

マルチエージェントによる頑健な自然言語処理の協調方式

小松英二 福本淳一 木下哲男

沖電気工業(株) マルチメディア研究所

e-mail: {komatsu, fukumoto, kino}@okilab.oki.co.jp

従来の自然言語処理は形態素解析、構文解析、意味解析、文脈解析を逐次的に行う処理モデルに基づいたものが多かった。このような逐次的な処理モデルでは、知識を有効に利用できず、逐次的な処理が頑健性を実現するための障害となっていた。この問題点を解決するため、知識を効率的かつ有効に利用できるモデルとして、マルチエージェントモデルが考えられる。本稿では、頑健性を曖昧性解消に絞り、例文を用いて知識利用の分析・考察を行ない、及び、マルチエージェントに基づく協調方式の枠組みについて検討する。

Cooperation in Multi-Agent Based Robust NLP

Eiji KOMATSU Jun'ichi FUKUMOTO Tetsuo KINOSHITA

Media Laboratories, Oki Electric Industry Co., Ltd.

In this paper, we discuss an robust NLP system architecture based on multi-agent model. Firstly, we will show some problems in current NLP systems whose processing modules are organized sequentially. Then we will analyze these problems in terms of how knowledge is handled in each agent to resolve ambiguities which are caused by sequential processing.

Conventional NLP systems is composed of morphological, syntactical, semantic and discourse analyzer, sequentially and knowledge in each module can not be used mutually. On the other hand, in a multi-agent based NLP system, knowledge is accessible from any agents due to cooperations among agents, therefore it can attain robustness.

1 はじめに

従来の自然言語処理のモデルは、形態素解析、構文解析、文脈解析等の各処理を逐次的に行うものが多かった。逐次的な処理モデルは、効率についてはすぐれている一方、処理系の持っている知識を十分有効に利用しているとはいえなかった。このような逐次的な解析の問題点を解決するため、解析を統合的にする研究が行なわれてきた [1] [2]。

筆者らは、従来の逐次的な自然言語処理では解析が難しかった様々な言語現象を扱うことのできる頑健な自然言語処理方式として、マルチエージェントに基づく処理モデルを研究している [3]。本モデルでは、個々の知識源を基本的なエージェントとし、さらに、知識源を組み合わせることにより、従来の処理方式のモジュールの機能を果たすエージェントを

構成する。エージェント間のデータの流れ、及び、制御の流れの自由度を高めることにより、システムが持っている知識を有効に利用することを目標としている。

以下、2 節で頑健性、3 節で逐次的な処理モデルとマルチエージェントモデルとの比較検討、4 節で曖昧性解消における知識利用の分析、5 節でマルチエージェントモデルに基づく協調方式についての枠組みについて述べる。

2 頑健性

自然言語処理の頑健性には、様々な定義がある [4]。本稿では、[4] に従って、できるだけ多くの言語現象に対して結果を返せることとする。正解を求める(理解する)ための障害となるような言語現象を不適格

性と呼び、解析の主体が人間である場合には絶対的な不適格性、プログラムである場合には相対的な不適格性と呼ぶ。不適格性を含む文を不適格文と呼び、相対的な不適格性を含む文を相対的な不適格文、絶対的な不適格文を含む文を相対的な不適格文と呼ぶ。次に、絶対的な不適格文と相対的な不適格文の例を示す。

絶対的な不適格文： ファイル見る 書きを食べる。 MT(機械翻)訳 ○○党が××党に噛みつく。 昨日内定した人に会った。	(助詞抜け) (変換誤り) (タイプ誤り) (暗喩) (曖昧文)
相対的な不適格文： 1. まえがき あることが明らかになった。 それを作った会社が調査した。	(箇条書き) (形態素解析での曖昧文) (構文解析での曖昧文)

図 1: 絶対的な不適格文と相対的な不適格文の例

本稿では、不適格性を相対的な曖昧性に絞って検討する。

3 逐次的なモデルとマルチエージェントモデルとの比較検討

2 節で述べた、相対的な曖昧性に関連して、逐次的な処理モデルと、マルチエージェントモデルを比較する。図 2 に示す通り、モジュール間のデータ、及び、制御の流れは一方向、かつ、固定的である。

