ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 2 - 2026

Synergistic Reasoning between Large Language Models and Conventional NLP Methods in Low-Resource Settings

Hanyang He[1,*]

[1] HuaXin Institute of Software Engineering, Tianjin University of Technology, Tianjin, 300000, China

SUMMARY: *The NLP systems with low resources frequently undergo the combined restrictions of few labeled data, unstable word shape expressions, and not whole knowledge covering ranges. Under these circumstances, the reasoning of pure large language models is easy to be affected by confidence drift and evidence separation, while traditional NLP methods still hold strong points only under local restrictions and sparse matching. This manuscript puts forward CLEAR, which is a cooperative inference framework that unites lexical fixed points, structure prior knowledge, sparse search proof, and multilingual LLM inference into one single decision loop. The framework changes traditional outputs into one structured evidence card and furthermore combines consistent gating, limited modification, and high-confidence pseudo-label feedback for repeated improvement. This research has been arranged under unified 16/64/256-shot settings, and it covers named entity recognition, intent classification, question answering and sentiment analysis by using multilingual public benchmark data sets. The obtained writing and graphic plan displays that the main benefits come from two mutual supplementary effects: traditional NLP compresses the candidate space into verifiable local decisions, and the LLM branch solves semantic ambiguity and cross-language implicit relations inside this restricted space. Extra ablation and efficiency study show that evidence compression and gate control hold a determining function in the balance among accuracy, calibration, and inference cost. Therefore, the framework we put forward is fitting for low-resource deployment situations which need controllable forecast outcomes rather than generation without constraints.*

KEYWORDS: *low-resource settings; large language models; conventional NLP; structured evidence; multilingual reasoning*

1 Introduction

Local government Q&A, cross-border e-commerce customer service, regional media monitoring, and local knowledge retrieval often operate under the same constraints: there are few annotated samples available, and the text is interspersed with colloquialisms, provincialisms, transcription variants, and script switches, but the system still has to output verifiable entity boundaries, intent labels, and answer fragments. For high-resource languages, such problems can usually be slowly corrected through continuous annotation and large-scale iteration; for low-resource languages, long data collection cycles, high labor costs, and fluctuating corpus distributions make it difficult to form a sufficiently thick supervised set before going live for many tasks. Such pressures are particularly prominent in African news

*hhyang1651@stud.tjut.edu.cn
https://doi.org/10.65102/is2026772
Published: 30 April, 2026

monitoring, Southeast Asian local customer service, and regional medical hotlines, where text lengths are short, the same concept is often spelled in multiple ways, and manual proofreading queues often fail to cover the full range of language variants. As a result, models often miss at high-frequency locations in the business: entity recognition slices the name of an organization into place names and function words, intent categorization drifts back and forth between similar service labels, and question-and-answer systems generate fluent answers but fail to give reliable sources. What's missing in low-resource scenarios is a mechanism to maintain decision boundaries under conditions of evidence scarcity and expressive instability.

The multilingual big language model, therefore, gives a novel starting point for the solution of this problem. Open multilingual teaching models have expanded inference and generation abilities to a broader language covering scope, and low resource languages can obtain available outcomes from fewer sample hints; Linguistically various signals take English proficiency as a fulcrum, they have likewise proved that cross-language movements can still obtain obvious benefits in low resource environments [1-3]. These progression have altered the beginning point of low-resource NLP: whereas in previous times, models must be trained one by one for every task, nowadays label determination, piece picking, and cross-sentence summarization can all be finished by utilizing an unified generation interface, hence a small number of task data can be utilized to set scenario restrictions. In the same time, the attention on the business side has had a change: whether it can be controlled, whether it can be rolled back, and how to explain it have begun to be as important as the accuracy rate itself. Especially in the stage that system begins operation.

This direction quickly shifted from “can it work” to “can it be stably deployed”. Around low-resource language alignment, researchers started to construct self-built cross-language instruction sets, pivot language-assisted alignment, multi-language feedback, and multi-language preference optimization, in the hope of narrowing the gap between the English-dominant training distribution and the real target language [4-7]. At the same time, reliability issues have been exposed in advance. Multilingual rejection and causal rejection studies have shown that the main risk in low-resource languages comes not only from knowledge gaps, but also from a disconnect between confidence and correctness: models may give high-confidence answers when there is insufficient evidence, or may show too much subjective consistency in semantically close labeling [8, 9]. Once the system needs to enter high-risk scenarios such as political consulting, financial services, or medical triage, these kinds of errors are difficult to fully remediate through simple post-human auditing.

Under this situation, the engineering worth of conventional NLP methods has appeared again. When resources are not plenty, cross-language movement, continuous pre-training, and local structure modeling can still give constraint information which generative models are not good at, such as boundary a posteriori, word anchors, alias mapping, label prototyping, and sparse recall [10, 11]. This information is not enough all alone to make complex reasonings by itself, hence it can narrow the candidate space before it passes semantic judgments over to more powerful language models. The actual question is that current mixed methods are inclined to use traditional branches only as search attachments or hint prefixes, being short of unified proof objects, clear gating, and fault retreat interfaces. The candidate boundary, local rule and inter-class restriction that traditional roads give have not been formally made into one part of the decision chain, hence the output that LLM gives lacks the check of system-level consistency. The system that we get seems to be “multi-module collaborative”, but in its essence it is still a single path that generates results with some loose hints.

This gap is further magnified under low resource conditions. First, although sequence annotation, short text categorization, and Q&A tasks are different in form, they all need to make joint judgments on local evidence and global semantics; if each type of task is designed with

its own set of hybrid logic, it is difficult to migrate the approach, and the cost of deployment and maintenance will rise rapidly. Second, low-resource systems are more concerned about when to give the results to the LLM, when to ask it to rejudge among restricted candidates, and when to fall back directly to the traditional path, a strategic issue that is closer to real scenarios than a single bare-point boost. Third, unlabeled corpus is not scarce in many low-resource languages, what is scarce is high-quality manual annotation; if the system is unable to identify which high-confidence results are worthy of backward training, it will be difficult to convert the limited annotation budget into sustained gains.

According to the above judgment, this article reduces the scope of the research problem to: how to convert the lexical, structural, retrieval proofs produced by traditional NLP into inference objects which can be used by LLMs, checked by the system, and shared among multiple tasks under the situations of low-resource languages and low annotation funds, and therefore, build a cooperative decision-making mechanism which can be rolled back and expanded. Regarding this problem, this article puts forward the CLEAR framework. This framework firstly carries out compression on boundary posterior probability, carries out marking on prior label, candidate fragments, and sparse recall results into structured evidence cards, hence then incorporates traditional branches and generative branches into the identical decision loop through consistency gating, constrained corrections, and high-confidence pseudo-labeling reflow. When we compare with projects that only depend on LLM or old models alone, CLEAR pays attention to how the two kinds of abilities are allocated in the situation of low resources: traditional roads take charge of candidate collection and partial verifiable limits, and LLM takes charge of cross-language semantic supplement, unclear meaning explanation, and hidden relation deduction.

The work in this paper focuses on three aspects. One, to construct a unified data organization for low-resource multi-task environments by compressing supervised core sets, unlabeled pools, and sparse retrieval indexes into the same object layer to reduce the data interface differences between different tasks. Second, propose a collaborative reasoning mechanism for structured evidence, so that entity recognition, intent classification, question and answer, and sentiment classification can share the same gating and correction logic. Third, systematic analysis of collaborative gains, reliability changes, cost locations and deployment boundaries under unified 16, 64 and 256 labeling budgets closes the method mechanism, result interpretation and application implications in the same set of diagrams, and provides a direct basis for risk stratification and threshold configuration in low-resource scenarios.

2 Methods

2.1 Low-Resource Task Construction and Data Organization

The cooperative inference work under low-resource condition, the most primary thing it depends on is the method that how the objects get arranged. It has been proven that translation combination, knowledge combination, retrieval-raised hints, code-raised syntactic resources, and context study for extremely low-resource languages all reduce the performance decline from not enough supervision through different methods [12-18]. These research results give a shared conclusion: the thing that really has migration value is the capability to place local restrictions, candidate information, and cross-language semantics onto the identical decision interface. In this research, we therefore do not construct independent procedures from a alone task, but rather define unified data entities before compressing different tasks inside a shared interface.

