Bridging the Gap between Different Vocabularies for LLM Ensemble

Anonymous ACL submission

Abstract

Ensembling different large language models (LLMs) to unleash their complementary potential and harness their individual strengths is highly valuable. Nevertheless, vocabulary discrepancies among various LLMs have constrained previous studies to either selecting or blending completely generated outputs. This limitation hinders the dynamic correction and enhancement of outputs during the generation process, resulting in a limited capacity for effective ensemble. To address this issue, we propose a novel method to **E**nsemble LLMs via **V**ocabulary **A**lignment (EVA). EVA bridges the lexical gap among various LLMs, enabling meticulous ensemble at each generation step. Specifically, we first learn mappings between the vocabularies of different LLMs with the assistance of overlapping tokens. Subsequently, these mappings are employed to project output distributions of LLMs into a unified space, facilitating a fine-grained ensemble. Finally, we design a filtering strategy to exclude models that generate unfaithful tokens. Experimental results on commonsense reasoning, arithmetic reasoning, machine translation, and data-to-text generation tasks demonstrate the superiority of our approach compared with individual LLMs and previous ensemble methods conducted on complete outputs. Further analyses confirm that our approach can leverage knowledge from different language models and yield consistent improvement.

1 Introduction

Large language models (LLMs) have demonstrated impressive performance across various natural language processing tasks (Anil et al., 2023; Touvron et al., 2023; Chiang et al., 2023). These models, spanning diverse datasets, architectures, and training methodologies, exhibit different strengths and weaknesses (Jiang et al., 2023). Therefore, ensembling these LLMs to unleash their complementary potential and harness their individual strengths is highly valuable (Jiang et al., 2023; Lu et al., 2023; Shnitzer et al., 2023).

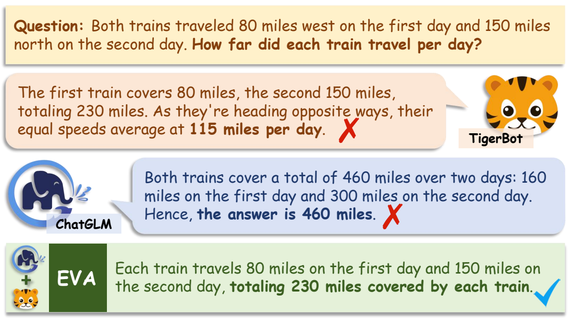

Figure 1: **Motivation of EVA.** For the problem of *train travel distance*, both TigerBot and ChatGLM provide wrong answers. Ensembling over completely generated outputs cannot derive the correct answer. EVA achieves correct answers by performing fine-grained ensemble at each generation step, allowing each token to benefit from the ensemble.

Previous studies typically concentrate on the ensemble of completely generated outputs, which involve either ranking multiple outputs to select the best one (Lu et al., 2023; Shnitzer et al., 2023) or incorporating additional fusion models to blend these outputs (Jiang et al., 2023). Therefore, these methods usually lead to ensemble outcomes confined to the space of several completely generated outputs. As shown in Figure 1, for the problem of *train travel distance*, both TigerBot and ChatGLM provide incorrect reasoning processes, resulting in wrong answers. Ensembling over completely generated outputs cannot produce correct answer if all the candidate complete outputs are wrong.

One potential solution to this problem involves incorporating ensembling into the generation process of LLMs. As indicated by Zhang et al. (2023), early errors in LLMs tend to snowball, leading to subsequent errors that might not have otherwise occurred. Ensembling during generation helps prevent the generation of inaccurate tokens at each

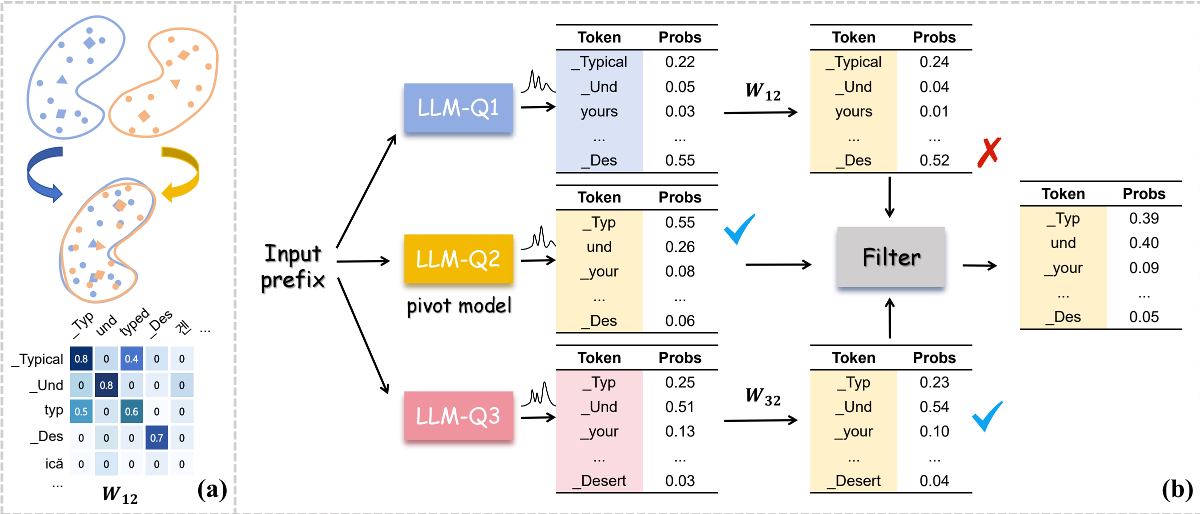

Figure 2: **The EVA framework.** EVA consists of two steps. (a) Firstly, we establishes alignment between the vocabularies of different models. (b) Next, we project the output distributions of different LLMs into a unified space using the established vocabulary alignment and exclude unfaithful tokens to perform fine-grained ensemble.

