Journal Title
XX(X):1–10
©The Author(s) 2025
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

# Decoupling Formal and Natural Language Token Embeddings in Fine-Tuned Models for Improved Ontology Integration[a]

[a] This work is an extended version of a paper published in the proceedings of *NeSy 2025 – the International Conference on Neurosymbolic Learning and Reasoning* titled ”Grounding Terms from an Ontology for use in Autoformalization: Tokenization is All You Need.”

**Abstract**

Large Language Models (LLMs) have shown strong performance in translating natural language into programming languages like Python or Java. However, for niche computer languages, where there is limited training data, fine-tuning a base model is often necessary. A key challenge arises when the pretrained embeddings of natural language terms interfere with the intended syntax and semantics of formal language terms. This issue is especially pronounced in the logical language of SUO-KIF, which is used in the Suggested Upper Merged Ontology (SUMO). SUMO contains thousands of terms that closely resemble everyday English words or phrases. As a result, models often produce syntactic errors or hallucinate non-existent terms due to conflicting embeddings learned during base training.

This work introduces a tokenization-based technique to mitigate these issues. By altering how formal terms are tokenized, we can decouple their embeddings from similar natural language words, significantly reducing syntax errors and term hallucinations in the generated formal language output.

**Keywords**

Formal Semantics, Natural Language Processing, Transformers, Automated Natural Language Translation, Neurosymbolic AI, Ontology, Logic and Learning, Tokenization, Fine-tuning

## Introduction

Formalizing natural language into logic-based representations is a long-standing goal in artificial intelligence as a way to enable machines to reason about the world. Among other benefits, formalization is a promising avenue for verifying the accuracy of large language model (LLM) output, detecting hallucinations, and providing provably correct responses (Huang, Ruan, Huang, Jin, Dong, Wu, Bensalem, Mu, Qi & Zhao, 2024). LLMs have excelled at translating natural language prompts into programming languages such as Python or Java. However, less attention has been paid to niche formal languages that support knowledge representation and logical inference (Xu, Alon, Neubig & Hellendoorn, 2022). Fine-tuning is a frequent solution for adding capability to an LLM, such as enhancing an LLM's ability to support niche formal languages.

One such language is SUO-KIF, a higher-order logic language used in the Suggested Upper Merged Ontology (SUMO) (Niles & Pease, 2001)*. The language and ontology together allow for the formal specification of entities and their attributes, enabling automated reasoning. For example, the sentence *"An apple is red."* may be formalized as:

```
(exists (?C)
  (and
    (instance ?C Apple)
    (attribute ?C Red)))
```

Although the logical language is syntactically quite simple, with keywords for just a few logical operators (`and, or, not, forall, exists, <=>, =>` and `equals`), we also must ensure that logical statements use the correct terms from the SUMO library. There are approximately 20,000 of these, which have been created by hand, each with detailed definitions in logic, over a 25 year period. These named concepts do not necessarily mirror the set of lexicalized tokens in English or any other human language. We cannot simply use words as though they are logical symbols since they may not align with the inventory of concept symbols in SUMO. For example, the verb phrase in "The money *changes hands*." is expressed in SUMO with the logical concept `ChangeOfPossession`†. In order to take advantage of the formulas in SUMO that detail what happens in that action, we have to generate a formula that uses that precise identifier and not just the surface English text. The problem may be considered similar to not only generating the correct syntax for Python or Java, but also selecting the right function or method names from tens of thousands of possible choices in libraries.

Another challenge that arises is that SUMO terms often resemble everyday English words. Despite fine-tuning, LLMs frequently produce syntax errors and hallucinated

*SUO-KIF specification, SUMO, and source code is found at https://github.com/ontologyportal

†The formulas associated with SUMO terms can be viewed on line at e.g. https://sigma.ontologyportal.org:8443/sigma/Browse.jsp?lang=EnglishLanguage&flang=SUO-KIF&kb=SUMO&term=ChangeOfPossession

terms due to statistical associations learned during pretraining. LLMs break input into subword tokens and embed them in a semantic space, where related terms cluster, as shown simplistically in Figure 2.