The main experiment covers four types of tasks and keeps an external validation set. MasakhaNER 2.0 for named entity recognition, MASSIVE for intent classification, TyDi QA for extractive Q&A, NusaX for sentiment classification, and Indic-QA for external validation of Q&A [19-23]. These datasets simultaneously cover sequence annotation, short text categorization, and evidence fragment extraction tasks, and are suitable for testing whether collaborative mechanisms can be reused across tasks. The usage of each dataset and the caliber of low resource division are shown in Table 1.

Table 1: Datasets and Low-Resource Split Settings

Dataset	Task	Language Coverage	Official Scale	Role in This Paper
MasakhaNER 2.0	Named entity recognition	20 African languages	Human-annotated multilingual NER corpus	Main sequence labeling benchmark
MASSIVE	Intent classification	51 languages	1M utterances, 18 domains, 60 intents	Main intent benchmark
TyDi QA	Extractive question answering	11 languages	204K question-answer pairs	Main QA benchmark
NusaX	Sentiment classification	10 Indonesian local languages	Multilingual parallel sentiment corpus	Robustness benchmark
Indic-QA	Context-grounded question answering	11 Indic languages	Multilingual QA benchmark	External validation

The low-resource set follows a uniform budget. For each language-task combination, only 16, 64 or 256 manually labeled samples are sampled from the training set to form the supervised core set; the validation and test sets are kept in the official division; the unsampled training corpus is not directly discarded, but is divided into the unlabeled pool and sparse retrieval index according to the usage. This is because this paper focuses on how different capabilities work together when the annotation budget is limited; at the same time, low-resource scenarios usually have no shortage of unlabeled text, and retaining the remaining corpus is conducive to subsequent retrieval and pseudo-annotation reflow.

The preprocessing follows the unicode approach. First, Unicode normalization, full- and half-corner unification, punctuation cleaning and duplicate blank compression are performed on the original text to reduce the interference of script noise on character statistics. Second, the original script is retained without uniform romanization to avoid further smoothing out the already thin word shape differences. Again, under the condition of unstable lexer or ambiguous language boundary, character n-gram statistics, lexical hits and local positional features are preferentially extracted without basing the method on the assumption of high-quality lexers. For question-answer tasks, the retrieval index preserves question, context, and answer boundary information; for sequence annotation tasks, the index additionally preserves entity span and alias mappings; and for categorization tasks, the index preserves labeled prototypical phrases and high-weighted trigger words. After this process, the original samples of different tasks are organized into three parts: input text - task label space - external evidence cache.

Further, this paper splits each sample into three maintainable objects: a supervised core set $\mathcal{S}$, an unlabeled pool $\mathcal{U}$, and a sparse retrieval index $\mathcal{R}$. The supervised core set carries only manual annotations and is used to learn the smallest usable boundary under low budget; the

unlabeled pool holds unlabeled text for subsequent pseudo-annotation reflow; and the sparse retrieval index is responsible for providing instance-level similarity samples, candidate answer snippets, and labeled prototypes to support. The overall structure of this organization is given in Figure 1.

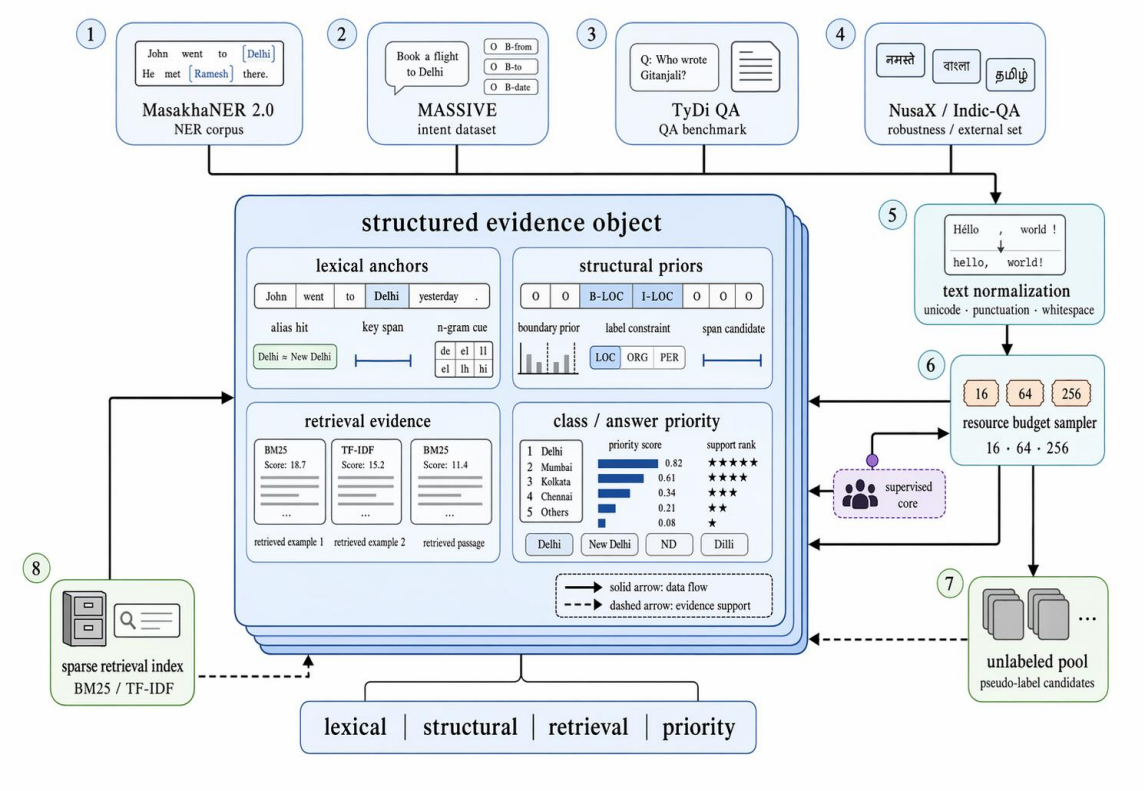

Figure 1: Low-Resource Sample Organization and Structured Evidence Construction.

In Figure 1, the raw corpus first enters the unified normalization module, then diverts to three types of objects based on budget sampling and task type, and finally converges into the construction interface for structured evidence objects. In this way, what traditional NLP produces is no longer just discrete predictions, but a collection of evidence that can be traced to the source text, boundary candidates, and labeled a priori.

This object-oriented design also assumes the role of cross-task alignment. Sequence annotation, short text categorization, and extractive Q&A vary greatly in labeling form, but they can all be abstracted as combinations of "input + candidate space + local constraints + semantic discriminations". As long as this information can be encoded uniformly at the object level, subsequent gating, fallback, and pseudo-annotation rules do not need to be rewritten separately for each type of task.

2.2 Structured Evidence–LLM Synergistic Reasoning Framework

After completing the object organization, the next step is to transcribe the local information generated by traditional branching into LLM-consumable and system-checkable evidence objects. In this paper, we name this mechanism CLEAR. its core is to make the two types of paths work around the same piece of structured evidence and form a unified decision at the gating level. For named entity recognition, the traditional branch consists of character n-gram linear chaining CRF, alias dictionary matching, and spanning a posteriori; for intent and sentiment tasks, the traditional branch consists of TF-IDF linear classifiers, labeled prototype

similarity, and high weighted trigger words; and for question-answer tasks, the traditional branch consists of BM25 sparse retrieval, candidate fragment ranking, and answer boundary a priori. Different tasks share the same principle: the traditional branch only outputs verifiable local evidence and does not directly occupy the final decision.

Eq. (1) defines the structured evidence object corresponding to sample x_i.

$$E_i = Concat\left(e_i^{\text{lex}}, e_i^{\text{str}}, e_i^{\text{ret}}, e_i^{\text{pri}}\right) \tag{1}$$

where E_i denotes the structured evidence card of the ith sample; e_i^{lex} is the lexical anchors, including character n-gram high-weight fragments, alias dictionary hits, and local keywords; e_i^{str} is the structural a priori, including entity boundary a posteriori, label transfer constraints, and syntactic templates; e_i^{ret} is the sparse candidate examples or candidate answer fragments returned by the search; e_i^{pri} is the category a priori or candidate prioritization given by traditional branching. Evidence cards use a fixed field order in the cue word, keeping only short textbook fragments, boundary positions, and candidate labels to control the cue length.