step, thereby reducing misleading cues for subsequent token generation. However, such an ensemble approach is unfeasible for LLMs due to vocabulary discrepancies. As illustrated in Figure 2, the three LLMs use distinct vocabularies, leading to different output distributions over tokens. This divergence hinders the straightforward token-level ensemble at each generation step.

To tackle this issue, we propose a simple yet effective method named **E**nsemble via **V**ocabulary **A**lignment (EVA), facilitating the fine-grained ensemble of LLMs at each generation step. EVA stems from a straightforward observation: although various LLMs have distinct vocabularies, they commonly share a significant number of overlapping tokens. By leveraging these tokens as bridges, EVA can achieve vocabulary alignment. Specifically, for vocabularies $\mathcal{V}^{Q_1}$, $\mathcal{V}^{Q_2}$ used in LLM-Q_1 and LLM-Q_2, we first extract embeddings of the overlapping tokens and learn a mapping matrix to project these embeddings into a shared space. Subsequently, by computing similarity scores between tokens in these vocabularies, we derive the semantic projection $\boldsymbol{W} \in \mathbb{R}^{|\mathcal{V}^{Q_1}|\times|\mathcal{V}^{Q_2}|}$. This enables the projection of output distributions from LLM-Q_1 to LLM-Q_2 and generates reasonable tokens based on the fused distribution of these LLMs at each inference step. Finally, we further enhance our approach by devising a filtering strategy capable of excluding models that generate unfaithful tokens.

Our method successfully overcomes the vocabulary discrepancy between different LLMs and facilitates fine-grained ensemble during generation. Significantly, our method necessitates solely an additional projection matrix $\boldsymbol{W}$, eliminating the necessity of extra fusion models or supervised training corpora. We evaluate our method on various NLP tasks, including Commonsense Reasoning, Arithmetic Reasoning, Machine Translation, and Data-to-Text Generation. Experimental results demonstrate the superiority of our approach compared with individual LLMs and previous ensemble methods conducted on complete outputs. Further analyses confirm that our approach can leverage knowledge from different language models and yield consistent improvement.

Briefly, our contributions can be summarized as follows:

- We propose a novel LLM ensemble method to achieve fine-grained ensemble at each generation step. Our method aims to bridge the lexical gap between LLMs, thereby unleashing their complementary potentials.
- We devise an effective filtering strategy to exclude models generating unfaithful tokens, preventing underperforming models from misleading the overall judgment.
- Empirical results demonstrate the effectiveness and superiority of our method, which significantly improves overall performance on various natural language processing tasks.

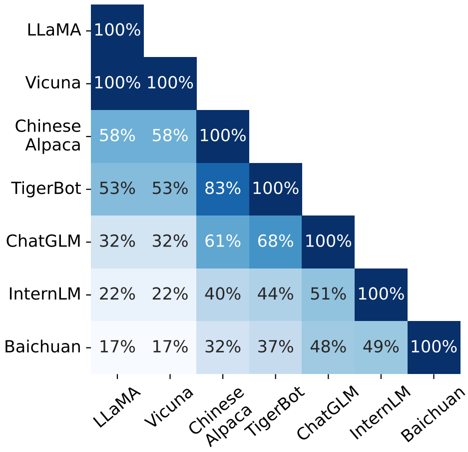

Figure 3: The rate of overlapping tokens between different LLMs vocabularies. The models are arranged in ascending order based on vocabulary size. Each cell represents the proportion of shared tokens between the horizontal and vertical models, relative to the vocabulary size of the vertical model.

2 Vocabulary Overlap Phenomenon

2.1 Impact of Vocabulary Distinction

Current LLMs accomplish various tasks through language generation, where LLMs receive the input prompt and generate succeeding tokens. Suppose the input tokens are $x_1, \cdots, x_{i-1}$, LLMs decode the next token x_i based on the conditional distribution $p(\cdot|x_{\leq i}) \in \mathbb{R}^{|V|}$ over the corresponding vocabulary.

However, different LLMs usually independently learn sentencepiece (Kudo and Richardson, 2018) models from different training corpora, leading to different vocabularies. For instance, the vocabulary size of LLaMA is 32,000, whereas ChatGLM has a vocabulary length of 125,696. Such a discrepancy makes the output distributions of different models noncomparable, thereby impeding direct ensembling, as commonly practiced in conventional classification tasks.

