

Natural Language Analysis Using a Network Model

- Modification Deciding Network -

Masahiko Ishikawa Ryoichi Sugimura

Matsushita Electric Industrial Co.,Ltd.
1006, Kadoma, Kadoma-shi, Osaka, 571, Japan
ishikawa@isl.mei.co.jp sugimura@isl.mei.co.jp

Abstract

We have developed an original analyzing method using the network structure called the MDN. The network is similar to that of a neural network. In the MDN, all of the modification candidates can be compared in parallel, and it can decide the most appropriate interpretation effectively. It allows high quality of natural language analysis, and high analyzing speed. In this paper we will describe Japanese sentence analysis using the MDN, and then describe discussions about the MDN, comparing with sequential analysis, neural networks, and interactive analysis.

1 Introduction

As an application of natural language processing technology, a machine translation system from Japanese into English is under development. Our machine translation system is based on the transfer method that includes a Japanese sentence analysis module, a transfer module, and a generation module. The system has an interactive module to disambiguate sentence meaning.

Our main concern in Japanese sentence analysis is how to integrate syntactic constraints, semantic constraints, pragmatic constraints and user knowledge for sentence analysis which are traditionally applied in a sequential manner [Tsuji 84].

As a solution to this problem, we have developed an original method of analysis using a network structure. This network structure is called the *MDN (Modification Deciding Network)*. In the MDN, because all of the modification candidates can be compared in parallel, it becomes possible to ensure the correct analysis.

Significant characteristics of MDN are shown below.

- Because all of the modification candidates can be compared in parallel, it becomes possible to ensure the correct analysis.
- MDN does not need learning. If we use a neural network method which needs learning, we have to get a vast amount of pre-analyzed example sentences. However, it is almost impossible to get a

vast amount of data. Therefore, the neural network method is not suited to a large-scale practical system. Whereas, because MDN does not need learning, it is suitable for a large-scale practical NLP system.

- Because the number of nodes in the MDN is reduced to the minimum necessary, MDN gives the improvement of efficiency in memory space, and a high speed of analysis.
- The modification candidates in the MDN are selected gradually using four rounds of activation. This method is useful for avoiding a local minimum problem.

In this paper, we describe Japanese sentence analysis using the MDN, and then discuss the MDN, comparing with sequential analysis, neural networks, and so on. Though we describe Japanese analysis, the basic idea of the MDN is also useful for other natural languages.

2 Modification Analysis of Japanese

"Modification Analysis" of Japanese is the analysis of what kinds of phrase modify a verb phrase and the relationships that hold between phrases in terms of meaning. In this paper, a modifying phrase is **a phrase which is composed in terms of a noun phrase and postposition particles ("が", "を", etc. the postpositions which relate to the modification of a verb).**

One commonly used method of modification analysis is *case analysis* [Fillmore 68, Tsujii 84]. In case analysis, modification relations are decided by expressing the relationship between a verb and the phrases that modify it in terms of *deep case* (time, location, etc) and then by matching the semantic information of the modifying phrases with the case frame of the verb (which describes what semantics or what deep case a noun must have to be compatible with a verb).

3 Japanese Analysis Using MDN

This system also uses the case analysis method. With approx. 18,000 frames and over 3,000 semantic categories, our system embodies detailed case frames. However, because of the great vagueness caused by the structural and semantic freedom of Japanese, using case analysis alone, the system cannot resolve semantic ambiguities of the input sentence with the result that many interpretations are left. As a solution, the application of grammar rules based on heuristics has been considered, but the sequence of application of rules has presented a problem.

As a solution to this problem, we have developed the MDN. In this method, all of the modification candidates are first compiled as nodes of the network structure. The network has a structure similar to that of a neural network (a connectionist model) [Rumelhart 86, Selman 89]. Each node has a numerical value called the *activation level* and is connected to the other nodes by a *cooperative* link or by an *exclusive* link. Each link has a *weight*. However, there is no stratified structure, it is a mutually connected network like a Hopfield network [Hopfield 85].

A node updates its activation level with each time step. The new activation level is influenced by the sum of the weighted outputs from other nodes connected to it. This is called *spreading activation* [Collins 75, Rumelhart 86, Selman 89]. If the node A is connected to node B by a cooperative link,

an increase in the activation level of node A will increase the activation level of node B. If nodes are connected by an exclusive link, an increase in the activation level of node A will decrease that of node B.

