ShZZaM: An LLM+ATP Natural Language to Logic Translator

First1 Last1, First2 Last2
Affiliation
City, Country

Abstract

This paper describes the ShZZaM tool that uses Large Language Models (LLMs) and Automated Theorem Proving (ATP) tools to translate natural language to typed first-order logic in the TFF syntax of the TPTP World.

Large Language Models (LLMs) (Peykani et al. 2025) have shown themselves to be useful in a broad range of applications (Sajjadi Mohammadabadi et al. 2025). However, it is well known that LLMs make mistakes (Huang et al. 2025), and this is acknowledged on LLMs' web interfaces, e.g., ChatGPT admits "ChatGPT can make mistakes. Check important info". In the face of such unreliability, the results from LLMs in mission-critical applications require verification. One approach is to translate the LLM input and output to a logical form that can be checked using Automated Theorem Proving (ATP) tools, e.g., (Yang et al. 2025; Cheng et al. 2025).[1] A key step in this verification pipeline is the faithful translation of the natural language to an appropriate logical form. This task is difficult due to the ambiguous nature of natural language statements, especially informally expressed statements. Work in this area includes LINC (Olausson et al. 2023), FOLIO (Han et al. 2024), and LINA (Li et al. 2024). This paper makes another contribution in this area, taking a new interactive approach to the translation process, zigzagging (hence the 'ZZ' in the tool name) between natural language and logic until convergence is achieved. A key feature of ShZZaM is its use of LLMs and Automated Theorem Proving (ATP) tools, which complement each other in the translation steps.

Figure 1 shows the overall process implemented of ShZZaM. Starting with the natural language, a combination of LLMs and ATP tools make a first translation (LLM-L+ATP - a "Zig") to the typed first-order logic in the TFF syntax (Sutcliffe et al. 2012; Blanchette and Paskevich 2013) of the TPTP World (Sutcliffe 2024). An LLM is then used to translate the logic back to natural language (LLM-NL - a "Zag"). An LLM is then used to judge (LLM-S) the similarity in meaning of the new and previous natural language statements. If they are adequately similar - above a "convergence threshold", the logic inbetween them is accepted as the translation. This zizagging continues until the natural language pairs converge to the required level of similarity (or a limit is reached). Upon convergence the logic is sent to an ATP system, either a model finder if there are only axioms in the logic, or a theorem prover if there is also a conjecture. The results from the ATP system is reported in the SZS format (Sutcliffe 2008). If the similarity between the final natural language and the original natural language (which is computed in the zigzag step - see below) is above the "zigzagging threshold" the translation is complete. Otherwise the entire process repeats (or a limit is reached). This outermost loop ensures the final natural language of the converged pair is adequately similar in meaning to the original natural language.

The translation from natural language to TFF logic and back to natural language - one zigzag, is an iterative one involving LLMs and ATP tools. Figure 2 shows the details. LLM-L is used to translate from natural language to logic. The translation is successively checked using ATP tools for syntax errors and type errors (recall the logic is *typed* first-order logic). If an error occurs in either check the error message is captured and passed back into the LLM-L for another attempt. When a syntactically and type correct logic is created, LLM-NL translates the logic back to the provisional natural language, and LLM-S is used to compare this to the original natural language. If the similarity is below an "acceptance threshold" the provisional natural language is rejected and the error is passed back into the LLM-L for another attempt. This prevents the new natural language straying too far in meaning from the original natural language. If the similarity is below the acceptance threshold the provisional natural language becomes the accepted result of the zigzag. It is this accepted natural language that is compared to the previous natural language for convergence, as explained above.

The LLM translation from natural language to logic and back again works because the LLM has been exposed to enough natural language and enough TPTP format TFF logic. The former is the natural result of scraping the world's web sites, etc. The latter might be surprising, as TFF is a

[1] For a more comprehensive survey, just ask your favourite LLM to "show me some research on how LLMs make mistakes, and the need for symbolic checking of LLM output".

