Sparse-Checklist Prompting for Arabic Grammar Tutoring: Fast, Token-Efficient Feedback

Zayan Hasan
Rock Ridge High School
Ashburn, VA
zayanuhasan@gmail.com

Abstract

We explore token-efficient prompting for Arabic grammar tutoring, where time and cost-efficient approaches to feedback are important for Muslim community classes. Rather than producing free-form explanations, we restrict the model to providing a single pedagogical hint tag from a set of 5 possible tags, (Sparse-Checklist), and implement a simple router that sends clearly correct outputs down a short path. On 180 items with skill-labeled responses in the categories of agreement, pronoun clitics, prepositions and definiteness, Sparse-Checklist enhanced correctness over a Direct feedback baseline (81.1% versus 76.1%), reduced median latency (0.530s versus 0.807s) and half the completion tokens, which we consider a realization of reasoning cost (11.9 versus 22.7). A combined Router variant achieves 79.4% accuracy, while achieving 18.2 completion tokens and 0.639s median latency. On incorrect responses, Sparse-Checklist and Router both select the appropriate skill tag 100% of the time.

1 Introduction

Muslim community classrooms, including weekend schools, after-school programs, and introductory Arabic programs, frequently have large numbers of students and limited scheduled time and resources. This necessitates feedback from teachers that is quick, inexpensive, and pedagogically accurate for learners to practice more and better for teachers to triage. A further challenge with Arabic is that the rich morphology and agreement, clitic attachment, and definiteness patterns lead to multiple, minor errors, unless they receive short, specific clues instead of detailed explanation [1]. In this context, the social impact of inexpensive, scalable formative feedback is direct where more students get timely guidance, and teachers reclaim time for high-value interaction.

Educational technology shows that well-designed tutoring support can approach the effectiveness of human tutoring, especially when feedback is immediate and focused [2]. LLMs promise to deliver such support at scale, yet common prompting styles (e.g., chain-of-thought) trade speed for verbose reasoning, inflating latency and token costs [3, 4]. Recent efficiency work explores pruning and compression on the prompt/response side [5], while sparsity and expert routing in model architectures demonstrate that choosing among a small set of alternatives can preserve accuracy at lower compute [6, 7]. We bring those intuitions to tutoring by constraining *outputs*, not models.

We categorize Arabic feedback as a small judgement: choose `CORRECT/INCORRECT` and optionally select one hint from a small, classroom-specific taxonomy (agreement, pronoun clitics, prepositions, definiteness). This *Sparse-Checklist* response treats explanation as selection, keeping feedback short and interpretable for teachers. A simple router short-circuits the obviously correct cases to an even simpler response path. Our aim is indeed practical. Lower token/latency cost while delivering the right feedback pointing learners to the right concept or idea: a cost lever for programs operating with limited budgets and shared devices.

This study is an initial step in using synthetic items to compare token-efficient prompting styles. Synthetic data enables controlled comparisons, but also limits external validity. We therefore view the results as hypothesis generating and plan to evaluate on authentic learner responses with teacher rated helpfulness, and also broaden skill coverage and tagging granularity.

2 Methods

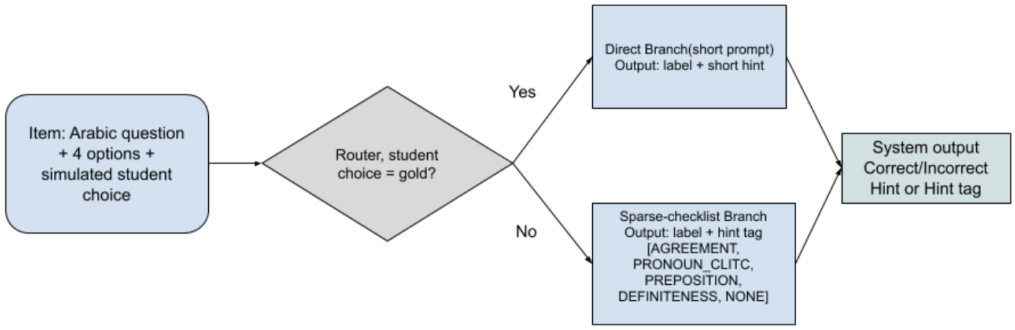

Figure 1: Method overview. An Arabic MCQ + student choice enters a deterministic router: if the choice equals the gold, use a short *Direct* prompt; otherwise use *Sparse-Checklist* (select one tag from {AGREEMENT, PRONOUN_CLITIC, PREPOSITION, DEFINITENESS, NONE}. The system outputs `Label` and `Hint`/`HintTag`

We evaluate a standard free-form prompting baseline with two token-efficient variants, both of which are designed to promote short and pedagogically focused feedback. We evaluate on 180 synthetic multiple-choice items, constructed to represent common Arabic grammar error types. In all cases the model receives the Arabic question, 4 options, and the students chosen option, and it outputs a binary judgment (`CORRECT/INCORRECT`) and an optional pedagogical hint. We do include the option, though it is optional after the model output correctness.

