

Self-Evolving Multi-Model Memory Loop (SEMML)

Thomas Felber

vedanta.study82@gmail.com

Abstract

The Self-Evolving Multi-Model Memory Loop (SEMML) is a proposed approach to machine translation in which several language models work with a shared collection of reviewed bilingual resources.

The collection begins with prepared translations, terminology and contextual information. For a new request, the system either reuses a suitable translation or retrieves material to support the generation of alternatives. One translation is selected for the current request. A separate review determines whether it should enter the memory for later use. SEMML thus connects the preparation of language resources with the continuing use and revision of model outputs. Its contribution is the organization of this recurring process: several models draw on the same resources, reviewed results become available to later requests, and their origin and conditions of use remain traceable. This paper presents the approach, explains how its parts work together and discusses its relationship to existing translation and memory methods.

1 Introduction

SEMML brings several language models into a translation process that retains and maintains useful results. The shared memory is the connection between individual requests. It provides the models with relevant material, receives selected translations after review and preserves the information needed to decide where those translations can be used again. Over time, the available resources reflect both the material prepared at the outset and the work completed through the system.

The starting problem is familiar in repeated translation work. A text may contain terminology already resolved in an earlier document, yet a new model call can produce a different rendering. Supplying previous material can help, but generated translations also need scrutiny before they become references for further work. Once an error enters the reference collection, later translations may repeat it. The question is therefore how to carry useful work forward while retaining control over what is stored and reused.

SEMML addresses this problem by making review the link between a current answer and future resources. Selecting an answer settles the immediate request. Storing that answer gives it a further role: it may guide later generation or be returned when an appropriate request recurs. These decisions have different purposes. A proposal can be the most suitable available answer without being sufficiently checked for continued use.

The full name, Self-Evolving Multi-Model Memory Loop, describes this arrangement. Several models generate alternatives, while the external memory changes as translations are accepted, corrected or withdrawn. The language models themselves remain unchanged during the process. The intended development occurs in the collection of usable resources and in the record of how they were obtained.

2 SEMML Loop Architecture

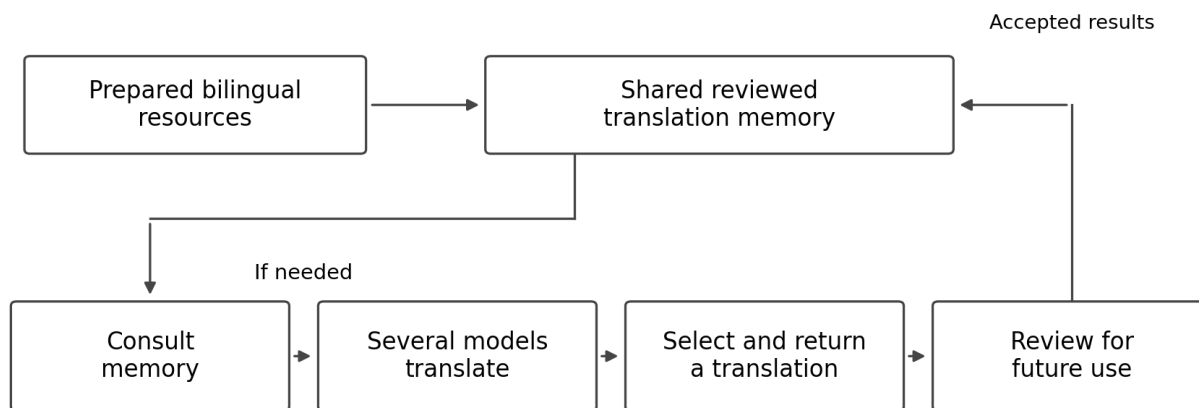


Figure 1 The SEMML loop. The figure shows the path used when a new translation is generated. A reviewed translation that fits the request and its context can also be returned directly from memory.

2.1 Preparing the shared resources

The process starts with a collection prepared for a particular language pair and field of use. It can include parallel text passages, approved terminology, explanations of abbreviations and other relevant language material. Preparation establishes what each entry contains, where it comes from and when it is appropriate to use. Conflicting translations can be retained if their different contexts are recorded.

