

Neural sentence generation from formal semantics

Kana Manome¹

manome.kana@is.ocha.ac.jp

Masashi Yoshikawa²

yoshikawa.masashi
.yh8@is.naist.jp

Hitomi Yanaka³

hitomi.yanaka@riken.jp

Pascual Martínez-Gómez^{4*}

gomepasc@amazon.com

Koji Mineshima¹

mineshima.koji@ocha.ac.jp

Daisuke Bekki¹

bekki@is.ocha.ac.jp

¹Ochanomizu University ²Nara Institute of Science and Technology
³RIKEN ⁴Amazon

Abstract

Sequence-to-sequence models have shown strong performance in a wide range of NLP tasks, yet their applications to sentence generation from logical representations are underdeveloped. In this paper, we present a sequence-to-sequence model for generating sentences from logical meaning representations based on event semantics. We use a semantic parsing system based on Combinatory Categorical Grammar (CCG) to obtain data annotated with logical formulas. We augment our sequence-to-sequence model with masking for predicates to constrain output sentences. We also propose a novel evaluation method for generation using Recognizing Textual Entailment (RTE). Combining parsing and generation, we test whether or not the output sentence entails the original text and vice versa. Experiments showed that our model outperformed a baseline with respect to both BLEU scores and accuracies in RTE.

1 Introduction

In recent years, syntactic and semantic parsing has been developed and improved significantly. Syntactic parsing based on syntactic theories has been accomplishing reasonable accuracy to support various application tasks (Clark and Curran, 2007; Lewis and Steedman, 2014; Yoshikawa et al., 2017). Mapping sentences to logical formulas automatically has also been studied in depth, so there are semantic parsing systems that can produce high quality formulas (Bos, 2008, 2015; Martínez-Gómez et al., 2016).

One advantage of using logical formulas in semantic parsing is that they have expressive power that goes beyond simple representations such as predicate-argument structures. More specifically, logical formulas can capture aspects of sentence meanings that arise from complex syntactic structures such as coordination, functional words such as negation and quantifiers, and the scope of interactions between them (Steedman, 2000, 2012). In combination with the restricted use of higher-order logic (HOL) developed in formal semantics, those logical formulas have recently been used for RTE (Mineshima et al., 2015; Abzianidze, 2015) and Semantic Textual Similarity (STS) (Yanaka et al., 2017) and achieved high accuracy.

Compared with these recent developments in syntactic and semantic parsing, automatic generation of sentences from expressive logical formulas has received relatively less attention, despite a long and venerable tradition of work on surface realization, including those based on Minimal Recursion Semantics (MRS) (Carroll et al., 1999; Carroll and Oepen, 2005) and CCG (White, 2006; White and Rajkumar, 2009). If one can generate sentences from formulas, it would be possible to perform other NLP tasks in combination with RTE, including those challenging tasks such as paraphrase extraction (Levy et al., 2016) and sentence splitting and rephrasing (Narayan et al., 2017; Aharoni and Goldberg, 2018).

Meanwhile, sequence-to-sequence models showed high performance in machine translation and many other areas in NLP (Sutskever et al., 2014), yet their applications to sentence generation from logical meaning representations are still underdeveloped, mainly due to a lack of data and the structural complexity of meaning representations (Konstas et al., 2017). To address this challenge, we introduce a first sequence-to-sequence model for sentence generation from

*This work was done prior to joining Amazon.

logical formulas. We use the semantic parsing system `cg2lambda` (Martínez-Gómez et al., 2016)¹ to obtain data annotated with logical formulas including higher-order ones. Since the distinction between content words and function words plays an important role in parsing and generation, we augment the sequence-to-sequence model with masking for predicates, so that content words in input logical formulas occur in output sentences with a list of function words utilized in the parsing system.

We also propose a novel evaluation method for sentence generation. BLEU (Papineni et al., 2002) is widely used to evaluate the quality of decoded sentences, but it has difficulties in assessing fine-grained meaning relations between sentences. Instead, we use an RTE system for evaluation. We test whether or not the output sentence entails the original text and vice versa. This idea is motivated by the assumption that unlike surface-based methods such as BLEU, textual entailment is sensitive to syntactic and semantic aspects of sentences, thus making it possible to distinguish fine-grained meaning relations between original and output sentences. RTE has also been shown to be effective for evaluation of machine translation (Padó et al., 2009). Experiments show that our model outperforms a baseline with respect to both BLEU scores and accuracy in RTE.