2.2 Overlap between Vocabularies

Although different LLMs have distinct vocabularies, given that these diverse vocabularies are learned from comparable corpora collected from the web, a substantial number of overlapping tokens naturally emerge. To illustrate this phenomenon, we record the rate of overlapping tokens between vocabularies of LLMs. As shown in Figure 3, the number of overlapping tokens is adequate. For example, TigerBot and LLaMA have 53% overlapping tokens. Intuitively, these overlapping tokens play a crucial role as a bridge to project diverse output distributions into a shared space and establish the corresponding relations, facilitating the ensemble of LLMs.

3 Our Method

EVA comprises two key components: *cross-model vocabulary alignment* (Section 3.1) and *LLMs ensemble* (Section 3.2). The framework is shown in Figure 2, *(a) cross-model vocabulary alignment* establishes the relations between tokens of distinct vocabularies. *(b) LLMs ensemble* projects the output distributions into the same space via the established vocabulary relations and achieves fine-grained ensembling at each generation step.

Considering a set of N large language models denoted as $\mathcal{M} = \{Q_1, Q_2, \cdots, Q_{N-1}, P\}$, where P represents the chosen pivot model. We empirically select the model with the largest vocabulary as the pivot model P.

3.1 Cross-Model Vocabulary Alignment

3.1.1 Vocabulary Projection

As shown in the upper part of Figure 2(a), We first utilize the overlapping tokens as supervised labels to map token embeddings from different models to a common vector space. Taking $N = 2$ as an example, let $\mathcal{V}^P$ and $\mathcal{V}^Q$ represent the vocabularies of the pivot model and the non-pivot model, and $\boldsymbol{E}^P$ and $\boldsymbol{E}^Q$ be the word embedding matrices of the respective models. The training objective is to find transformation matrices $\boldsymbol{U}_{QP}$ such that:

$$\boldsymbol{U}_{QP} = \underset{\boldsymbol{U}_{QP}}{\mathrm{argmin}} \sum_i \sum_j \mathcal{D}_{ij} \left\| \boldsymbol{E}^Q_{i*} \boldsymbol{U}_{QP} - \boldsymbol{E}^P_{j*} \right\|^2 \tag{1}$$

where $\mathcal{D}$ is the overlapping dictionary of $\mathcal{V}^Q$ and $\mathcal{V}^P$, and $\mathcal{D}_{ij} = 1$ indicates that the i-th word in $\mathcal{V}^Q$ and the j-th word in $\mathcal{V}^P$ are identical. We utilize the supervised setting of the open-source toolkit VecMap[1] to achieve the training process. This involves applying normalization, whitening, orthogonal mapping, re-weighting, and de-whitening operations to the word embeddings (Artetxe et al., 2018). The optimal $\boldsymbol{U}_{QP}$ minimizes the Euclidean distance between identical words from different model vocabularies in the mapped common space.

Subsequently, we establish vocabulary mappings between models based on the similarity relation-

[1] https://github.com/artetxem/vecmap

ships between tokens:

$$\boldsymbol{W}^{QP} = \operatorname{SIM}\left(\boldsymbol{E}^{Q}\boldsymbol{U}_{QP}, \boldsymbol{E}^{P}\right) \tag{2}$$

Specifically, we adopt the cross-domain similarity local scaling (CSLS) (Lample et al., 2018) score as the token similarity from $\mathcal{V}^{Q}$ to $\mathcal{V}^{P}$ and derive the similarity matrix $\boldsymbol{W}^{QP} \in \mathbb{R}^{|\mathcal{V}^{Q}| \times |\mathcal{V}^{P}|}$.

3.1.2 Noise Reduction

Since the similarity matrix obtained above is excessively large and contains substantial noise, we calculate the alignment across various similarity intervals (as shown in Table 1, with detailed analysis in Appendix A) and devise three steps to reduce noise and retain the pertinent and concise alignment information.

Step-1: Top-t Truncation. The complete similarity matrix is redundant, as each token should only align with a small subset of other tokens. Thus, for each token in $\mathcal{V}_{Q}$, we retain top-t tokens in $\mathcal{V}_{P}$ that exhibit the highest similarity to it.

$$\boldsymbol{W}_{ij}^{QP} = \begin{cases} \boldsymbol{W}_{ij}^{QP}, & \boldsymbol{W}_{ij}^{QP} \in \text{top-}t\left(\boldsymbol{W}_{i*}^{QP}\right) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Step-2: Threshold Truncation. When the similarity between two tokens is too low, aligning them becomes meaningless. Therefore, we set a threshold to discard the portion of similarity scores that are below the threshold.

$$\boldsymbol{W}_{ij}^{QP} = \begin{cases} \boldsymbol{W}_{ij}^{QP}, & \boldsymbol{W}_{ij}^{QP} \geq threshold \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Step-3: Variance Truncation. Through the observation of Table 1, we found that tokens without actual meaning exhibit similar and high similarity scores with multiple tokens, which cannot represent the semantic similarity. We use variance to determine and eliminate this noise, taking into account the number of non-zero similarity scores as well to avoid low variance resulting from insufficient quantity.

$$\boldsymbol{W}_{ij}^{QP} = \begin{cases} 0, & \operatorname{Var}\left(\boldsymbol{W}_{i*}^{QP}\right) \leq \sigma, \operatorname{count}\left(\boldsymbol{W}_{i*}^{QP} \neq 0\right) \geq c \\ \boldsymbol{W}_{ij}^{QP} & \text{otherwise} \end{cases} \tag{5}$$

Following these three processes, we obtain a sparse and efficient mapping matrix $\boldsymbol{W}^{QP}$, which is only about 1MB. This matrix maps the output distribution of the non-pivot model to the pivot model.

