Computational linguistic grammar theory and its application to artificial intelligence

Tomoki Yoshihashi[1]

[1]Affiliation not available

October 1, 2020

Abstract

The goal of this paper is to establish a new approach with graph networks, which we refer to as the cluster grammar, as a computational linguistic grammar. The main content of this paper is the construction of the theory and the verification of its potential for application in computer science.

We begin by introducing the graph networks, which we refer to as cluster graphs. The graphs are considered as the essential elements for the language ability. The innate abilities that we introduce in order to elucidate the acquisition of the language ability, generation and processing of sentences are given as graph theoretical algorithms. The theory includes a sort of verification problem as well as the Universal Grammar introduced by Chomsky, however the theory is intended to be applied for information technology where the given algorithms and methods can be developed further to have computers process languages with more developed data structure and more human.

Contents:

Introduction

Background and intention

This paper attempts to establish a new approach with graph networks, which we refer to as the cluster grammar, as a computational linguistic grammar. The main content of this paper is the construction of the theory and the verification of its potential for application to computer science.

We begin by introducing the graph networks. The graphs are considered as the essential elements for language abilities. The innate abilities which we introduce in order to elucidate the acquisition of the language ability, generation and processing of sentences are given as the graph theoretical algorithms. The theory includes a sort of verification problem as well as the Universal Grammar introduced by Chomsky, however the theory is

intended to be applied to information technology where the given algorithms and methods can be developed further to have computers process languages with more developed data structure.

This sort of graph networks has a similarity in the semantic network introduced by Richard Richens in 1956. The cluster grammar not only gives such a structure on morphemes, as well as a different structure on signifiés.

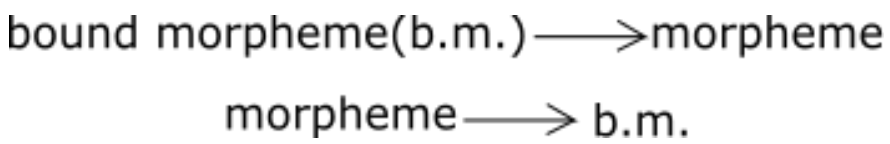
I Grammar structures

1.1 Overview of the grammar structure

The cluster grammar introduces graph networks, which we refer to as the cluster graphs, which have the directed graph structure instead of the trees used in classical grammar theory such as the Generative Grammar. The cluster graphs consist of two graphs: the morpheme graph which consists of the whole morphemes of a language and the signifié graph which consists of the signifiés that correspond to the morphemes.

The morpheme graph is a directed graph which consists of the morphemes as its nodes and the edges connecting them. The signifié graph has signifiés which correspond to the morphemes which are the nodes of the morpheme graph. The correspondence between these two graphs is not the signification as the morpheme graph only deals with morphemes, not signifiants. Morphemes that correspond to several different signifiés are distinguished from each other. Because signifiants are only signs and the structure of the graph is deeply concerned with the sentence generation process, the nodes of such a graph should be morphemes.

The directed edges of the morpheme graph are determined by the grammatic order of the morphemes, and vice versa. The following graph a and graph b are the examples of the morpheme graph. The arrows show the order of the enumeration of the morphemes. (q.v. 1.2)



Graph a

Graph b

1.2 Fundamental processes of sentence generation

Sentence generation is performed by enumeration of the morphemes according to the directed edges. The processing can be executed not only in order of the directed edges, but also by the morpheme alternation under a certain rule in order to proceed the enumeration with morphemes that have no edges directed from the last morpheme (q.v. 1.3).

The sentence generation requires an ability to select signifiés which construct the meaning of the sentence. This ability and the process, which we refer to as the signifiés selection, deeply concerns with one's cognition. Videlicet, the selection of the signifiés that describe a phenomenon depends on the speaker. Differences in the use of languages are created in the phase the speaker selects the signifiés, before the selection of the

morphemes. Diction is considered to derive from the differences in cognition. A photograph of a room can be *living room* to someone while the others describe it as *home* .

In the following phase, the morphemes which construct the sentence are selected from the morphemes which correspond to the signifiés. This is the morpheme selection. Unnatural sentence structures, such as unsuitable conjugation, are obviated as the directed edges for those structures are not formed.

