A Sequence Labeling Method for First-order Logic Representation

Hanxiu Chen[1, *]

[1]vivo AI Lab, Shenzhen, Guangdong Province, China

*Corresponding author: hanxiu.chen@vivo.com

Abstract. Extracting logic knowledge is an important part of natural language understanding. This study introduces a new dataset for the purpose of measuring the model's ability to transfer natural language to logic form. Since logic represents how various concepts infer and reason from/to each other, it is essential to learn logical structure in any form of language while trying to comprehend its meaning of it. There have recently been many interests in deploying machine learning and deep learning methods for extracting logical knowledge and representing logic from natural language. The promising generative model that applied to recent sequence labeling work is used in the task of representing logic. Experiments have shown that generative methods can extract logic form from natural language texts. To be more specific, a sequence-to-sequence model is quite a convenient option when it deals with extraction tasks. The model provided in this study would be inspiring for future studies to work out more efficient solutions.

Keywords: Sequence-to-sequence model; Logic representation; Generative Model; Natural Language Processing.

1. Introduction

Logic representation is the task of transferring normal texts to a well-structured logical form, which is a sub-task in the field of semantic parsing [1, 2]. Since logic represents how various concepts infer and reason from/to each other, it is essential to learn logical structure in any form of language while trying to comprehend its meaning of it. There have recently been many interests in deploying machine learning and deep learning methods for extracting logical knowledge and representing logic from natural language, the main progress presented in Section 3 [3]. Many previous systems rely on encoder-decoder architecture to bridge natural language and logical forms [4]. In this study, a more complicated generative sequence labeling structure is used to derive a better performance on the task.

Sequence labeling tasks such as named entity recognition and relation extraction infer entity spans and types in texts and extract the relations between them [5, 6]. This study adapts generative methods which have been successfully applied to sequence labeling to the task of logic representation [7]. Different from previous works, the task of logic representation is converted to the task of extracting logical relations among recognized logical entities. The study also introduces a new dataset TREF that can be applied to measure the performance of logic representation methods.

Results of experiments demonstrate that the generative methods are competitive for the task of extracting logic form from texts. And since there are few corpus and benchmarks for the task, this study also contributes a new benchmark that is available for further research.

2. Logic Representation

2.1 The First-order Logic Variant

The logic presented in this paper is a variant of the first-order logic with:

An infinite set of *variables* is often denoted with lowercase letters such as $(g_1, g_2, g_3, g_4, \dots, g_n)$.

Predicates with *n*-valence $(n \geq 0)$ representing relations between n variables or functions, denoted with uppercase letters like $P(x)$ and $Q(x, y)$.

Functions with *n*-valence$(n \in \{0,1,2\})$ denoted with function symbols $f(x_1, x_2)$. Particularly, nullary functions are regarded as *constants.*

Logical connectives include unary connective negation $\neg$ and binary connectives conjunction $\wedge$, disjunction $\vee$, conditional $\rightarrow$, and biconditional $\leftrightarrow$.

Quantifiers include universal quantification $\forall$ and existential quantification $\exists$.

The equality symbol $=$ represents the equality between two elements.

2.2 Representing Logic from Natural Language

The procedure of representing logical expressions from natural language through logical symbols and variables is symbolization [8].

The concept of symbolization lies back in Frege who first introduced quantification, and Peano interpreted logic as propositions and classes composed under certain principles [9, 10]. And the concept of symbolic logic is first represented in Principia Mathematica, where Russell introduced a set of axioms [11, 12]. Later Church clarified and assigned all variables to an r-type, which constrained strictly on the types of variables [13]. On the contrary, Quine gave an objection to this denotation that he argued that the clarification of r-types relies on the definition of the variables, but may also bounded to thr higher-order variables, which indicates that there is an unmeasurable gap between the semantic interpretation of the notion of propositional function and the natural language expressions of the variables involved in their definition [14].

It is reasonable to symbolize:

I went to the garden as Harry did yesterday. (1)

as

$$G(i, g_1) \wedge G(h, g_2) \wedge (g_1 = g_2) \quad (2)$$

However, sentence (1) is apparently not aligned to (2) as there are different quantities of variables. Looking into the process, the sentence is most likely converted to

I went to the garden today and Harry went to the garden yesterday and the garden I went to is identical to the garden that Harry went to. (3)

first where extra modules such as names (I, Harry) and actions (went to, are identical to) are added to make the expression complete. Then, all the logic modules are projected to logic variables thus making up (2).