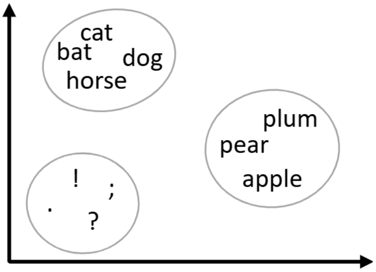


**Figure 1.** Simplistic view of an embedding space, where similar tokens are grouped together.

Strong embeddings can create a problem when translating to SUO-KIF. For example, variables begin with “?” and have no whitespace in SUO-KIF, while English treats “?” as punctuation. Sentences translated from English often have a space incorrectly inserted after the “?” in SUO-KIF, creating invalid syntax. For instance, when employing a fine-tuned Flan-T5 model trained on millions of examples, variable identifiers such as “?C” are consistently rendered as “? C”, introducing an unintended space between the question mark and the variable label.

In addition to invalid syntax, models may confuse formal SUMO terms like BatMammal with invalid hallucinated ones like BatAnimal, driven by natural language co-occurrence. Table 1 shows a list of hallucinated terms generated by a fine-tuned Flan-T5 model (Raffel, Shazeer, Roberts, Lee, Narang, Matena, Zhou, Li & Liu, 2023). Since these hallucinated formal terms are absent from the SUMO ontology, the logical formulas generated by the LLM translation exhibit limited inferential utility and cannot leverage the pre-existing knowledge provided by the ontology.

**Table 1.** List of hallucinated terms generated by a fine-tuned Flan-T5 model

| | | |
|---|---|---|
| Homes | AtmosphericAerosol | BiologicalSynthesis |
| CoolingEffect | Productivity | During |
| CommunicationTechnology | OutputPerHourWorked | AnnualReturn |
| UniversalTransferinstance | Disarmament | Habitat |

To address both invalid syntax and hallucinated types, we introduce a method to re-ground formal SUMO terms in a way that decouples them from natural language embeddings: SUMO terms are mapped to new, unrelated tokens. During the fine-tuning process these new terms are grouped appropriately in the embedding space, but lack the undesired positional encodings and statistical correlations of previous tokens learned by LLMs from English texts. This reduces both syntax errors and term hallucinations, significantly improving SUO-KIF generation quality.

## Background

### *SUO-KIF*

SUO-KIF (Standard Upper Ontology - Knowledge Interchange Format) is a formal language designed for expressing ontological and logical statements in a machine-readable way (Pease, 2009). Unlike more widely adopted languages such as the OWL family of description logics, SUO-KIF is a more expressive higher order logic (Benzmüller & Pease, 2010). It is particularly suited for encoding complex conceptual knowledge (Pease, 2021).

### *SUMO*

The Suggested Upper Merged Ontology (SUMO) is a comprehensive formal ontology originally intended to provide a foundation for more specific domain ontologies (Niles & Pease, 2001; Pease, 2011) but now expanded to include dozens of domain specific ontologies (Pease & Benzmüller, 2010) supported by a mid-level ontology. Developed to support automated reasoning, SUMO includes thousands of terms and axioms written in SUO-KIF that describe abstract concepts (e.g., `Attribute`, `Quantity`), relations (e.g., `part`, `orientation`) and concrete entities (e.g., `Apple`, `Book`, `Talking`). The Sigma Knowledge Engineering Environment (Pease & Schulz, 2014) translates SUMO to several languages in the TPTP family of logical languages (Sutcliffe, 2010) used by modern automated theorem provers. These include First Order Form (FOF) (Trac, Sutcliffe & Pease, 2008), Typed First order form with Arithmetic (TFA) (Pease, 2023) and Typed Higher Order Form (THF) (Benzmüller & Pease, 2010; Benzmüller & Pease, 2012). These languages are used for logical deduction over SUMO in Vampire (Kovács & Voronkov, 2013), Eprover (Schulz, 2002) and other theorem provers.