The LLM branch takes the original input and the evidence card as common inputs and outputs the candidate result y_i^L and its confidence c_i^L. Traditional branch synchronization gives the prediction y_i^T and confidence (c_i^T. In this paper, we use consistency gating to synthesize evidence coverage, two-branch consistency, and two types of confidence, and the gating scores are defined in equation (2).

$$g_i = \sigma(\alpha q_i + \beta a_i + \gamma c_i^L + \delta c_i^T) \tag{2}$$

where g_i is the consistency gating score of the first i sample; q_i denotes the evidence coverage, which is used to measure the degree of consistency between the output of the LLM and the evidence card; a_i denotes the consistency score between the LLM and the traditional branch; c_i^L and $(\alpha q_i + \beta a_i + \gamma c_i^L + \delta c_i^T)$ are the confidence levels of the two branches, respectively; $\alpha, \beta, \gamma, \delta$ is the weight estimated based on the development set; $\sigma(\cdot)$ is the weight estimated based on the development set; $\sigma(\cdot)\gamma, \delta$ is the weight estimated based on the development set; and α, β, γ, δ are the weights estimated based on the development set; $\sigma(\cdot)$ is the Sigmoid function. For NER, a_i consists of span overlap together with labeling consistency; for classification tasks, a_i consists of labeling identity together with prototype similarity; and for QA, a_i takes into account both answer string overlap, source fragment consistency, and retrieval support strength.

The decision logic after gating is designed with dual thresholds to distinguish the three states of direct acceptance, restricted rejudgement, and traditional fallback, as defined in Equation (3).

$$\hat{y}_i = \begin{cases} y_i^L, g_i \geq \tau_1, \\ \phi(y_i^L, E_i), \tau_0 \leq g_i < \tau_1, \\ y_i^T, g_i < \tau_0, \end{cases} \qquad 0 < \tau_0 < \tau_1 < 1 \tag{3}$$

where $\hat{y}_i$ is the final output, τ_0 and τ_1 are the double thresholds, and $\phi(\cdot)$ is the constrained correction operator. The system accepts the LLM output when the gating score is above τ_1, triggers the constrained correction when it falls in the middle interval, and falls back to the conventional path when it is below τ_0. The implementation of the constrained correction varies with the task. For NER, the corrector allows reselection of boundaries and labels only within the set of candidate spans; for the Intention and Emotion task, the corrector compresses

the label space to the top-k candidates given by the traditional branch; for QA, the corrector requires that the answer must come from the retrieval of support fragments and prioritizes the retention of text fragments that can be accurately aligned. Figure 2 illustrates the relationship between evidence cards, dual branching, gating, and reflow memory.

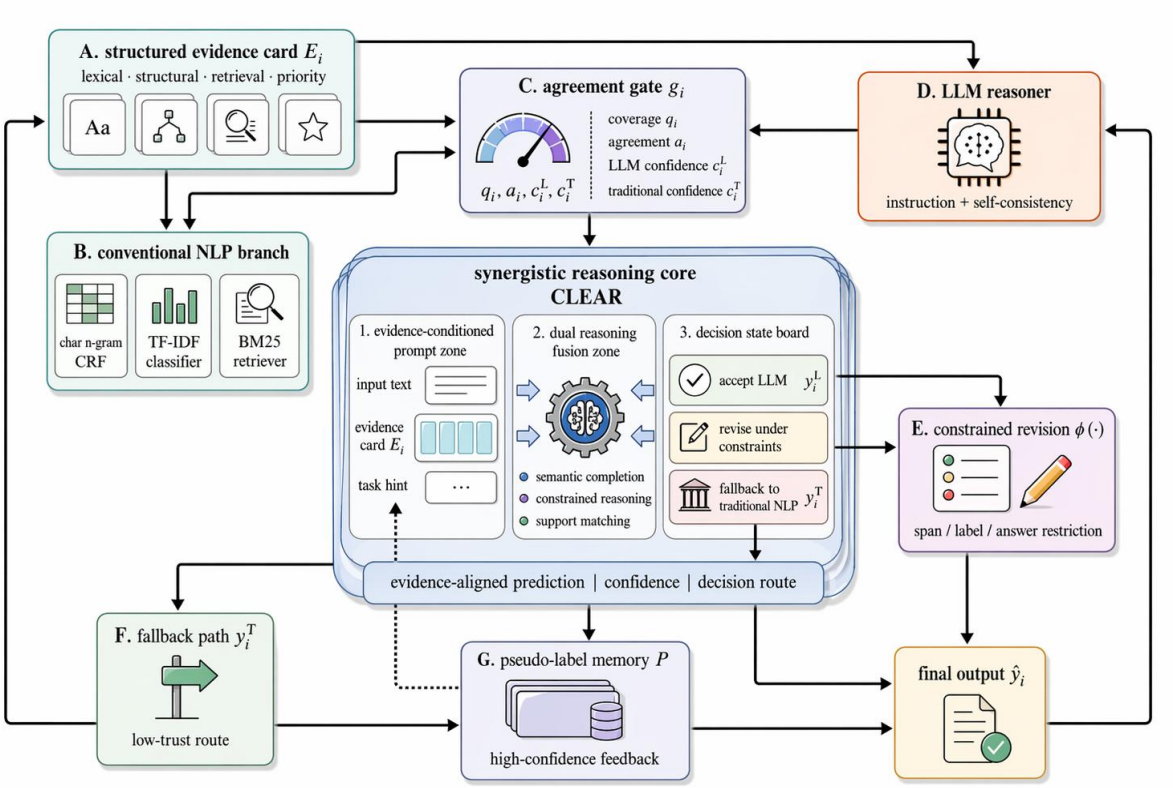

Figure 2: CLEAR: Synergistic Reasoning between Structured Evidence and LLM.

Considering that there are far more unlabeled texts than manually labeled in low-resource scenarios, CLEAR further introduces a high-confidence pseudo-labeling reflow mechanism. The filtering rule of the pseudo-labeled set is shown in equation (4).

$$\mathcal{P} = \{x_i \mid g_i > \tau_p,\ H(p_i^L) < \tau_h,\ viol(y_i^L, E_i) = 0\} \tag{4}$$

where $\mathcal{P}$ is the set of pseudo-labeled samples; τ_p is the reflux gating threshold; p_i^L is the output distribution of the LLM; $H(\cdot)$ is the information entropy; τ_h is the entropy threshold; and $viol(\cdot)$ is the value used to detect whether the output violates the evidence constraints. Samples are absorbed into the pseudo-labeling memory only if the gating score is high enough, the output entropy is low enough and the result does not violate the evidence card constraints. The number of new pseudo-annotations per round is no more than twice the size of the manual annotation set, and at most 2 rounds are executed to control error accumulation.

This design allows traditional paths to become formal evidence providers, while long-range semantic inference for LLM is done in a restricted candidate space, and unlabeled pools can be transformed into additional supervised sources through gated filtering.CLEAR is thus suitable for low-resource multitasking deployments, and is closer to a maintainable system scenario.

2.3 Experimental Protocol and Evaluation

The design of experiment needs to give reply to three questions: whether CLEAR can steadily do better than single-path methods in all tasks and budgets, through which mechanisms the gains are mainly obtained, and whether these gains are located in acceptable intervals of efficiency. For guaranteeing a uniform standard of comparison, the present paper establishes four kinds of comparison methods: The Traditional NLP merely uses traditional branches, which include character-level CRFs, TF-IDF linear classifiers, and BM25 retrieval, to finish the decision; The LLM baseline utilizes a unified template for few-shot prompts, and thus it does not obtain access to structured evidence cards; RAG-LLM carries out a sparse extraction of alike samples into few-shot prompts; and RAG-LLM carries out a sparse searching for similar samples to add into the few-shot hints. the similar samples which are given back by sparse retrieval; and CLEAR makes possible evidence cards, consistency gating, and restrained corrections by pseudo-labeled reflow on this identical backbone model. Figure 3 provides the overall arrangement of the experimental flow plan, baseline relationships and assessment interface.