2.3 Logic and Prior Knowledge

The example in 2.2 has thrown two potential problems: How to clarify the semantic consistency e.g., the meaning of "as Harry did" is identical to" and Harry went to the garden yesterday…"; How to pick the most proper projection e.g. project "went to" as a relational predicate between names "I" and "garden" rather than a function "went to the garden". The solution leads to prior knowledge. Unlike the concept of action prediction, the prior knowledge in this study is seen as empirical linguistic results. The language models have shown that the meaning of words can be defined by their surroundings, and so is logic. Suppose there is a large corpus that can contain all the text language expressions, and find the expression "as someone did" means repeating the action that is already mentioned in most cases, then it shall be confident about the consistency between "as Harry did" and" …and Harry went to the garden yesterday…" in (1) and (3). The concept of prior knowledge may be defined as frequency and sensitivity but generally a consequence of them, as it also contains lexical and grammatical restrictions [6, 8].

Therefore, the flow of symbolization can be interpreted as compositions of logic modules, and on modulating logic expressions, the symbolization of natural language turns out to be a probabilistic problem.

3. Related Works

The problem is closely related to the research of semantic parsing and the model presented in this paper is of generative sequence-to-sequence architecture with neural networks.

The problem of learning logical information and parsing it can be dated back to Woods [15]. Researchers have come up with various modeling strategies. Researches include the use of parsing models, inductive logic reasoning, and grammar inspection [16-18]. These strategies are lexicon-based, manual-linguistic based, or domain-specific. Universal logic representations are proposed recently in tasks of building knowledge graphs, promoting question-answering systems, and natural language inference [19, 20].

This study gets inspiration from reasoning logic in knowledge graph and builds a sequence labeling model to present the logic expression. The sequence-to-sequence neural network framework this study adopts has been succeeded in other NLP fields such as semantic parsing, machine translation, and question answering [21]. Moreover, generative approaches based on a sequence-to-sequence framework have been proven to progress in most tasks above [22]. Our model works on generative sequence-to-sequence methods on logic representations. The main difference between this study to previous ones is that this study focuses on relating logic symbolization with sequence labeling, and this study shows it is competitive to consider symbolizing logic as entity and relation prediction.

4. The Method

4.1 Problem Objective

The research aims to modulize the prior knowledge, which maps natural language sequence $t = x_1 \dots x_n$, $n = |t|$ first to logically complete sequence $l = y_1 \dots y_m$, $m = |l|$ then to a symbolic representation $a = z_1 \dots z_k$, $k = |a|$. The conditional probability $p(t)$ is:

$$p(t) = \prod_{i=1}^{k} p(z_1 \dots z_{i-1}, l) \cdot \prod_{j=1}^{m} p\left(y_1 \dots y_{j-1}, t\right) \quad (4)$$

The method includes a sequence-to-sequence model which encodes input t to vector space and decides to generate rewrite sequence l conditioned on the encoding vector, as well as a projection model which converts rewrite sequence l to symbolic representation $z_1 \dots z_k$ in a graph structure.

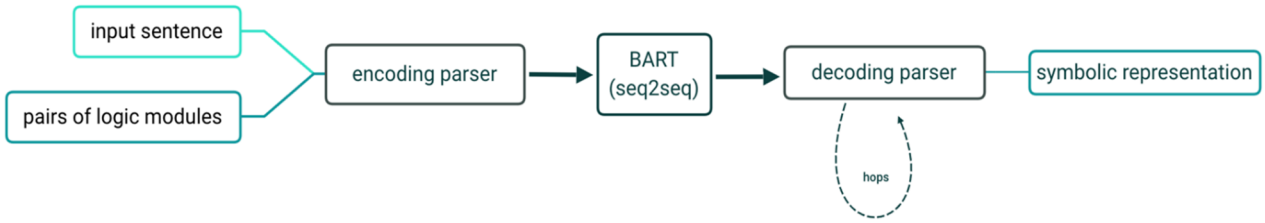


Fig. 1 Overview of the model

4.2 A Generative Method

Considering the task as a sequence labeling task, the symbolization task is treated as a combination of entity extraction and relation extraction. Given a sentence, *logic modules* are detected as entities in pairs. Relationships between the pairs of entities are then identified to show whether it is a unary function (predicate), a binary function, a logic connective, or an equation.

This study uses the generative model in Ni et al. for relation extraction and classification tasks. In Figure 1, the general view of the model is shown. An input sequence is encoded to a source sequence with its pairs of logic modules via an encoding parser. Then, the source sequence is passed to a sequence-to-sequence model to produce a target sequence containing relations and their types. Finally, a target parser is used to formulate a standard symbolic representation based on the entity-relation triples.

