do not evaluate whether to call the large LLM for these 20 queries, we only obtain outputs from the auxiliary models, or the margin value for Margin Sampling. This may require calling the small LLM depending on the strategy. We then use this distribution to calculate an initial p_r or p_c-th percentile value. For all subsequent queries, the decision to call the large LLM is made, and the threshold is dynamically updated based on all past queries. We test the effect of the dynamic threshold in additional experiments (Table 10) and conclude it does not result in major performance inefficiencies.

4 Experimental setup

4.1 LLMs

We study three pairs of small and large LLMs in our experiments: Mistral 7B (Jiang et al., 2023) and Mixtral 8x7B (Jiang et al., 2024); Llama-2 of size 13B and Llama-2 of size 70B (Touvron et al., 2023); GPT-3 (Brown et al., 2020) and GPT-4 (OpenAI, 2023). We selected these pre-trained LLMs because of their popularity among practitioners, as well as to show robustness across different scales. We have arranged the pairs this way to keep the family of LLMs similar; however, this is not a requirement for any of the calling strategies.

For the open-source families (Mistral, Llama-2), all our experiments are done locally in one NVIDIA A100 GPU (80 GB), after applying a 4-bit quantisation. Section C contains more details about the LLMs used.

4.2 Datasets

We draw inspiration from Liang et al. (2022) and Ramírez et al. (2023) and choose a wide range of tasks, showcasing different difficulties. The sizes of the datasets, along with the label distribution and accuracy of the LLMs, can be found in Appendix (Section B, Section C).

Classification tasks For emotion classification we use ISEAR (Shao et al., 2015); for fact-checking we use FEVER (Thorne et al., 2018); for sentiment analysis we use RT-Polarity (Pang & Lee, 2005), CR (Ni et al., 2019), and SST-2 (Socher et al., 2013). All of these datasets are balanced.

Multiple-choice We use Openbook (Mihaylov et al., 2018), a popular multiple-choice dataset that involves common knowledge of the world.

QA - short generation We use NaturalQuestions (Kwiatkowski et al., 2019), that contains real questions from human users; we use Wikifact (Petroni et al., 2019; Goodrich et al., 2019), which consists of a knowledge base completion problem to test factual knowledge; we use bAbI (Weston et al., 2016), that tests language understanding and reasoning.

4.3 Experiment details

For each dataset we set aside $1,000$ data-points for a train dataset that is used only to train the auxiliary models. The remainder is used for the online test set. We use $n = 500$ data-points from the train dataset in the Routing, HybridLLM and FrugalGPT strategies when training the auxiliary model, unless stated otherwise. For the auxiliary models, we fine-tune DistilBERT (Sanh et al., 2019), as per Sakota et al. (2023); Ding et al. (2024); Chen et al. (2023); Madaan et al. (2023); Ding et al. (2024). We further split the training data into 80% train and 20% validation, and fine-tune for 100 epochs with early stopping and patience of 20 epochs.

Unless stated otherwise, our results assume a simple cost scheme with $c_s(x_i) = 1$ and $c_l(x_i) = 10$, consistent with the pricing of commercial APIs (Wang et al., 2024) and similar

to the cost schemes of related work (Madaan et al., 2023; Chen et al., 2023; Lu et al., 2023; Sakota et al., 2023). We do not take into account in our experiments the latency of running DistilBERT, which we deem negligible compared to running the LLMs. To evaluate accuracy across budgets, we report Area Under the Curve (AUC) of the accuracy divided by $\hat{c}_l - \hat{c}_s$. Bolded results mark best performance, and underlined results mark second-best. We run our experiments with three random seeds, and report average results.