In the MDN, the activation level is not only influenced by the weight of the links, but also by a value called the *activation level parameter*. The activation level parameter is set on each node. It represents the syntactic and semantic priority of the node.

The weight of the links and the activation level parameter are determined by a set of grammar rules called *control rules*. Also, by carrying out spreading activation in a manner similar to a neural network, the MDN can judge that the candidates that enter the activated state (whose activation level is higher than the threshold level) are correct.

Consider the following sentence.

Example sentence:

この カメラ で 子供 が 遊んでいる 場面 を 撮影した。 (1)
Kono kamera de kodomo ga asondeiru bamen wo satsueishita.
This camera child playing scene filmed
"The child who was playing was filmed with this camera."

Figure 1 shows a part of the network for this sentence. This sentence has many ambiguities. (a1, b1, etc. represent modification candidates in Figure 1)

a1,b2: suitable "The child who was playing was filmed with this camera."

a2,b2: unsuitable "The child who was playing with this camera was filmed."

a1,b1: unsuitable "(Someone) who was playing was filmed with this camera by the child."

By means of spreading activation, activation energy spreads over the network. Eventually, the node *a1* ("kono kamera de" modifies "satsueishita") and the node *b2* ("kodomo ga" modifies "asondeiru") are activated. Then the first interpretation ("The child who was playing was filmed with this camera.") is chosen.

4 The Process of Modification Analysis with MDN

The process of modification analysis as carried out in the MDN is shown below:

1. The sentence to be analyzed is input.
2. The input sentence is morphologically analyzed (word segmentation, supplementation of dictionary information etc).
3. Syntactic analysis is carried out. Verbs and any modifying phrases are recognized in this process.
4. Case Analysis is carried out and all possible modification candidates are extracted.
5. Modification candidates are compiled as nodes of the network structure.

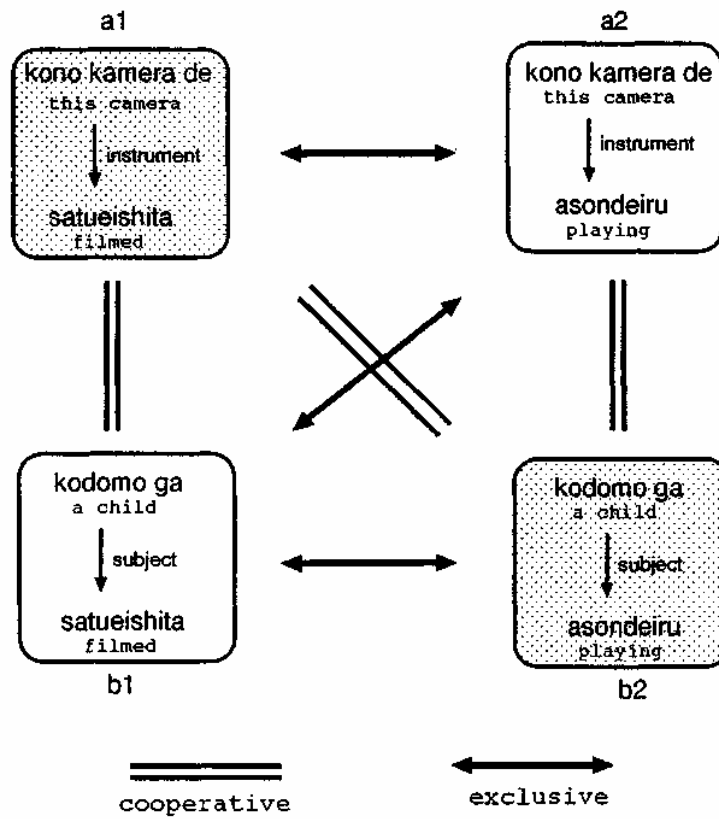
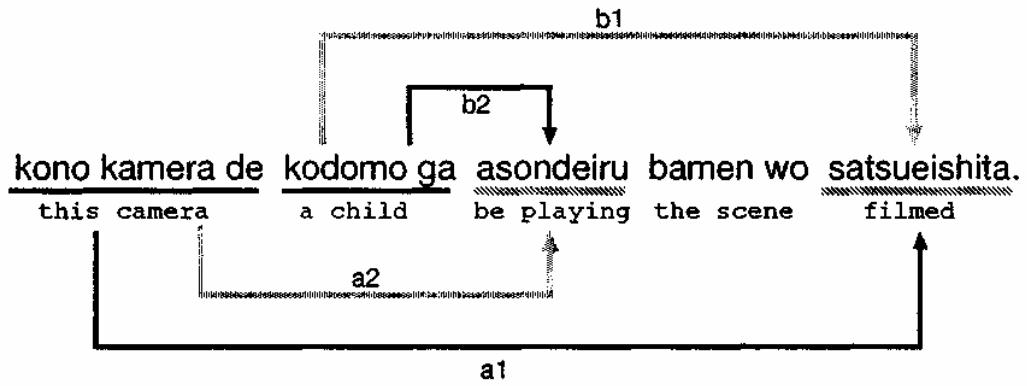