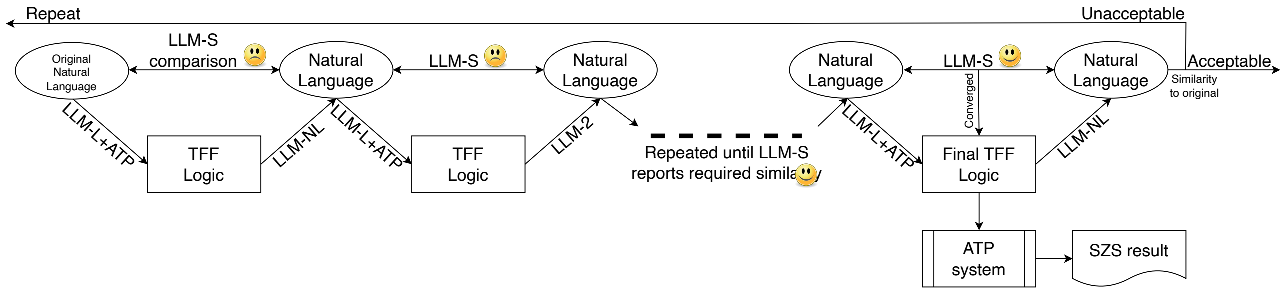

Figure 1: ShZZaM process

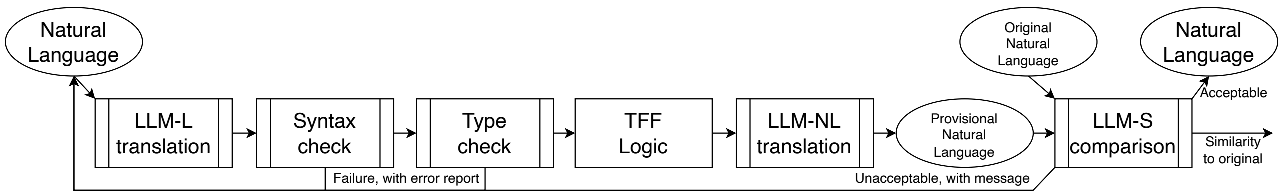

Figure 2: Details of one zigzag

comparatively small fragment of the data used to train LLM models. Evidently there are adequate corpora that use TFF that are exposed on the web, e.g., the TPTP problem library (Sutcliffe 2017), exports from the Isabelle Archive of Formal Proofs (Blanchette et al. 2015), exports of the Mizar Mathematical Library (Urban 2003), etc.

ShZZaM is implemented in Python. The LLMs available are OpenAI's `gpt-5-chat-latest` and Google's `gemini-2.5-flash`. Access to ATP tools is provided via the TPTP World's SystemOnTPTP service (Sutcliffe 2000). ShZZaM has parameters that allow the selection of LLMs (default OpenAI) and ATP systems (default Vampire (Bártek et al. 2025) for both theorem proving and model finding), setting the acceptance, convergence, and zigzagging thresholds (defaults 0.74, 0.94, 0.94), setting the maximal number of failures in a Zig (default 10), number of zigzags in a sequence (default 10), and number of zigzagging repetitions (default 3). ShZZaM is available, with test files and the results discussed below, from `https://github.com/GeoffsPapers/ShZZaM`. It can also be run in default mode in SystemB4TPTP, at `https://tptp.org/cgi-bin/SystemB4TPTP`, by selecting "English" as the "Input as in" option. The user has to provide their OpenAI API key as an initial comment line, e.g.,

```
# OPENAI_API_KEY=the_user_api_key
ShZZaM is a nice translation tool.
```

No tool needs to be selected – just "ProcessProblem".

Initial testing has been done on 12 test texts from SUMO-based research (Niles and Pease 2001; Thompson et al. 2025). There are four texts that are expected to produce axioms with a provable conjecture, three texts that that are expected to produce axioms with an unprovable conjecture, and five contradictory texts that are expected expected to produce unsatisfiable axioms. An example of the first type is: "If a country is a member of NATO, it will protect any member that is attacked. Sweden is a member of NATO. Germany is a member of NATO Russia attacked Germany. Will Sweden protect Germany?". An example of the second type is: "Terry possesses a Traditional Savings Account. Terry withdrew from the Traditional Savings Account. The bank penalized Terry. Did the withdrawal cause a penalty?". An example of the third type is: "The tornado damaged the house. The tornado did not damage the house.". Testing used the default settings, plus additionally used the Google LLM for the language translations. ShZZaM was run on each text three times in each of the two configurations so that stochastic variations could be analysed, for a total of 72 runs. Of the 72 runs, 65 ended in convergence, and 36 converged above the 0.94 zigzagging threshold. Over the 72 runs, 26 required only one zigzagging repetition, 6 required two repetitions, and 40 used all three repetitions. For the first type of test problem – the theorems, 21 of the 24 runs produced results that were confirmed by the ATP system. For the second type – the non-theorems, 17 of the 18 runs were confirmed, and for the third type – the unsatisfiable axioms, 18 of the 30 runs were confirmed. The OpenAI model produced the best results, with 32 of its 36 runs producing confirmed results, while Google achieved only 24 out of 36. The stochastic variations are interesting: Of the 24 problem-LLM sets of three runs, 17 produced the same correctly confirmed result in all three runs. Of those 17, HOWMANY produced the same logic in each of the three runs.