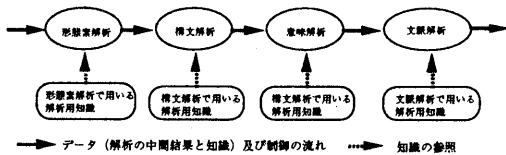


図 2: 逐次的な処理モデル

各モジュールでは、モジュール内の限られた知識で曖昧性を減らさなくてはならないため、解析に必要な知識が他のモジュールに存在する場合、それを十分に利用することができず、曖昧性が発生していた。また、次の処理へすべての中間結果を渡すこと困難であるため、ヒューリスティックス等の知識を用いて結果を絞り込まなければならないため、正しい

解析結果が途中の段階で削除される等の問題があった。さらに、知識自体も、各モジュールに重複して持たざるを得ず、知識の作成・管理を複雑にする可能性があった。

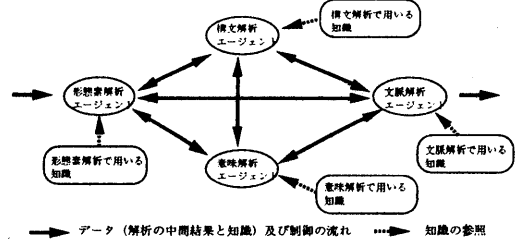


図 3: マルチエージェントモデル

一方、図 3 に示すようにマルチエージェントモデルでは、データ及び制御の流れは特に制限しない。このため、必要ときに、必要な中間結果及び知識を利用できるため、相対的な曖昧性の発生が減少、及び、知識の作成・管理の困難を改善することが期待され、頑健性の向上に寄与すると予想される。

4 曖昧性解消における知識利用の分析

3 節で示したマルチエージェントモデルにおける、エージェント間のデータ及び制御の流れを明らかにするために曖昧性が発生した例文の曖昧性発生原因の分析を行った。分析方法は、エージェントとして、形態素解析、構文解析、意味解析、文脈解析の 4 つのエージェントを想定し、曖昧性の生じたエージェントが他のすべてのエージェントに曖昧性解消のための処理を依頼するものとして、処理の流れを机上でシミュレーションを行なった。また、曖昧性解消に有効な知識は必要に応じて具体的に構成し、曖昧性が解消できるかどうかを調べた。

4.1 形態素解析の曖昧性解消の分析

例 1. 構文解析の知識の利用例

- 原文
「話し合いをして、結論を出す。」
- 曖昧性¹

¹「/」は文節区切り以外の形態素区切り、「//」は文節区切りである形態素区切りを表す。

- (a) 「話し合い/を//し(動詞)/て(接続助詞)
//、//結論/を//出/す/。」
- (b) 「話し合い/を//して(名詞)/、//結論/を
//出/す/。」

3. 曖昧性解消に有効と思われる知識²

- W_1 : m (見出し = 話し合い, 品詞 = 名詞)³
- W_2 : m (見出し = する, 品詞 = 動詞)
- W_3 : m (見出し = 結論, 品詞 = 名詞)
- W_4 : m (見出し = 出す, 品詞 = 動詞, 必須の表層格 = (が, を, と))
- W_5 : m (見出し = して, 品詞 = 名詞)
(付属語の知識は省略した)
- D_1 : p (自立語 = m (品詞 = 名詞), 後接付属語 = m (見出し = を, 品詞 = 格助詞))
← p (自立語 = m (品詞 = 動詞)), ただし、「を」格は重複しない。⁴
- D_2 : p (自立語 = m (品詞 = 動詞), 後接付属語 = m (見出し = て, 品詞 = 接続助詞))
← p (自立語 = m (品詞 = 動詞))
- D_3 : p (自立語 = m (品詞 = 名詞))
← p (自立語 = m (品詞 = 動詞))

(a), (b) の曖昧性解消を他のすべてのエージェントに依頼する。

構文解析: (a) は、 W_1 、 W_2 、 W_3 、 W_4 、 D_1 、 D_2 を用いて、構文解析が成功する。(b) は、「話し合いを」が「して(名詞)」に係るようにする規則がないため、 W_1 、 W_4 、 D_1 により「出す」に係るしかない。しかし、「結論を」も W_3 、 W_4 、 D_1 により「出す」に係るため、「を」格が重複するため、構文解析は失敗する。従って、構文解析で (b) は誤りであると分かり、形態素解析の曖昧性解消ができる。