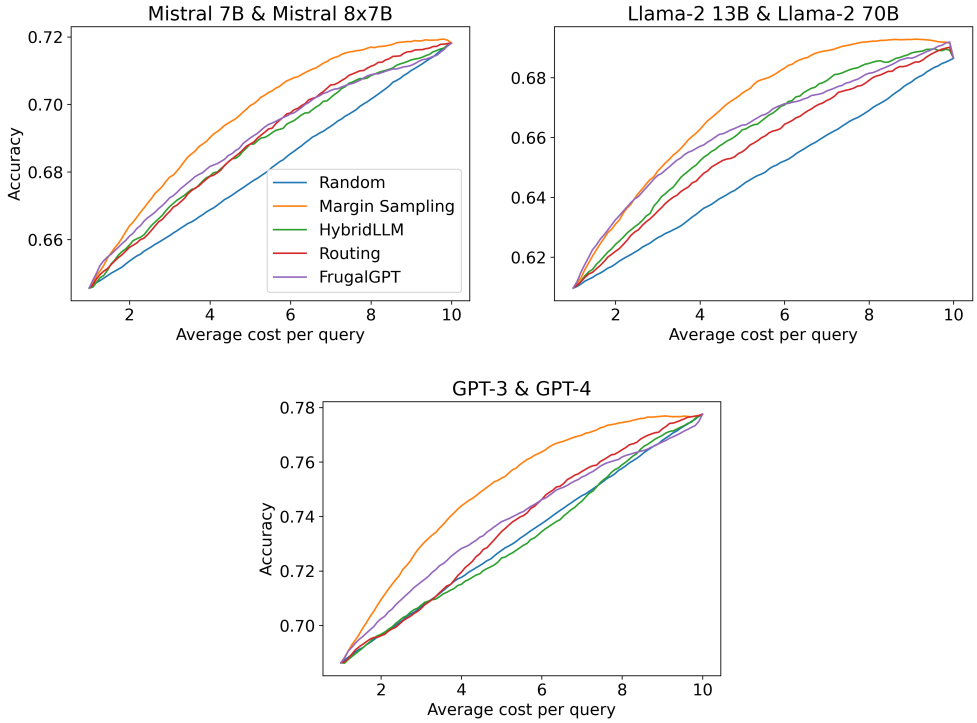

Figure 2: Accuracy curve with respect to budgets. We have averaged results for all the tasks.

5 Results

5.1 Comparison of calling strategies

Table 1 shows the AUC for all five evaluated calling strategies, across the nine tasks and for the three pairs of LLMs. Averaged across all nine tasks (the final column), we see that Margin Sampling outperforms all strategies for all LLM pairs. Across individual tasks, we see that it performs best or second best for seven of the nine tasks consistently across all three LLM pairs. Of all the nine tasks and three LLM pairs, it does not achieve best or second-best performance for only two of the 27 combinations: for ISEAR and for NaturalQuestions for the Mistral 7B and Mixtral 8x7B LLM pair. This is likely due to the poor accuracy of the small LLM Mistral 7B on these tasks (Appendix C). Figure 2 shows how performance changes with budget. We see that Margin Sampling dominates across all budgets, despite not using any training data.

Indeed, the performance of Margin Sampling seems to improve as the performance of the cheaper LLM improves, which is to be expected; it achieves its best results when applied on top of GPT-3.

FrugalGPT is on average the second-best performing strategy. This is due to its good performance on classification tasks, which is expected as it uses a classifier trained on the task as the auxiliary model. However, it performs worse than the random baseline on the more

	ISEAR	RT-Pol	FEVER	CR	SST-2	Openbook	Wikifact	bAbI	NaturalQ	Average
Mistral 7B - Mixtral 8x7B										
Random	0.606	0.876	0.773	0.923	0.880	0.843	0.443	0.597	0.193	0.681
Routing	0.618	0.876	**0.777**	0.924	0.890	0.844	0.492	0.606	0.177	0.689
HybridLLM	0.618	0.876	0.776	0.924	0.886	0.849	0.454	**0.612**	**0.199**	0.688
FrugalGPT	**0.632**	**0.887**	**0.777**	0.931	**0.901**	0.835	0.477	0.596	0.172	0.690
Margin Sampling	0.617	0.885	**0.777**	**0.933**	0.899	**0.868**	**0.499**	0.606	0.187	**0.697**
Llama-2 13B - Llama-2 70B										
Random	0.630	0.809	0.653	0.885	0.873	0.617	0.505	0.600	0.259	0.648
Routing	0.639	0.836	0.662	0.909	0.883	0.621	0.514	0.593	0.254	0.657
HybridLLM	0.641	0.844	0.681	0.899	0.874	0.626	0.514	0.608	0.264	0.661
FrugalGPT	**0.662**	**0.856**	0.668	**0.918**	**0.899**	0.598	0.507	0.602	0.258	0.663
Margin Sampling	0.645	0.853	**0.691**	0.912	0.893	**0.640**	**0.516**	**0.609**	**0.270**	**0.670**
GPT-3 - GPT-4										
Random	0.747	0.914	0.816	0.931	0.898	0.877	0.552	0.574	0.281	0.732
Routing	0.769	0.915	0.815	0.936	0.898	0.878	0.558	0.584	0.277	0.737
HybridLLM	0.744	0.914	0.821	0.932	0.899	0.882	0.556	0.558	0.278	0.732
FrugalGPT	0.767	0.922	0.819	**0.940**	**0.903**	0.876	0.564	0.572	0.283	0.738
Margin Sampling	**0.771**	**0.925**	**0.826**	**0.940**	0.899	**0.918**	**0.584**	**0.598**	**0.294**	**0.751**

Table 1: Accuracy (AUC) for the three LLM model pairs across the five classification tasks (columns 2-6), the multiple-choice task (column 7) and the three generation tasks (columns 8-10).

challenging multiple-choice task, Openbook. FrugalGPT also performs inconsistently for QA tasks; we conclude that FrugalGPT may be satisfactory on relatively easy classification tasks and struggle with harder generation tasks.

Finally, Routing and HybridLLM seem to have a good performance in QA tasks while having a worse performance in classification tasks. We note that HybridLLM on average has the same performance as random for the OpenAI models, which is a surprising finding.