A morpheme which follows another morpheme is hereinafter referred to as a subsequent morpheme. The morpheme in the beginning of a sentence is equally considered to be a subsequent morpheme of another morpheme; the morpheme is referred to as the α-morpheme (alpha-morpheme). The α-morpheme has no influences in the meaning of the sentence and has no signifiants. It has a function to lead the morpheme in the beginning of the sentence as its subsequent morpheme. Some morphemes in a sentence behave as subsequent morphemes of a morpheme before those (not the previous morpheme) e.g. the concord in Early Middle Japanese, present perfect in German. Those subsequent morphemes are referred to as the quasi-subsequent morphemes. The sentence generation stops when there are no morphemes needed.

The following are examples of the processes of sentence generation of a sentence: 'I saw a yellow house.' in English, Japanese and 'I've seen the yellow house.' in German. In each graph, the morphemes that are selected through the signifiés selection and the morpheme selection to be required to generate the sentences are highlighted in yellow.

Graph c describes the example sentence generation in English. First, the suitable signifiés are selected. Subsequently, the morphemes that correspond to the signifiés are selected. The morpheme 'house' is a subsequent morpheme of the adjective 'yellow', and the quasi-subsequent morpheme of the article 'a.' In the enumeration phase, all the morphemes are ordered starting with 'I' led by the α-morpheme in order of the directed edges of the graph.

Graph d describes the sentence generation in Japanese:

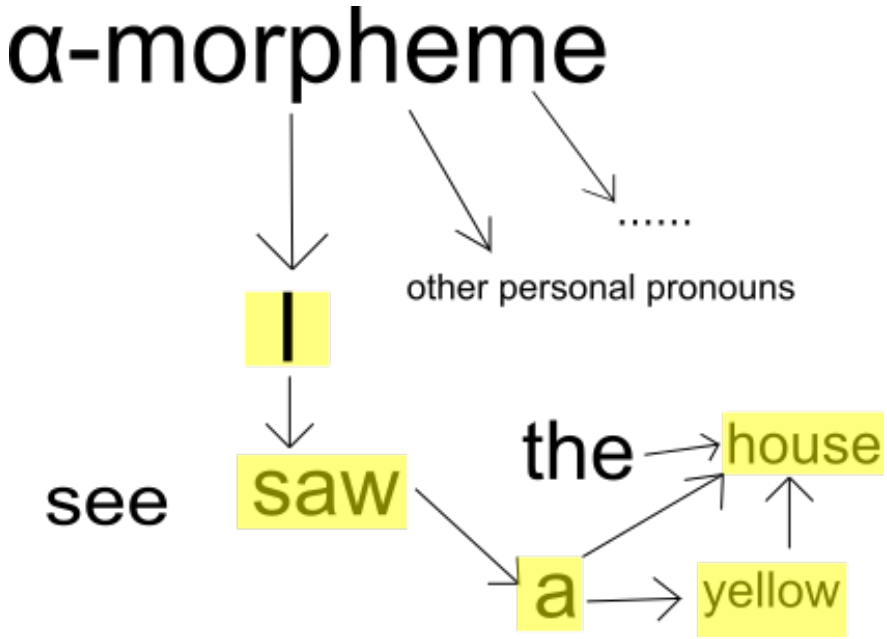
(私ha) 黄色i 家wo ta.

I.$_{\mathrm{NOM}}$ yellow house.$_{\mathrm{ACC}}$ saw.

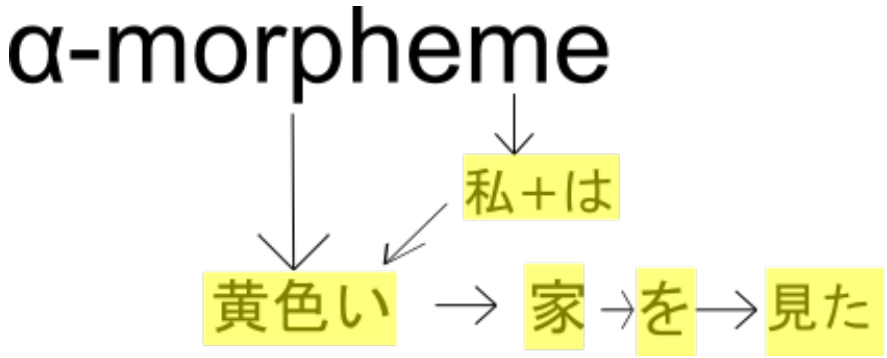
'I saw a yellow house.'