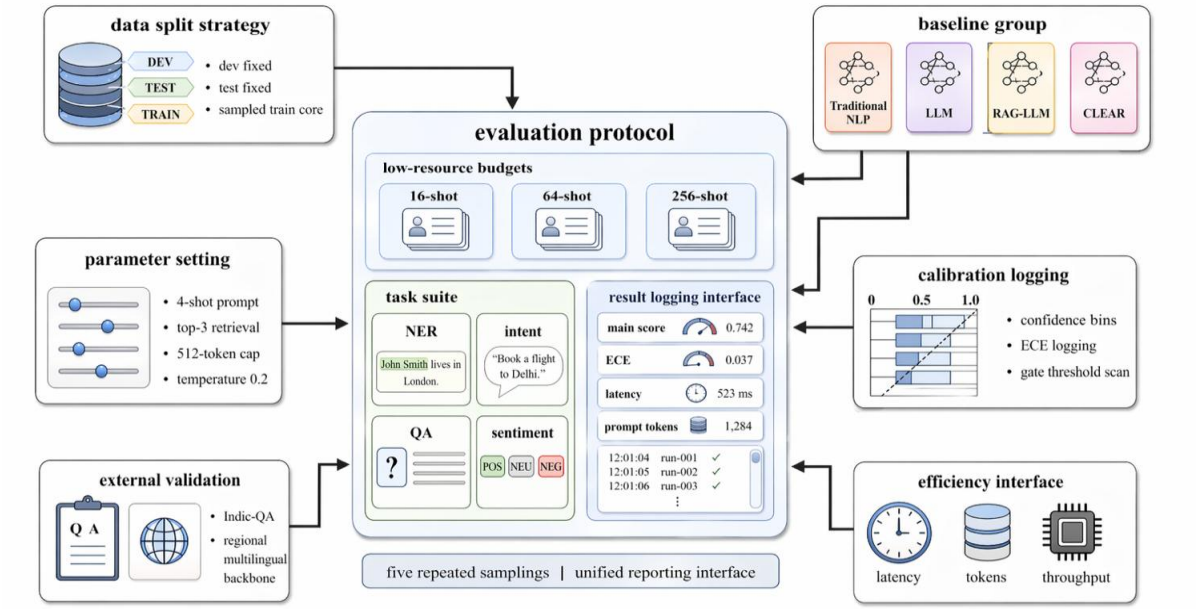

Figure 3: Experimental Protocol, Baselines, and Metric Interface.

The experiments have been carried out with independent sampling by each language-task combination. For each budget establishment, we repeat 5 random sampling processes and report the mean value, hence to reduce the fluctuations caused by low-resource sampling. The quantity of few-shot examples is being fixed to 4, the maximal quantity of candidates which are got back by BM25 is 3, the maximal quantity of lexical anchors in the evidence card is 6, and the maximal length of cues is 512 tokens. To the model that possesses open token probability, the LLM confidence degree is obtained from the length-normalized logarithmic probability which has temperature calibration. As for the model that does not possess open token probability, we utilize five low-temperature samples in order to estimate the output agreement rate which acts as the approximate confidence level. The temperature is fixed to 0.2 to reduce the effect of generation randomness on gating. The gating thresholds are determined through the development set grid search, and $\tau_0 = 0.45$, $\tau_1 = 0.65$, $\tau_p = 0.70$ are used by default; the pseudo-labeling reflow is executed in 2 rounds at most, and the new samples in each round do not exceed the size of the manual labeling by more than 2 times.

Evaluation metrics are set hierarchically according to task attributes. Named Entity Recognition and Sentiment Classification report Macro-F1, Intent Classification report Accuracy, Question and Answer report F1, and Exact Match is kept in the Appendix for verification. In order to compare the efficiency gains across tasks, Macro Score is additionally defined in this paper: the main metrics of each task are linearly mapped to the interval [0,100], and then averaged over the four tasks.Macro Score is only used to compare the cost-benefit ratios of the different methods under a unified budget, and is not a substitute for the original metrics of the tasks. The reliability section uses Expected Calibration Error to assess the deviation between prediction confidence and true correctness; the efficiency section records the single-sample average latency and average cue length; and the deployment section provides further statistics on the relationship between selective coverage and risk to see whether the system can maintain a more stable risk boundary under different fallback thresholds. In addition to the main experiment, we keep two other experiments in this paper.

3 Results and Discussion

3.1 Main Results across Tasks and Resource Levels

After completing the unified protocol setting, the first question that needs to be answered is whether collaborative reasoning can stably outperform single-path approaches under different tasks and different labeling budgets. The main results are shown in Table 2, and the corresponding resource-performance trends are shown in Figure 4. In Table 2, CLEAR achieves the highest scores under all four types of tasks and three budgets, and the advantage is not concentrated in a single task. Based on the Macro Score obtained from the average of the four tasks, CLEAR achieves 62.2, 72.9, and 79.6 under 16, 64, and 256-shot conditions, respectively, while the strongest baselines under the same budgets are 58.2, 69.4, and 77.1, with 4.1, 3.5, and 2.5 points of gain, respectively. it can be seen that the more scarce the labeling is, the more concentrated the synergistic gain is; after the budget is raised, the advantage is not concentrated in a single task. the more concentrated the synergy gains are; the advantage narrows but does not disappear after the budget is raised.

Table 2: Main Comparative Results under Different Label Budgets

Dataset / Metric	Labels per language	Traditional NLP	LLM	RAG-LLM	CLEAR
MasakhaNER 2.0 (F1)	16	58.4	54.1	60.3	64.8
MasakhaNER 2.0 (F1)	64	68.9	66.5	70.1	74.0
MasakhaNER 2.0 (F1)	256	76.8	74.9	78.2	80.
MASSIVE (Acc)	16	63.2	60.8	65.1	69.4
MASSIVE (Acc)	64	77.5	75.2	78.6	81.9
MASSIVE (Acc)	256	85.1	83.8	86.0	88.2
TyDi QA (F1)	16	38.7	41.4	44.9	48.6
TyDi QA (F1)	64	49.5	52.3	55.8	59.7
TyDi QA (F1)	256	58.1	60.4	63.5	66.8
NusaX (F1)	16	60.5	58.7	62.4	66.1
NusaX (F1)	64	71.8	70.2	73.1	76.0
NusaX (F1)	256	79.9	78.6	80.7	82.9

Figure 4 further gives the ordering of the curves for the four types of tasks under different

budgets.

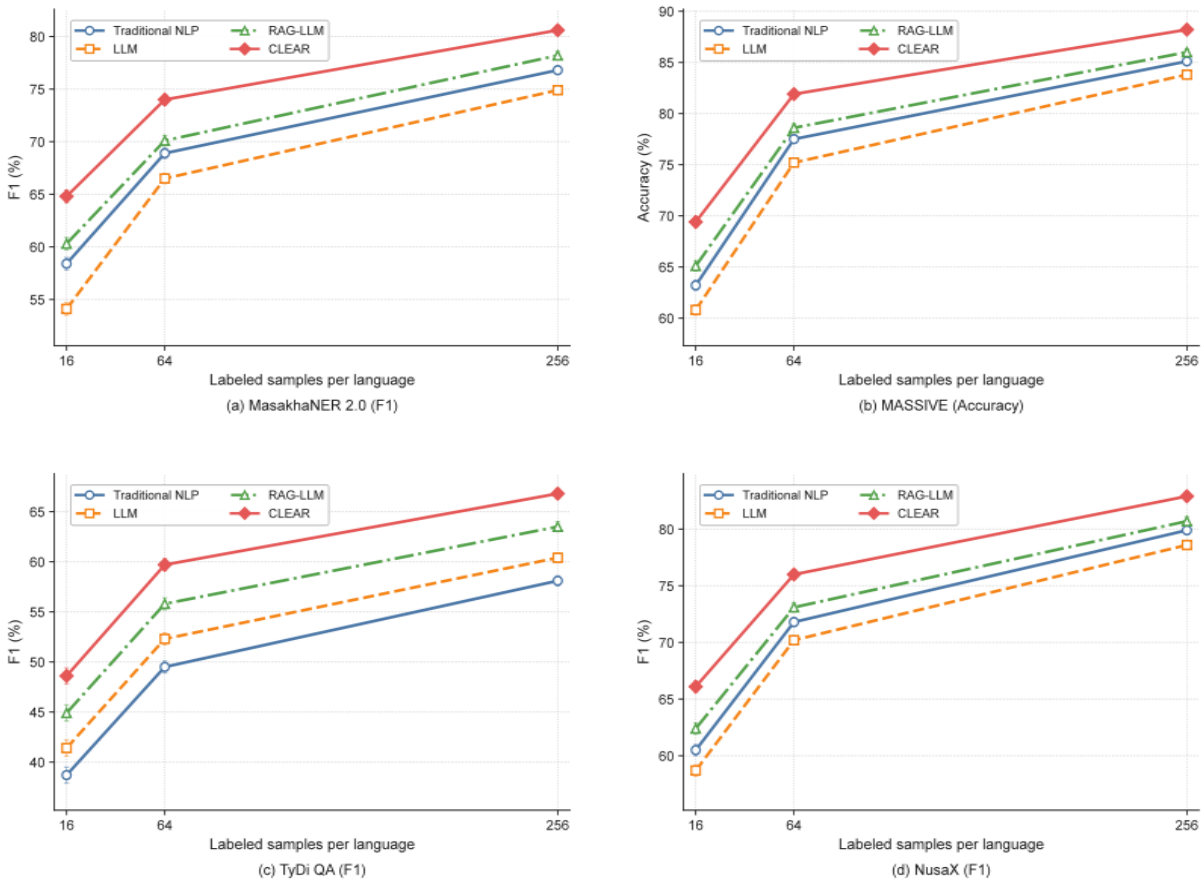

Figure 4: Performance-Resource Curves across NER, Intent Classification, QA, and Sentiment.