Figure 1: MDN diagrams for example sentence (1)

6. The link control rules are applied to the network. As a result, each node is connected with others by either a cooperative or exclusive link.
7. The level control rules are applied to the network. As a result, an activation level parameter is set on each node.
8. Spreading activation is carried out on the network (refer to Section 7) and only those modification candidates that are in the activated state remain.
9. If modification candidates that are in a complete exclusive relation (eg. Rule1 and Rule3 in Section6.1) remain, the process is repeated from stage 8. At such a time, the present activation level is halved and the process continued, the nodes whose activation level is below the threshold level are deleted and spreading activation is repeated. If no modification candidates in a complete exclusive relation remain, the modification candidates that are in the activated state at that time are used as the correct interpretation. Then the analysis tree is built.

The possible number of repetitions of stages 8 and 9 is restricted to four. If it is the case that, even after spreading activation has been carried out four times, there are still modification candidate pairs with complete exclusive relations, then the average value of the activation level up to that point is compared and the higher score is the one used.

5 Data Structure of MDN

Using case analysis, all the candidates with possible modification relations are extracted. Some of the modification candidates for example sentence (1) are shown in Figure 2.

[8]	advp: <i>kamera de</i> , vp: <i>asondeiru</i> , frame: 1, slot: 13, deep case: <i>instrument</i> , case frame: <i>play progressive</i>
[9]	advp: <i>kodomo ga</i> , vp: <i>asondeiru</i> , frame: 1, slot: 1, deep case: <i>subject</i> , case frame: <i>play progressive</i>

Figure 2: Modification candidates

In Figure 2, *advp* represents a modifying phrase, and *vp* a verb phrase. For example, the modification candidate [8] of Figure 2 represents the modification relation in which the modifying phrase "kamera de" modifies the 13th slot (which indicates the deep case, here "instrument") of the first frame (the verb meaning, here "play progressive") of the verb phrase ("asondeiru").

In this system, the modification candidate node is compiled as a structure of C language. Figure 3 shows a partial MDN for example sentence (1). The meaning of each term in Figure 3 is shown below.

advp: ID number of modifying phrase,	vp: ID number of verb phrase
frame: frame number of case frame,	slot: slot number (deep case)
asems: semantic code of advp,	level: activation level

act: activation level parameter,
link: cooperative / exclusive link

trace: history of activation level

[4]	advp: 5	vp: 7	frame: 1	slot: 1	
	asems: s5230 s5130				
	level: 2000.000	act: 7000			
	trace:				
	1000.000	1107.796	1211.992	1312.976	1411.123
	1506.797	1600.351	1692.129	1782.469	1871.700
	1960.148	2000.000	2000.000	2000.000	2000.000
	2000.000	2000.000	2000.000	2000.000	2000.000
	2000.000				
	link:				
	(0) 0	(1) 0	(2) 0		
	(3) 0	(4) 0	(5) -1000		
	(6) -800	(7) 0	(8) 0		

Figure 3: example of MDN (a part)

6 Control Rules

Control rules come in the form of *link control rules* and *level control rules*. In the current version, there are 22 control rules.

6.1 Examples of Link Control Rules

Nodes are connected by either cooperative or exclusive links according to the link control rules. In a mutually cooperative relationship, partner nodes are connected with a cooperative link, and partner nodes in an exclusive relationship are connected with an exclusive link. The weight of a link is changed according to the degree of cooperativeness or exclusiveness. The weight of a cooperative link is a positive value, while the weight of an exclusive link is a negative value.

Examples of link control rules are as follows.

Rule 1: A single verb cannot simultaneously express two meanings.

When comparing two modification candidates, if the *vp* number is the same, but the *frame*, number (the number corresponding to the meaning of the verb) is different, these are connected by an exclusive link. These are in a complete exclusive relation.