4.2.1 Entity-Relation Triples

In the task of relation extraction, an entity-relation triple is in the order of "subject, relation, object". Under the logic symbolization task, this study modified the triple to get a better fit. This study demonstrates the details of the modifications in this and the next section.

Let sentence $t = x_1 \dots x_n$ with n tokens. Let $s = x_{s1} \dots x_{s2}$ and $o = x_{o1} \dots x_{o2}$ be a pair of variables in the sentence, where s is the main variable and o is the sub-variable which can be set empty. To define the margin, this study use entity begins (<st>) and end (</st>) markers to encode the location information of the variables. As there are no entity types, this study removes the type markers but preserves the direction subsequence to highlight the main variable.

Let $R = \{ (rn_1, rt_1) \dots (rn_K, rt_K) \}$be the predefined relation sets. Each element of the set consists of the relation name rn and the relation type rk where rk is in $\{ c, n, r, i, p, u, q \}$ denoting conjunction, negation, binary-function, conditional, unary-function, disjunction, and equation respectively. The relation name of conjunction, negation, conditional, disjunction, and equation is set to null. Therefore, the encoding of the source sequence is concatenated with 1) marked sentence tokens, 2) direction sub-sequence and 3) relation pairs.

As for the target sequence, this study uses a similar triple structure with the order "main variable, sub-variable, relation" :

$$tar = [s \mid o \mid r(s, o)] \quad (5)$$

4.2.2 Dealing with logic hops

Most sentences are composed of logic hops, which indicates that the symbolization is more complicated than a simple (variable, variable, relation) triple but a backbone triple (main logic module, sub-module, relation) where each logic module is represented by either a single variable or a sub-triple of logic. Li used a sequence-to-tree method to recognize the hops, and this study deployed a similar recurrence method to denote the structure [23]. Instead of the variable tokens shown in 4.2.1, all the entities are split into two entity clusters to form the backbone triple. For each cluster, the entities are split recursively to form sub-triples until the size of the cluster is 1. The entity begin and end markers are also labeled with rank to clarify the hop e.g. <$sh1$> and </$sh1$> mark the start and the end location of the backbone hop.

4.2.3 Sequence-to-Sequence model

This study applies a pretrained sequence-to-sequence model BART to bridge between the source and the target sequence [24]. The BART model is based on the Transformer architecture composed of a multi-bidirectional encoder and a multi-autoregressive decoder. The model has reached state-of-the-art performance on relation extraction tasks. This study modifies the training data as described in the following section and fine-tunes the pre-trained model for logic form generation.

5. Experiments

5.1 Datasets

TREF is a sequence labeling corpus that contains 1000 queries to a database of Australian geography and traveling feedback [25]. TREF has 1000 instances that are unsplit. This study modifies and uses 343 instances of the corpus, with 321 instances as training examples and 22 as test sets.

5.2 Experiment Setup

This study builds and deploys the model based on the GREC on top of the BART model, with Hugging Face's PyTorch. Regarding the size of the training data, this study chose the base model. The optimizer of the model is Adam with a learning rate of $1e-5$. The experiment is conducted on a single V100 GPU with 10 epochs.

5.3 Results

This section firstly demonstrates the general output performance of the model on TREF, then discusses detailed performance on different logic hops. Table 1 presents the performance of full

models based on two pre-trained models with different amounts of parameters and one variant, which flattened all the entity-type labels to a concatenation of logic types and entity names.

Table 1. Scores on the dev set

Model	P	R	F1
Bart-base	62.3	68.1	65.5
Bart-base with flattened r-type	59.8	67.2	63.6
Bart-large	63.1	68.8	66.2

Overall, the model that is trained on Bart-large is superior to the one trained on Bart-base. This is to be expected since Bart-large has more parameters and thus catches better features. The difference is smaller than expected, however, which may be accounted for by the small size of the corpus, making it difficult for large pre-trained models to catch more lexical features. The model trained on Bart-base with flattened r-type fails to outperform the one trained on Bart-base with a 1.9% lower score. The reason lies in the fact that flattening type labels results in the reduction of samples in each label. To make the model learn sufficient knowledge, it is better to reduce the labels and supplement with a similarity search.

Table 2. Performance of the model with different hops

hops h	P	R	F1
h <=2	84.3	87.8	85.2
2 <= h <= 4	62.1	63.3	62.9
h > 4	43.0	55.4	49.7

For different hops, we analyze the result of the model trained on Bart-base. Table 2 shows that the performance of the model falls rapidly while logic hops are added. The result corresponds to the experiments by Li that indicates sentences with longer lengths are more difficult to get correct logic forms [25]. The decline is caused by the sophistication of logic structure with large hops, which raises the difficulty in clarifying the span margins of different logic modules [26]. The decline may also be caused by the ambiguity between logic hops since first-order logic is unable to cover all logic structures in natural language [27].