Range	Description	Percentage	Examples			
0.6 ~ 1.0	*Completely Alignment*	54.3%	0.90	_your	→	_your
	Meaningless Alignment	0.4%	0.77	<0x0D>	→	<0xF9>
0.4 ~ 0.6	*Semantic Alignment*	11.9%	0.40	_use	→	_uses
	With Minor Inconsistencies		0.51	_use	→	_Use
0.1 ~ 0.4	*Partial Alignment*	16.4%	0.32	_use	→	_utilize
	Cross-Lingual Alignment		0.32	_use	→	使用的
0.0 ~ 0.1	*Mis-Alignment*	17.1%	0.03	hex	→	菱形
			0.08	ică	→	겐

Table 1: Statistics of token alignment from LLaMA to Baichuan. Similarity scores are divided into four subsets based on alignment performances. We intend to retain the pairs highlighted in green and discard those highlighted in red.

3.2 LLMs Ensemble

As shown in Figure 2(b), given the mapping matrix (e.g., $\boldsymbol{W}^{12}$ and $\boldsymbol{W}^{32}$) from non-pivot models (Q_1 and Q_3) to the pivot model (Q_2), we align the output distribution of non-pivot models at the current time step with the pivot model.

$$p_{\ell}\left(\cdot \mid x_{<i}\right) = q_{\ell}\left(\cdot \mid x_{<i}\right) \boldsymbol{W}^{\ell\rho} \quad \forall \ell \neq \rho. \tag{6}$$

where ρ is the identifier for the pivot model, $q_{\ell}\left(\cdot \mid x_{<i}\right)$ and $p_{\ell}\left(\cdot \mid x_{<i}\right)$ separately denote the original output distribution of the ℓ-th model in $\mathcal{M}$ and its corresponding mapping in the unified space.

A straightforward ensemble approach involves deriving the succeeding token by averaging the mapped output distributions of all models:

$$p\left(\cdot \mid x_{<i}\right) = \frac{1}{N} \sum_{\ell=1}^{N} p_{\ell}\left(\cdot \mid x_{<i}\right) \tag{7}$$

However, this approach is susceptible to outliers, which can mislead overall judgments. Hence, we devise a filtering strategy to enforce a requisite consistency among tokens generated by diverse models. Specifically, if the top-1 token predicted by a model falls outside the top-n tokens predicted by any other model, it is excluded from the ensemble.

$$p(\cdot \mid x_{<i}) = \frac{1}{\sum_{\ell=1}^{N} I\left(\ell\right)} \sum_{\ell=1}^{N} I\left(\ell\right) \cdot p_{\ell}(\cdot \mid x_{<i}) \tag{8}$$

$$I\left(\ell\right) = \begin{cases} 1 & \text{if } \operatorname{top-1}\left(p_{\ell}\right) \in \bigcup_{o \neq \ell} \operatorname{top-}n\left(p_{o}\right) \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

As shown in Figure 2(b), When we directly average the probability distributions of the three models,

	Machine Translation				Data-to-Text
	Flores-Zh-En		Flores-En-Zh		E2E
System	BLEU	ChrF	BLEU	ChrF	ROUGE-L
LLaMA2-7B-Chat	24.49	52.37	13.99	22.78	33.58
ChatGLM2-6B	24.17	51.71	<u>23.77</u>	31.14	<u>40.57</u>
Baichuan2-7B-Chat	<u>29.18</u>	56.63	<u>30.56</u>	35.95	30.61
InternLM-7B-Chat	22.59	51.81	23.58	31.18	<u>41.11</u>
TigerBot-7B-Chat-V3	<u>26.81</u>	54.34	<u>30.59</u>	35.58	20.37
Vicuna-7B-V1.5	<u>26.37</u>	53.83	20.61	28.98	<u>37.08</u>
ChineseAlpaca2-7B	<u>28.54</u>	54.42	<u>27.66</u>	33.87	<u>38.24</u>
MBR (Farinhas et al., 2023)	30.72(+*1.54*)	56.97(+*0.34*)	31.29(+*0.70*)	36.84(+*0.89*)	41.47(+*0.36*)
PairRanker (Jiang et al., 2023)	29.73(+*0.55*)	56.58(- *0.05*)	29.45(- *1.41*)	35.25(- *0.70*)	38.90(- *2.21*)
LLM-Blender (Jiang et al., 2023)	27.18(+*1.54*)	53.89(+*0.34*)	-	-	**43.62(+*2.51*)**
EVA (*ours*)	**31.16(+*1.98*)**	**57.77(+*1.14*)**	**32.68(+*2.09*)**	**38.16(+*2.21*)**	42.62(+*1.51*)

Table 2: Main results of machine translation and data-to-text tasks. Best results are highlighted in bold and the model employed within the ensemble is underlined for distinction. LLM-Blender is not trained on Chinese corpora, thus unable to produce meaningful translations from English to Chinese.

the ensemble result is *_Typ*. Upon incorporating the filtering strategy with $n = 3$, the top-1 token for model Q_1 is *_Des*, which is not within the top-3 tokens of Q_2 or Q_3, hence excluded from ensemble. On the contrary, the top-1 token of Q_2 is *_Typ*, falling within the top-3 tokens of Q_1 and Q_3. The top-1 token of Q_3 is *und*, within the top-3 tokens of Q_2. Consequently, we ensemble only Q2 and Q3, resulting in the correct output *und*.