Future work includes adding access to more LLMs, e.g., Anthropic's `claude-sonnet-4-5`, testing over larger datasets, e.g., the FOLIO (Han et al. 2024) abd ProofWriter (Tafjord, Mishra, and Clark 2021) datasets. The main weakness of ShZZam (and speculatively all other natural language to logic translators) is its stochastic nature – the logic produced can vary between runs. This is a matter for further and deeper research.

References

Blanchette, J., and Paskevich, A. 2013. TFF1: The TPTP Typed First-order Form with Rank-1 Polymorphism. In Bonacina, M., ed., *Proceedings of the 24th International Conference on Automated Deduction*, number 7898 in Lecture Notes in Artificial Intelligence, 414–420. Springer-Verlag.

Blanchette, J.; Haslbeck, M.; Matichuk, D.; and Nipkow, T. 2015. Mining the Archive of Formal Proofs. In Kerber, M.; Carette, J.; Kaliszyk, C.; Rabe, F.; and Sorge, V., eds., *Proceedings of the 8th Conference on Intelligent Computer Mathematics*, number 9150 in Lecture Notes in Computer Science, 3–17. Springer-Verlag.

Bártek, F.; Bhayat, A.; Coutelier, R.; Hajdu, M.; Hetzenberger, M.; Hozzová, P.; Kovács, L.; Rath, J.; Rawson, M.; Reger, G.; Suda, M.; Schoisswohl, J.; and Voronkov, A. 2025. The Vampire Diary. In Piskac, R., and Rakamaric, Z., eds., *Proceedings of the 37th International Conference on Computer Aided Verification*, number 15933 in Lecture Notes in Computer Science, 57–71.

Cheng, F.; Li, H.; Liu, F.; Van Rooij, R.; Zhang, K.; and Lin, Z. 2025. Empowering LLMs with Logical Reasoning: A Comprehensive Survey. In Kwok, J., ed., *Proceedings of the 34th International Joint Conference on Artificial Intelligence*, 10400–10408. AAAI Press.

Han, S.; Schoelkopf, H.; Zhao, Y.; Qi, Z.; Riddell, M.; Zhou, W.; Coady, J.; Peng, D.; Qiao, Y.; Benson, L.; Sun, L.; Wardle-Solano, A. Szabó, H.; Zubova, E.; Burtell, M.; Fan, J.; Liu, Y.; Wong, B.; Sailor, M.; Ni, A.; Nan, L.; Kasai, J.; Yu, T.; Zhang, R.; Fabbri, A.; Kryscinski, W.; Yavuz, S.; Liu, Y.; Lin, X.; Joty, S.; Zhou, Y.; Xiong, C.; Ying, R.; Cohan, A.; and Radev, D. 2024. FOLIO: Natural Language Reasoning with First-Order Logic. In Al-Onaizan, Y.; Bansal, M.; and Chen, Y.-N., eds., *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 22017–22031. Association for Computational Linguistics.

Huang, L.; Yu, W.; Ma, W.; Zhong, W.; Feng, Z.; Wang, H.; Chen, Q.; Peng, W.; Feng, X.; Qin, B.; and Liu, T. 2025. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. *ACM Transactions on Information Systems* 42:1–55.

Li, Q.; Li, J.; Liu, T.; Zeng, Y.; Cheng, M.; Huang, W.; and Liu, Q. 2024. Leveraging LLMs for Hypothetical Deduction in Logical Inference: A Neuro-Symbolic Approach.