### *T5 and Flan-T5*

T5 (Text-to-Text Transfer Transformer) and its variant Flan-T5 are transformer-based language models that frame NLP tasks as text generation problems (Raffel, Shazeer, Roberts, Lee, Narang, Matena, Zhou, Li & Liu, 2023). In our work, we experiment with both T5 and Flan-T5 as the base models for translating informal English prompts into formal SUO-KIF expressions. Trained on a wide range of tasks using a unified format, T5 and Flan-T5 have shown strong performance across translation benchmarks, and are considered state of the art (Longpre, Hou, Vu, Webson, Chung, Tay, Zhou, Le, Zoph, Wei & others, 2023). Tokenization was conducted using the native T5Tokenizer, which is based on the popular SentencePiece tokenizer (Kudo & Richardson, 2018).

### *Alternative approaches*

Several alternative solutions that may enhance translation accuracy exist beyond the scope of this study. These approaches are complementary to the term re-grounding method presented here, and exploring the synergistic benefits of their integration represents a promising direction for future research.

*Vocabulary expansion* With vocabulary expansion, whole SUMO terms are explicitly added to the token set, and then selection of only these tokens is forced in the output. This approach has been shown to significantly increase memory requirements, cost, and training time (Toraman, Yilmaz, Sahinuc & Ozcelik, 2023). Additionally, many SUMO terms align exactly with existing tokens, thus vocabulary expansion mitigates, but does not eliminate the issue.

*Retrieval-augmented generation (RAG)* A RAG is an information retrieval framework that can be populated with SUMO terms. When presented with a prompt, the LLM queries the RAG system to fetch relevant SUMO terms, which are then integrated with the model's pre-existing knowledge to produce more accurate and contextually informed responses (Gao, Xiong, Gao, Jia, Pan, Bi, Dai, Sun, Wang & Wang, 2024). While the focus of this research is restricted to fine-tuning, RAGs have been shown to be effective at improving output accuracy of responses and can work in conjunction with fine-tuning (Balaguer, Benara, Cunha, Hendry, Holstein, Marsman, Mecklenburg, Malvar, Nunes, Padilha & others, 2024). Additionally, changes in SUMO are more easily reflected in a RAG, which can be updated without further training of the model, alleviating a weakness of the fine-tuning approach. While a RAG would likely assist greatly with type accuracy, it is less likely that it would alleviate syntax problems such as inserting spaces after question marks in variables.

## Related work

Researchers have shown fine-tuning to be an effective way to improve generation of programming language statements (Shypula, Madaan, Zeng, Alon, Gardner, Hashemi, Neubig, Ranganathan, Bastani & Yazdanbakhsh, 2024). Progress has also been made in auto-formalization of mathematical problems that have been expressed semi-formally (Wu, Jiang, Li, Rabe, Staats, Jamnik & Szegedy, 2022). In translating from one language to another, improper tokenization has been shown to increase the rate of hallucinations of terms (Wang, Li, Jiang, Ding, Luo, Jiang, Liang & Yang, 2024).

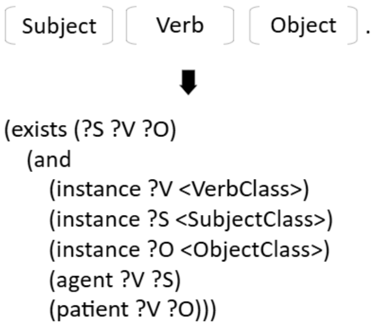
Ontology alignment and grounding have traditionally focused on matching human concepts across knowledge bases, but not on decoupling formal representations from their natural language counterparts within generative models. Previous work in semantic grounding, such as retrofitting embeddings using ontology structures (Faruqui, Dodge, Jauhar, Dyer, Hovy & Smith, 2015) or concept embeddings from WordNet (Camacho-Collados, Pilehvar & Navigli, 2016), aligns knowledge bases based on semantic similarity. However, these methods still preserve natural language proximity and do not address the situations where that proximity is problematic.