Processes of the sentence generation are identical in any languages; however different morphemes and signifiés are used in different languages. For instance, a subject of a verb in a sentence can be determined by the context in Japanese. There are other morphemes whose corresponding signifiés are determined by the context such as demonstrative adjectives and demonstrative pronouns. Therefore, the signifié graph is considered to be consisted of the long-term signifié graph as the knowledge of a language and the short-term signifié graph which is similar to the dynamic memory of computers. In this example, the subject 'I' is included in the verb with reference to the short-term signifié graph.

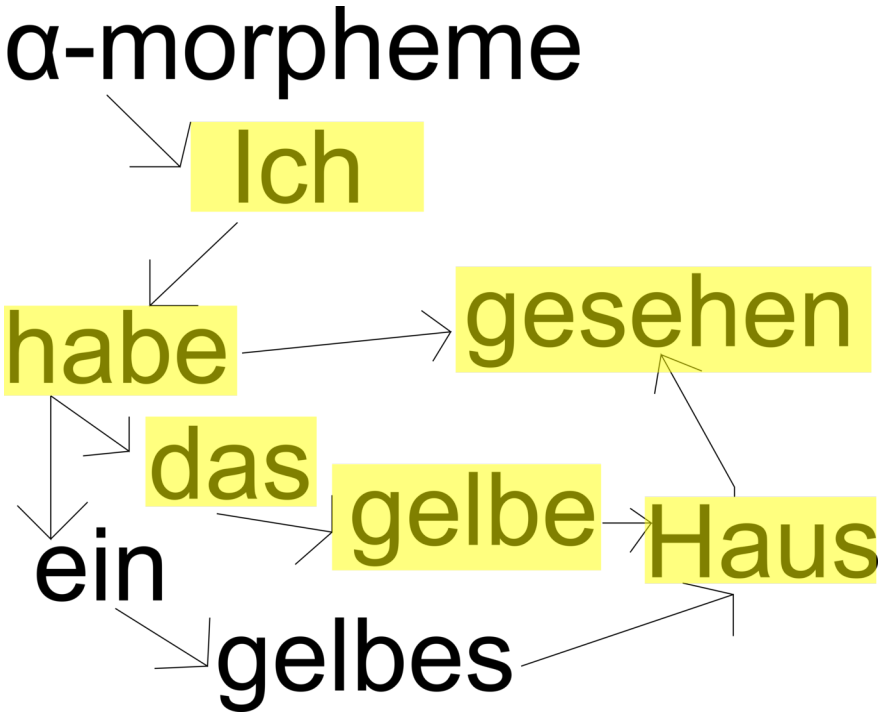
Graph e shows the processes in the sentence generation in German. The conjugated verb 'gesehen' is a quasi-subsequent morpheme of 'habe.'



Graph c



Graph d



Graph e

1.3 Morpheme alternation

Morphemes can be categorized by its grammatical role, the structures of the morpheme graph and its signifié. The morpheme alternation is done based on this categorization: a morpheme can be alternated with another morpheme which belongs to the same category. This categorization can be represented by labeling the nodes of the morpheme graph. The morpheme alternation is executed according to a certain algorithm based on the graph structure. For example, a sentence 'We saw a blue bird' can be generated using the directed edges used in generating the sentence 'I saw a yellow house'.

The following is the flow of the processing in the morpheme alternation.

1. The nodes of the morpheme graph are labelled by a certain algorithm.

2. After the selection of the morphemes which shall construct the sentence, the existence of the path that includes the whole nodes will be checked.

3. If a such path does not exist, search an alternate node which has the same label as the node to which no edges are directed.

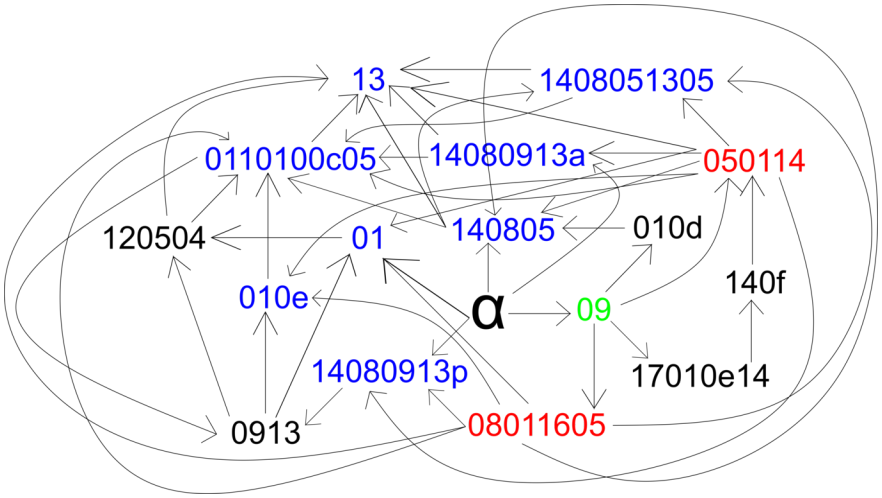
4. The sentence generation follows the structure of the graph of the alternate node; and it continues its processing.

