Toward Human-like Sentence Interpretation — a Syntactic Parser Implemented as a Restricted Quasi Bayesian Network —

Naoto Takahashi and Yuuji Ichisugi

Artificial Intelligence Research Center,
National Institute of Advanced Industrial Science and Technology (AIST)
Tsukuba, Japan
naoto.takahashi@aist.go.jp

Abstract. Most sentences expressed in a natural language are ambiguous. However, human beings effortlessly understand the intended message of the sentence even when a computer program finds out countless possible interpretations. If we want to create a computer program that understands a natural language in the same way as human beings do, a promising way would be implementing a human-like mechanism of sentence processing instead of implementing a "list exhaustively then select" method. By the way, it is highly likely that human's language ability is realized mostly by the cerebral cortex, and recent neuroscientific studies hypothesize that the cerebral cortex works as a Bayesian network. Then it should be possible to reproduce human's language ability using a Bayesian network. Based on this idea, we implemented a syntactic parser using a restricted quasi Bayesian network, which is a prototyping tool for creating models of cerebral cortical areas. The parser analyzes a sequence of syntactic categories based on a subset of combinatory categorial grammar. We confirmed that the parser correctly parsed grammatical sequences and rejected ungrammatical sequences.

Keywords: syntactic analysis, Bayesian networks, combinatory categorial grammar, language area

1 Introduction

Most sentences expressed in a natural language are lexically and syntactically ambiguous. This fact is easily demonstrated by parsing natural languages with computer programs; you will be surprised by seeing countless possible interpretations that you have never imagined. Therefore it is necessary to assess each interpretation based on a certain criterion to select the most probable one.

Nevertheless, human beings effortlessly understand the intended message of the sentence without being troubled by possible but unintended interpretations.

If we want to create a computer program that understands a natural language in the same way as human beings do, one of the most promising way

This is a pre-copyedited version of a contribution published in Samsonovich A. (eds) Biologically Inspired Cognitive Architectures 2018 (BICA 2018) published by Springer.
The definitive authenticated version is available online via https://doi.org/10.1007/978-3-319-99316-4_40

would be implementing a human-like mechanism of sentence processing instead of implementing a “list exhaustively then select” method.

By the way, medical case studies of aphasia and observations of brain activities measured by recent technologies strongly suggest that specific areas of cerebral cortex, so called language areas, play crucial roles in language processing [7]. At the same time, neuroscientific studies hypothesize that the cerebral cortex works as a Bayesian network [10] or a kind of probabilistic graphical model [1–6, 8, 9, 11–14]. If human’s language ability is realized by the cerebral cortex, and if the cerebral cortex works as a Bayesian network, then it should be possible to reproduce human’s language ability using a Bayesian network.

Based on this idea, the current authors formerly implemented a syntactic parser for a context free grammar using a restricted quasi Bayesian network [16]. We present another syntactic parser for a different type of grammar, i.e. combinatory categorial grammar, in this article.

2 Restricted Quasi Bayesian Networks

One way to study the mechanism of information processing in the brain is to create computational models of the targeted function. However, creating realistic models using machine learning techniques, e.g. Bayesian networks, forces the designers to resolve inessential problems, like tuning hyper parameters. Moreover, it is often difficult to trace the real cause of unsatisfying results when the created model does not behave as expected.

It is often helpful to create prototypes before creating a realistic model. By creating prototypes, we can estimate the hopefulness of the fundamental design of the realistic model that we are going to create.

Restricted quasi Bayesian network [16] is a prototyping tool for creating models of cerebral cortical areas. It is a simplified Bayesian network that only distinguishes probability value zero from other values. Its conditional probability tables are restricted to fulfill certain mathematical conditions to avoid combinatorial explosion.

Restricted quasi Bayesian network provides *gates*, which control the flow of information. Thus the designer can design generative models in a similar way as designing logical circuits.

Since restricted quasi Bayesian network does not have learning ability, conditional probability tables must be prepared by the designer. Because of its limited capabilities, restricted quasi Bayesian network may not be applied to practical, real-world problems. However, it releases the designers from inessential problems and allows them to concentrate on the essential part of model design. As a result of agile prototyping activities, designers would find potential problems in the model, which can be extremely difficult to find in a complicated, realistic model.