4 Experimental Settings

4.1 Datasets

We evaluate our proposed ensemble method from the perspective of natural language generation (NLG) and reasoning. For NLG, we choose machine translation (Flores-101 Chinese↔English) (Goyal et al., 2022) and data-to-text generation task (E2E) (Novikova et al., 2017). For commonsense reasoning, we employ Natrual Question (NQ) (Kwiatkowski et al., 2019) and TriviaQA (Joshi et al., 2017) for evaluation. For arithmetic reasoning, we adopt GSM8K (Cobbe et al., 2021), AddSub (Hosseini et al., 2014) and ASDiv (Miao et al., 2020) for evaluation.[2]

4.2 Candidate LLMs

We select seven open-source chat LLMs of approximately 7B size as the candidate LLMs for the ensemble as follows: LLaMA2-7B-Chat (Touvron et al., 2023), ChatGLM2-6B (Zeng et al., 2022), Baichuan2-7B-Chat (Baichuan, 2023), InternLM-7B-Chat (Team, 2023), TigerBot-7B-Chat-V3[3], Vicuna-7B-V1.5 (Chiang et al., 2023), ChineseAlpaca2-7B (Cui et al., 2023).

These models originate from distinct institutions and have different vocabularies. Each model is aligned by supervised instruction tuning and leverages large-scale, high-quality data to establish a powerful knowledge base, thus performing well on public benchmarks.

4.3 Baselines

We compare EVA with existing selection-based methods and fusion-based methods.

MBR Farinhas et al. (2023) use the average similarity between one output and the rest to select the best output. We utilize BERTScore to measure the similarity between two outputs to adapt across different tasks.

PairRanker Jiang et al. (2023) employ a specialized pairwise comparison method to distinguish subtle differences between candidate outputs.

LLM-Blender Jiang et al. (2023) utilize a 3b-parameter model fine-tuned on an instruction dataset to merge the ranking outcomes from PairRanker and generate the final output.

4.4 Implement Details

Configurations. For each task, we selected the top-performing four models out of seven for the ensemble. We employ greedy decoding in all

[2]Please refer to the appendix B for details of the tasks.

[3]https://github.com/TigerResearch/TigerBot

System	Commonsense Reasoning		Arithmetic Reasoning		
	NQ	TriviaQA	GSM8K	AddSub	ASDiv
LLaMA2-7B-Chat	28.59	62.77	24.64	55.05	55.02
ChatGLM2-6B	14.93	31.77	30.78	49.54	60.52
Baichuan2-7B-Chat	24.07	55.62	29.95	55.05	58.74
InternLM-7B-Chat	17.20	44.05	32.30	62.39	58.58
TigerBot-7B-Chat-V3	11.33	23.87	27.29	24.77	41.75
Vicuna-7B-V1.5	26.84	61.21	18.88	44.04	44.17
ChineseAlpaca2-7B	22.58	50.86	13.12	23.85	28.64
MBR (Farinhas et al., 2023)	28.61(*+0.02*)	63.75(*+0.98*)	36.47(*+4.17*)	58.72(*-3.67*)	61.00(*+0.48*)
PairRanker (Jiang et al., 2023)	29.81(*+1.22*)	63.24(*+0.47*)	39.58(*+7.28*)	58.72(*-3.67*)	62.62(*+2.10*)
LLM-Blender (Jiang et al., 2023)	**32.19(*+3.60*)**	62.77(*+0.00*)	34.80(*+2.50*)	58.72(*-3.67*)	59.71(*-0.81*)
EVA (*ours*)	30.64(*+2.05*)	**64.29(*+1.52*)**	**42.91(*+10.61*)**	**64.22(*+1.83*)**	**65.05(*+4.53*)**

Table 3: Main results of commonsense reasoning (measured by Exact Match) and arithmetic reasoning tasks (measured by Accuracy). Best results are highlighted with bold and the model employed within the ensemble is underlined for distinction.

experiments since it generally produces higher-quality outputs. To mitigate the impact of long-tail noise accumulation, we perform top-k truncation on the original output distributions of each candidate model.

Hyperparameters. Unless otherwise stated, the same hyper-parameters are used in all experiments. Concerning the three steps mentioned in Section 3.1.1, we empirically set $t = 10$, $threshold = 0.1$, $sigma = 0.0001$ and $c = 5$ based on observations. For top-k truncation on the output distributions, we always set $k = 320$ for the main results in the paper, which is quite robust across various tasks. Due to variations in task characteristics, we empirically set $n = 40$ for NLG tasks and $n = 3$ for reasoning tasks in our experiments.