Example sentence:

私	は	1年生	の	英語	を	持って	いる。	(2)
<i>Watashi</i>	<i>wa</i>	<i>1nensei</i>	<i>no</i>	<i>eigo</i>	<i>wo</i>	<i>matte</i>	<i>iru.</i>	
<i>I</i>		<i>freshmen</i>		<i>English</i>		<i>take charge of</i>		

"I teach English to the freshmen."

The verb 'motte' has several meanings (have, take charge of, etc.) in general, but in this sentence (2) the word has only one meaning (that is 'take charge of').

In this case, the node (vp = 'motte', meaning = 'have') and the node (vp='motte', meaning = 'take care of') are connected with an exclusive link.

Rule 2: When a single modifying phrase simultaneously modifies a number of verb phrases, there are many cases when the phrase's deep case is identical.

When comparing two modification candidates, if their *advp* number is the same, their *vp* number is different and their deep case is the same, then they are connected by a cooperative link.

Example sentence:

彼 は 本 を 買い、それを 読んだ。 (3)
Kare wa hon wo kai, sore wo yonda.
He book buy it read
"He bought a book and read it."

In this sentence, the noun 'kare' simultaneously modifies the verb 'kai' and the verb 'yonda'. The deep case of 'kare' and 'kai' is 'subject', and the deep case of 'kare' and 'yonda' is also 'subject'. In most cases, they are identical.

Rule 3: Modification relationships must not cross each other (The Non-Intersection Condition).

Pairs of modification candidates that break the non-intersection condition are connected by an exclusive link. These are in a complete exclusive relation.

Example sentence:

私 は 彼 が 昨日 作曲した 曲 を 聞いた。 (4)
Watashi wa kare ga kinou sakkyokushita kyoku wo kiita.
I he yesterday composed song listened
"I listened to the song he composed yesterday."

The node ('kare' modifies 'kiita') and the node ('watashi' modifies 'sakkyokushita') break the non-intersection condition (their modification relationships cross each other). They are connected by an exclusive link.

6.2 Examples of Level Control Rules

An activation level parameter is set on each node according to level control rules. In the case that the priority of a node is to be increased, the activation level parameter is increased. In the case that the priority of a node is to be lowered, the activation level parameter is decreased.

Examples of level control rules are as follows.

Rule 4: Obligatory case is given precedence over optional case and free case.

The activation level parameter is changed according to the degree of precedence of the deep cases.

Rule 5: Precedence is given to the modification candidate whose modifying phrases are closer in distance to verbs than others.

The activation level parameter is changed according to the degree of precedence of distance.

Rule 6: It is rare that a modifying phrase with a *touten* punctuation mark ("_") modifies an embedded verb phrase.

When checking a modification candidate, if the modifying phrase contains a *touten* mark and the verb phrase it modifies is embedded, then the activation level parameter is lowered.

Rule 7: Precedence is given to a modification candidate in which the modifying phrase and the verb phrase have strong connection in terms of semantics (for example, "camera" and "filmed").

7 Spreading Activation

Spreading activation is performed as shown below (refer to [Williams 86, Collins 75]). In the general spreading activation algorithm, the new activation level is a function of its previous activation level and the sum of the weighted outputs from other nodes connected to it. However, in the MDN, the activation level parameter is also added at each time step.

$$\begin{aligned}
 A_j(t+1) &= A_j(t) + (act \times J_DELTA1 \times n) \\
 &\quad + J_DELTA2 \times \left(\sum_{i=0}^{n-1} (E_{ij} \times A_i(t)) + \sum_{i=0}^{n-1} (I_{ij} \times A_i(t)) \right) \\
 &\quad (0 \leq A_j(t+1) \leq J_MAX_LEVEL)
 \end{aligned}$$

where

$A_j(t)$: the activation level of the j th MDN node at time t

act : the activation level parameter

$E_{ij}(\geq 0)$: the weight of the cooperative link from i node to j node

$I_{ij}(< 0)$: the weight of the exclusive link from i node to j node

n : the total number of nodes

J_DELTA1 : the delta coefficient for activation level parameters

J_DELTA2 : the delta coefficient for link control rules

J_MAX_LEVEL : the maximum value of activation level

In the current version, the number of repetitions of spreading activation (the maximum value of t) is set at 20. As mentioned above, this is repeated for a maximum of 4 rounds (refer to Section 4).

8 Discussions about MDN

Let us discuss characteristics of MDN comparing sequential analysis, neural networks, etc.