意味解析: 意味関係をチェックするためには、構文解析が必要であるが、構文解析からは、(a) の結果しか戻らず、既に曖昧性が解消されているため、意

²以下の知識の記述では、単語辞書の知識を W 、係り受け規則の知識を D 、意味関係辞書の知識を S 、文脈解析の照応規則を A と表すものとする。

³ $m()$, $p()$ は、それぞれ形態素、文節を示すものとする。また、そのなかの記法として、 $XX = YY$ の XX はその要素の属性名、 YY はその値を表す。

⁴ p_1 ← p_2 は、文節 p_2 が文節 p_1 に係ることを示す。

味解析は必要ない。従って、この例では意味解析は曖昧性解消に役立たない。

文脈解析: この例では文脈解析は曖昧性解消に役立つ知識が想定できず、曖昧性解消には役立たないと考えられる。

この例では、構文解析のみが曖昧性解消に必要なことが分かる。意味解析から構文解析を呼び出したように、連鎖的にエージェントが呼び出される処理が起こる。この例では、構文解析で曖昧性解消が行えるため、意味解析、文脈解析の処理は冗長な処理になる。

例 2. 意味解析の知識の利用例

1. 原文

「丘のうえに立って考えた。」

2. 曖昧性

- (c) 「丘のうえに//立(動詞)/つ(語尾)/て(接続助詞)//考え/た/い /。」
- (d) 「丘のうえに//立(名詞)/つて(格助詞)//考え/た/い /。」

3. 曖昧性解消に有効と思われる知識

- W_6 : m (見出し = 丘, 品詞 = 名詞, 意味カテゴリ = [丘])
- W_7 : m (見出し = のうえに, 品詞 = 助詞相当語, 意味カテゴリ = location)
- W_8 : m (見出し = 立つ, 品詞 = 動詞, 必須の表層格 = (が, を, と or のうえに), に格の深層格は location, 意味カテゴリ = [立つ])
- W_9 : m (見出し = 考える, 品詞 = 動詞, 必須の表層格 = (が, を, と or って) or (が, を), 意味カテゴリ = [考える], 「って」格の深層格は role, 意味カテゴリ = [考える], 「のうえに」格の深層格は location)
- W_{10} : m (見出し = 立, 読み = りっとる, 品詞 = 名詞, 意味カテゴリ = [リットル])
(「のうえに」以外の付属語の知識は省略した)
- D_4 : p (自立語 = m (品詞 = 名詞), 後接付属語 = m (見出し = のうえに))
← p (自立語 = m (品詞 = 動詞))

- D_5 : p (自立語 = m (品詞 = 動詞), 後接付属語 = m (見出し = て))
 ← p (自立語 = m (品詞 = 動詞))
- D_6 : p (自立語 = m (品詞 = 名詞), 後接付属語 = m (見出し = のうえに))
 ← p (自立語 = m (品詞 = 名詞))
- D_7 : p (自立語 = m (品詞 = 名詞), 後接付属語 = m (見出し = のうえに))
 ← p (自立語 = m (品詞 = 動詞))
- D_8 : p (自立語 = m (品詞 = 名詞), 後接付属語 = m (見出し = って))
 ← p (自立語 = m (品詞 = 動詞))
- S_1 : [丘]is_a[具体物]
 S_2 : [立つ]locate_at[具体物]
 S_3 : 「立つ」と「考える」の意味関係は認める。

(c)、(d)の曖昧性解消を他のすべてのエージェントに依頼する。

構文解析: (c)は W_6 、 W_7 、 W_8 、 W_9 、 D_4 、 D_5 (または、 D_7 、 D_8)を用いて解析が成功する。(d)は、 W_6 、 W_7 、 W_9 、 W_{10} 、 D_6 、 D_8 を用いて解析が成功するため、構文解析では曖昧性解消はできない。構文解析では、(c)、(d)のそれぞれについて曖昧性が発生する。構文解析は曖昧性を解消して、その結果を形態素解析へ返すことになる。(構文解析での曖昧性解消は、例4参照)

意味解析: 意味関係をチェックするためには、構文解析が必要であるため構文解析を呼び出す。構文解析からは、(c)について構文解析結果が戻ってくる。(この時、構文解析でも曖昧性解消が必要となっている。(この曖昧性解消については、例4参照)