6. Conclusion

This study adapts generative methods which have been successfully applied to sequence labeling to the task of logic representation. Different from previous works, the task of logic representation is converted to the task of extracting logical relations among recognized logical entities. The study also introduces a new dataset TREF that can be applied to measure the performance of logic representation methods. This paper presents a generative sequence-to-sequence model for transferring natural language to logic forms. The study shows it is competitive to consider symbolizing logic as sequence labeling tasks such as entity and relation extraction. Further work will concentrate on larger domains and corpora to continually experiment with the performance of generative methods in representing logic.

Acknowledgments

This study is under the guidance of Owen Griffiths. The device used to run the experiments is supported by vivo AI Lab.

References

[1] Moore R C. Logic and representation. Center for the Study of Language (CSLI), 1995.

[2] Baral C, Gelfond M. Logic programming and knowledge representation. The Journal of Logic Programming, 1994, 19: 73-148.

[3] Gershman S J. The successor representation: its computational logic and neural substrates. Journal of Neuroscience, 2018, 38(33): 7193-7200.

[4] Thompson A, Mooney J, Tang R. Learning to parse natural language database queries into logical form. 1997.

[5] Settles B, Craven M. An analysis of active learning strategies for sequence labeling tasks. proceedings of EMNLP. 2008: 1070-1079.

[6] Nguyen N, Guo Y. Comparisons of sequence labeling algorithms and extensions. Proceedings of the 24th ICML. 2007: 681-688.

[7] Chowdhary K R. Natural language processing. Fundamentals of artificial intelligence, 2020: 603-649.

[8] Cohen S B, Smith N A. Variational inference for grammar induction with prior knowledge. Proceedings of the ACL-IJCNLP 2009 Conference Short Papers. 2009: 1-4.

[9] Ni J, Rossiello G, Gliozzo A, Florian R. A Generative Model for Relation Extraction and Classification. ArXiv, *abs/2202.13229*, 2022.

[10] Zalta, Edward N, Gottlob Frege. The Stanford Encyclopedia of Philosophy (Fall 2022 Edition), Edward N. Zalta (ed.), 2022.

[11] Quine O. Peano as logician, Hist. Philos, Logic, 1987, 8(1): 15-24.

[12] Russell B. Introduction to Mathematical Philosophy, London: George Allen & Unwin, 1919.

[13] Church A. Russellian Simple Type Theory, Proceedings and Addresses of the American Philosophical Association, 1974, 47: 21–33.

[14] Quine O. Word and Object, Cambridge: MIT Press, 1960.

[15] Woods W A. 1973. Progress in natural language understanding: An application to lunar geology. Proceedings of the National Computer Conference and Exposition. 1973: 441–450.

[16] Zhao K, Huang L. Type-driven incremental semantic parsing with polymorphism. Proceedings of the NAACL, 2015: 1416–1421.

[17] Thomspon C A, Mooney J. Acquiring word-meaning mappings for natural language interfaces. JAIR, 2003, 18:1–44.

[18] Abzianidze L, Kogkalidis K. A Logic-Based Framework for Natural Language Inference in Dutch. CLIN Journal, 2021, 11: 35-58.

[19] Zhu Z et al. Neural-Symbolic Models for Logical Queries on Knowledge Graphs. ICML 2022.

[20] Evans R et al. Can Neural Networks Understand Logical Entailment? ArXivabs/1802.08535, 2018.

[21] Raffel C et al. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer, JMLR, 2020, 21(140):1−67.

[22] Tom B B et al. Language models are few-shot learners. Proceedings of the 34th International Conference on Neural Information Processing Systems. Curran Associates Inc., 2020, 1877–1901.

[23] Li D, Lapata M. Language to Logical Form with Neural Attention. Proceedings of the 54th ACL, 2016: 33–43.

[24] Lewis M et al. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. Proceedings of the 58th ACL, 2020: 7871–7880.

[25] Trisedya B D, Weikum G, Qi J, Zhang R. Neural relation extraction for knowledge base enrichment. Proceedings of the 57th ACL, 2019: 229-240.

[26] Josifoski M, Nicola D C, Peyrard M, West R. GenIE: Generative Information Extraction. Proceedings of the NACCL: Human Language Technologies, 2022: 4626-4643.

[27] Wolf T et al. Transformers: State-of-the-Art Natural Language Processing. EMNLP, 2020.