We have not shown comparisons to approaches that require repeated calls to a small LLM throughout this work, as in preliminary experiments we found they do not perform well (see Section D.2).

5.2 Multi-task setting

LLMs are often used to handle various tasks simultaneously. To simulate this scenario, we create an artificial multi-task setting by merging the datasets from all nine tasks. We then sample 10,000 data-points. We split this dataset into a 10% train set ($n = 1,000$ data-points) and a 90% online test set.

Figure 3 and Table 2 show the results for our multi-task experiments. We see again that Margin Sampling has the best performance. This shows the versatility of this method, that it can be applied across tasks with ease. In contrast, HybridLLM has a poor performance both for Llama-2 and OpenAI models. We found these results still hold when using 5,000 data-points as training data (Appendix D.1).

5.3 Robustness of experiments

5.3.1 Investigating the effect of training data

To ensure that auxiliary models are not being unfairly handicapped by a low-data setting, we train them with double the data points as before ($n = 1,000$), in spite of this being a possibly infeasible amount of data for many researchers and practitioners.

	Mistral	Llama-2	OpenAI
Random	0.718	0.681	0.775
Router	0.731	0.696	0.786
HybridLLM	0.726	0.676	0.773
FrugalGPT	0.733	0.694	0.781
Margin Sampling	**0.736**	**0.704**	**0.792**

Table 2: Accuracy (AUC) in the multi-task setting. Methods Router, HybridLLM and FrugalGPT have been trained with $n = 1,000$ data-points.

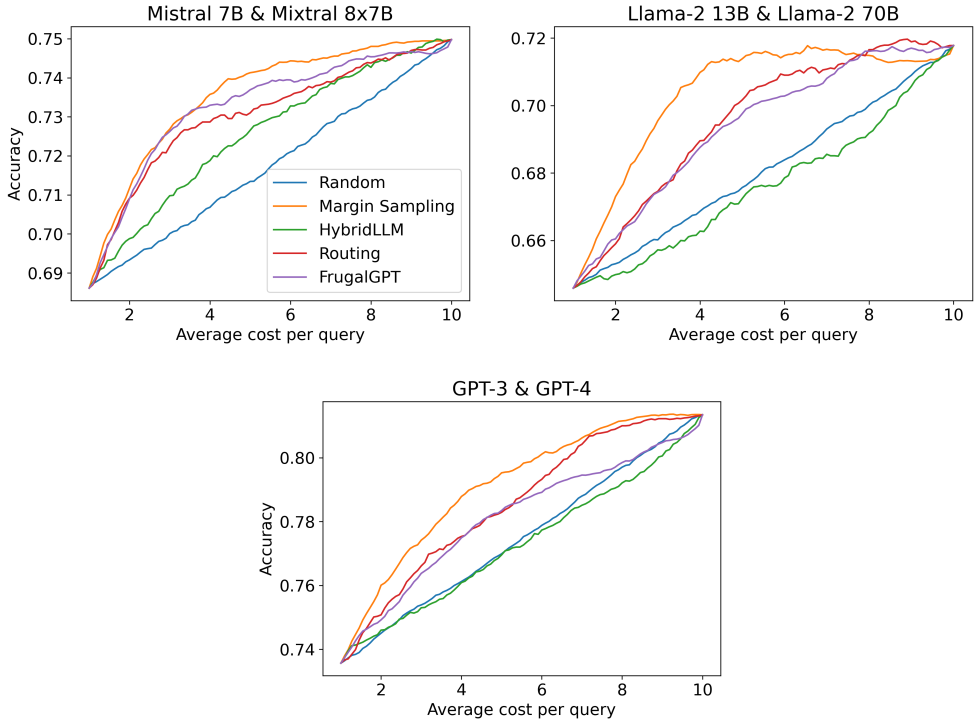

Figure 3: Accuracy curve with respect to budgets, in the multi-task setting.

Table 3 shows the averaged AUC results across tasks for the three LLM pairs. The additional data generally improves the performance of the auxiliary models, however, Margin Sampling still performs competitively, achieving the best performance for two of the three pairs. Table 3 also shows that the gap with FrugalGPT closes and that HybridLLM has again a limited performance on the OpenAI models. Additionally, we run experiments in settings with more data ($n = 5,000$), and confirm that Margin Sampling still performs competitively (Table 11).

5.3.2 Investigated the effect of cost

While previous studies typically only study a single cost setting (Sakota et al., 2023; Chen et al., 2023; Madaan et al., 2023; Ding et al., 2024), this could influence findings. We carry out further experiments with alternative cost schemes for the three LLM pairs across all tasks. [2] The cost ratio $r_{\text{cost}} = c_l/c_s$ is sufficient to parameterise the effect of cost. We need only vary c_l, and maintain $c_s = 1$. We study values of $c_l = 2$, 5 and 20.

