Towards Open Domain English to Logic Translation

Adam Pease[1,2] and Josef Urban[2]

[1] Parallax Research, Beavercreek, OH, USA
adam.pease@parallaxresearch.com
[2] CIIRC, Czech Technical University, Prague, Czechia
jurban@cvut.cz

1 Introduction

Efforts in statistical machine processing of natural language text have made great advances in recent years. Logic-based system on the other hand have been less popular but have high degrees of trustworthiness due to their ability to document their reasoning and sources. It is possible that logic-based systems may be used in the future to provide plausible explanations of answers provided by machine learning systems or to test their outputs against known facts. There have been several barriers to language to logic translation. Often very simple resulting logics are used, limiting the generality and power of the portion of natural language semantics that can be captured[12]. Approaches that have used linguistic elements as though they were logical terms suffer from the absence of background knowledge that anchors the meaning of those terms and ensures that machine inference conforms to human understanding of linguistically-expressed concepts. Rule-based parsing remains challenging due to the complexity of language, and therefore the difficulty of scaling up the manual creation of language-to-logic interpretation rules. On the other hand, approaches to training a machine learning based language to logic system have been hampered by the challenge of creating training pairs of language and their equivalent logical translations.

Our approach attempts to address these issues by using an expressive higher order logic and a very large theory of world knowledge with a comprehensive mapping to linguistic tokens. The focus of this short paper is on the generation of training data of language and logic pairs.

Previous work in *auto-formalization* of mathematics has shown how it is possible to take a comprehensive set of informal descriptions and turn them into fully formal logic expressions [17]. Those efforts are part of the inspiration for our current work, and also inform the machine learning model that we use.

We utilize the Suggested Upper Merged Ontology (SUMO)[7, 10][1], a comprehensive ontology of around 20,000 concepts and 80,000 hand-authored logical statements in a higher-order logic[3] called SUO-KIF[9], that has an associated integrated development environment[14] integrated with leading theorem provers such as Eprover [15] Vampire [6] and LEO-III [1], and manually-created links[8] to all word senses in the WordNet lexico-semantic database[5]. We have described [14] elsewhere how we translate SUMO to the strictly first order language of TPTP [16], as well as TF0/TFA [11, 13] and TH0/THF[2].

2 Synthetic Corpus

We create a simple frame structure of linguistic elements that can be turned into a sentence and a logical expression. We started with a simple subject-verb-object structure that corresponds to the most common English sentences, and then added extra features incrementally. SUMO

[1] https://www.ontologyportal.org

has such a large set of concepts and their corresponding linguistic equivalents, we can generate millions of sentences even for some of the simplest variations. Thanks to SUMO's collection of higher-order relationships we can include statements of authorship, belief, normative force and many other constructs that have been conspicuously absent in prior efforts at language to logic translation.

Our conceptual and lexical library allows us to generate

- 1323 Process types - roughly equivalent to verbs, describing types of actions
- 67 CaseRole(s) - that describe the roles that entities play in processes
- 930 Object types - that can be subjects, direct objects or indirect objects
- 323 Social Roles - that refer to people by their professions or other social characteristics
- 100 names of people - biased heavily to Western names

Linguistic features are generated according to probabilities that favor more common constructs. These include:

- "You understood" forms - imperatives, with or without politeness phrases
- epistemics such as "believes" and "knows"
- modals such as "possibility" and "necessity"
- normative force such as "obligation", "permission" and "prohibition"
- numbers and units, quantities, qualifiers including non-numeric forms like "some"
- calendar days and clock times and past, present, future and progressive tenses
- negation
- desires
- authorship statements such as "said", "wrote", "quoted" or "unquoted'

Some sample outputs are

- *On Sat, 23 Dec 2023 at 5AM the cardinal infected Mariam.*
- *You should locate the waiter.*
- *The linguist is reading about eight novels.*
- *A reptile will not choke the professor.*
- *Kenneth doesn't say "The major was smelling the dancer."*

The last sentence has the formalization in the SUO-KIF language:

```
(not
  (exists (?HA)
    (and
      (instance ?HA Human)
      (names "Kenneth" ?HA)
      (says ?HA
        (exists (?H ?P ?DO ?IO)
          (and
            (attribute ?H Major)
            (instance ?P Smelling)
            (experiencer ?P ?H)
            (instance ?DO Dancer)
            (patient ?P ?DO)))))))
```

Just for subject-verb-object-indirectObject sentences we can theoretically generate 200 trillion combinations, and that does not include most of the additional linguistic features we can generate as listed above. However, not all combinations make sense. While SUMO has logical definitions that could restrict many such spurious combinations (for example, the "John" can't be Eating a Table) it is impractical in terms of the time required to run theorem proving to test all combinations. So we use SUMO's relation capability which relates types of processes to the types of things that can play specific roles in those processes. We also added the relation requiredRole to express combinations that make sense and exclude others. Each of these relations are defined axiomatically so they can also support theorem proving, but are in a standard form that is read into a table that can be checked very quickly during sentence generation. Reviewing generated sentences for bad combinations has been an important part of this work and creates a useful byproduct - preventing nonsensical sentences from being generated requires an understanding of why these combinations do not accord with common sense, thereby adding more knowledge to SUMO about how the world does or does not work. Finally, the generation of a certain percentage of nonsense sentences has an impact only on the efficiency of the data set, rather than the resulting correctness of the trained system. It simply allows the neural network to learn plausible logical equivalents for nonsense sentences. As long as those examples are not so prevalent as to dominate training time, there is no impact.