The *Direct* baseline asks for correctness judgment and a one-sentence justification. We chose a minimal two-line header (the label (`Label:`) and hint (`Hint:`)), but otherwise left the hint free-form natural language. This is similar to common tutoring prompts, and elicits longer completions.

The *Sparse-Checklist* variant uses a single tag from a small taxonomy in lieu of free-form hints: AGREEMENT, PRONOUN_CLITIC, PREPOSITION, DEFINITENESS or NONE. The output is strictly `Label: CORRECT|INCORRECT` and `HintTag: <one tag>`. In this design, we turned an explanation into a selection, thus limiting maximum completion length, as well as forcing the model to choose a single pedagogical dimension. This is similar to sparsity/expert selection in model architectures [6, 7] vs. open-ended chain-of-thought reasoning [3].

As an even cheaper alternative for easy items, we use a simple *Router* to check the student's choice equals the gold, e.g., if so, the model gets to a very short version of Direct model with a tight completion cap, if not it gets a Sparse-Checklist prompt. In evaluation we give a merged *Router* model as a single method since it is one policy system. Throughout we consider *completion tokens* (the generated output) as our basis for "reasoning cost", in addition to wall clock latency and we report total tokens as subordinate to wall clock latency for reasons that have more to do with fixed length prompts than the completion as a function of the reasoning budget used by the model.

3 Experimental Setup

The dataset used for evaluation consisted of 180 *synthetic* multiple-choice items. Each item was generated from handcrafted templates with randomized lexical fillers and distractors designed to emulate real learner mistakes. All experiments depicted here were run in a Jupyter Notebook (Colab) assigning use of a singly an A100 gpu. For the totality of these experiments, we use `gpt-4o-mini` with temperature 0, and short completion caps chosen to reflect each style, where Direct complete

uses up to 80 completion tokens to be able to return a natural-language hint that is brief, while Sparse-Checklist and each of the Router branches cap out at 40 tokens since they only return a label and tag. The prompts are also fixed as before, with Sparse-Checklist using the five-tag inventory that helps lengthen the prompt such as brevity can be used to cut back on completions. The Router short Branch uses direct path only if the student choice and gold are equivalent; otherwise using that checklist path. When we report results, we combine each of the branches as one *Router* method.

We report four quantities total. Correctness is labeled accuracy and macro-F1 on the binary judgment, or the known gold answer. Pedagogical targeting is measured on *incorrect* item by checking if predicted hint tag is exactly the goal item ground truth skill. Latency is wall-clock-time per item as variant of summary measure (median). Reasoning cost is estimated by completion tokens; when consumption fields are provided we record directly, otherwise by using a tokenization based on use prior to prevent missing values. This tokenization focus to measuring response tokens is consistent with very recent work on prompt/response compression and efficiency of LLM inference [5]. Each run uses the same seed and exactly one pass to each item.

4 Results

Table 1: Main results. Completion tokens are our proxy for reasoning cost.

Method	Acc. (%)	Macro-F1 (%)	Med. Lat. (s)	Mean Comp. Tok.	Hint Acc. (%)
Direct	76.1	76.1	0.807	22.7	0
Sparse-Checklist	**81.1**	**81.1**	**0.530**	**11.9**	**100**
Router (merged)	79.4	79.4	0.639	18.2	100

These differences should be interpreted descriptively given single runs and unequal completion caps. The 100% hint-tag accuracy reflects exact string matching within synthetic templates; we will test robustness on non-templated and real learner data. Table 1 provides an aggregate comparison. Sparse-Checklist is able to achieve correctness gains against the Direct baseline (81.1% vs. 76.1%) as well as reduced median latency (0.807s to 0.530s) and completion tokens (22.7 to 11.9). This is nearly a 50% reduction. As a cost–accuracy trade-off, the merged Router version is able to achieve 79.4% accuracy, 0.639s median latency, and 18.2 completion tokens. Given that the checklist prompt outlines the five allowable tags, this gets a little longer than Direct, so we present completion tokens as the primary measure of reasoning cost; total tokens are provided for completeness, though they are not critical to the interpretation.

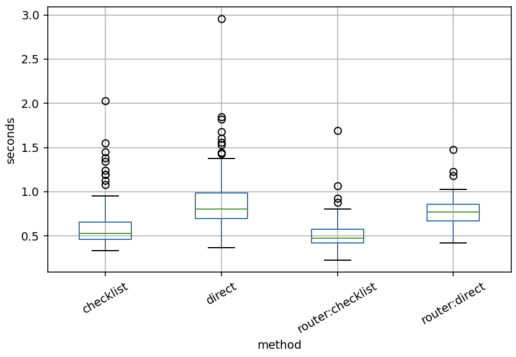

Figure 2: Latency by method. Boxplots over ∼180 Arabic grammar items. Boxes show IQR, center line the median, whiskers 1.5×IQR; dots are outliers.

