

Towards Engineering Design Principles for Role-Based AI Translation Systems

Pavel Petrov¹, Vassil Milev², Filipp Safronov², Vladimir Filatov⁴

^{1,2}Department of Computer Systems and Technologies, Faculty of Mathematics and Informatics, St Cyril and St Methodius University of Veliko Tarnovo, Bulgaria

³Faculty of Modern Languages, St. Cyril and St. Methodius University of Veliko Tarnovo, Bulgaria

⁴Independent Software Researcher, Russia

Abstract

This paper explores the engineering principles of role-based AI translation systems. A configurable experimental framework was developed based on a review of existing multi-agent translation architectures to evaluate role allocation, model selection, glossary integration, and review strategies. Controlled experiments have identified the key factors that influence translation quality, and have formulated practical design principles for the development of efficient, domain-specific Role-Based AI translation systems.

Keywords: role-based machine translation, multi-agent systems, AI translation engineering, terminology integration, local language models, computational efficiency

1 Introduction

Multi-agent architectures promise higher translation quality by distributing responsibilities among specialised roles such as translator, editor, terminologist, and reviewer. While most studies compare agent-based and non-agent-based systems [1, 13] or propose new architectures [2, 14, 15], few investigate the engineering trade-offs of assigning roles to several specialised agents under fixed computational constraints [1, 7, 11].

The previous work on personalised AI translation assistants demonstrated that translation effectiveness depends not only on the underlying language model but also on workflow organisation, prompting, and knowledge integration [5]. Building upon these findings, the present study shifts the focus from assistant personalisation to architectural design. Existing multi-agent translation systems are first analysed to identify common engineering principles, which are then implemented in a configurable local experimental platform for controlled comparison of translation models, role allocation strategies, glossary integration, and review workflows.

All experiments were conducted on locally deployed models running on consumer-grade hardware. To evaluate architectures under realistic computational constraints, all multi-agent experiments employed lightweight 3B–4B language models for agent roles [12].

Multiple architectural configurations translated the same Bulgarian academic texts into English to evaluate the influence of model selection, functional role distribution, terminology

support, review strategies, execution time, and computational cost on translation quality.

Rather than introducing another multi-agent translation system, this work formulates experimentally validated engineering principles for designing local role-based AI translation systems. The resulting recommendations balance translation quality with computational efficiency, providing practical guidance for future domain-specific AI translation assistants.

2 Methods and Algorithms

2.1 Analysis of Existing Architectures

To identify common engineering approaches to role-based AI translation, several representative multi-agent systems were reviewed. These included TRANSAGENTS for literary translation [6]; MAATS for machine-aided translation [14]; HiMATE for hierarchical translation quality evaluation [15]; ReCodeAgent for source code translation at the repository level [2]; and HNCLMT for domain-specific machine translation [3]. While these systems differ in their application domains, workflow organisation, and evaluation strategies, they all employ specialised agents that collaborate to accomplish complex translation tasks.

Table 1 presents a comparative overview of the selected generative AI multi-agent systems.

Architecture	Translator	Reviewer/ Validator	Terminology role	Hierarchical coordination
TRANSAGENTS	Translator	Junior Editor (critic), Proofreader	(Localization Specialist, not terminology- specific)	Yes (CEO orchestrates Senior/Junior Editors, Translator, Localization Specialist, Proofreader)
MAATS	Translator Agent	MQM Evaluator Agents (multi-dimensi onal) [4]	Yes. dedicated MQM Terminology evaluator [4]	Partial (central Editor Agent synthesises evaluator output)
HiMATE	(evaluates existing translations, no generation role)	Tier-1 / Tier-2 agents	Implicit (possible Tier-2 error subtype)	Yes (two-tier structure with cross-tier discussion)
ReCodeAgent	Translator	Validator (test execution)	(not applicable; code translation)	Yes (Analyzer and Planning Agent coordinate the pipeline)
HNCLMT	Translator (translation layer)	Validator, Translation Confirmer	Yes (domain-extr action layer (Analyst, Validator, Confirmer))	Yes (two-layer structure (domain extraction + translation))

Table 1. Comparison of role structures in existing multi-agent translation architectures

The reviewed architectures exhibit several recurring design patterns, including:

- functional layering, which involves separating workflow management, translation execution, and quality assurance;
- role specialisation, whereby individual agents are assigned to specific translation functions;
- producer–validator separation, whereby translation generation and translation evaluation are performed by different agents;
- quality-driven role granularity, whereby specialised agents focus on particular quality dimensions, such as terminology, adequacy, or fluency;
- hierarchical coordination, which allows higher-level agents to resolve conflicts and validate final outputs.