This processing can be executed continuously.

1.4 An example of morpheme graph

The following graph is a subgraph of a morpheme graph of a language.

Graph f



The two morphemes highlighted in red are labeled identically. It is difficult to draw all the nodes and the edges of a complexed morpheme graph. However the morpheme graph can be represented simply in a form of adjacency list. The nodes of a morpheme graph can be represented in any forms as long as they can be distinguished from each other. Labeling and the representation are discussed later.

1.5 Grammar data

It is uncertain if there are data which only contain the connections of the nodes with certain labels. If such data do not exist, it is unknown which node is referred to in order to alternate the morphemes. There must be an algorithm to determine which node to refer to in either case and such data must be generated by the speaker itself.

In the application to computers, such data might enable more efficient sentence generation.

II Acquisition of language and application of the theory to artificial intelligence

In this chapter, the algorithms and the data structures discussed above are given concretely. Verification experiments for those are beyond the scope of this paper; however, those shall be tested in further investigations and studies.

2.1 Overview

In Government and Binding Theory(Chomsky 1981), the principles and parameters are assumed to constitute the innately given knowledge which constrains first language acquisition. The cluster grammar attempts to elucidate the acquisition of the language ability by the construction of the cluster graphs and other data

through reinforcement learning. The construction of the cluster graphs on human brains can be interpreted as a sort of connectionism.

Interpretation of the acquisition of the language ability by the connectionism causes a problem; it is difficult to elucidate the short amount of time in which an infant acquires a language. This problem will be solved by the assumptions that there is a part of the brain specialized to the acquisition of the language ability i.e. the algorithms, or that there are more efficient algorithms.

2.2 The phases in the acquisition of language ability

In the first phase, spectrum data of signifiés are made by innate abilities (data processing algorithms). Signifiés are related to signifiants and the morphemes are analyzed in order to form the nodes of the morpheme graph. Ultimately, the edges of the morpheme graph are formed. Artificial Intelligence must first analyze the primary inputted data in order to produce metadata such as the data that describe the correspondence between one specific input and a signifiant.

2.3 Computer and algorithms

The abilities required for the language processing in the cluster grammar are interpreted as algorithms. I. e, this theory has an application potential for computers. More concretely, computers programmed the data processing algorithms based on the cluster grammar are assumed to be able to analyze, learn and use languages and generate sentences like humans.

2.4 The differences of the cluster graphs in human brain and computer

The acquisition of the language ability is related to the generation of the cluster graphs. Both the morpheme graph and the signifié graph must be generated. Human brains and computers do not necessarily process the graphs with the same methods because of the structural differences.

There are two main possibilities for computers to express the morpheme graph: matrix and list

Adjacency list is generally unordered. By attaching the adjacency information only to the starting nodes of the directed edges, the list structure can be formed as a singly linked list in order to express the directed graph. This structure makes it possible for the list to contain the labels of each node within the list.

On the other hand, the adjacency matrix is unlikely to be suitable for the expression of the morpheme graph. The problem is not simply the data size, but the time O(n) required to search an adjacent node from another node: in sentence generation, this is a process to find the subsequent morphemes.

2.5 Algorithms to produce the signifiés graph

The cluster structures can be constructed through the signifiés data learning. The signifiés data relate arbitrary data such as video, image, audio, temperature and so on, which can be inputted, to morphemes. AI learns which data are related to which morphemes and vice versa. This process is similar to the handwritten character recognition but with larger and more various data. Similarities between several signifiés can also be learned. For example, some “white” data can also be learned as “bright” data and the relevance between these can be learned as well.

It is difficult to learn complexed ideas through this sort of learning. Those complexed ideas can be learned with the help of knowledge graphs, for example the one by Google, 2012.

2.6 Required algorithms for the cluster grammar

Following is the list of algorithms required to learn and process languages, construct sentences, and analyze their meanings.