3 Combinatory Categorial Grammar

Combinatory categorial grammar [15] generalizes classical categorial grammar by introducing functional composition. It is suitable to describe syntactic rules of natural languages because its weak generative capacity locates between context-free grammar and context-sensitive grammar.

In traditional phrase structure grammar, each syntactic category is represented as a unique non-terminal symbol. For example, sentence, noun phrase and verb are often represented as S, NP and V, respectively.

In combinatory categorial grammar, on the other hand, syntactic categories are represented by *ground categories* and *operators*. When X and Y are syntactic categories, $X\backslash Y$ represents a syntactic category that constitutes X when preceded by Y. Likewise, X/Y represents a syntactic category that constitutes X when followed by Y.

In English, for example, a verb phrase is composed with a preceding noun phrase (NP) to constitute a sentence (S). Thus the category for verb phrase is represented as $S\backslash NP$, assuming that S and NP are ground categories. Furthermore, a transitive verb is composed with a succeeding noun phrase (NP) to constitute a verb phrase ($S\backslash NP$). Thus the category for transitive verbs is represented as $(S\backslash NP)/NP$.

4 Implementation

In this section, we explain a restricted quasi Bayesian network that implements the forward/backward functional application rules of combinatory categorial grammar.

4.1 Representation of Syntactic Categories in a Bayesian Network

Theoretically, the length of a syntactic category in combinatory categorial grammar is unlimited. However, we suppose it is limited because of the information processing ability of human. In this article, we represent a syntactic category by five nodes. Each node takes a ground category or an operator as its value.

We adopt prefix notation to eliminate parentheses. In a sequence of five nodes, first comes the operator, then the category that was originally at the right side, and finally the category that was at the left side. Five nodes are used in the flush-left mode; unused nodes take a special value to explicitly indicate its inactivity (Fig. 1).

4.2 Forward/Backward Functional Application Rules

Figure 2 shows a restricted quasi Bayesian network that implements the forward functional application rule. The value combinations that appear with a probability greater than zero are listed in Table 1.

Ordinary notation	Prefix notation				
X/Y	/	Y	X	.	.
$(X/Y)\backslash Z$	\	Z	/	Y	X
$X/(Y\backslash Z)$	/	\	Z	Y	X

Fig. 1. The ordinary notation of syntactic categories in combinatory categorial grammar (left) and their prefix notation in this article (right). A full stop (.) represents an unused node

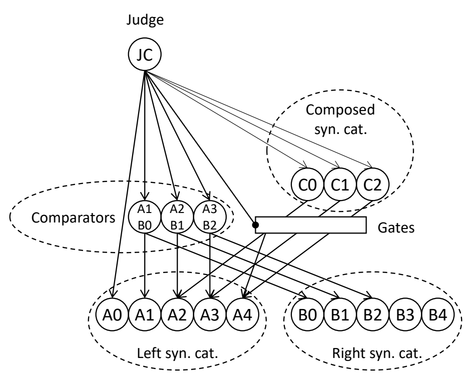

Fig. 2. A restricted quasi Bayesian network that implements the forward functional application rule. Each node in a syntactic category takes a ground category or an operator as its value. Each comparator node has two children. If both children have the same value, the parent comparator takes that value. Otherwise it takes a special value that represents "unmatched". The judge node controls the gates between the left syntactic category and the composed syntactic category based on the values of the comparators. Only the first three nodes of the composed syntactic category are depicted since the length of the syntactic category is always shorten when a function application rule is applied

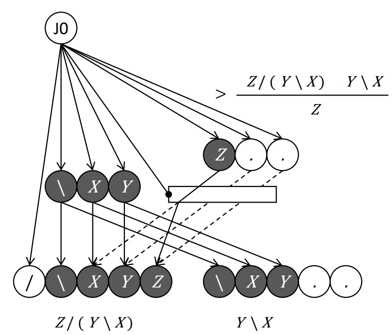

Fig. 3. An example of the forward functional application rule in which the first three nodes of the right syntactic category match a part of the left syntactic category. The gates operate so that the final node of the left syntactic category and the first node of the composed syntactic category have the same value

Table 1. The value combinations that have a probability greater than zero in Fig. 2. X is an arbitrary ground category and can be different from column to column. A full stop (.) means "unused" or "unmatched". An underscore (_) is an arbitrary value. The rightmost four columns indicate connection/disconnection between nodes controlled by the gates. The value of the judge node (JC) may be anything as long as each row can be distinguished