[2]We note that, at the time of writing, the pricing ratio for the OpenAI models that we used (GPT-3 and GPT-4) is $\frac{\$30/\text{million tokens}}{\$2/\text{million tokens}} = 15$

	Mistral	Llama-2	OpenAI
Random	0.681	0.648	0.732
Router	0.694	0.661	0.741
HybridLLM	0.691	0.664	0.734
FrugalGPT	0.695	**0.672**	0.743
Margin Sampling	**0.697**	0.670	**0.751**

Table 3: Accuracy (AUC, averaged across tasks) when Router, HybridLLM and FrugalGPT have been trained with 1,000 data-points.

	Mistral			Llama-2			OpenAI		
	c_1=2	c_1=5	c_1=20	c_1=2	c_1=5	c_1=20	c_1=2	c_1=5	c_1=20
Random	0.681	0.681	0.681	0.648	0.648	0.648	0.732	0.732	0.732
Router	**0.690**	0.690	0.689	0.657	0.657	0.657	**0.737**	0.737	0.736
HybridLLM	0.689	0.688	0.688	**0.661**	0.661	0.661	0.732	0.732	0.731
FrugalGPT	0.677	0.687	0.691	0.650	0.660	0.664	0.721	0.735	0.740
Margin Sampling	0.683	**0.694**	**0.698**	0.654	**0.667**	**0.671**	0.734	**0.747**	**0.752**

Table 4: Accuracy (AUC, averaged across datasets) under cost schemes $c_s = 1$ and varying c_1.

Table 4 shows the results of these experiments. Independently of the cost, Margin Sampling appears the best cascading strategy (against FrugalGPT). In addition, we observe that Margin Sampling is the best overall strategy for $r_{\text{cost}} \geq 5$; for $r_{\text{cost}} = 2$, routing strategies could be preferred. Intuitively, cascading needs the cost of the small LLM to be relatively cheap enough to not sacrifice too many calls to the large LLM ($p_c < p_r$).

5.3.3 Mixing different families of LLMs

Margin Sampling can also be applied when the two LLMs are from different families; we further test its robustness with pairs Llama-2 13B - Mixtral 8x7B, Mistral 7B - Llama-2 70B. Our findings confirm the effectiveness of Margin Sampling (Table 5).

6 Discussion

In this paper, we have used a simple measure of confidence of the small LLM, known as Margin Sampling, to estimate the uncertainty of an LLM output in short-generation tasks, and we leave for future work the generalisation of this approach to long-generation tasks. Our experiments show that Margin Sampling performs consistently well on a range of short-generation tasks relevant to researchers and practitioners, such as QA and multiple-choice/classification tasks.

Existing approaches on the optimisation of LLM calls require training an auxiliary model. We hypothesised that these approaches introduce additional complexity that obfuscates their

	Llama-2 13B - Mixtral 8x7B	Mistral 7B - Llama-2 70B
Random	0.664	0.666
Router	0.673	0.688
HybridLLM	0.672	0.687
FrugalGPT	0.678	0.688
Margin Sampling	**0.685**	**0.690**

Table 5: Accuracy (AUC, averaged across datasets) when the LLMs are from different families.

understanding, and our findings that HybridLLM performs poorly with the OpenAI models lends credibility to this hypothesis. In contrast, we propose moving towards research that leverages information within the LLM itself. Classic notions of the uncertainty of the LLM's generation, such as perplexity, margin or entropy, could be relevant signals to help optimise LLM calls. A natural extension of our work is to generalise it for an arbitrary task length, for which it may be that a global notion of uncertainty also heavily depends on the first token. Our preliminary results with Machine Translation (Appendix D.5) indicate that some further adaptation of our method may be required for tasks where the first token may not be directly or partially the answer.

It was beyond the scope of this work to investigate cascading and routing strategies of three or more LLMs. However, we hypothesise our approach may still perform well, as Margin Sampling has shown to be robust to different-sized LLMs.

7 Conclusions

We have proposed a method for LLM call optimisation that achieves superior performance without the need of an auxiliary model. To the best of our knowledge, the field of LLM call optimisation has not yet gained widespread adoption among practitioners. This may be due to the limited versatility and increased complexity of previous solutions. In contrast, our proposed simple approach can be easily and quickly implemented with most commercial LLMs. We believe that our findings could encourage a new direction in the research of LLM call optimisations.

Acknowledgements

We acknowledge Yumnah Mohamied for her help and discussions while writing this paper. GR is supported by the UKRI Centre for Doctoral Training in Natural Language Processing, funded by the UKRI (grant EP/S022481/1), the University of Edinburgh, School of Informatics and School of Philosophy, Psychology & Language Sciences. IT acknowledges support by the Dutch National Science Foundation (NWO Vici VI.C.212.053). AB has been supported by the European Union's Horizon Europe Research and Innovation programme under Grant Agreement No 101070631 (UTTER). The computations described in this research were performed using the Baskerville HPC service, funded by the EPSRC and UKRI (grant EP/T022221/1) and the Digital Research Infrastructure programme (EP/W032244/1). This work also used the Cirrus UK National Tier-2 HPC Service at EPCC, funded by the University of Edinburgh and EPSRC (EP/P020267/1). The project that gave rise to these results received the support of a fellowship from "la Caixa" Foundation (ID 100010434, grant code LCF/BQ/EU22/11930079).