In Figure 4, CLEAR's curve is at the highest position in all four subplots of NER, Intent Categorization, Q&A, and Sentiment Categorization, and maintains a similar interval from the strongest baseline. Taking the 64-shot as an example, CLEAR achieves 74.0 F1, 81.9 Accuracy, 59.7 F1, and 76.0 F1 on MasakhaNER 2.0, MASSIVE, TyDi QA, and NusaX, which is an improvement of 3.9, 3.3, 3.9, and 2.9 points over the strongest baseline at the same budget, respectively.The gains of NER and QA gains are more stable, suggesting that structured evidence compresses the candidate space more significantly when the task relies on both local boundaries and contextual semantics; the gains for intent and sentiment classification are slightly smaller, but remain above 2 points, suggesting that short text labeling decisions benefit equally from labeling prototypes and high-weighted trigger words.

This gain is not equally distributed across all language conditions. To further determine where the synergistic advantage comes from, Figure 5 divides the languages into four intervals from Q1 to Q4 by vocabulary coverage and counts the gain in CLEAR relative to the strongest baseline.

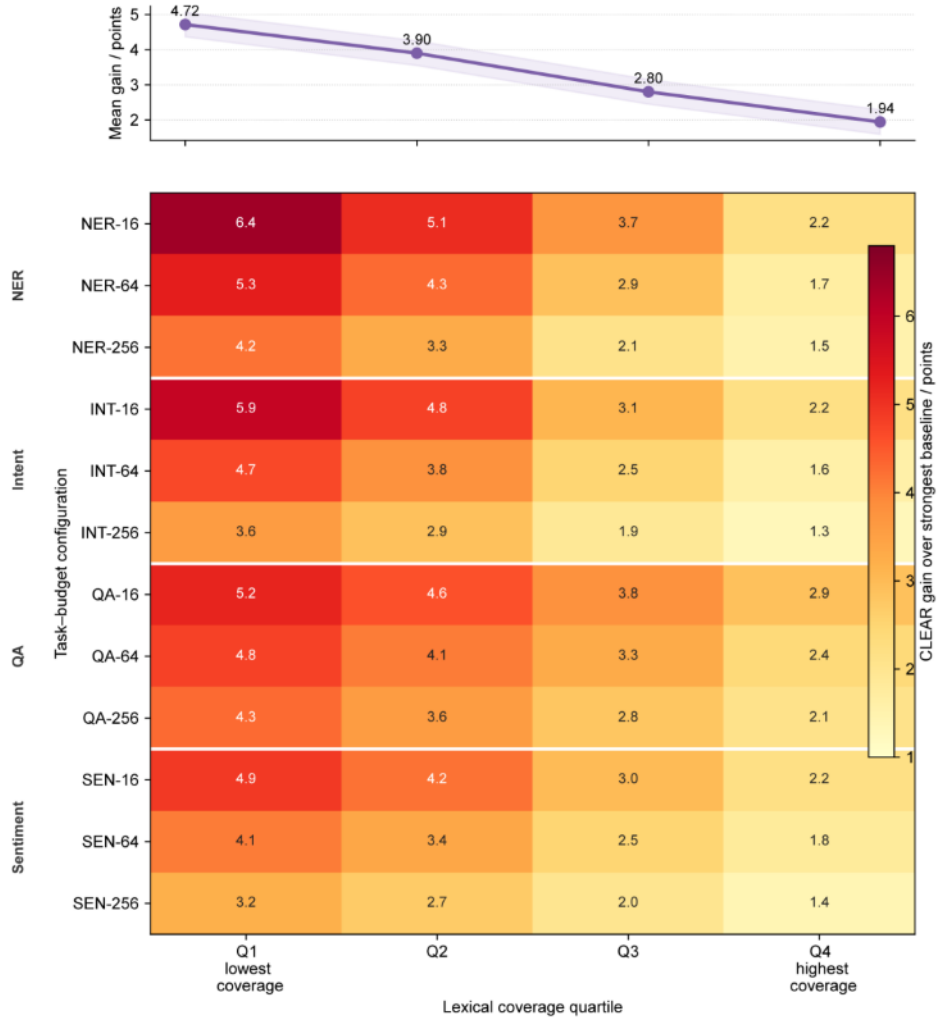

Figure 5: Gain Heatmap by Lexical Coverage and Resource Budget.

In Fig. 5, the high values are mainly concentrated in the low-coverage area. For example, in the 16-shot case, the average gain of the four tasks is 5.6 points in Q1, while it is only 2.4 points in Q4; in the 64-shot and 256-shot cases, the average gain is 4.7 and 3.4 points in Q1, while it is 1.9 and 1.6 points in Q4, respectively. The distribution presented in the heatmap implies that low-coverage languages benefit first from candidate constraints: character fragments, aliases, and local structure a priori narrow down the search scope, and then the LLM accomplishes semantic discrimination in a smaller uncertainty space. For high-coverage languages, the synergy gain is more in the form of stability compensation.

This result answers the central question of this section.The advantages of CLEAR do not depend on a single task, nor are they limited to very low budgets. It maintains performance margins across budgets and stems mainly from the candidate compression and semantic complementation division of labor on low-coverage languages. The next question is what modules actually bring this advantage and what reliability and efficiency costs the system has to pay in order to obtain these gains.

3.2 Ablation, Calibration, and Efficiency Analysis

After identifying the overall gains, this section further answers two questions: which components CLEAR's enhancement mainly comes from, and whether the additional costs introduced by these components are still in the deployable range.The ablation results for the

64-shot condition are shown in Table 3, the gating thresholds correspond to the calibration changes in Fig. 6, and the fractional-delay-cue-length of the 3D relationship is shown in Fig. 7, the reliability curves are shown in Fig. 8, and the results of the module response surface coupled to the efficiency are shown in Fig. 9. Together, the sets of graphs illustrate that the benefits of CLEAR come from the combined effects of evidence compression, gating shunting, and constrained corrections.

Table 3: Ablation and Efficiency Analysis of CLEAR at 64-shot

Variant	Macro Score	MasakhaNER F1	TyDi QA F1	ECE	Latency (s/sample)
CLEAR	72.9	74.0	59.7	0.081	1.78
w/o lexical & structural evidence	69.8	70.7	58.2	0.117	1.64
w/o retrieval evidence	70.5	72.6	56.8	0.109	1.66
w/o agreement gate	70.6	71.8	58.9	0.134	1.75
w/o pseudo-label loop	71.4	72.9	58.8	0.094	1.74
w/o constrained revision	71.1	72.2	58.3	0.102	1.71

Let's look at the module contributions first. In Table 3, after removing the lexical & structural evidence, the Macro Score decreases from 72.9 to 69.8, and the MasakhaNER F1 decreases from 74.0 to 70.7, which is the largest decrease among all ablation settings. This suggests that the most crucial role of conventional branching under low resource conditions is still to give local verifiable structures. After removing retrieval evidence, TyDi QA F1 decreases from 59.7 to 56.8, a drop of 2.9 points, which is higher than the other tasks, suggesting that the Q&A task is more sensitive to candidate fragment support. After removing the agreement gate, the decrease in score is not in the first place, but the ECE rises to 0.134, an increase of 0.053 compared with the full model, indicating that the gating module firstly undertakes to control the penetration of high confidence errors, and its main role is not in the bare score pulling up. After removing the pseudo-label loop and constrained revision, the score decreases more slowly, but both of them steadily weaken the upper limit under low budget, implying that label-free reflux and intermediate interval reclassification are equally important for system convergence. The coupling between gating thresholds and calibration is given in Figure 6.