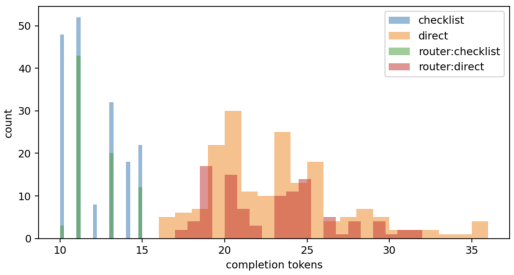

Figure 3: Completion-token usage (reasoning cost). Histogram of response tokens per item.

The distributional perspectives in Figures 2 and 3 agree with the table: both Sparse-Checklist and Router shift latency to the left compared to Direct, and the completion-token histogram indicates significant collapse under Sparse-Checklist while Router is between the two. Even with these positives, pedagogical targeting is retained. For items the student is incorrect on, Sparse-Checklist

and Router selected the ground-truth skill tag 100% of the time, meaning the constrained output does not lessen the model's ability to nudge learners towards the correct skill/concept.

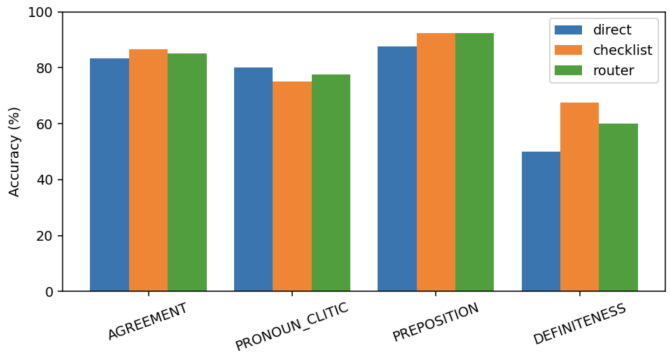

Figure 4: Per-skill correctness (accuracy, %). Results for AGREEMENT, PRONOUN_CLITIC, PREPOSITION, and DEFINITENESS (45 items per skill).

Figure 4 details correctness by skill. Performance is generally similar within *agreement*, *pronoun clitics*, *prepositions*, and *definiteness*, with Sparse-Checklist equaling or exceeding Direct performance in all skills and Router still competitive. We do not see one whole skill contributing to the aggregate advances, which implies that overall the output constraint is more crucial than the peculiarities of a specific template family. When paired with the table and figures, this evidence leads back to the main takeaway: constraining outputs to a tiny taxonomy and routing clear-cut cases across a minimal path achieves faster feedback and much less reasoning cost, while being equally correct and hint-targeted.

5 Limitations

Our system focuses on just four skills and requires the model to produce one tag. Authentic learner errors typically involve multiple skills at play and pragmatic/contextual factors which a single label may not encapsulate. The Router also presumes a clear gold answer, but even for learners, there may be multiple defensible choices, so it may not drive performance. We review a single model; the generalized external validity of models, suppliers, and classroom distributions remain ambiguous. Our evaluation size is also modest at around 180 items, larger sets may produce benefit for future work.

6 Conclusion

We created a token-efficient approach for Arabic grammar feedback tutoring that restricts feedback to just one classroom-applicable hint tag (Sparse-Checklist) and is augmented with a two-possible options router for ambiguous cases. Using a $\sim$180 item skill-labeled dataset, Sparse-Checklist improved correctness compared to a Direct benchmark (81.1% compared to 76.1%), decreased median latency (0.530s compared to 0.807s), and halved the number of completion tokens (11.9 compared to 22.7), while the combined Router provided a speed-cost trade-off (79.4% correctness; 0.639s; 18.2 tokens). These benefits held true for agreement, pronoun clitics, prepositions, and definiteness, and the limited output provided (100% of hint-tag accuracy) pedagogically targeted wrong answer feedback. The methodology is model-agnostic, easily deployable in community classrooms, and acknowledges the imperative for quick and low cost formative feedback. Future work will further the skill taxonomy, include real learner data with teacher-rated helpfulness, and assess more complicated routing and compression in tighter boundaries. Beyond simply technical efficiencies, we want to contribute to the scaling of low-cost feedback for under-resourced learning environments like Muslim weekend schools. In this regard, token-efficient tutoring could serve as a bridge between high-tech Artificial Intelligence methods and the everyday educational priorities of communities.

References

[1] Nizar Habash. *Introduction to Arabic Natural Language Processing*. Morgan & Claypool, 2010.

[2] Kurt VanLehn. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, 46(4):197–221, 2011.

[3] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, et al. Chain-of-thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022.

[4] Takeshi Kojima, Keisuke Sakaguchi, Yizhong Deng, Michihiro Yasunaga, et al. Large language models are zero-shot reasoners. *arXiv preprint arXiv:2205.11916*, 2022.

[5] Zhengbao Jiang, Tianyi Zhang, Amir Gholami, Trevor Darrell, and Joseph E. Gonzalez. Llmlingua: Compressing prompts for accelerated inference of large language models. *arXiv preprint arXiv:2310.05736*, 2023.

[6] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, et al. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

[7] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *arXiv preprint arXiv:2101.03961*, 2021.