References

Joris Baan, Nico Daheim, Evgenia Ilia, Dennis Ulmer, Haau-Sing Li, Raquel Fernández, Barbara Plank, Rico Sennrich, Chrysoula Zerva, and Wilker Aziz. Uncertainty in natural language generation: From theory to applications. *CoRR*, abs/2307.15703, 2023. doi: 10.48550/ARXIV.2307.15703. URL https://doi.org/10.48550/arXiv.2307.15703.

Cenk Baykal, Khoa Trinh, Fotis Iliopoulos, Gaurav Menghani, and Erik Vee. Robust active distillation. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=ALDM5SN2r7M.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models

are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Cristian Bucila, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In Tina Eliassi-Rad, Lyle H. Ungar, Mark Craven, and Dimitrios Gunopulos (eds.), *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, August 20-23, 2006*, pp. 535–541. ACM, 2006. doi: 10.1145/1150402.1150464. URL https://doi.org/10.1145/1150402.1150464.

Lingjiao Chen, Matei Zaharia, and James Zou. Frugalgpt: How to use large language models while reducing cost and improving performance. *CoRR*, abs/2305.05176, 2023. doi: 10.48550/ARXIV.2305.05176. URL https://doi.org/10.48550/arXiv.2305.05176.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Rühle, Laks V. S. Lakshmanan, and Ahmed Hassan Awadallah. Hybrid LLM: Cost-efficient and quality-aware query routing. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=02f3mUtqnM.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang,

Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Ekaterina Fadeeva, Aleksandr Rubashevskii, Artem Shelmanov, Sergey Petrakov, Haonan Li, Hamdy Mubarak, Evgenii Tsymbalov, Gleb Kuzmin, Alexander Panchenko, Timothy Baldwin, Preslav Nakov, and Maxim Panov. Fact-checking the output of large language models via token-level uncertainty quantification. *CoRR*, abs/2403.04696, 2024. doi: 10.48550/ARXIV.2403.04696. URL https://doi.org/10.48550/arXiv.2403.04696.

Ben Goodrich, Vinay Rao, Peter J. Liu, and Mohammad Saleh. Assessing the factual accuracy of generated text. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pp. 166–175, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362016. doi: 10.1145/3292500.3330955. URL https://doi.org/10.1145/3292500.3330955.

Neha Gupta, Harikrishna Narasimhan, Wittawat Jitkrittum, Ankit Singh Rawat, Aditya Krishna Menon, and Sanjiv Kumar. Language model cascades: Token-level uncertainty and beyond. *CoRR*, abs/2404.10136, 2024. doi: 10.48550/ARXIV.2404.10136. URL https://doi.org/10.48550/arXiv.2404.10136.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015. URL http://arxiv.org/abs/1503.02531.

Yuheng Huang, Jiayang Song, Zhijie Wang, Huaming Chen, and Lei Ma. Look before you leap: An exploratory study of uncertainty measurement for large language models. *CoRR*, abs/2307.10236, 2023. doi: 10.48550/ARXIV.2307.10236. URL https://doi.org/10.48550/arXiv.2307.10236.

Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew G. Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 2704–2713. Computer Vision Foundation / IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00286. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Jacob_Quantization_and_Training_CVPR_2018_paper.html.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *CoRR*, abs/2310.06825, 2023. doi: 10.48550/ARXIV.2310.06825. URL https://doi.org/10.48550/arXiv.2310.06825.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts. *CoRR*, abs/2401.04088, 2024. doi: 10.48550/ARXIV.2401.04088. URL https://doi.org/10.48550/arXiv.2401.04088.

Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X: Papers*, pp. 79–86, Phuket, Thailand, September 13-15 2005. URL https://aclanthology.org/2005.mtsummit-papers.11.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: a benchmark for question answering research. *Trans. Assoc. Comput. Linguistics*, 7:452–466, 2019. doi: 10.1162/TACL_A_00276. URL https://doi.org/10.1162/tacl_a_00276.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara

Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19274–19286. PMLR, 2023. URL https://proceedings.mlr.press/v202/leviathan23a.html.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana Acosta-Navas, Drew A. Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue Wang, Keshav Santhanam, Laurel J. Orr, Lucia Zheng, Mert Yüksekgönül, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri S. Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. Holistic evaluation of language models. *CoRR*, abs/2211.09110, 2022. doi: 10.48550/ARXIV.2211.09110. URL https://doi.org/10.48550/arXiv.2211.09110.