Prompting. For machine translation tasks, we utilize a 4-shot in-context learning setting, whereas for other tasks, we conduct zero-shot inference. Additionally, we include a chain of thought prompt in arithmetic reasoning tasks. We adhere to the specific format required by each chat model and employ task-specific prompts.

5 Experimental Results

The main results on NLG tasks and reasoning tasks are shown in Table 2 and Table 3, respectively.

EVA demonstrates superiority. Our proposed EVA consistently outperforms individual models and selection-based ensemble methods across all types of tasks, showcasing its cross-task versatility. Especially in the GSM8K task, EVA achieves a significant 10.61 improvement compared with the best-performing individual model, ChatGLM2-6B. Remarkably, EVA also outperforms LLM-Blender, which leverages an additional 3b-parameter fusion model, on six out of eight tasks, demonstrating the effectiveness of our approach. We attribute this success to the EVA which conducts fine-grained ensembles at each generation step, ensuring precision in token generation and thereby mitigating subsequent errors in the generation of following tokens.

LLMs have diverse strengths and weaknesses. Additionally, observing the performance of individual models on each task, we find that no models participate in every task ensemble. However, each model contributes to at least three task ensembles. This highlights the distinct knowledge possessed by each LLM and emphasizes the significance of ensembling LLMs.

6 Analysis

6.1 Effect of Model Filtering Intensity

Recall in Section 3.2, we introduced the hyperparameter n as a way to control how strict our model filtering is. In this section, we investigate the sensitivity of our method to n. As shown in Table 4, all tasks, except for arithmetic reasoning, are not sensitive to n. Any variations within these ranges lead to reasonable performance. For E2E tasks, a looser filtering approach results in better text flexibility, leading to slight performance improvements. Notably, arithmetic reasoning tasks exhibit unique behavior. Tighter filtering significantly improved

	Arithmetic Reasoning			Commonsense Reasoning		Machine Translation		Data-to-Text
System	**GSM8K**	**AddSub**	**ASDiv**	**NQ**	**TriviaQA**	**Zh-En**	**En-Zh**	**E2E**
$\text{EVA}_{n=40}$	31.39	58.72	61.33	30.86	64.59	31.16	32.68	42.62
$\text{EVA}_{n=20}$	31.54	59.63	60.68	30.61	64.48	31.20	32.78	42.64
$\text{EVA}_{n=10}$	35.03	59.63	63.27	30.83	64.41	31.13	32.78	42.59
$\text{EVA}_{n=5}$	37.30	62.39	65.86	30.75	64.26	31.01	32.67	42.00
$\text{EVA}_{n=3}$	42.91	64.22	65.05	30.64	64.29	31.13	32.64	41.98

Table 4: Effect of model filtering intensity.

	Flores-Zh-En
Input	他补充道：“我们现在有 4 个月大没有糖尿病的老鼠，但它们曾经得过该病。”
Output prefix	He added, "We have 4-month-
Continuations	old mice that have never had diabetus, but they have had it in the past."
Next token distribution	’old’, ’olds’, ’ old’, ’Old’, ’older’, ’ Old’, ’OLD’, ’ olds’, ’旧’, ’olding’, ...
	GSM8K
Input	Janet\u2019s ducks lay 16 eggs...How much in dollars does she make every day at the farmers’ market?
Output prefix	First, we need to determine how many eggs Janet has left after she eats three for breakfast and bakes
Continuations	four muffins...The answer: 10.
Next token distribution	’ four’, ’ muff’, ’ the’, ’ some’, ’ ’, ’ a’, ’ three’, ’ her’, ’ for’, ’ two’, ...

Table 5: Examples of the distribution of the next token for GSM8K and Flores-Zh-En tasks.

the performance on the GSM8K, AddSub, and ASDiv datasets.

We believe that these differences in sensitivity arise from the nature of the tasks. The outputs of tasks other than arithmetic reasoning exhibit a certain level of determinism (specific answers to questions, sentences conveying the same semantics in the target language, or restaurant reviews containing specific information). Hence, the output distributions of different models will demonstrate strong consistency. As illustrated in Table 5, in the case of Chinese→English translation task, models exhibit marginal differences in predicting the next token. As a result, the filtering strategy has minimal impact here. In contrast, arithmetic reasoning tasks generate a series of intermediate reasoning steps. Since the same answer can be derived from multiple distinct reasoning paths, the output tokens exhibit inconsistency. As shown in Table 5, there is a significant semantic difference between the distributions of the next token in the GSM8K task. Employing tighter filtering here can effectively eliminate models generating unfaithful tokens.

To verify our hypothesis, we conduct further experimental analysis on tasks with the highest sensitivity (GSM8K) and lowest sensitivity (Machine Translation). Since tokens are very fine-grained units, spelling variations can directly represent semantic differences. Hence, We specifically define diversity as the average edit distance between the top-n tokens and the top-1 token generated by a model. We conducted a statistical analysis on the outputs at 10,000 positions in both datasets. As depicted in Figure 4, across various top-n ranges, the edit distance for the GSM8K task consistently exceeds that of Flores, confirming our hypothesis.