2 Background and Related Work

2.1 Input logical formula

For input, we use logical formulas obtained from `cg2lambda` (Martínez-Gómez et al., 2016), a parsing and inference system that can be used for RTE. This system parses sentences into syntactic trees based on CCG (Steedman, 2000), a syntactic theory suitable for semantic composition from syntactic structures. The meaning of each word is specified using a lambda term. Logical formulas are obtained compositionally, by combining lambda terms in accordance with the meaning composition rules specified in the CCG tree and semantic templates. Semantic templates are defined manually based on formal semantics (Mineshima et al., 2015).

For logical formulas, we use standard Neo-Davidsonian event semantics (Parsons, 1990). For instance, the sentence *Eddy walked on the green grass* is represented as $\exists e.(\text{walk}(e) \wedge \text{subj}(e) =$

$\text{eddy} \wedge \exists x.(\text{green}(x) \wedge \text{grass}(x) \wedge \text{on}(e, x)))$. In this semantics, content words such as nouns and verbs are represented as predicates, and function words such as determiners, negation, and connectives are represented as logical operators with scope relations.

We decided not to include the following linguistic information in the input formulas: the definite–indefinite and singular–plural distinctions for NPs and tense and aspect for VPs. The intention is to normalize these semantic differences, so that the resulting formulas are easily usable in reasoning tasks based on RTE, where such fine-grained linguistic distinctions may sometimes make it more difficult to establish entailment relations between sentences. While more fine-grained linguistic information in logical formulas is readily obtainable by modifying semantic templates, we leave testing formulas with such additional semantic information for future work.

2.2 Related Work

A large amount of work has been done to convert meaning representations to their surface forms. In addition to those works mentioned in Section 1, there has been also a line of work on generating sentences from meaning representations used in semantic parsing systems (Wong and Mooney, 2007; Lu et al., 2009). Recently, Mei et al. (2016) has proposed an end-to-end neural sentence generation model from such meaning representations. These studies use datasets annotated with meaning representations, such as ROBOCUP (www.robocup.org) and GEOQUERY (Zelle and Mooney, 1996). However, these meaning representations are much simpler than logical formulas used in formal semantics in that they do not contain logical operators such as disjunction and quantifiers nor variable binding structures in standard first-order logic.

Recent rule-based approaches to generation using formal semantics and higher-order logic include a type-theoretic system based on Grammatical Framework (GF) (Ranta, 2011) and a system called Treebank Semantics based on event semantics (Butler, 2016).

Closest to our work is the one based on AMRs (Konstas et al., 2017), which has achieved high performance in sentence generation using neural networks from AMR-graphs (Banarescu et al., 2013). While AMR has been used as an

¹<https://github.com/mynlp/cg2lambda>

intermediate meaning representation for a wide range of tasks, it has less descriptive power than standard first-order logic (Bos, 2016). In addition, current AMRs do not support inference systems and thus cannot deal with logical inference as handled by RTE systems.

3 Method

We present a sequence-to-sequence model with attention for formula-to-sentence conversion.

3.1 Embedding

In using the sequence-to-sequence model, the point to address is how to linearize logical formulas. We test two ways of embedding: one is a token-based method where a formula is separated by each token (predicate and operator) and the other is based on graph representations converted from input logical formulas.

The token-based method tokenizes logical expressions. Below is an example of token-separated linearization:

[exists, e, (, walk, (, e,), &, Subj, (, e,), ...]

On the other hand, a graph representation reflects the structure of a logical formula. We use the formula-to-graph conversion presented in Wang et al. (2017). This method converts a formula to a tree structure and then obtains its graph representation by identifying the nodes for a same variable and replacing edges of the tree accordingly. See Wang et al. (2017) for the detail.

3.2 Sequence-to-Sequence with Attention

Our baseline model is a sequence-to-sequence with attention mechanism. Let $\mathbf{x} = (x_1, \dots, x_{|\mathbf{x}|})$ and $\mathbf{y} = (y_1, \dots, y_{|\mathbf{y}|})$ be an input formula and an output sentence, respectively. Then, the probability of the sentence \mathbf{y} given a formula \mathbf{x} is

$$P_{\Theta}(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{|\mathbf{y}|} P_{\Theta}(y_i|\mathbf{y}_{<i}, \mathbf{x}),$$

$$y_i|\mathbf{y}_{<i}, \mathbf{x} \sim \text{softmax}(\mathbf{f}(\Theta, \mathbf{y}_{<i}, \mathbf{x})), \quad (1)$$

where $\mathbf{y}_{<i}$ denotes the previously generated sequence of words at step i , \mathbf{x} is the input formula and Θ are the model parameters. The function \mathbf{f} is defined as