The level of granularity in specialised roles varies across architectures. Simpler systems typically distinguish only between translation and review, whereas more sophisticated frameworks introduce dedicated agents responsible for terminology management, stylistic editing, workflow coordination, or domain analysis. Several architectures also use hierarchical control, whereby higher-level agents coordinate the work of specialised execution agents or resolve conflicting recommendations before producing the final output.

2.2 Experimental Hypothetical Design Principles

Analysis of existing role-based translation systems reveals four engineering principles that are consistently applied across different architectures.

Firstly, successful systems replace monolithic translation models with functional role allocation, whereby individual agents perform specialised tasks such as translation, editing, terminology management, or quality evaluation. Rather than increasing the number of agents, these architectures aim to allocate responsibilities according to the translation workflow.

Secondly, most architectures maintain a clear separation between producers and validators. The agent responsible for generating the translation differs from those responsible for evaluating and refining it. This reduces self-confirmation bias and enables quality assessment to focus on specific criteria.

Thirdly, translation is generally organised as an iterative translate–review–refine process rather than a single-pass generation process. Reviewer feedback is used to improve the translation with targeted corrections before the final output is produced.

Finally, existing systems commonly emulate professional translation workflows by assigning agents to clearly defined functional roles. This modular organisation simplifies context management, improves terminology consistency, and provides greater transparency than monolithic translation approaches.

These recurring principles represent common engineering design patterns rather than a single prescribed architecture. However, many published systems implement these patterns using complex, orchestrator-based topologies that require substantial computational resources. As the aim of this study is to evaluate role-based translation under local hardware constraints, the experimental platform adopts these same principles within a simplified sequential workflow. This design preserves functional role separation, reduces computational overhead, and enables the controlled comparison of alternative models and role configurations.

2.3 Experimental Platform and Evaluation Methods

Based on the engineering principles identified in the previous section, a lightweight, sequential, multi-agent translation platform was developed for controlled evaluation under

local deployment constraints. Rather than using complex orchestrator-based topologies, the platform implements role separation within a linear processing pipeline, thus preserving the key principles of role-based translation while reducing computational overhead.

Figure 1 presents the experimental workflow of the platform, which consists of six stages:

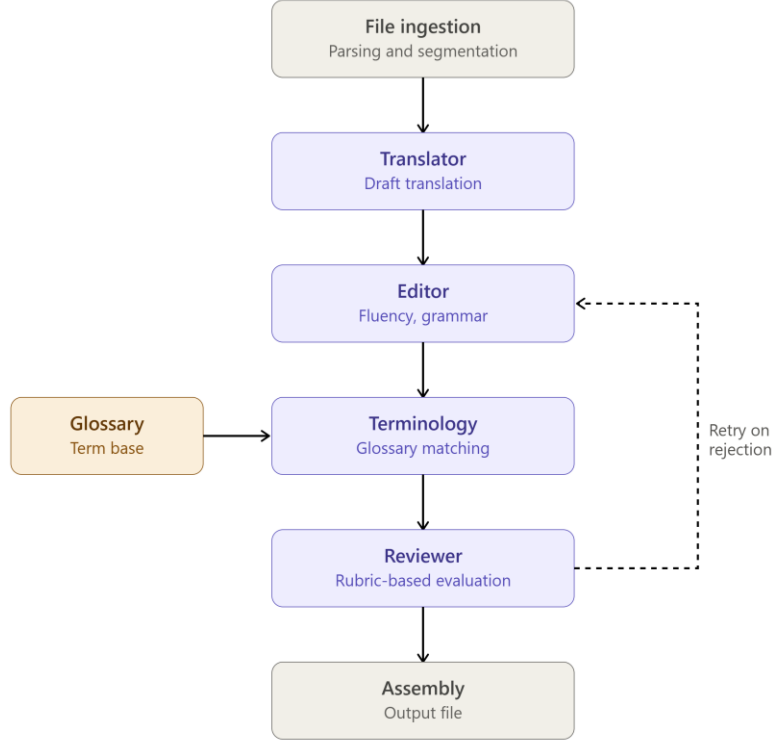


Figure 1. The experimental workflow of multi-agent translation architectures

The specialised NLLB-200-600M model served as the LLM-core translation backbone, while the lightweight Qwen2.5-3B model was assigned all auxiliary roles (Editor, Terminologist, and Reviewer). This reflects a realistic local deployment scenario in which compact 3B–4B language models are feasible for multi-agent execution [12].

An experimental evaluation was conducted on a relatively small corpus of domain-specific academic texts to provide initial validation of the proposed architecture. Future studies will assess the scalability and statistical robustness of the proposed design principles using larger datasets and longer documents.