A terminology entry can guide the translation of a word without supplying a complete sentence. A short explanation can clarify a meaning without providing an approved target-language expression. SEMML keeps these distinctions visible so that a fragment of supporting information is not silently treated as a complete translation. The prepared collection also needs review; manual inclusion alone does not establish accuracy.

2.2 Consulting the memory

A request supplies the source text, the target language and the available document context. The system first checks whether the memory contains a translation suitable for direct reuse. The same source wording is not enough when its meaning or required terminology changes with the surrounding text. Language pair, subject area and target-language conventions also affect this decision.

If a stored answer does not fit, the system retrieves material that may help produce a new one. This can be a related passage, an approved term or an explanation. Retrieval supplies information to the models; it does not turn every related entry into a ready-made answer. Where the available material is contradictory, the conflict must be resolved or carried into the review rather than hidden by the search.

2.3 Generating and selecting a translation

Several models receive the same source text, context and retrieved material. They produce alternatives separately, without seeing one another's drafts. Their outputs may differ in terminology, phrasing or interpretation. This offers a basis for selection, although models can still share the same errors.

SEMML uses an established selection method to choose among the usable alternatives. Failed or incomplete outputs are excluded, and the system needs a stated procedure for ties or the absence of a

usable response. Selection returns an existing candidate. It does not by itself establish the candidate's accuracy or authorize storage.

2.4 Reviewing results for future use

The selected translation is considered separately for inclusion in the memory. The review examines it against the source text and the requirements of the task. It may be performed by a qualified bilingual reviewer, by suitable task-specific checks or by an automatic procedure whose reliability has been established for that use. The important requirement is an actual basis for acceptance beyond the fact that a candidate was selected.

An accepted entry is stored with its source, context, review information and version. It becomes available only after review is complete. A rejected or pending translation may be retained for documentation, but it is excluded from the resources supplied to later requests. Whether the current answer is also reviewed before delivery is a separate decision for the application; review for future storage does not imply that it has been.

Corrections produce a new version. The previous version remains identifiable but is withdrawn from active use. Records of the resources consulted during generation make it possible to find later translations that may also need attention. Those records show where an error could have travelled; a reviewer still has to determine whether a later translation is actually affected.

3 Methodological Novelty and Contributions

SEMML is organized around a shared body of bilingual work that both supports generation and receives reviewed results. Preparing resources, producing alternatives and maintaining translations belong to the same recurring process. A correction can therefore change the material used for subsequent requests, while the models continue to operate without retraining. This gives the approach a clear focus: retaining useful translation decisions without treating every generated answer as a new reference.

The combination matters because each part serves a different need. Prepared material gives the process a domain-specific starting point. Several models provide alternative translations of the current text. Review determines which results are suitable for continued use. Context records limit where those results may be reused, and version history allows a translation decision to be revised. The proposed value comes from the way these decisions are connected across requests.

Existing work provides the foundations. Translation-memory integration combines earlier translations with machine translation [1]. Retrieval-augmented generation supplies external information during generation [2], and semantic caching returns previously produced answers [3]. These methods explain how earlier material can enter a new response. SEMML adds an explicit account of how a selected translation is reviewed, incorporated into the shared collection and subsequently maintained as part of that continuing process.

Multi-agent translation also has precedents. CRAT, for example, combines specialized agents, bilingual retrieval and information checking [4]. SEMML concentrates on the shared translation memory that connects successive requests, including the future use of accepted results. A-MEM and Mem0 show how external memories can be organized and updated [5, 6]. AECG addresses the reliability and revision of reusable procedural knowledge in multi-agent systems [7]. These are closely related ideas. SEMML applies such concerns to bilingual records, where the source passage, translation, terminology and document context determine whether reuse is appropriate.

The contribution proposed here is this particular arrangement for translation. It does not require a new retrieval method, a new similarity function or a new language model. Its claim to originality rests on bringing shared resources, alternative generation and reviewed reuse into a coherent translation approach. The paper makes no claim that each of these elements is unique in isolation, or that the combination has already demonstrated better performance.