$$\mathbf{f}(\Theta, \mathbf{y}_{<i}, \mathbf{x}) = W_o \text{MLP} \left(\begin{bmatrix} \mathbf{g}_i \\ \mathbf{c}_i \end{bmatrix} \right) + \mathbf{b}_o,$$

$$\mathbf{g}_i = \text{LSTM}_{dec}(\mathbf{v}_T(y_{i-1}), \mathbf{g}_{i-1}),$$

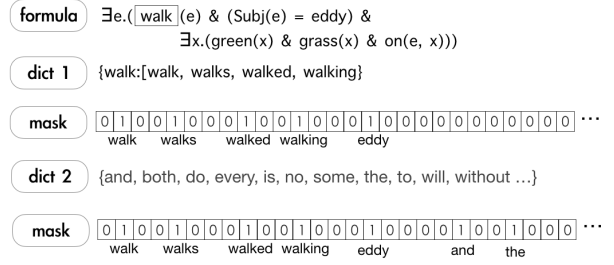


Figure 1: Masking

where MLP is a multi-layer perceptron with tanh activation, $\mathbf{v}_T(y)$ is an embedding vector of y , and \mathbf{c}_i is a weighted average (attention) $\sum_{j=1}^{|\mathbf{x}|} \alpha_{ij} \mathbf{h}_j$ of the hidden vectors \mathbf{h}_j for each x_j , where

$$\alpha_i = \text{softmax}(\mathbf{e}_i),$$

$$\mathbf{e}_j = \text{MLP}(\mathbf{g}_{i-1}, \mathbf{h}_j),$$

$$\mathbf{h}_j = \text{LSTM}_{enc}(\mathbf{v}_S(x_j), \mathbf{h}_{j-1}),$$

and LSTM_{enc} is an LSTM encoder, which calculates a hidden state \mathbf{h}_j using embedding vector $\mathbf{v}_S(x_j)$ and its previous hidden state.

We train the entire model by optimizing the log-likelihood with respect to the training data.

3.3 Masking

Logical formulas contain predicates for content words that should invariably appear in decoded sentences. For instance, in the sentence *Eddy walked on the green grass*, its content words are *Eddy*, *walked*, *green*, and *grass*, while *on* and *the* are function words. Using *ccg2lambda*, we obtain the following formula for this sentence:

exists e.(walk(e) & (Subj(e) = eddy) & exists x.(green(x) & grass(x) & on(e,x))).

To utilize the information available in a logical formula, we use a masking vector $\mathbf{m} \in \{0, 1\}^N$, where N is the size of the output vocabulary, which zeroes out the probabilities of words that do not appear in the formula (see Figure 1). Thus, instead of Eq. 1, we take the element-wise multiplication of the softmax probability and mask \mathbf{m} as

$$y_i|\mathbf{y}_{<i} \sim \mathbf{m} \otimes \text{softmax}(\mathbf{f}(\Theta, \mathbf{y}_{<i}, \mathbf{x})).$$

To construct the masking vector, we use a dictionary that maps a lemma to a list of its inflected forms, since logical formulas contain only lemmatized forms of words. The idea of using a masking vector can be seen as a simplified method of

employing a coverage vector, as has been widely used in a line of work on chart realization by Kay (1996). Our method provides a simple adaptation to sequence-to-sequence models. We obtained the dictionary by applying the lemmatizer implemented in C&C parser (Clark and Curran, 2007) to all training data used in the experiment.

In the previous example, there is a dictionary entry that maps *walk* to the list *walk*, *walks*, *walked* and *walking*. We set 1 in **m** at positions that correspond to these inflected forms (see dict1 in Figure 1). Additionally, we made functional words always available at decoding, by using a predefined list of those words (see dict2 in Figure 1).

4 Experiment

4.1 Dataset

We create a dataset annotated with logical formulas from the SNLI corpus (Bowman et al., 2015), a collection of 570,000 English sentence pairs manually labeled with an entailment relation. We use 50,000 hypothesis sentences from its training portion and split them into 42,000, 4,000, and 4,000 sentences for our training, development, and test sets, respectively. We map the sentences into logical formulas using *ccg2lambda*. We use C&C parser for converting tokenized sentences into CCG trees.

Table 1 shows the number of words in the constructed corpus (vocab), the max length (max-len) and average length (ave-len) of sequences obtained for the token-based (**token**) and graph-based methods (**graph**). Here **output** shows information on the output sentences.

As a baseline, we use Treebank Semantics (Butler, 2016)², a rule-based system for parsing and generation with logical formulas based on event semantics.