Translation quality is evaluated against human reference translations using three complementary automatic metrics: BLEU [8], chrF [9], and COMET [10]. BLEU measures lexical similarity; chrF evaluates character-level correspondence, being particularly suitable for morphologically rich languages; and COMET estimates semantic adequacy using multilingual neural representations trained on human-quality judgments. Together, these metrics provide a balanced assessment of lexical accuracy, linguistic quality, and semantic fidelity. In addition to translation quality, execution time, and computational cost were recorded to analyse the practical efficiency of different architectural configurations.

3 Results and Discussion

3.1 Validation of the Experimental Hypothetical Design Principles

In this role-ablation study, functional stages were added incrementally to a single-model baseline in order to assess their individual contribution. The complete quantitative results are presented in Table 2:

Configuration	chrF	BLEU	COMET	Time (min)	Tokens
Baseline LLM Core Translator (no multi-agents)	82.97	58.40	86.51	1.2	2,136
Editor only	82.11	56.78	86.94	1.8	10,091
Terminology only	84.22	61.56	87.10	0.8	7,755
Editor + Terminology	83.33	59.01	87.15	1.6	15,713
Full pipeline (with reviewer)	83.12	60.87	87.17	2.3	20,476

Table 2. Evaluation of the experimental hypothetical design: role-ablation study

The experiments suggest that the contribution of individual roles is not cumulative. The terminology stage produced the largest improvement in translation quality. In contrast, the editor stage provided only marginal benefit, and the addition of the reviewer increased computational cost without improving translation quality correspondingly.

Figure 2 presents the distribution of token consumption across the individual pipeline roles for each experimental configuration:

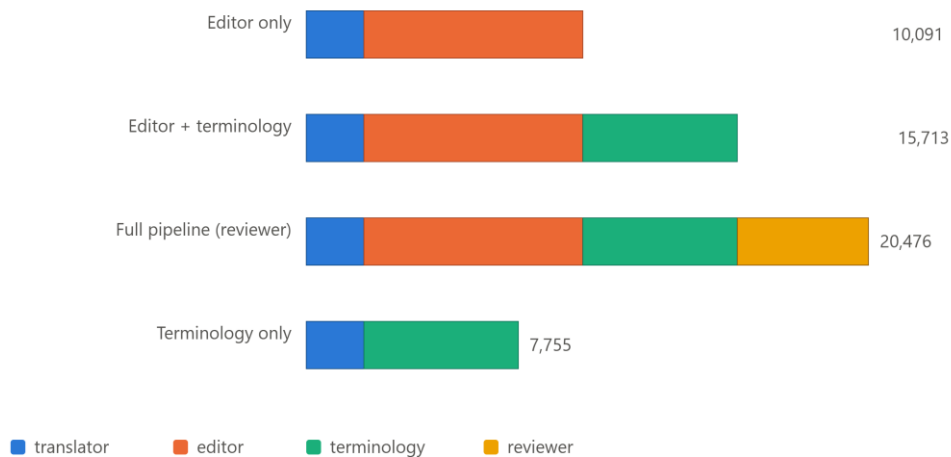


Figure 2. Distribution of token consumption across the evaluated pipeline

The token consumption shows that adding extra processing stages substantially increases the computational cost while only providing limited improvements in translation quality. These results suggest that the effectiveness of role-based architectures depends more on allocating high-impact specialised functions than on increasing the number of agents involved, thereby highlighting the importance of balancing translation quality with computational efficiency.

3.2 Key findings on engineering design principles

The experiments demonstrate that the effectiveness of role-based translation systems is more dependent on functional role allocation than on the overall complexity of the agent architecture. The specialised machine translation model was found to consistently provide the strongest contribution to translation quality, confirming its role as the backbone of the system.

In contrast, replacing or extending the translation process with general-purpose language model agents often degraded translation quality as these models introduced unnecessary modifications and inconsistencies. While larger language models could potentially improve these results, they would require substantially greater computational resources and execution time. The most effective use of language-model agents was therefore in complementary validation roles, particularly for terminology verification, where they refined rather than replaced the output of the translation model.

These findings support a lightweight engineering approach to local AI translation; whereby specialised machine translation models perform the core translation task, and language models are reserved for post-processing and quality assurance functions where they add demonstrable value. Rather than increasing the number of agents, future architectures should optimise the functional contribution of individual roles, balancing translation quality with computational efficiency.

3.3 Future research directions

Future research should extend the evaluation of role-based translation systems beyond translation quality to include translation productivity. This should consider execution time, token consumption, hardware requirements and overall computational efficiency, as well as traditional quality metrics. Such an evaluation framework would better reflect the practical constraints of locally deployed AI translation assistants.

The contribution of individual pipeline components should be investigated by evaluating the roles of translator, editor, terminologist, and reviewer independently with dedicated models. Finally, the proposed engineering principles should be validated using larger multilingual datasets, additional language pairs, and diverse hardware platforms to establish their general applicability.