Several approaches have been proposed to handle domain-specific vocabularies in neural language models. Vocabulary expansion techniques add domain-specific terms directly to the model's token vocabulary (Toraman, Yilmaz, Sahinuc & Ozcelik, 2023), but this approach significantly increases memory requirements, computational cost, and training time. Additionally, when domain terms closely resemble existing vocabulary (as is often the case with SUMO terms), vocabulary expansion only mitigates the embedding interference problem.

## Methodology

### Training data generation

We used training data consisting of approximately 6 million English sentences paired with their SUO-KIF logic equivalent. Sentences were synthetically generated by iterating over sets of appropriate SUMO terms, using the manually created SUMO-WordNet mappings (Niles & Pease, 2003) to create a natural language English equivalent of formal SUMO terms, and inserting them into a sentence frame structure as shown in Figure 2. Corresponding SUO-KIF logic statements are generated by building a frame structure, or template, for each sentence and filling in the slots of the frame from the contents of SUMO; the objects, processes, and relationships. Because the intended semantics of the frame is understood, precisely equivalent English and logic expressions of the frame can be generated. Note that while we show a quite simple sentence here, our generated training set makes use of the full expressivity of SUMO and corresponding English syntax, including prepositions, units and quantities, temporal restrictions, conditionals and modal expressions.



```
(exists (?S ?V ?O)
  (and
    (instance ?V <VerbClass>)
    (instance ?S <SubjectClass>)
    (instance ?O <ObjectClass>)
    (agent ?V ?S)
    (patient ?V ?O)))
```

**Figure 2.** Example sentence frame structure used to synthetically generate training data.

### Term and key word regrounding

SUO-KIF keywords, variables, and unique SUMO terms were extracted from the training data and assigned a corresponding label composed of five random capital letters. An example is shown in Table 2. For example, the SUMO term `Historian` was mapped to AJOFN. In this way, we semantically separate the SUMO term from its English token, without a strong grounding in the model. During training, the new mappings are moved to appropriate locations in the embedding space. Characters such as parentheses and whitespace were not translated, as the base model already handles them properly.

**Table 2.** An example of re-grounding

| English sentence | The historian is not awake right now. |
|---|---|
| Logic translation | `(not`<br>` (exists (?H)`<br>`  (and`<br>`   (attribute ?H Historian)`<br>`   (equal ?T Now)`<br>`   (holdsDuring ?T`<br>`    (attribute ?H Awake)))))` |
| Re-grounded translation | `(SIRQJ`<br>` (LOAXA (UGQJQ)`<br>`  (QAGRM`<br>`   (RJGUO UGQJQ AJOFN)`<br>`   (ANNFF KABBQ OFHBH)`<br>`   (ILEGC KABBQ`<br>`    (RJGUO UGQJQ LZEJO)))))` |

Fine-tuning is conducted using both the Flan-T5 and T5 models. A baseline model was fine-tuned for each, using the English sentences and the normal logic translations for training data. Re-grounded models were also trained for both Flan-T5 and the T5 models, using the English sentences and the re-grounded translations.

*Evaluation metric* Testing was conducted on a standard, static set of 100 test sentences, chosen for their grammatical diversity that remained constant through the experiment. Additionally, testing was conducted with 100 sentences randomly chosen from a corpus of policy documents comprising approximately 60,000 sentences. For the models trained using re-grounded SUMO terms, postprocessing was conducted to translate the model output back to the corresponding SUMO terms. A translation was considered syntactically correct if it followed the SUO-KIF grammar rules, without regard to type accuracy. A translation was considered type correct if all terms in the translation had a corresponding term in SUMO.

Baseline data was gathered using models that did not employ the re-grounding technique and compared with models that used re-grounding. The percent of correct translations was calculated using the following formula:

$$\text{Percent Correct} = \frac{\text{Number Correct}}{\text{Total Sentences}} \times 100\%$$

## Results

Results using a static set of sentences are shown in Table 3. Results with sentences pulled randomly from a large corpus are shown in Table 4.