4.2 Evaluation

For evaluation, AMR generation tasks (Konstas et al., 2017) use BLEU, which does not directly consider the meaning and structure of a sentence. For instance, two sentences *No one visited the old man to greet him* and *Someone visited the old man to greet him* are similar but differ in meaning. To avoid this problem, we propose an evaluation method using parsing and RTE. Namely, we first

²<http://www.compling.jp/ajb129/generation.html>

	vocab	max-len	ave-len
token	6,822	419	44
graph	6,747	145	17
output	8,875	40	8

Table 1: Information about sequences.

	BLEU	$S_1 \Rightarrow S_2$	$S_2 \Rightarrow S_1$	$S_1 \Leftrightarrow S_2$
token	43.0	87.3	87.3	87.3
+mask	60.0	92.3	90.8	89.8
graph	42.2	86.3	90.0	86.3
+mask	50.0	92.5	92.3	90.8
rule	38.3	61.5	62.3	58.8

Table 2: Evaluation results.

parse an input sentence S_1 to obtain a formula P and then generate a sentence S_2 from the formula P . Finally, we check whether S_1 entails S_2 and vice versa. Our method based on RTE can detect differences in meaning in cases like the above.

We measure the accuracy of RTE for unidirectional and bidirectional entailments: $S_1 \Rightarrow S_2$, $S_2 \Rightarrow S_1$ and $S_1 \Leftrightarrow S_2$. We use *ccg2lambda* for parsing original and generated sentences and proving entailment relations between them. We use 400 pairs of sentences taken from the test set for RTE experiments. The inference system outputs *yes* (entailment), *no* (contradiction) or *unknown*. The gold answer is set to *yes*. The parsing and inference system of *ccg2lambda* achieved high precision in RTE tasks; Martínez-Gómez et al. (2017) reported that the precision was nearly 100% for the SICK dataset (Marelli et al., 2014). Thus, a predicted entailment (*yes*) judgement can serve as a reliable measure for evaluating the entailment relation between S_1 and S_2 .

4.3 Results

Table 2 shows BLEU scores and RTE accuracy. Here, **token** and **graph** show the results for a token-based model with attention and the graph-based model with attention, respectively, and **+mask** means the model with masking. The baseline is shown by **rule**, which is the performance of Treebank Semantics. As shown here, all the models outperformed the baseline with respect to both BLEU score and RTE accuracy. For the RTE accuracy, the increase in the score of the **graph + mask** model was slightly larger than the increase for the **token + mask** model.

Table 3 shows examples of decoded sentences

	Input sentence (S_1)	Decoded sentence (S_2)
(1)	the girls are swimming in the ocean.	the girls are swimming in the ocean.
(2)	a dog is playing fetch with his owner.	a dog is playing fetch with owner.
(3)	a man is sitting on the couch.	the men are sitting on a couch.
(4)	a tall man.	the man is tall.
(5)	a child is standing.	the children are standing together.
(6)	there are several people in this picture.	people are pictured in a picture.

Table 3: Examples of decoded sentences obtained from the **graph + mask** model

obtained from the **graph + mask** model. (1) and (2) are examples that preserve the form of input sentences. (3) is an example where singular form is changed to plural form, as well as articles *a* and *the*. This is because our semantics neutralizes these distinctions. The decoded sentence is grammatically correct, accommodating *be*-verbs. In (4), the input is a noun phrase, while the decoded result is a sentence. Example (5) contains an unnecessary word *together*, but the subject is also changed so that the decoded sentence is meaningful. In example (6), the *there*-construction in the input is removed while preserving the same content.

5 Conclusion

To our knowledge, this is the first study to describe a neural sentence generation model from logical formulas. We also proposed a new evaluation method based on RTE. In future work, we will refine our model for generation of longer sentences and test formulas with richer semantic information.

Acknowledgement We would like to thank the reviewers for their helpful comments and suggestions. We are also grateful to Ribeka Tanaka, Yurina Ito, and Yukiko Yana for helpful discussion and Fadoua Ghourabi for reading an earlier draft of the paper. This work was partially supported by JST AIP-PRISM Grant Number JPMJCR18Y1, Japan.

References

Lasha Abzianidze. 2015. A Tableau Prover for Natural Logic and Language. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2492–2502.

Roei Aharoni and Yoav Goldberg. 2018. Split and rephrase: Better evaluation and stronger baselines.

In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 719–724.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for Sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186.

Johan Bos. 2008. Wide-coverage semantic analysis with Boxer. In *Proceedings of the 2008 Conference on Semantics in Text Processing*, pages 277–286.

Johan Bos. 2015. Open-domain semantic parsing with Boxer. In *Proceedings of the 20th Nordic Conference of Computational Linguistics*, pages 301–304.