4 Conclusion

This study examined the engineering principles underlying role-based AI translation systems, evaluating their practical validity in the context of local, resource-constrained deployment. A review of representative multi-agent architectures, including TRANSAGENTS, MAATS, HiMATE, ReCodeAgent, and HNCLMT, revealed several recurring design patterns. These included functional layering, role specialisation, producer–validator separation, quality-driven role granularity, and hierarchical coordination. These patterns were then reformulated as a simplified, sequential experimental pipeline suited to consumer-grade hardware, thereby avoiding the complexity associated with orchestrator-based systems.

Controlled ablation experiments revealed that the contribution of individual roles is not cumulative; translation quality improved substantially only when a role introduced a capability lacking in the core translation model, as was most clearly demonstrated by the terminology stage. The editor role provided negligible benefit, and adding a reviewer increased execution time and token consumption without a corresponding improvement in quality. These results suggest that the engineering effort in role-based translation systems would be better directed towards identifying and allocating high-impact functional roles rather than expanding the number of agents in the pipeline. Accordingly, the study's central design principle is targeted specialisation: a compact core translation model complemented by a small number of validation agents whose functional value has been demonstrated empirically, rather than full emulation of professional translation workflows.

Taken together, these findings offer practical guidance to developers of domain-specific AI translation assistants with limited computational resources: prioritise terminology

integration as the most reliable means of improving quality; treat editing and review roles as optional additions, justified by measurable gains rather than assumed by architectural convention; and evaluate translation quality alongside execution time and token cost, rather than in isolation. Although the results are based on a single language pair and a limited set of academic texts, the proposed engineering principles — allocating roles based on demonstrated functional contribution rather than architectural complexity — provide a transferable framework for future work.

References

- [1] V. Briva-Iglesias, *Are AI Agents the New Machine Translation Frontier? Challenges and Opportunities of Single- and Multi-Agent Systems for Multilingual Digital Communication*, Proceedings of Machine Translation Summit XX, vol. 1 (2025), 365-377.
- [2] A.R. Ibrahimzada, et al., *ReCodeAgent: A Multi-Agent Workflow for Language-Agnostic Translation and Validation of Large-Scale Repositories*, arXiv preprint arXiv:2604.07341 (2026).
- [3] M. Li, K. Zhang, *A Multi-Agent System Based on HNC for Domain-Specific Machine Translation*, Scientific Reports, vol. 15, no. 1 (2025), 20820.
- [4] A. Lommel, et al., *The Multi-Range Theory of Translation Quality Measurement: MQM Scoring Models and Statistical Quality Control*, arXiv preprint arXiv:2405.16969 (2024).
- [5] V. Milev, F. Safronov, M. Racheva, *A Model for Customizing AI Assistants for Specialized Translation for Educational Purposes*, Digital Educational Technologies (2026), 55-67, <https://doi.org/10.54664/NYFS9761>.
- [6] W. Minghao, et al., *(Perhaps) Beyond Human Translation: Harnessing Multi-Agent Collaboration for Translating Ultra-Long Literary Texts*, Transactions of the Association for Computational Linguistics, vol. 13 (2025), 901-922.
- [7] A. Nuraeni, N.S. Saptaningsih, D. Adika, *Role-Based Prompting for Translation: A Study on Language Style and Translation Technique of Airline Slogan Translation Generated by ChatGPT*, LiNGUA, vol. 21, no. 1 (2026), 49.
- [8] K. Papineni, et al., *Bleu: A Method for Automatic Evaluation of Machine Translation*, Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (2002), 311-318.
- [9] M. Popović, *chrF++: Words Helping Character n-Grams*, Proceedings of the Second Conference on Machine Translation (2017), 612-618.
- [10] R. Rei, et al., *COMET: A Neural Framework for MT Evaluation*, Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP) (2020), 2685-2702.
- [11] R. Sajja, et al., *Artificial Intelligence-Enabled Intelligent Assistant for Personalized and Adaptive Learning in Higher Education*, Information, vol. 15, no. 10 (2024), 596.
- [12] R. Sharma, M. Mehta. Small language models for agentic systems: A survey of architectures, capabilities, and deployment trade-offs. *arXiv preprint arXiv:2510.03847*, (2025).
- [13] B. Tianchi, et al., *Multi-Agent Learning for Neural Machine Translation*, Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) (2019), 856-865.
- [14] G. Wang, J. Hu, S. Ali, *MAATS: A Multi-Agent Automated Translation System Based on MQM Evaluation*, arXiv preprint arXiv:2505.14848 (2025).

- [15] S. Zhang, et al., *HiMATE: A Hierarchical Multi-Agent Framework for Machine Translation Evaluation*, Findings of the Association for Computational Linguistics: EMNLP (2025), 11121-11145.

Copyright © 2026 Pavel Petrov, Vassil Milev, Filipp Safronov, Vladimir Filatov. This is an open-access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.