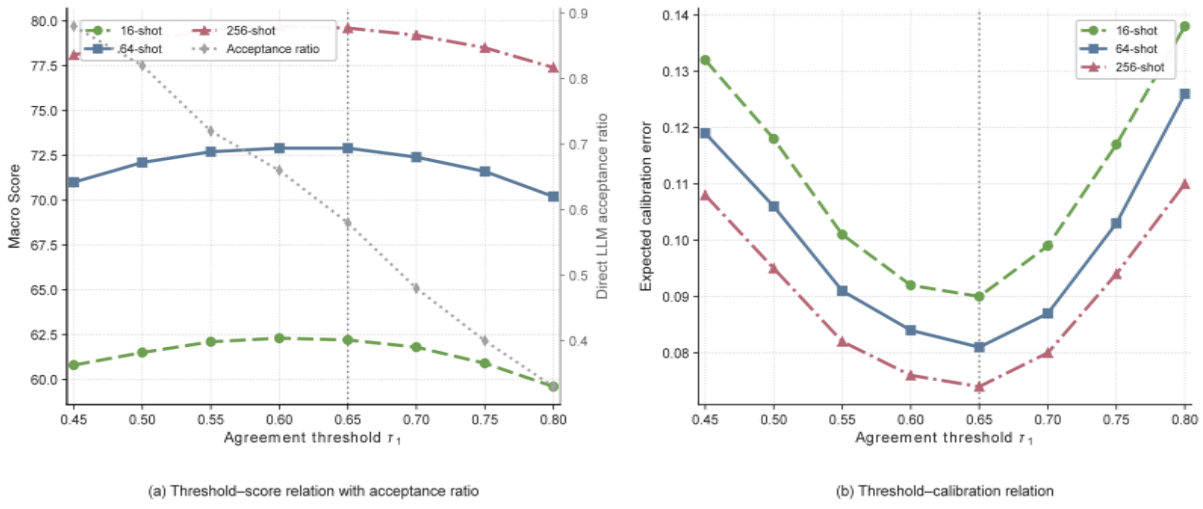

Figure 6: Threshold–Calibration Coupling under Agreement Gate.

In Fig. 6, the Macro Score peaks between \(\tau_1=0.60\) and 0.65 for all three budget conditions, while the ECE stays the lowest in the same range. In the case of the 64-shot, for example, when \(\tau_1=0.65\), the Macro Score is 72.9 and the ECE is 0.081; after the threshold is raised to 0.80, the Macro Score decreases to 70.2, and the acceptance ratio decreases significantly. This suggests that too high a threshold will cause the system to fall back too much to the traditional path, while too low a threshold will make it easier for high-confidence errors to enter the final output. What gating really regulates is the ratio between "accepting generative results" and "triggering constrained corrections or fallbacks", which is the most sensitive control interface in low-resource deployments. Whether the increase in accuracy is worth the extra cost needs to be judged in conjunction with Figures 7 through 9.

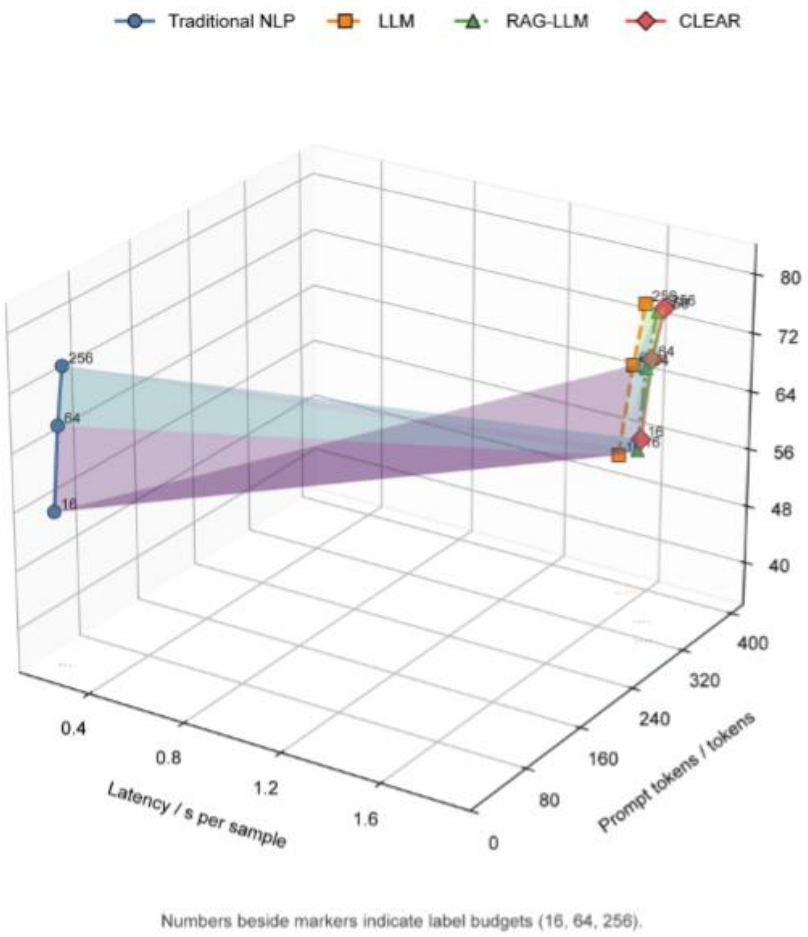

Figure 7: Pareto Relation among Score, Latency, and Prompt Tokens.

In the Fig. 7, the Traditional NLP is placed in the region of lowest time delay and shortest cue length, but therefore it also has the uppermost limit of lowest Macro Score; Large Language Model and Retrieval-Augmented Generation Large Language Model thus move in the direction of higher time delay and longer cue areas; and CLEAR lies still more forward of the Pareto front. Take 64-shot as the example, CLEAR possesses a Macro Score of 72.9, average time delay of 1.80 s/sample, and average cue length of 318 tokens, this is only an increment of about 0.09 s/sample when compared with RAG-LLM, therefore it obtains 3.5 points, hence the cue length is still lower than that of the direct few-shot LLM. LLM. The reliability curves which are in Figure 8 further make explanation that this cost was not traded in exchange for higher confidence drift.