6.2 Effect of Number of Ensemble Models

As shown in Figure 5, we demonstrate the changes in ensemble performance on the GSM8K dataset as the number of ensemble models increases. We observe that even as the performance of newly added models gradually decreases, EVA consistently brings further improvements, which indicates that EVA effectively unleashes the complementary potential of different models by unifying the vocabulary space. Moreover, this confirms that different models possess distinct knowledge. The knowledge within underperforming models is not entirely covered by better-performing ones, leaving space for further enhancement via ensembling.

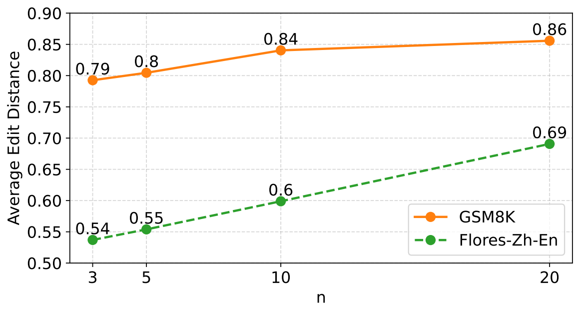

Figure 4: The average edit distance of GSM8K (orange solid line) and Flores-Zh-En (green dotted line) tasks across various top-n ranges. The average edit distance indicates the output token diversity.

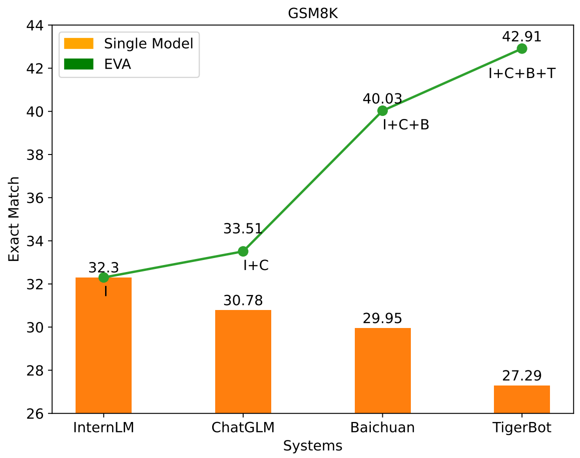

Figure 5: Effect of number of ensemble models. The orange bars represent the performance of individual models, while the green line denotes the result of ensembling multiple models, denoted by their initials.

7 Related Work

Ensemble learning is a widely adopted technique to improve performance on a given task and provide robust generalization by leveraging multiple complementary systems. Existing ensemble methods can be divided into two categories: selection-based ensemble and generation-based ensemble.

Selection-based Ensemble Selection-based ensemble methods select the best output from multiple outputs. Shnitzer et al. (2023) employs benchmark datasets to learn a router model responsible for selecting the best LLM out of a collection of models for a given task. FrugalGPT (Chen et al., 2023) calls LLMs sequentially until a dedicated scoring model deems the generation acceptable to effectively and efficiently leverage different LLMs. Ravaut et al. (2022a);Liu and Liu (2021);Liu et al. (2022) train dedicated scoring or ranking models for text summarization. Farinhas et al. (2023) demonstrated that minimum Bayes risk decoding is an effective ensemble method for LLM-based machine translation.

However, such methods are limited by the output quality of the candidate models and are unable to generate outputs superior to those of existing models. Nevertheless, the distinctions among candidates could be quite subtle. A model's output might outperform one part compared to another model's output yet lag behind in other parts. Selecting among existing answers limits the release of the complementary potential of the ensemble.

Fusion-based Ensemble Compared to selection-based methods, fusion-based ensemble approaches bypass the limitation of existing complete outputs, often yielding superior outputs. Jiang et al. (2023) presents a general ensemble framework utilizing a pair ranker to filter the top K optimal outputs, followed by a fusion model to merge and generate the final output. Furthermore, Izacard and Grave (2021) enhances question answering by amalgamating retrieved text, while Ravaut et al. (2022b) applies generative fusion methods to text summarization. However, a fusion model typically needs to have a size comparable to the base model. For instance, Jiang et al. (2023) employs a 3B-sized model as a fusion model, significantly elevating the training and inference costs.

Our proposed EVA conducts fine-grained ensemble at each generation step, not only obtaining new results distinct from individual model outputs but also incurring almost negligible training costs for mapping vocabularies. Furthermore, our approach exhibits strong performance without the need for training on specific task datasets, demonstrating excellent generalization capabilities.

8 Conclusion

In this paper, we propose a novel ensemble method named EVA, which effectively bridges the lexical gap between different LLMs and facilitates fine-grained ensemble at each generation step. Compared to ensemble methods that select or fuse completely generated results, EVA provides intermediate ensemble results to candidate models, enabling them to benefit from higher-quality output prefixes, thereby unleashing their complementary potentials. Experimental results on NLG tasks and reasoning tasks demonstrate the effectiveness of our approach, which significantly improves overall performance on various natural language processing tasks.

Limitation

Due to the inherent nature of the ensemble, our approach, like previous ensemble methods, requires performing inference N times when ensembling N models. However, we want to argue that those inferences can be executed in parallel because they are totally independent.

References

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Baichuan. 2023. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*.

Lingjiao Chen, Matei Zaharia, and James Zou. 2023. Frugalgpt: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

OpenCompass Contributors. 2023. Opencompass: A universal evaluation platform for foundation models. https://github.com/open-compass/opencompass.