Keming Lu, Hongyi Yuan, Runji Lin, Junyang Lin, Zheng Yuan, Chang Zhou, and Jingren Zhou. Routing to the expert: Efficient reward-guided ensemble of large language models. *CoRR*, abs/2311.08692, 2023. doi: 10.48550/ARXIV.2311.08692. URL https://doi.org/10.48550/arXiv.2311.08692.

Tong Luo, K. Kramer, S. Samson, A. Remsen, D.B. Goldgof, L.O. Hall, and T. Hopkins. Active learning to recognize multiple types of plankton. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 3, pp. 478–481 Vol.3, August 2004. doi: 10.1109/ICPR.2004.1334570. ISSN: 1051-4651.

Aman Madaan, Pranjal Aggarwal, Ankit Anand, Srividya Pranavi Potharaju, Swaroop Mishra, Pei Zhou, Aditya Gupta, Dheeraj Rajagopal, Karthik Kappaganthu, Yiming Yang, Shyam Upadhyay, Mausam, and Manaal Faruqui. Automix: Automatically mixing language models. *CoRR*, abs/2310.12963, 2023. doi: 10.48550/ARXIV.2310.12963. URL https://doi.org/10.48550/arXiv.2310.12963.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? A new dataset for open book question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pp. 2381–2391. Association for Computational Linguistics, 2018. doi: 10.18653/v1/d18-1260. URL https://doi.org/10.18653/v1/d18-1260.

Jianmo Ni, Jiacheng Li, and Julian McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 188–197, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1018. URL https://aclanthology.org/D19-1018.

OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023. doi: 10.48550/ARXIV.2303.08774. URL https://doi.org/10.48550/arXiv.2303.08774.

Bo Pang and Lillian Lee. Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pp. 115–124, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. doi: 10.3115/1219840.1219855. URL https://aclanthology.org/P05-1015.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pp. 311–318. ACL, 2002. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040/.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander H. Miller. Language models as knowledge bases? In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pp. 2463–2473. Association for Computational Linguistics, 2019. doi: 10.18653/V1/D19-1250. URL https://doi.org/10.18653/v1/D19-1250.

Guillem Ramírez, Matthias Lindemann, Alexandra Birch, and Ivan Titov. Cache & distil: Optimising API calls to large language models. *CoRR*, abs/2310.13561, 2023. doi: 10.48550/ARXIV.2310.13561. URL https://doi.org/10.48550/arXiv.2310.13561.

Marija Sakota, Maxime Peyrard, and Robert West. Fly-swat or cannon? cost-effective language model choice via meta-modeling. *CoRR*, abs/2308.06077, 2023. doi: 10.48550/ARXIV.2308.06077. URL https://doi.org/10.48550/arXiv.2308.06077.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108, 2019. URL http://arxiv.org/abs/1910.01108.

Tobias Scheffer, Christian Decomain, and Stefan Wrobel. Active Hidden Markov Models for Information Extraction. In Frank Hoffmann, David J. Hand, Niall Adams, Douglas Fisher, and Gabriela Guimaraes (eds.), *Advances in Intelligent Data Analysis*, Lecture Notes in Computer Science, pp. 309–318, Berlin, Heidelberg, 2001. Springer. ISBN 978-3-540-44816-7. doi: 10.1007/3-540-44816-0_31.

Bo Shao, Lorna Doucet, and David R. Caruso. Universality Versus Cultural Specificity of Three Emotion Domains: Some Evidence Based on the Cascading Model of Emotional Intelligence. *Journal of Cross-Cultural Psychology*, 46(2):229–251, February 2015. ISSN 0022-0221. doi: 10.1177/0022022114557479. URL https://doi.org/10.1177/0022022114557479. Publisher: SAGE Publications Inc.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pp. 1631–1642. ACL, 2013. URL https://aclanthology.org/D13-1170/.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and verification. In Marilyn A. Walker, Heng Ji, and Amanda Stent (eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pp. 809–819. Association for Computational Linguistics, 2018. doi: 10.18653/v1/n18-1074. URL https://doi.org/10.18653/v1/n18-1074.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288, 2023. doi: 10.48550/ARXIV.2307.09288. URL https://doi.org/10.48550/arXiv.2307.09288.

Can Wang, Bolin Zhang, Dianbo Sui, Zhiying Tu, Xiaoyu Liu, and Jiabao Kang. A survey on effective invocation methods of massive LLM services. *CoRR*, abs/2402.03408, 2024. doi: 10.48550/ARXIV.2402.03408. URL https://doi.org/10.48550/arXiv.2402.03408.

Jason Weston, Antoine Bordes, Sumit Chopra, and Tomás Mikolov. Towards ai-complete question answering: A set of prerequisite toy tasks. In Yoshua Bengio and Yann LeCun (eds.), *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL http://arxiv.org/abs/1502.05698.