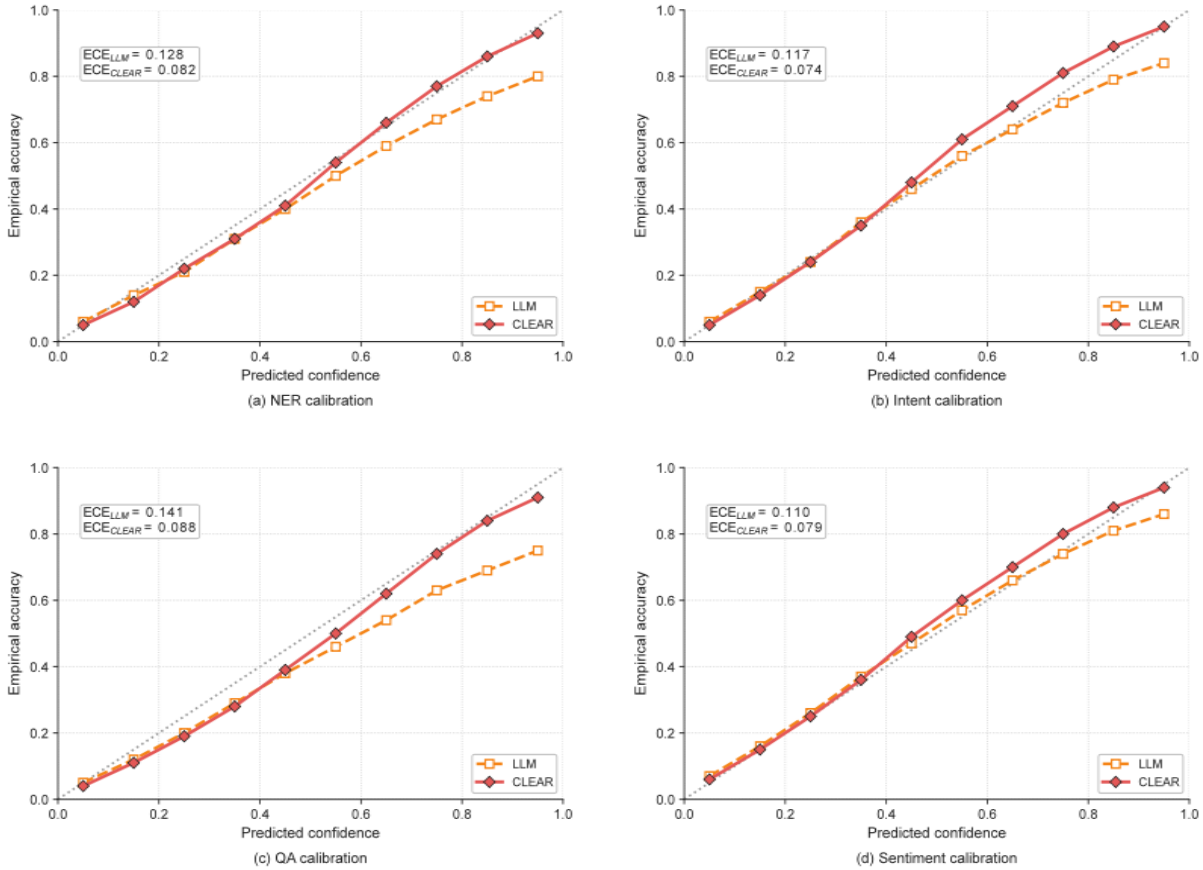

Figure 8: Representative Case Comparisons of Traditional NLP, LLM, and CLEAR.

The ECE of CLEAR on the four tasks of NER, Intent, QA, and Sentiment are 0.082, 0.074, 0.088, and 0.079, respectively, which are lower than those of LLM alone, which are 0.128, 0.117, 0.141, and 0.110.The plot of the error taxonomy versus the deployment threshold is shown in Fig. 9.

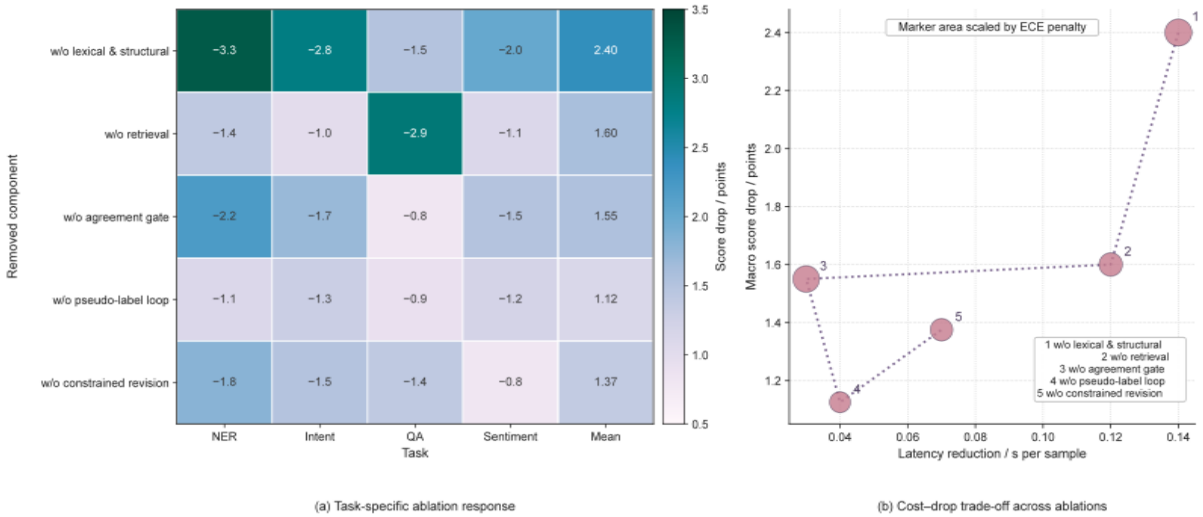

Figure 9: Error Taxonomy and Deployment Threshold Map.

Figure 9 next puts module contributions and efficiency punishments into the same coordinate system: taking away lexical and structural evidence brings the largest average

decrease, taking away agreement gate brings the largest ECE punishment, and taking away retrieval evidence mainly influences QA. Several change directions hold consistency with one another, hence this indicates that The performance, calibration, and efficiency superiorities of CLEAR originate from stable synergistic mechanisms which do not depend on accidental parameters or one single module.

3.3 Error Sources, Case Interpretation, and Deployment Meaning

The overall scores and ablation results already illustrate that CLEAR has an advantage at the statistical level, but the deployment is more concerned with where the errors fall, manageability, and how the fallback boundaries are set. Figure 10 puts the selective coverage-risk frontier in the same set of plots as the error type distribution and is used to answer this question.

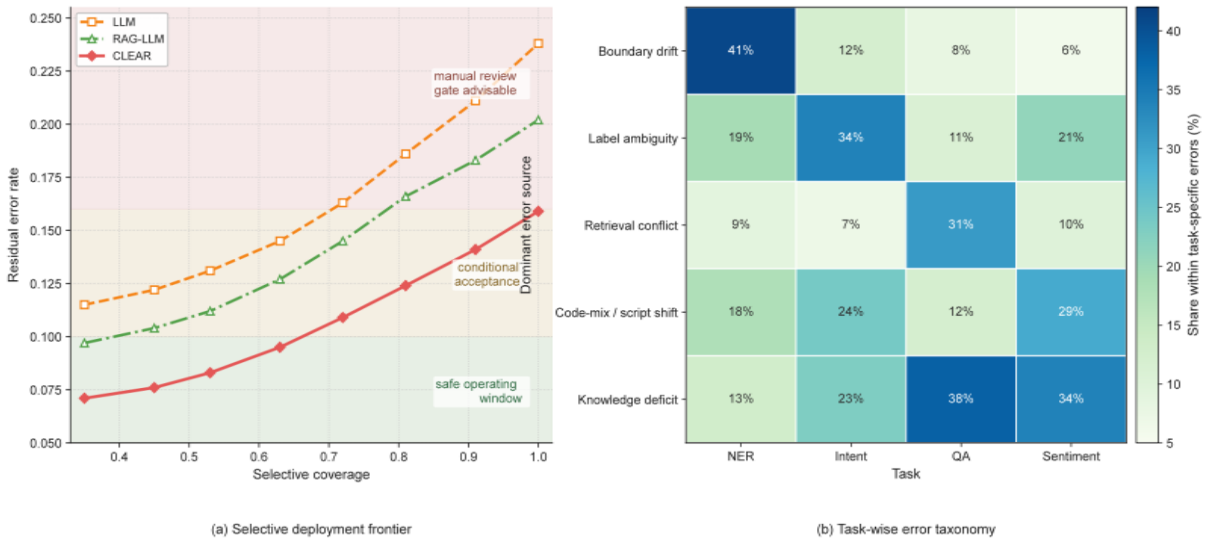

Figure 10: Selective Deployment Frontier and Task-Wise Error Taxonomy.

In the Figure 10, the coverage-risk curve of CLEAR is always lower than the curve of LLM only and RAG-LLM. When selective coverage is promoted to 1.00, the risk of CLEAR is 0.159, which is lower than the 0.202 of RAG-LLM and the 0.238 of LLM alone, hence suggesting that the synergistic mechanism can suppress the overall risk to a far lower level, even in the condition of not actively giving up the forecast. This gap becomes even wider under more conservative confidence covering intervals, hence this suggests that gating and fallback mechanisms do indeed provide operable risk controlling interfaces.

The error heatmap reveals that the dominant error is not the same across tasks. In Fig. 10, NER has the highest percentage of Boundary drift at 41%, indicating that low-resource entity recognition is firstly affected by boundary drift; Label ambiguity in the Intent task has a percentage of 34%, reflecting that semantic proximity between service intents is still the main interference; Retrieval conflict and Knowledge deficit in QA reach 31% and 38% respectively, indicating that candidate evidence conflict and insufficient background knowledge will jointly pull down the answer quality; Sentiment task is more likely to be affected by Code-mix/script shift and Knowledge deficit. This distribution is aligned with the analysis of module roles in the previous section: boundary and prototype problems rely mainly on lexical and structural evidence, Q&A conflicts rely more on retrieval evidence, and risk penetration is centrally suppressed by the agreement gate.