Yiming Cui, Ziqing Yang, and Xin Yao. 2023. Efficient and effective text encoding for chinese llama and alpaca. *arXiv preprint arXiv:2304.08177*.

António Farinhas, José GC de Souza, and André FT Martins. 2023. An empirical study of translation hypothesis ensembling with large language models. *arXiv preprint arXiv:2310.11430*.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.

Mohammad Javad Hosseini, Hannaneh Hajishirzi, Oren Etzioni, and Nate Kushman. 2014. Learning to solve arithmetic word problems with verb categorization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 523–533.

Gautier Izacard and Édouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023. Llm-blender: Ensembling large language models with pairwise comparison and generative fusion. In *Proceedings of the 61th Annual Meeting of the Association for Computational Linguistics (ACL 2023)*.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *International Conference on Learning Representations*.

Yixin Liu and Pengfei Liu. 2021. Simcls: A simple framework for contrastive learning of abstractive summarization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 1065–1072.

Yixin Liu, Pengfei Liu, Dragomir Radev, and Graham Neubig. 2022. BRIO: Bringing order to abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2890–2903, Dublin, Ireland. Association for Computational Linguistics.

Keming Lu, Hongyi Yuan, Runji Lin, Junyang Lin, Zheng Yuan, Chang Zhou, and Jingren Zhou. 2023. Routing to the expert: Efficient reward-guided ensemble of large language models. *arXiv preprint arXiv:2311.08692*.

Shen-Yun Miao, Chao-Chun Liang, and Keh-Yih Su. 2020. A diverse corpus for evaluating and developing english math word problem solvers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 975–984.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017. The e2e dataset: New challenges for end-to-end generation. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 201–206.

Maja Popović. 2015. chrf: character n-gram f-score for automatic mt evaluation. In *Proceedings of the tenth workshop on statistical machine translation*, pages 392–395.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Mathieu Ravaut, Shafiq Joty, and Nancy Chen. 2022a. Summareranker: A multi-task mixture-of-experts re-ranking framework for abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4504–4524.

Mathieu Ravaut, Shafiq Joty, and Nancy Chen. 2022b. Towards summary candidates fusion. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8488–8504.

Tal Shnitzer, Anthony Ou, Mírian Silva, Kate Soule, Yuekai Sun, Justin Solomon, Neil Thompson, and Mikhail Yurochkin. 2023. Large language model routing with benchmark datasets. *arXiv preprint arXiv:2309.15789*.

InternLM Team. 2023. Internlm: A multilingual language model with progressively enhanced capabilities. `https://github.com/InternLM/InternLM`.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2022. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*.

Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A Smith. 2023. How language model hallucinations can snowball. *arXiv preprint arXiv:2305.13534*.

A Effectiveness of Vocabulary Projection

We observe the results of vocabulary projection between different models and analyze the relationship between similarity scores and projection phenomena. In Table 1, we illustrate the observed results using the projection from LLaMA2-7B-Chat (Touvron et al., 2023) to Baichuan2-7B-Chat (Baichuan, 2023) as an example. For token pairs with similarity scores between 0.6 and 1, most of them are completely aligned. It should be noted that some special tokens demonstrate high similarity but lack semantic meaning in their alignment, clustering around a similarity score of 0.77. As the similarity decreases to the range of 0.4 to 0.6, minor inconsistencies that do not affect semantics begin to appear, such as singular and plural forms, uppercase and lowercase distinctions. Furthermore, as the similarity reduces to 0.1 to 0.4, phenomena shift towards partial alignment and cross-lingual alignment. When the similarity drops below 0.1, the majority of alignments are meaningless. Overall, approximately 82% of the vocabulary achieved meaningful mappings, indicating the effectiveness of our vocabulary projection.

B Datasets

GSM8K is a multi-step arithmetic reasoning dataset (Cobbe et al., 2021), consists of high quality linguistically diverse grade school math word problems created by human problem writers. Evaluation metrics are Accuracy.

AddSub consists of addition–subtraction word problems(Hosseini et al., 2014). Evaluation metrics are Accuracy.

ASDiv is a diverse (in terms of both language patterns and problem types) English math word problem corpus(Miao et al., 2020). Evaluation metrics are Accuracy.

Natural Questions (NQ) is a question answering dataset in which questions consist of real anonymized, aggregated queries issued to the Google search engine (Kwiatkowski et al., 2019). Following OpenCompass (Contributors, 2023), we repurposed the validation set for testing purposes. Evaluation metrics are Exact Match.

TriviaQA contains questions authored by trivia enthusiasts (Joshi et al., 2017). Again, we use the validation as test. Evaluation metrics are Exact Match.

Flores101 is a widely used benchmark dataset for machine translation (Goyal et al., 2022). Here we use the Chinese-English split and English-Chinese split for evaluation. Evaluation metrics are BLEU (Post, 2018) and ChrF (Popović, 2015).

E2E is a data-to-text dataset (Novikova et al., 2017). The input is a set of key-value attribute pairs, and the output is a description of the restaurant. Evaluation metrics are ROUGE-L[4].

[4] https://github.com/GrittyChen/NLG-evaluation