Murong Yue, Jie Zhao, Min Zhang, Liang Du, and Ziyu Yao. Large language model cascades with mixture of thought representations for cost-efficient reasoning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=6okaSfANzh.

Jieyu Zhang, Ranjay Krishna, Ahmed Hassan Awadallah, and Chi Wang. Ecoassistant: Using LLM assistant more affordably and accurately. *CoRR*, abs/2310.03046, 2023. doi: 10.48550/ARXIV.2310.03046. URL https://doi.org/10.48550/arXiv.2310.03046.

A Implementation details

For methods Routing, HybridLLM and FrugalGPT we use Huggingface's `AutoModelForSequenceClassification` and the model `distilbert/distilbert-base-uncased`. For methods Routing, HybridLLM we set the target number of classes to two; for FrugalGPT, we set it to either the number of target classes for classification/multiple-choice problems or two for QA problems. We perform a hyperparameter search (grid search) on a validation set of 500 examples of Openbook and Wikifact, and we find that the different methods have a similar convergence. We decide using learning rate $\mu = 5 \times 10^{-4}$, training batch size $m = 16$ and weight decay $\lambda = 0.01$, which is consistent with the reported values of Sakota et al. (2023) and seems to generalise well across tasks. Ding et al. (2024), Chen et al. (2023) and Lu et al. (2023) do not report hyperparameter values.

A.1 Training

Routing We generate an answer using the small LLM. Then, we compare it to the gold label. The target class is 1 or 0 depending on the correctness of the answer from the small LLM.

- **Input**: 'Who wrote 'If There Is I Haven't Found It Yet?'
- **Output**: '0'
- Explanation: Llama-2 13B produces an incorrect answer.

HybridLLM We generate an answer using the small and the large LLMs. Then, we use the first token y_1 of the gold answer y, to obtain the quality gap $H(x) = P_{\text{Small}}(y_1 \mid x) - P_{\text{Large}}(y_1 \mid x)$. Following Ding et al. (2024), we subtract the median of $H(x)$; then we map it to 1 or 0 depending on $H(x) > H_{\text{median}}$, thus leaving a binary classification problem of uniform classes.

- **Input**: 'Who wrote 'If There Is I Haven't Found It Yet?'
- **Output**: '0'
- Explanation: Llama-2 13B is less likely than Llama-2 70B to produce the right answer.

FrugalGPT (classification and multiple-choice) We train DistilBERT with gold data. During inference, we generate an answer with the small LLM. The score is the probability DistilBERT associates to this class.

	ISEAR	RT-Polarity	FEVER	CR	SST-2	Openbook	Wikifact	bAbI	NaturalQ
Total datapoints	6132	8529	5289	3393	7000	5457	5733	3000	2876
Number of classes	7	2	2	2	2	4	QA	QA	QA

Table 6: Datasets used.

	ISEAR	RT-Polarity	FEVER	CR	SST-2	Openbook	Wikifact	bAbI	NaturalQ
Mistral 7B	0.557	0.862	0.770	0.911	0.854	0.813	0.359	0.560	0.125
Mixtral 8x7B	0.655	0.889	0.779	0.936	0.906	0.875	0.530	0.634	0.260
Llama-13b	0.599	0.798	0.613	0.902	0.867	0.556	0.416	0.525	0.212
Llama-70b	0.661	0.820	0.691	0.871	0.882	0.681	0.590	0.676	0.307
GPT-3	0.699	0.900	0.777	0.921	0.897	0.798	0.486	0.462	0.251
GPT-4	0.796	0.929	0.852	0.943	0.899	0.956	0.622	0.695	0.324

Table 7: Accuracy of the LLMs in the studied tasks.

FrugalGPT (QA) We train a binary classifier that predicts if an answer is correct. To do so, we use as the positive class the gold labels. We generate answers with Llama-2 13B and tag them as either positive or negative class depending on whether they match the gold labels.

- **Input**: 'Who wrote 'If There Is I Haven't Found It Yet? ANSWER: Anna Funder'
- **Output**: '0'
- Explanation: Anna Funder is an incorrect answer.

B Datasets

Table 6 contains some statistics on the datasets used. All the classification datasets are uniformly distributed. For our experiments, we have reserved 1,000 datapoints for training the scorers (Router, HybridLLM and FrugalGPT) and used the remaining of the datasets for online inference.

C LLMs used

We load the open-source LLMs with Huggingface's `AutoModelForCausalLM.from_pretrained`, activating `load_in_4bit`.
We use models `meta-llama/Llama-2-13b-hf`, `meta-llama/Llama-2-70b-hf`, `mistralai/Mistral-7B-Instruct-v0.2` and `mistralai/Mixtral-8x7B-Instruct-v0.1`. We set the temperature to 0 and look at the most likely token.

For ISEAR, RT-Polarity, FEVER, OpenbookQA, SST-2 and CR, we use the prompts from Ramírez et al. (2023) (0-shot). For Wikifact, NaturalQuestions and bAbI we use the prompts from the HELM benchmark (Liang et al., 2022).