Looking further at the representative sample, the improvements in CLEAR are mainly reflected in three types of decision interfaces. The first category is the sample where entity

boundaries and entity types are implicated with each other. In this case, CLEAR first accepts the span candidates provided by the traditional branch, and then requires the LLM to complete the type judgment only within the restricted boundaries, so the errors are more likely to change from “cross-boundary errors” to “cross-boundary errors”. “cross-boundary error” to “within-boundary fine-grained classification error”. The second category is samples with highly proximity of intent labels, e.g., the neighboring labels of fund transfer, bill payment and account management. Labeling prototypes and high-weighted trigger words can significantly reduce label drift by narrowing down the optional labels to a small range, and then the LLM can discriminate them based on the contextual role relationships and target objects. The third category is the conflicting evidence samples in Q&A. When two retrieval fragments give different dates, aliases, or locations, the LLM alone tends to select the linguistically smoother one; CLEAR triggers a constrained correction due to a drop in the gating score, so that the answer has to fall within the supported fragment, thus avoiding unsourced generation.

These error patterns correspond directly to deployment strategies. For NER and short text categorization, system maintenance should focus on lexicon, alias repository, and tag prototype updates; for Q&A systems, retrieval de-weighting, source hierarchies, and conflicting fragment management need to be dealt with first; and for mixed script corpus, character normalization and script tagging should be enhanced in the preprocessing stage. The coverage-risk frontier given in Figure 10 allows the system to set thresholds stratified by business risk: high-risk Q&A can increase the percentage of fallbacks, while low-risk categorization tasks can retain more generative output. The emergence of regional multilingual models with large-scale multilingual resource infrastructures provides external conditions for this deployment route. The regional model route for Southeast Asian languages illustrates that a more linguistically relevant backbone further reduces semantic drift [24], while a 200-language level machine translation and resource extension system provides a more stable base for alias complementation, cross-language retrieval, and weakly supervised constructions [25]. The value of CLEAR also lies in the integration of decentralized localized evidence, generative capabilities, and risk control in low-resource scenarios into the same maintainable interface.

4 Conclusion

In this paper, we construct CLEAR, a collaborative reasoning framework for large language models and traditional NLP methods, around the contradiction between evidence scarcity and decision-making controllability in low-resource scenarios, and validate it under multi-task and multi-budget conditions.

(1) This paper gives interfaces at the object organization level. Supervised core sets, unlabeled pools, and sparse retrieval indexes are incorporated into the same organization, and lexical anchors, structural priors, and candidate fragments are compressed into structured evidence cards to enable sequence annotation, short text classification, and question-answer tasks to share evidence representations.

(2) In this article, we carry out verification on the effect of cooperative decision making on both method level and result level. CLEAR links the candidate collection abilities of classical branching and the semantic distinguishing abilities of LLM into the identical decision loop by means of consistency gating, restricted corrections, and high confidence pseudo-labeling reflow. The Macro Score which CLEAR gets attains 62.2, 72.9, and 79.6 under 16-shot, 64-shot, and 256-shot conditions, respectively, all of these are higher than the strongest baseline that has the same budget.

(3) The limitation of this paper is that dictionary quality, retrieval de-duplication and threshold calibration still need to be refined by task, and the validation focuses on text tasks.

Follow-up work can further automate alias maintenance, conflict evidence management and pseudo-labeling screening, and extend collaborative gating to speech transcription and OCR noisy text.

About the Author

Hanyang He was born in Wuhu, Anhui, China, in 2003. He is currently an undergraduate student at Tianjin University of Technology. His recent research interests include deep learning, natural language processing, and image processing.

References

[1] Üstün, A., Aryabumi, V., Yong, Z., et al. (2024). Aya model: An instruction finetuned open-access multilingual language model. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 15894-15939).

[2] Cahyawijaya, S., Lovenia, H., & Fung, P. (2024). LLMs are few-shot in-context low-resource language learners. In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers) (pp. 405-433).

[3] Nguyen, X.-P., Aljunied, M., Joty, S., et al. (2024). Democratizing LLMs for low-resource languages by leveraging their English dominant abilities with linguistically-diverse prompts. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 3501-3516).

[4] Li, C., Yang, W., Zhang, J., et al. (2024). X-Instruction: Aligning language model in low-resource languages with self-curated cross-lingual instructions. In Findings of the Association for Computational Linguistics: ACL 2024 (pp. 546-566).

[5] Zhang, Z., Lee, D.-H., Fang, Y., et al. (2024). PLUG: Leveraging pivot language in cross-lingual instruction tuning. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 7025-7046).

[6] Lai, W., Mesgar, M., & Fraser, A. (2024). LLMs beyond English: Scaling the multilingual capability of LLMs with cross-lingual feedback. In Findings of the Association for Computational Linguistics: ACL 2024 (pp. 8186-8213).

[7] Dang, J., Ahmadian, A., Marchisio, K., et al. (2024). RLHF can speak many languages: Unlocking multilingual preference optimization for LLMs. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (pp. 13134-13156).

[8] Feng, S., Shi, W., Wang, Y., et al. (2024). Teaching LLMs to abstain across languages via multilingual feedback. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (pp. 4125-4150).

[9] Sun, Y., Zuo, A., Gao, W., et al. (2025). CausalAbstain: Enhancing multilingual LLMs with causal reasoning for trustworthy abstention. In Findings of the Association for

Computational Linguistics: ACL 2025 (pp. 14060-14076).

[10] Ansell, A., Parović, M., Vulić, I., et al. (2023). Unifying cross-lingual transfer across scenarios of resource scarcity. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (pp. 3980-3995).

[11] Zheng, W., Pan, W., Xu, X., et al. (2024). Breaking language barriers: Cross-lingual continual pre-training at scale. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (pp. 7725-7738).

[12] Chen, Y., Shah, V., & Ritter, A. (2025). Translation and fusion improves cross-lingual information extraction. In Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 7744-7764).

[13] Lai, P., Gan, J., Ye, F., et al. (2025). Improving low-resource sequence labeling with knowledge fusion and contextual label explanations. In Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (pp. 5655-5674).

[14] Merx, R., Mahmudi, A., Langford, K., et al. (2024). Low-resource machine translation through retrieval-augmented LLM prompting: A study on the Mambai language. In Proceedings of the 2nd Workshop on Resources and Technologies for Indigenous, Endangered and Lesser-resourced Languages in Eurasia (EURALI) @ LREC-COLING 2024 (pp. 1-11).

[15] Pei, R., Liu, Y., Lin, P., et al. (2025). Understanding in-context machine translation for low-resource languages: A case study on Manchu. In Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 8767-8788).

[16] Zhang, C., Lin, J., Liu, X., et al. (2025). Read it in two steps: Translating extremely low-resource languages with code-augmented grammar books. In Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 3977-3997).

[17] Li, Y., Zhao, Z., & Scarton, C. (2025). It's all about in-context learning! Teaching extremely low-resource languages to LLMs. In Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (pp. 29544-29559).

[18] Lu, K., Yang, Y., Yang, F., et al. (2025). Low-resource language expansion and translation capacity enhancement for LLM: A study on the Uyghur. In Proceedings of the 31st International Conference on Computational Linguistics (pp. 8360-8373).

[19] Adelani, D. I., Neubig, G., Ruder, S., et al. (2022). MasakhaNER 2.0: Africa-centric transfer learning for named entity recognition. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (pp. 4488-4508).

[20] FitzGerald, J., Hench, C., Peris, C., et al. (2023). MASSIVE: A 1M-example multilingual natural language understanding dataset with 51 typologically-diverse languages. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 4277-4302).

[21] Clark, J. H., Choi, E., Collins, M., et al. (2020). TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. Transactions of the Association for Computational Linguistics, 8, 454-470.

[22] Winata, G. I., Aji, A. F., Cahyawijaya, S., et al. (2023). NusaX: Multilingual parallel sentiment dataset for 10 Indonesian local languages. In Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics (pp. 815-834).

[23] Singh, A. K., Kumar, V., Murthy, R., et al. (2025). INDIC QA benchmark: A multilingual benchmark to evaluate question answering capability of LLMs for Indic languages. In Findings of the Association for Computational Linguistics: NAACL 2025 (pp. 2607-2626).

[24] Nguyen, X.-P., Zhang, W., Li, X., et al. (2024). SeaLLMs - Large language models for Southeast Asia. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations) (pp. 294-304).

[25] NLLB Team. (2024). Scaling neural machine translation to 200 languages. Nature, 630, 841-846.