8.1 MDN vs Sequential Analysis

Most conventional natural language processing systems use a sequential analysis method. Syntactic and semantic processing are separated in these systems. However, this separation is not good in many cases, and the sequence of applying grammar rules may cause a problem.

Consider the following sentences.

Example sentences:

私 は この カメラ で 子供 が 遊んでいる 場面 を 撮影した。 (5)
Watashi wa kono kamera de kodomo ga asondeiru bamen wo satsueishita.
I this camera child playing scene filmed
"I filmed with this camera the child who was playing"

私 は 子供 が この カメラ で 遊んでいる 場面 を 撮影した。 (6)
Watashi wa kodomo ga kono kamera de asondeiru bamen wo satsueishita.
I child this camera playing scene filmed
"I filmed the child who was playing with this camera."

Both sentences have the following ambiguities.

modification candidate 1: "kamera de" modifies "asondeiru"

modification candidate 2: "kamera de" modifies "satsueishita"

Candidate 1 has precedence for distance (the modifying phrase is closer to the verb phrase; refer to Rule 5 in Section 6.2). Candidate 2 has precedence for strong connection ("kamera" and "satsueishita" have a strong connection; refer to Rule 7 in Section 6.2).

In sentence (5), candidate 2 is preferable to candidate 1 (Rule 7 is stronger than Rule 5). However, in sentence (6), candidate 1 is preferable (Rule 5 is stronger than Rule 7). Therefore, the order of applying rules should be changed according to the situation. Sequential analysis systems are at a disadvantage in this problem.

Syntactic, semantic, and pragmatic rules should be processed in a highly integrated manner. The MDN is suitable for modeling highly integrated forms of processing. In the MDN, because all of the possible modification candidates can be compared in parallel, compared to systems that use sequential analysis, more accurate analyses can be obtained.

8.2 MDN vs Neural Network

Using a neural network method, the weight of a link is determined using learning [McClelland 86]. If we want to develop a practical MT system, we need a vast amount of sentence data whose analyzed results are already known. They are to be used as learning data. However, it is almost impossible to get such a vast amount of analyzed sentences. Consequently, most natural language processing systems using a neural network method can handle only a small number of input sentences. In the MDN, because the

weight of a link and activation level parameter are determined by the control rule, there is no need to get example sentences for learning. Therefore, the MDN is useful for a practical (i.e. large-scale) natural language processing system.

Using a neural network method, it is not clear what sort of operation each of the nodes is performing or how many nodes are needed. Consequently, not all of the nodes are functioning effectively. Using the MDN, however, because the nodes correspond to the modification candidates, whose number has been previously reduced by case analysis, the number of nodes in the network is reduced to the minimum necessary, and efficiency of processing and use of memory space are improved. Therefore, the MDN achieves a high speed of analysis.

In a neural network, the weight of a link is determined by using a single learning rule (such as a *back-propagation rule* [Rumelhart & Hinton 86]). Whereas in the MDN, it is determined by a number of control rules, and so the network can be controlled much more precisely.

8.3 Interactive analysis and MDN

Because natural languages, such as Japanese and English, have a lot of ambiguities, it is difficult to analyze all input sentences perfectly. So interaction with users can be useful [Maruyama 90]. However, if the system asks the user about all modification ambiguities, they may become tired of the interaction. A system using an MDN can make use of only a little help from the user. For example, take the case where the user teaches the system: "This modification ('kamera' modifies 'asondeiru') is wrong. I'm not sure about the other modifications." In the MDN, the initial activation level of the wrong modification candidate is set at zero, and then spreading activation is carried out again. As a result, the new interpretation which satisfies the intention of the user will be obtained. Therefore, the MDN is suitable for interactive analysis. Indeed, our MT system has an interactive function using the MDN.

8.4 Other characteristics of MDN

- In the MDN, the modification is determined using four rounds of activation rather than one. Because deactivated modification candidates are removed after each round of spreading activation, the candidates are selected gradually using this method. As a result, a more precise interpretation can be obtained. This is useful for avoiding a local minimum problem.
- The MDN can embed different type of rules easily. We have implemented link control rules and level control rules. If you want to handle rules concerning mutual relations between modification candidates, you use link control rules. In the case of rules concerning the precedence level of the modification candidates, you use level control rules.
- Because the reliability of the grammar rules can be expressed by the weight of a link and the activation level parameter, the influence of each grammar rule on the network varies with the reliability of the rule. Therefore, even heuristic rules whose reliability is not very high are sufficient for use in the system.