**Table 3.** Percent of sentences translated correctly on standard test set.

| | Baseline | | SUMO Terms Re-grounded | |
|---|---|---|---|---|
| | **T5** | **Flan-T5** | **T5** | **Flan-T5** |
| Syntax correct | 45% | 72% | 94% | 93% |
| Types correct | 8% | 9% | 66% | 74% |
| Both correct | 7% | 8% | 61% | 70% |

**Table 4.** Percent of sentences translated correctly on randomized test set.

| | Baseline | | SUMO Terms Re-grounded | |
|---|---|---|---|---|
| | **T5** | **Flan-T5** | **T5** | **Flan-T5** |
| Syntax correct | 53% | 58% | 90% | 85% |
| Types correct | 3% | 4% | 63% | 74% |
| Both correct | 3% | 4% | 59% | 62% |

An important caveat to the baseline data is that the syntactic accuracy was only achieved with postprocessing by ensuring there was white space before the "?" character, and removing any white space between it and the variable name. Without this postprocessing step, not a single sentence would be syntactically correct when using either baseline model. This postprocessing step is not conducted (or needed) for the re-grounded models.

Re-grounding SUMO terms had a significant positive impact in both the T5 and Flan-T5 models. Type accuracy improved greatly, and fewer terms were hallucinated. It is hypothesized that greater grammar and vocabulary diversity in the training data will further reduce remaining type errors. Future work should include a study on how the extent of term re-grounding affects the semantic accuracy of the translation. While re-grounding did not appear to have had much influence on semantic accuracy, the impact will become clearer as the diversity of the training set grows.

## Conclusion

This study demonstrates that formal terms that overlap semantically or syntactically with everyday English significantly hinder the accurate translation of natural language into formal languages like SUO-KIF using LLMs. We introduced a simple yet effective tokenization-based re-grounding technique that disconnects formal SUMO terms from their natural language embeddings. By mapping these terms to randomized token sequences, we reduce the influence of pretrained embeddings and improve both syntactic and type accuracy. Our experiments with T5 and Flan-T5 models show that this approach dramatically outperforms baseline fine-tuning in terms of correctness and robustness. This method itself is model agnostic, adds minimal computational overhead to the fine-tuning, and requires no vocabulary expansion. Future work will explore its effects on semantic fidelity and extend its application to more diverse language inputs.

## References

Balaguer, A., Benara, V., Cunha, R. L. d. F., Hendry, T., Holstein, D., Marsman, J., Mecklenburg, N., Malvar, S., Nunes, L. O., Padilha, R. et al. (2024). RAG vs Fine-tuning: Pipelines, Tradeoffs, and a Case Study on Agriculture. *arXiv preprint arXiv:2401.08406.*

Benzmüller, C. & Pease, A. (2010). Progress in automating higher-order ontology reasoning. In B. Konev, R. Schmidt & S. Schulz (Eds.), *Workshop on Practical Aspects of Automated Reasoning (PAAR-2010)* (pp. 22–32). Edinburgh, UK: CEUR Workshop Proceedings.

Benzmüller, C. & Pease, A. (2012). Higher-Order Aspects and Context in SUMO. In I. J. V. Jos Lehmann & A. Bundy (Eds.), *Special issue on Reasoning with context in the Semantic Web*, Volume 12-13. Science, Services and Agents on the World Wide Web.

Camacho-Collados, J., Pilehvar, M. T. & Navigli, R. (2016). Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities. In *Artificial Intelligence* (pp. 36–64).

Faruqui, M., Dodge, J., Jauhar, S. K., Dyer, C., Hovy, E. & Smith, N. A. (2015). Retrofitting word vectors to semantic lexicons. In *Proceedings of NAACL* (pp. 1–31).

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M. & Wang, H. (2024). Retrieval-Augmented Generation for Large Language Models: A Survey. *arXiv preprint arXiv:2312.10997.*

Huang, X., Ruan, W., Huang, W., Jin, G., Dong, Y., Wu, C., Bensalem, S., Mu, R., Qi, Y. & Zhao, X. (2024). A survey of safety and trustworthiness of large language models through the lens of verification and validation. *Artificial Intelligence Review*, *57(7)*, 175.