4 Illustrative Translation Task

Consider a manufacturer translating a series of German maintenance manuals into English. The prepared collection contains reviewed passages and a terminology list for its equipment. A new manual uses a term for a component that has already been discussed in earlier documents. The system retrieves the relevant examples and supplies them, together with the new passage, to several language models.

The models produce different drafts. One follows the preferred component name but uses awkward wording; another is fluent but chooses a broader term. The selected answer addresses the current request. During the separate storage review, a bilingual reviewer checks the intended component and revises the wording if necessary. Only the accepted version enters the shared memory, together with the product context and the decision about terminology.

When a related passage appears in another manual, the accepted entry is available as supporting material. A matching passage can be reused directly if the context is still appropriate. The stored translation does not become a universal rendering of the term: a different product or field may require another choice.

Suppose the manufacturer later changes its approved component name. The previous entry is withdrawn from use and replaced with a reviewed version. The record of earlier consultations helps locate translations that may need revision. In this example, the loop carries a translation decision forward, preserves the reason for using it and allows it to be changed. The example illustrates the proposed operation; it is not a report of an implemented experiment.

5 Intended Use and Practical Limitations

SEMML is intended for work in which related requests recur and reviewed language material can remain useful across documents. Technical documentation and other recurring specialist texts are plausible settings. An organization can begin with a modest collection and define its review practices according to the subject matter and the consequences of error.

The main practical constraint is the work required to prepare and maintain the collection. Reusing translations can reduce repeated generation, but review and correction also take time. Several models add further expense. The benefit depends on whether the retained work is useful often enough to justify that effort. A growing memory alone says little about its value.

Quality remains dependent on the resources and the people or procedures that assess them. Models may agree on a wrong interpretation, and a reviewed entry may become unsuitable when the context changes. Recording its origin makes the decision inspectable; it does not guarantee correctness. The collection therefore needs a way to withdraw unreliable material and to revisit translations that relied on it.

The design leaves model choice, retrieval and candidate selection open to the needs of an application. Review and reuse rules must be made concrete when it is implemented. Practical evidence would then be needed to determine whether the approach improves consistency, translation quality or the amount of repeated work. The present paper establishes the concept and its intended operation.

6 Conclusion

SEMML proposes a continuing translation process in which several large language models use a shared, reviewed memory and selected results can return to that memory after assessment. The prepared collection gives the process its starting point; accepted translations extend what later requests can draw on. Keeping context, review decisions and corrections with the stored material makes this reuse accountable. The approach offers a way to connect established generation and memory methods around the accumulated work of translation.

References

- [1] Koehn, P., and Senellart, J. (2010). Convergence of Translation Memory and Statistical Machine Translation. Proceedings of the Second Joint EM+/CNGL Workshop, 21–31. <https://aclanthology.org/2010.jec-1.4/>
- [2] Lewis, P., et al. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. Advances in Neural Information Processing Systems. <https://arxiv.org/abs/2005.11401>
- [3] Bang, F. (2023). GPTCache: An Open-Source Semantic Cache for LLM Applications Enabling Faster Answers and Cost Savings. Proceedings of NLP-OSS, 212–218. <https://doi.org/10.18653/v1/2023.nlposs-1.24>
- [4] Chen, M., Meng, F., Zhang, Y., Zhang, Y., and Zhou, J. (2024). CRAT: A Multi-Agent Framework for Causality-Enhanced Reflective and Retrieval-Augmented Translation with Large Language Models. arXiv:2410.21067. <https://arxiv.org/abs/2410.21067>
- [5] Xu, W., Liang, Z., Mei, K., Gao, H., Tan, J., and Zhang, Y. (2025). A-MEM: Agentic Memory for LLM Agents. Advances in Neural Information Processing Systems. <https://arxiv.org/abs/2502.12110>
- [6] Chhikara, P., Khant, D., Aryan, S., Singh, T., and Yadav, D. (2025). Mem0: Building Production-Ready AI Agents with Scalable Long-Term Memory. arXiv:2504.19413. <https://arxiv.org/abs/2504.19413>
- [7] Tian, A., et al. (2026). AECG: Asymmetric Experience Consolidation and Governance In Multi-Agent Systems. arXiv:2610.05176. <https://arxiv.org/abs/2610.05176>