For the OpenAI models, we use `davinci-002` (GPT-3) and `gpt-4`. Annotating all the datasets has a cost of around $180.

Table 7 contains shows the accuracy of the LLMs across the different tasks.

	Mistral	Llama-2	OpenAI
Random	0.718	0.682	0.777
Router	0.734	0.696	0.790
HybridLLM	0.725	0.688	0.774
FrugalGPT	**0.739**	**0.706**	0.791
Margin Sampling	**0.739**	0.705	**0.794**

Table 8: Accuracy (AUC) in the multiple-task setting. Methods Router, HybridLLM and FrugalGPT have been trained with $n = 5,000$ datapoints.

	ISEAR	RT-Polarity	FEVER	CR	SST-2	Openbook	Wikifact	bAbI	NaturalQ
Random	0.630	0.809	0.653	0.885	0.873	0.617	0.505	0.600	0.259
Committee	0.620	0.797	0.591	0.886	0.863	0.587	0.473	0.538	0.234

Table 9: Accuracy (AUC) for the Committee method (Yue et al., 2024), for Llama-2 13B and Llama-2 70B.

D Additional results

D.1 Multi-task setup

We experiment to see how far we can get with Routing, HybridLLM and FrugalGPT with more data. We train these methods with $n = 5,000$ datapoints. We find that Margin Sampling still outperforms them in this setup (Table 8).

D.2 Multiple calls to the LLM

We experiment with the method from Yue et al. (2024), which estimates the uncertainty of the generation by doing multiple calls to the small LLM. We make 5 calls, sampling with temperature $T = 1$. Our results (Table 9) reveal this method does badly in our setup. The reason for this relies that doing the multiple calls to the small LLM is relatively expensive in our setup with $c_{\text{Small}} = 1, c_{\text{Large}} = 10$.

D.3 Investigating the effect of the dynamic threshold

To investigate the effect of the dynamic threshold used in our main experiments, we simulate an offline setup, where we can obtain the values for the different methods, order them and find the *best* threshold value. Table 10 reveals that these gold threshold values only lead to a slim improvement; we conclude that dynamic threshold does not result in major performance inefficiencies.

	Mistral	Llama-2	OpenAI
Random	0.681	0.648	0.732
Router	0.690	0.657	0.737
HybridLLM	0.689	0.661	0.732
FrugalGPT	0.690	0.664	0.739
Margin Sampling	**0.697**	**0.670**	**0.751**

Table 10: Accuracy (AUC, averaged across tasks) for the offline setup. The slim differences to the online setup (Table 1) suggest that the dynamic threshold does not lead to inefficiencies.

	Openbook	Wikifact
Mistral 7B - Mixtral 8x7B		
Random	0.848	0.453
Router	0.850	**0.538**
HybridLLM	0.851	0.491
FrugalGPT	0.846	0.498
Margin Sampling	**0.865**	0.497
Llama-2 13B - Llama-2 70B		
Random	0.607	0.506
Router	0.613	**0.532**
HybridLLM	0.602	0.529
FrugalGPT	0.617	0.515
Margin Sampling	**0.632**	0.518
GPT-3 - GPT-4		
Random	0.872	0.556
Router	0.875	**0.604**
HybridLLM	0.883	0.559
FrugalGPT	0.876	0.577
Margin Sampling	**0.914**	0.588

Table 11: Accuracy (AUC). Methods Router, HybridLLM and FrugalGPT have been trained with n = 5, 000 data-points.

D.4 Experiments with additional data

We test the limits of our results when additional data is available for a particular task. We run experiments on OpenbookQA and Wikifact, which are two of the harder tasks, and we train the supervised methods with 5,000 datapoints. Table 11 shows the results. The trend, as expected, is that supervised methods improve the performance, but Margin Sampling still has a relevant performance.

D.5 Longer generation (Machine Translation)

To test the effectiveness of Margin Sampling in a task that requires longer text generation, we run experiments on the Europarl Parallel Corpus (Koehn, 2005), specifically on the German-French direction. We use Llama-3.1 of size 8B and Llama-3.1 of size 70B (Dubey et al., 2024). For this experiment setup, we find that Margin Sampling achieves 21.9 BLEU score (Papineni et al., 2002), underperforming the random baseline (22.2 BLEU). We conclude that some further adaptation of our method is required for tasks where the first token may not be directly or partially the answer. Our previous results with tasks generating multiple tokens, ie. Wikifact and NaturalQuestions, did show that Margin Sampling improves over the random baseline where a significant portion of the answers started with stop words such as articles. Therefore, we hypothesise that individual token uncertainty -not constrained to the first token- may be a suitable metric for the optimisation of API calls. This is validated by the results from Fadeeva et al. (2024), which found that token uncertainty may be used for hallucination detection, and by the work from Gupta et al. (2024), who trained a neural model to aggregate the individual token uncertainties.