Niles, I., and Pease, A. 2001. Towards A Standard Upper Ontology. In Welty, C., and Smith, B., eds., *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems*, 2–9.

Olausson, T.; Gu, A.; Lipkin, B.; Zhang, C.; Solar-Lezama, A.; Tenenbaum, J.; and Levy, R. 2023. LINC: A Neurosymbolic Approach for Logical Reasoning by Combining Languag Models with First-Order Logic Provers. In Pino, J., and Bali, K., eds., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 5153–5176. Association for Computational Linguistics.

Peykani, P.; Ramezanlou, F.; Tanasescu, C.; and Ghanidel, S. 2025. Large Language Models: A Structured Taxonomy and Review of Challenges, Limitations, Solutions, and Future Directions. *Applied Sciences* 15(14):8103.

Sajjadi Mohammadabadi, S.; Kara, B.; Eyupoglu, C.; Uzay, C.; Tosun, M.; and Karakuş, O. 2025. A Survey of Large Language Models: Evolution, Architectures, Adaptation, Benchmarking, Applications, Challenges, and Societal Implications. *Electronics* 14(18):3580.

Sutcliffe, G.; Schulz, S.; Claessen, K.; and Baumgartner, P. 2012. The TPTP Typed First-order Form with Arithmetic. In Bjørner, N., and Voronkov, A., eds., *Proceedings of the 18th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning*, number 7180 in Lecture Notes in Artificial Intelligence, 406–419. Springer-Verlag.

Sutcliffe, G. 2000. SystemOnTPTP. In McAllester, D., ed., *Proceedings of the 17th International Conference on Automated Deduction*, number 1831 in Lecture Notes in Artificial Intelligence, 406–410. Springer-Verlag.

Sutcliffe, G. 2008. The SZS Ontologies for Automated Reasoning Software. In Sutcliffe, G.; Rudnicki, P.; Schmidt, R.; Konev, B.; and Schulz, S., eds., *Proceedings of the LPAR Workshops: Knowledge Exchange: Automated Provers and Proof Assistants, and the 7th International Workshop on the Implementation of Logics*, number 418 in CEUR Workshop Proceedings, 38–49.

Sutcliffe, G. 2017. The TPTP Problem Library and Associated Infrastructure. From CNF to TH0, TPTP v6.4.0. *Journal of Automated Reasoning* 59(4):483–502.

Sutcliffe, G. 2024. Stepping Stones in the TPTP World. In Benzmüller, C.; Heule, M.; and Schmidt, R., eds., *Proceedings of the 12th International Joint Conference on Automated Reasoning*, number 14739 in Lecture Notes in Artificial Intelligence, 30–50.

Tafjord, O.; Mishra, B.; and Clark, P. 2021. ProofWriter: Generating Implications, Proofs, and Abductive Statements over Natural Language. In Zong, C.; Xia, F.; Li, W.; and Navigli, R., eds., *Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing and Reasoning*, 3621–3634. Association for Computational Linguistics.

Thompson, R.; Pease, A. Toutsios, A.; Milanese, R.; and Singley, J. 2025. Formalizing Natural Language: Cultivating LLM Translations Using Automated Theorem Proving. In Komendantskaya, E.; Polgreen, E.; Saemann, C.; Stark, K.; and Rawson, M., eds., *Proceedings of the EuroProofNet-WG5 Workshop on Theorem Proving and Machine Learning in the Age of LLMs*.

Urban, J. 2003. Translating Mizar for First Order Theorem Provers. In Asperti, A.; Buchberger, B.; and Davenport, J., eds., *Proceedings of the 2nd International Conference on Mathematical Knowledge Management*, number 2594 in

Lecture Notes in Computer Science, 203–215. Springer-Verlag.

Yang, X.-W.; J-J., S.; Guo, L.-Z.; Zhang, B.-W.; Zhou, Z.; Jia, L.-H.; Dai, W.-Z.; and Li, Y.-F. 2025. Neuro-Symbolic Artificial Intelligence: Towards Improving the Reasoning Abilities of Large Language Models. In Kwok, J., ed., *Proceedings of the 34th International Joint Conference on Artificial Intelligence*, 10770–10778. AAAI Press.