Johan Bos. 2016. Expressive power of abstract meaning representations. *Computational Linguistics*, 42(3):527–535.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642.

Alastair Butler. 2016. Deterministic natural language generation from meaning representations for machine translation. In *Proceedings of the 2nd Workshop on Semantics-Driven Machine Translation*, pages 1–9.

John Carroll, Ann Copestake, Dan Flickinger, and Victor Poznański. 1999. An efficient chart generator for (semi-) lexicalist grammars. In *Proceedings of the 7th European Workshop on Natural Language Generation*, pages 86–95.

John Carroll and Stefan Oepen. 2005. High efficiency realization for a wide-coverage unification grammar. In *Proceedings of International Joint Conference on Natural Language Processing*, pages 165–176.

Stephen Clark and James R. Curran. 2007. Wide-coverage efficient statistical parsing with CCG and log-linear models. *Computational Linguistics*, 33(4):493–552.

- Martin Kay. 1996. Chart generation. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pages 200–204.
- Ioannis Konstas, Srinivasan Iyer, Mark Yatskar, Yejin Choi, and Luke Zettlemoyer. 2017. Neural AMR: Sequence-to-sequence models for parsing and generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 146–157.
- Omer Levy, Ido Dagan, Gabriel Stanovsky, Judith Eckle-Kohler, and Iryna Gurevych. 2016. Modeling extractive sentence intersection via subtree entailment. In *Proceedings of the 26th International Conference on Computational Linguistics*, pages 2891–2901.
- Mike Lewis and Mark Steedman. 2014. A* CCG parsing with a supertag-factored model. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 990–1000.
- Wei Lu, Hwee Tou Ng, and Wee Sun Lee. 2009. Natural language generation with tree conditional random fields. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 400–409.
- Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of LREC2014*, pages 216–223.
- Pascual Martínez-Gómez, Koji Mineshima, Yusuke Miyao, and Daisuke Bekki. 2016. ccg2lambda: A Compositional Semantics System. In *Proceedings of ACL 2016 System Demonstrations*, pages 85–90.
- Pascual Martínez-Gómez, Koji Mineshima, Yusuke Miyao, and Daisuke Bekki. 2017. On-demand injection of lexical knowledge for recognising textual entailment. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 710–720.
- Hongyuan Mei, Mohit Bansal, and Matthew R. Walter. 2016. What to talk about and how? selective generation using LSTMs with coarse-to-fine alignment. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 720–730.
- Koji Mineshima, Pascual Martínez-Gómez, Yusuke Miyao, and Daisuke Bekki. 2015. Higher-order logical inference with compositional semantics. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2055–2061.
- Shashi Narayan, Claire Gardent, Shay Cohen, and Anastasia Shimorina. 2017. Split and rephrase. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 617–627.
- Sebastian Padó, Michel Galley, Dan Jurafsky, and Chris Manning. 2009. Robust machine translation evaluation with entailment features. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 297–305.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318.
- Terence Parsons. 1990. *Events in the semantics of English: a study in subatomic semantics*. The MIT Press.
- Aarne Ranta. 2011. *Grammatical framework: Programming with Multilingual Grammars*. CSLI Publications.
- Mark Steedman. 2000. *The Syntactic Process*. The MIT Press.
- Mark Steedman. 2012. *Taking scope: The natural semantics of quantifiers*. The MIT Press.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Advances in neural information processing systems*, pages 3104–3112.
- Mingzhe Wang, Yihe Tang, Jian Wang, and Jia Deng. 2017. Premise selection for theorem proving by deep graph embedding. In *Advances in Neural Information Processing Systems*, pages 2783–2793.
- Michael White. 2006. Efficient Realization of Coordinate Structures in Combinatory Categorical Grammar. *Research on Language & Computation*, 4(1):39–75.
- Michael White and Rajakrishnan Rajkumar. 2009. Perceptron reranking for CCG realization. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 410–419.
- Yuk Wah Wong and Raymond Mooney. 2007. Generation by inverting a semantic parser that uses statistical machine translation. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*, pages 172–179.
- Hitomi Yanaka, Koji Mineshima, Pascual Martínez-Gómez, and Daisuke Bekki. 2017. Determining Semantic Textual Similarity using Natural Deduction Proofs. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 692–702.
- Masashi Yoshikawa, Hiroshi Noji, and Yuji Matsumoto. 2017. A* CCG parsing with a supertag and dependency factored model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 277–287.

John M. Zelle and Raymond J. Mooney. 1996. Learning to parse database queries using inductive logic programming. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 1050–1055.