1. Signifié Graph Construction

2. Morpheme Graph Construction

3. Signifié Selection

4. Pathfinding

5. Path Verification

6. Semantic Analysis

As for the Signifié Graph Construction, existing deep learning and neural network methods can be applied. The Morpheme Graph Construction consists of three algorithms: Node, edge and labeling algorithms. The node algorithm produces the nodes of the morpheme graph 1 by 1 in reference to the last layer of the neural network of the signifié graph. The edge algorithm produces the edges of the morpheme graph by simple learning. The labeling algorithm labels several nodes of the morpheme graph that are subsequent morphemes of some identical morphemes and share identical subsequent morphemes.

Graph f

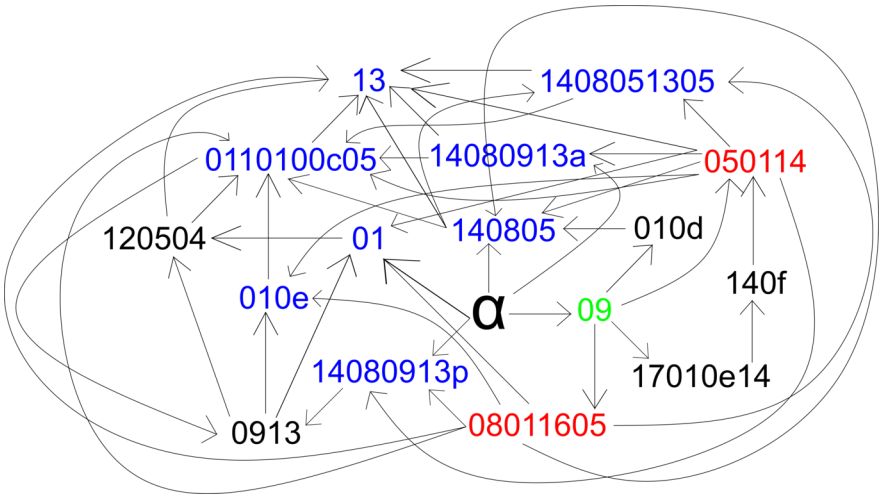


Figure 1:

In this example, the two morphemes highlighted in red are labeled identically. The morpheme graph shown in graph e is represented in a form of adjacency list as the following.

α	01, 09, 140805, 14080913a, 14080913p
01	120504
010d	140805
010e	0110100c05
0110100c05	0913, 13
050114	01, 010e, 0110100c05, 13, 140805, 1408051305, 14080913a, 14080913p
08011605	01, 010e, 0110100c05, 13, 140805, 1408051305, 14080913a, 14080913p
09	010d, 050114, 08011605, 17010e14
0913	01, 010e, 120504
120504	0110100c05, 13
13	
140805	0110100c05, 13, 1408051305
1408051305	0110100c05, 13
14080913a	0110100c05, 13
14080913p	0913
140f	050114
17010e14	140f

III Conclusion and further investigation

Although this theory is a pure interpretation on grammar structures, its aim is the application to computer science. The theory contends that the grammar structures can be represented in a different way from the traditional grammar theories and that it can be handled by artificial intelligence. The implementation experiments and the feasibility require further investigation.

Competing Interest

The author has no competing interests to declare.

References

Chomsky, N. (1986) *Knowledge of language: Its nature, origins and use* New York: Praeger.

Chomsky, N. (1981) *Government and binding theory* Cambridge, Mass.

White, L. (1989) *Universal grammar and second language acquisition* John Benjamins Publishing

Wexler, K., & Chien, Y. C. (1985) *The Development of Lexical Anaphors and Pronouns* Papers and reports on child language development, 24, 138-49.

Krashen, S. (1981) *Second language acquisition* Second Language Learning, 3(7), 19-39.

Collins, A. M. , & Loftus, E. F. (1975) *A spreading-activation theory of semantic processing Psychological review* , 82(6), 407

Guido van Rossum (1998) Python Patterns Implementing Graphs

(*https://www.python.org/doc/essays/graphs/*) Accessed April 18, 2020.

(*http://www.dais.is.tohoku.ac.jp/˜shioura/teaching/ad11/ad11-09.pdf*) Accessed April 16, 2020.

Rensselaer Polytechnic Institute, James, P. M. , Deborah, L. M. , John, S. E. and Katherine, C. (*https://www.authorea.com/users/6341/articles/107281)*Accessed April 18, 2020.