Kovács, L. & Voronkov, A. (2013). First-order theorem proving and Vampire. In *CAV* (pp. 1–35).

Kudo, T. & Richardson, J. (2018). SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. *arXiv preprint arXiv:1808.06226.*

Longpre, S., Hou, L., Vu, T., Webson, A., Chung, H. W., Tay, Y., Zhou, D., Le, Q. V., Zoph, B., Wei, J. et al. (2023). The Flan Collection: Designing Data and Methods for Effective Instruction Tuning. In *International Conference on Machine Learning* (pp. 22631–22648).

Niles, I. & Pease, A. (2001). Towards a standard upper ontology. In *Proceedings of the International Conference on Formal Ontology in Information Systems - Volume 2001*, FOIS '01 (p. 2–9). New York, NY, USA: Association for Computing Machinery.

Niles, I. & Pease, A. (2003). Linking Lexicons and Ontologies: Mapping WordNet to the Suggested Upper Merged Ontology. In *Proceedings of the IEEE International Conference on Information and Knowledge Engineering* (pp. 412–416).

Pease, A. (2009). SUO-KIF Reference Manual. https://github.com/ontologyportal/sigmakee/blob/master/suo-kif.pdf. retrieved 20 June 2020.

Pease, A. (2011). *Ontology: A Practical Guide*. Articulate Software Press.

Pease, A. (2021). Choosing a Logic to Represent the Semantics of Natural Language. In *In Proceedings of the 4th International Conference on Logic and Argumentation (CLAR2021).*

Pease, A. (2023). Converting the suggested upper merged ontology to typed first-order form. *arXiv preprint arXiv:2303.04148.*

Pease, A. & Benzmüller, C. (2010). Ontology Archaeology: Mining a Decade of Effort on the Suggested Upper Merged Ontology. *The ECAI-10 Workshop on Automated Reasoning about Context and Ontology Evolution*.

Pease, A. & Schulz, S. (2014). Knowledge Engineering for Large Ontologies with Sigma KEE 3.0. In *The International Joint Conference on Automated Reasoning*.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W. & Liu, P. J. (2023). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer.

Schulz, S. (2002). E - A Brainiac Theorem Prover. *AI Commun.*, *15(2-3)*, 111–126.

Shypula, A., Madaan, A., Zeng, Y., Alon, U., Gardner, J., Hashemi, M., Neubig, G., Ranganathan, P., Bastani, O. & Yazdanbakhsh, A. (2024). Learning Performance-Improving Code Edits. *arXiv preprint arXiv:2302.07867.*

Sutcliffe, G. (2010). The TPTP world - infrastructure for automated reasoning. In *LPAR (Dakar)* (pp. 1–12).

Toraman, C., Yilmaz, E. H., Sahinuc, F. & Ozcelik, O. (2023). Impact of Tokenization on Language Models: An Analysis for Turkish. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, *22(4)*.

Trac, S., Sutcliffe, G. & Pease, A. (2008). Integration of the TPTPWorld into SigmaKEE. In *Proceedings of IJCAR '08 Workshop on Practical Aspects of Automated Reasoning (PAAR-2008)*. CEUR Workshop Proceedings.

Wang, D., Li, Y., Jiang, J., Ding, Z., Luo, Z., Jiang, G., Liang, J. & Yang, D. (2024). Tokenization Matters! Degrading Large Language Models through Challenging Their Tokenization. *arXiv preprint arXiv:2405.17067.*

Wu, Y., Jiang, A. Q., Li, W., Rabe, M., Staats, C., Jamnik, M. & Szegedy, C. (2022). Autoformalization with large language models. *Advances in Neural Information Processing Systems*, *35*, 32353–32368.

Xu, F. F., Alon, U., Neubig, G. & Hellendoorn, V. J. (2022). A systematic evaluation of large language models of code. MAPS 2022 (p. 1–10). New York, NY, USA: Association for Computing Machinery.