JC	A0	C0	C1	C2	A1B0	A2B1	A3B2	C0A2	C0A4	C1A3	C2A4
J0	$/$	X	.	.	$/, \backslash$	X	X	off	on	off	off
J1	$/$	$/, \backslash, X$	$X, .$	$X, .$	X	.	_	on	off	on	on

When the first node, namely the topmost operator, of the left syntactic category has the value slash (/) and the values of the following three nodes match the values of the first three nodes of the right syntactic category, the gate that connects the final node of the left syntactic category and the first node of the composed syntactic category opens to make their values equal (Fig. 3). At the same time, all the other nodes in the composed syntactic category are marked as "unused".

When the first node of the left syntactic category has the value slash (/) and only one succeeding node matches the leftmost part of the right syntactic category, the gates operate so that the remaining three nodes of the left syntactic category and the nodes of the composed syntactic category have the same values (Fig. 4).

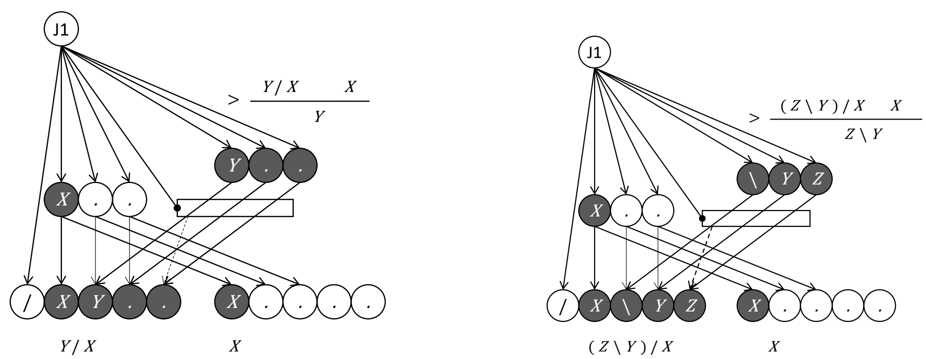

Fig. 4. Examples of the forward functional application rule in which only one node of the left syntactic category matches the leftmost part of the right syntactic category. The gates operate so that the last three nodes of the left syntactic category and the nodes of the composed syntactic category have the same values

The backward functional application rule is implemented similarly as the forward functional application rule (Fig. 5 and Table 2).

By combining the networks described in Figs. 2 and 5, we obtain a restricted quasi Bayesian network that performs the forward/backward function application rules between two syntactic categories.

Table 2. The value combinations that have a probability greater than zero in Fig. 5

JC	B0	C0	C1	C2	A0B1	A1B2	A2B3	C0B2	C0B4	C1B3	C2B4
J2	$\backslash$	X	.	.	$/, \backslash$	X	X	off	on	off	off
J3	$\backslash$	$/, \backslash, X$	$X, .$	$X, .$	X	.	_	on	off	on	on

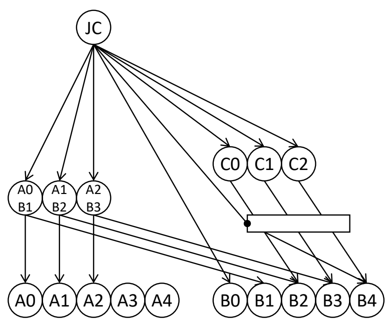

Fig. 5. A restricted quasi Bayesian network that implements the backward functional application rule

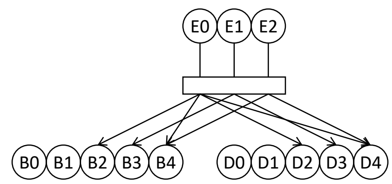

Fig. 6. The nodes for the third syntactic category (D0 to D4), the composed category (E0 to E2) and their connections for applying the functional application rules. The judge node and the comparator nodes are omitted

4.3 A Bayesian Network for Three Syntactic Categories

Now we explain how to construct a restricted quasi Bayesian network that applies the forward/backward functional application rules to three syntactic categories.

First, we introduce the third syntactic category for input, which is represented by the nodes D0 to D4. Then we connect the B nodes and the D nodes in the same way as we did for the A nodes and the B nodes. The composed syntactic category is represented by the nodes E0 to E2 (Fig. 6).