3 Training

We are using Google's Neural Machine Translation system[2] to train our translator on 10M synthetically generated language-logic pairs. In 4000 epochs we achieve a perplexity of 1.02[3]. We use a train/dev/test split of 80/10/10. We have created a simple evaluation function as part of the Sigma system that tests for the correct usage of SUO-KIF syntax and SUMO types and have used this to test the system on sentences from the Corpus of Contemporary American English (COCA)[4], a balanced, billion word corpus. That has helped us discover two area for improvement: (1) the need to include adjectives and adverbs, which present some interesting challenges; and (2) the need to simplify sentences. To address the latter issue, we have been using large language models and prompts that have had some success splitting complex sentences into simpler ones that are more easily interpreted by our trained model.

4 Conclusion and Future Work

We have shown that we can generate and train a transformer on a large corpus of language-logic pairs and learn them with high precision. However, some generated sentences are non-sensical, and many kinds of sentences that are representative of English constructions are not generated. Chiefly among them are sentences involving adjective and adverbs. Nouns and verbs have definitions that are usually independent of their context. A "table" or "lecturing" can be defined as long as we determine the word sense intended. SUMO has relied on concrete measurements rather than relative statements with respect to context so its treatment of adjectives and adverbs has been limited. The word "tall" only has meaning as a relationship between an individual and an explicit or implicit context to which the individual may be compared, but ">6ft" is decontextualized. These words are ubiquitous in humans' use of language but their interpretation from language to logic is more complex than other lexicalized tokens. Augmenting our synthetic corpus with these words, while employing a rigourous semantic interpretation of them is a focus of our next phase of effort.

[2] https://github.com/tensorflow/nmt

[3] corpus generation code is available at https://github.com/ontologyportal/sigmanlp in class com.articulate.nlp.GenSimpTestData which has a help screen. The training code is available at https://github.com/JUrban/sumonlp

Acknowledgement: The authors are grateful to Jan Hůla for his efforts to test on large language models.

References

[1] Christoph Benzmüller, Laurence Paulson, Frank Theiss, and A. Fietzke. (2008). *LEO-II - A Cooperative Automatic Theorem Prover for Higher-Order Logic. In Proceedings of the Fourth International Joint Conference on Automated Reasoning (IJCAR'08), LNAI volume*, 5195:162–170, 2008.

[2] Christoph Benzmüller and Adam Pease. Progress in automating higher-order ontology reasoning. In Boris Konev, Renate Schmidt, and Stephan Schulz, editors, *Workshop on Practical Aspects of Automated Reasoning (PAAR-2010)*. CEUR Workshop Proceedings, Edinburgh, UK, 2010.

[3] Chad E. Brown, Adam Pease, and Josef Urban. Translating SUMO-K to Higher-Order Set Theory. In *Frontiers of Combining Systems (FroCoS), to appear*, 2023.

[4] Mark Davies. The Corpus of Contemporary American English as the first reliable monitor corpus of English. *Literary and linguistic computing*, 25(4):447–464, 2010.

[5] Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[6] Laura Kovács and Andrei Voronkov. First-order theorem proving and vampire. In *Proceedings of the 25th International Conference on Computer Aided Verification*, volume 8044 of *CAV 2013*, pages 1–35, New York, NY, USA, 2013. Springer-Verlag New York, Inc.

[7] Ian Niles and Adam Pease. Toward a Standard Upper Ontology. In Chris Welty and Barry Smith, editors, *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001)*, pages 2–9, 2001.

[8] Ian Niles and Adam Pease. Linking Lexicons and Ontologies: Mapping WordNet to the Suggested Upper Merged Ontology. In *Proceedings of the IEEE International Conference on Information and Knowledge Engineering*, pages 412–416, 2003.

[9] Adam Pease. SUO-KIF Reference Manual. https://github.com/ontologyportal/sigmakee/blob/master/suo-kif.pdf, 2009. retrieved 20 June 2020.

[10] Adam Pease. *Ontology: A Practical Guide*. Articulate Software Press, Angwin, CA, 2011.

[11] Adam Pease. Arithmetic and Inference in a Large Theory. In *AI in Theorem Proving*, 2019.

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

[13] Adam Pease. Converting the Suggested Upper Merged Ontology to Typed First-order Form. arXiv:2303.04148 [cs.AI], 2023.

[14] Adam Pease and Stephan Schulz. Knowledge Engineering for Large Ontologies with Sigma KEE 3.0. In *The International Joint Conference on Automated Reasoning*, 2014.

[15] S. Schulz. E – A Brainiac Theorem Prover. *Journal of AI Communications*, 15(2/3):111–126, 2002.

[16] Steven Trac, Geoff Sutcliffe, and Adam Pease. Integration of the TPTPWorld into SigmaKEE. In *Proceedings of IJCAR '08 Workshop on Practical Aspects of Automated Reasoning (PAAR-2008)*. CEUR Workshop Proceedings, 2008.

[17] Qingxiang Wang, Chad E. Brown, Cezary Kaliszyk, and Josef Urban. Exploration of Neural Machine Translation in Autoformalization of Mathematics in Mizar. In *CPP*, pages 85–98. ACM, 2020.