9 Conclusions

We have discussed the basic idea of the MDN and compared natural language analysis approaches using a sequential process, a neural network and an interactive process.

We believe that syntactic, semantic, and pragmatic rules should be processed in an integrated manner. We wanted to handle rules of a different type including heuristic rules in a highly integrated manner. A neural network or a connectionist model is suitable for that purpose. However, it needs a vast amount of example sentences for learning, and it takes much time to compute. So most natural language processing systems with a neural network can only handle a small number of input sentences.

We have therefore developed the MDN. The network structure of the MDN is similar to that of a neural network, but the MDN does not need learning. Therefore, there is no need to get a lot of example sentences for learning. Also, the number of nodes in the MDN is reduced to the minimum necessary. Consequently, the MDN can handle a large number of sentences at high speed and is suitable for a large-scale natural language processing system.

We can also think of natural language analysis as a constraint satisfaction problem (a combinatorial optimization problem). There are a lot of constraints, such as syntactic, semantic and heuristic constraints. Hopfield proposed a mutually connected network called the Hopfield network as a solution to this problem [Hopfield 85]. The MDN is a mutually connected network. In the MDN, constraints are represented as control rules. The spreading activation mechanism plays an important role in deciding the most appropriate interpretation.

We tried to improve the disadvantages (speed, scale-problem) of a neural network, and apply its advantages (parallelism, suitability for a constraint satisfaction problem) to natural language processing. We have now developed many grammatical rules. The MDN is a method which amalgamates a neural approach and a rule-base approach. So our approach is a middle ground approach between empiricist and rationalist.

Acknowledgments

We would like to thank Timothy Cornish and Tatsuro Kyoden for useful comments and discussions.

References

- [Charniak 83] Charniak, E.: "Passing Markers: A Theory of Contextual Influence in Language Comprehension", *Cognitive Science* 7, pp.171-190, 1983
- [Collins 75] Collins, A.M. and Loftus, E.F.: "A spreading-activation theory of semantic processing", *Psychological Review* 82, pp.407-429, 1975
- [Faisal 90] Faisal, K.A. and Kwasny, S.C.: "Design of a Hybrid Deterministic Parser", *Coling-90*, pp.11-16, 1990
- [Fillmore 68] Fillmore, C.J.: "The Case for Case", In *Universals in Linguistic Theory*, Bach, E. and Harms, R.T.(eds), Holt, Rinehart and Winston, 1968

- [Hopfield 85] Hopfield, J.J. and Tank, D.W.: "Neural computation of decisions in optimization problems", *Biol. Cybernetics* 52, pp.141-152, 1985
- [Maruyama 90] Maruyama, H., and Watanabe, H.: "An Interactive Japanese Parser for Machine Translation", *Coling-90*, pp.257-262, 1990
- [McClelland 86] McClelland, J.L. and Kawamoto, A.H.: "Mechanisms of Sentence Processing: Assigning Roles to Constituents of Sentences", In *Parallel Distributed Processing*, Rumelhart, D.E. and McClelland, J.L., MIT Press, Volume 2, pp.272-325, 1986
- [Rumelhart 86] Rumelhart, D.E. and McClelland, J.L.: *Parallel Distributed Processing*, MIT Press, 1986
- [Rumelhart & Hinton 86] Rumelhart, D.E., Hinton, G.E. and Williams, R.J.: "Learning Internal Representation by Error Propagation", In *Parallel Distributed Processing*, Rumelhart, D.E. and McClelland, J.L., MIT Press, Volume 1, pp.318-362, 1986
- [Selman 89] Selman, B.: "Connectionist systems for natural language understanding", *Artificial Intelligence Review* 3, pp.23-31, 1989
- [Tsujii 84] Tsujii, J., Nakamura, J., and Nagao, M.: "Analysis Grammar of Japanese in the Mu-Project", *Coling-84*, pp.267-274, 1984.
- [Waltz 85] Waltz, D.L. and Pollack, J.B.: "Massively Parallel Parsing: A Strongly Interactive Model of Natural Language Interpretation", *Cognitive Science* 9, pp.51-74, 1985
- [Williams 86] Williams, R.J.: "The Logic of Activation Functions", In *Parallel Distributed Processing*, Rumelhart, D.E. and McClelland, J.L., MIT Press, pp.423-443, 1986