Then we connect the A nodes (the first input category) and the E nodes (the composition of the second and the third input categories), as well as the C nodes (the composition of the first and the second input categories) and the D nodes (the third input category). Both compositions are represented by the nodes F0 to F2 (Fig. 7).

We have implemented the above-mentioned network for three syntactic categories as a restricted quasi Bayesian network. We also confirmed that all the input combinations that are applicable to the functional application rules were correctly parsed, and inapplicable combinations were rejected, using up to three different ground categories.

Figure 8 is an example of parsing three syntactic categories. It shows how the created network parses a three-word sentence that consists of a subject noun phrase, a transitive verb and an object noun phrase. First, the transitive verb (*wrote*) and the object noun phrase (*plays*) compose a verb phrase ($S\backslash NP$). Next, the subject noun phrase (*Shakespeare*) and the composed verb phrase make a sentence (S).

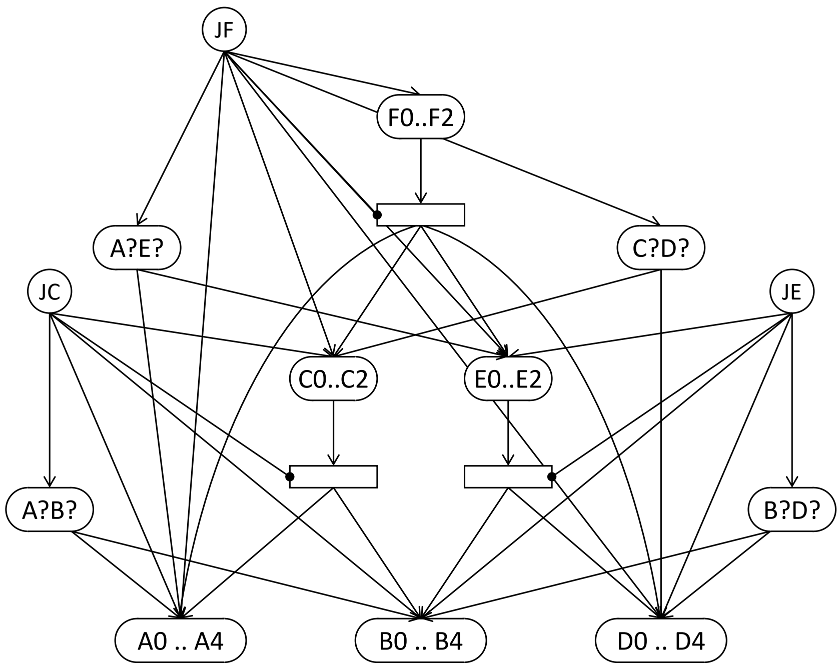

Fig. 7. A restricted quasi Bayesian network that parses three-word sentences. Multiple nodes that belong to the same group are depicted as a single oval. A0..A4, C0..C2, etc. are nodes to represent syntactic categories. A?B?, C?D?, etc. are comparators between two syntactic categories. JC, JE and JF are judge nodes. Rectangles are gates

5 Conclusion

We examined the feasibility of creating a syntactic parser for combinatory categorial grammar as a restricted quasi Bayesian network. So far we have only implemented the forward/backward functional application rules. The number of acceptable syntactic categories is also limited to a small number. To extend the current parser to a practical level, it is necessary to confirm that the presented

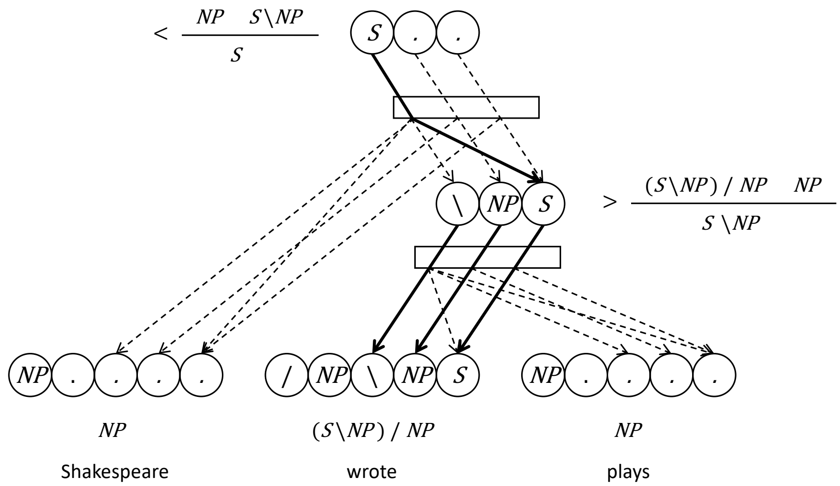

Fig. 8. A three-word sentence parsed by the restricted quasi Bayesian network. First, the transitive verb (*wrote*) and the object noun phrase (*plays*) compose a verb phrase ($S\backslash NP$). Next, the subject noun phrase (*Shakespeare*) and the composed verb phrase make a sentence (S). The judge nodes and the comparator nodes are not illustrated

design strategy does not cause combinatorial explosion with sufficient number of syntactic categories for input.

One of the advantages in using combinatory categorial grammar is that syntactical derivation and semantic composition can be associated elegantly; this property should be utilized in a practical parser.

Restricted quasi Bayesian networks perform exhaustive search to find all the combinations that have a probability greater than zero, but ordinary Bayesian networks can calculate the most probable combination quickly with approximation. It is possible that humans also use some kind of approximation to realize a real-time sentence interpretation because humans do not perform, at least consciously, exhaustive search, and they fail to interpret some types of grammatical sentences, e.g. deep centre-embedded sentences.

Our final goal is to reproduce human's language ability using an *ordinary* Bayesian network. For this purpose, we designed a prototype using a restricted quasi Bayesian network to see the feasibility of such networks. We plan to examine other implementations for comparison.

Acknowledgement

This paper is based on results obtained from a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

References

1. Chikkerur, S., Serre, T., Tan, C., Poggio, T.: What and where: A bayesian inference theory of attention. Vision Research **50**(22), 2233–2247 (2010)
2. Dura-Bernal, S., Wennekers, T., Denham, S.L.: Top-down feedback in an hmax-like cortical model of object perception based on hierarchical bayesian networks and belief propagation. PLoS ONE **7**(11) (2012)
3. George, D., Hawkins, J.: A hierarchical bayesian model of invariant pattern recognition in the visual cortex. In: 2005 International Joint Conference on Neural Networks (IJCNN) (2005)
4. Hosoya, H.: Multinominal bayesian learning for modeling classical and nonclassical receptive field properties. Neural Computation **24**(8) (2012)
5. Ichisugi, Y.: The cerebral cortex model that self-organizes conditional probability tables and executes belief propagation. In: 2007 International Joint Conference on Neural Networks (IJCNN) (2007)
6. Ichisugi, Y.: Recognition model of cerebral cortex based on approximate belief revision algorithm. In: 2011 International Joint Conference on Neural Networks (IJCNN) (2011)
7. Kemmerer, D.: Cognitive Neuroscience of Language. Psychology Press (2015)
8. Lee, T.S., Mumford, D.: Hierarchical bayesian inference in the visual cortex. J. Opt. Soc. Am. A **20**(7), 1434–1448 (2003)
9. Litvak, S., Ullman, S.: Cortical circuitry implementing graphical models. Neural Computation **21**(11), 3010–3056 (2009)
10. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1988)
11. Pitkow, X., Angelaki, D.E.: Inference in the brain: Statistics flowing in redundant population codes. Neuron **94**(5), 943–953 (2017)
12. Raju, R.V., Pitkow, X.: Inference by reparameterization in neural population codes. In: Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain, pp. 2029–2037 (2016)
13. Rao, R.P.: Bayesian inference and attention modulation in the visual cortex. Neuroreport **16**(16), 1843–1848 (2005)
14. Röhrbein, F., Eggert, J., Körner, E.: Bayesian columnar networks for grounded cognitive system. In: Proceedings of the 30th Annual Conference of the Cognitive Science Society, pp. 1423–1428 (2008)
15. Steedman, M.: The Syntactic Process. MIT Press, Cambridge, MA, USA (2000)
16. Takahashi, N., Ichisugi, Y.: Restricted quasi bayesian networks as a prototyping tool for computational models of individual cortical areas. In: Proceedings of Machine Learning Research, vol. 73. PMLR (2017)