(c)の構文構造を以下に示す。

(c1) ((丘のうえに, 立(動詞)って), 考えた。)

(c1)は、 W_6 、 W_7 、 W_8 、 W_9 から得られる意味カテゴリを用いると、 S_1 、 S_2 、 S_3 の意味関係を満たすため、意味解析が成功する。従って、(c)だけについて意味解析が成功するため、(d)は誤りであると分かり、曖昧性が解消できる。

文脈解析: この例では文脈解析は曖昧性解消に役立つ知識が想定できず、曖昧性解消には役立たないと考えられる。

この例では、意味解析が曖昧性解消に必要なことが分かる。例1と異なり、形態素解析から呼び出され

た意味解析がさらに構文解析を呼び出す処理が必要になり、構文解析も曖昧性解消が必要であるが、曖昧性解消に直接役立つのは、意味解析である。

例3. 構文解析、意味解析、文脈解析の知識の利用例

1. 原文

「最初がこの審議会である。」

2. 曖昧性

(e) 「最初/が(格助詞)//この(連体詞)/審議会/である/。」

(f) 「最初//がこ(名詞)/の(格助詞)//審議会/である/。」

「がこ」は、「臥虎」の読みで、「ふしている虎」の意。3つ前の文は、「冷夏対策審議会は、……」という文とする。

3. 曖昧性解消に有効と思われる知識

W_{11} : m (見出し = 最初, 品詞 = 名詞)

W_{12} : m (見出し = この, 品詞 = 連体詞)

W_{13} : m (見出し = 審議会, 品詞 = 名詞, 意味カテゴリ = [審議会])

W_{14} : m (見出し = がこ, 品詞 = 名詞, 意味カテゴリ = [臥虎])

D_9 : p (自立語 = m (品詞 = 名詞), 後接付属語 = m (見出し = が, 品詞 = 格助詞))

← p (自立語 = m (品詞 = 名詞), 後接付属語 = m (見出し = である))

D_{10} : p (自立語 = m (品詞 = 連体詞)

← p (自立語 = m (品詞 = 名詞))

D_{11} : p (自立語 = m (品詞 = 名詞)

← p (自立語 = m (品詞 = 名詞), 後接付属語 = m (見出し = である))

D_{12} : p (自立語 = m (品詞 = 名詞), 後接付属語 = m (見出し = の))

← p (自立語 = m (品詞 = 名詞))

S_4 : 「最初」、「この」、「審議会」、「がこ」については意味関係はすべて認める。

C_1 : m (見出し = この, 品詞 = 連体詞), m (品詞 = 名詞, 意味カテゴリ = [A]) は, m (品詞 = 名詞, 意味カテゴリ = [B]) を照応する。ただし、[A] = [B]、[A]is_a[B]、又は、[B]is_a[A]

C_2 : m (品詞 = 名詞, 意味カテゴリ = [A]) は
 m (品詞 = 名詞, 意味カテゴリ = [B]) を
 照応する。ただし、[A] = [B]、[A]is.a[B]
 、又は、[B]is.a[A]

(e)、(f) の曖昧性解消を他のすべてのエージェント
 に依頼する。

構文解析：(e) は W_{11} 、 W_{12} 、 W_{13} 、 D_9 、 D_{10} を用
 いて構文解析できる。(f) は W_{11} 、 W_{13} 、 W_{14} 、 D_{11} 、
 D_{12} を用いて構文解析できる。両方の候補共構文解析
 が成功するので曖昧性解消はできないが、「最初」と
 「である」の共起の確からしさに、何らかの尤度を定
 義することを認めれば、(e) を優先することができる。

意味解析：意味関係をチェックをするため、構文
 解析を呼び出す。(e)、(f) のどちらの候補も構文解析
 が成功する。 S_4 ではどちらも誤りでないが、構文解
 析と同様に、「がこ」と「審議会」の意味関係の確か
 らしさを認めれば、(e) を優先することができる。

文脈解析：構文解析、意味解析を呼び出した後で、
 (e)、(f) の照応処理を行う。(e) では、 C_1 を用いて、
 「この/審議会」から 3 文前の「冷夏対策審議会」への
 照応が見つかる。(f) では、 C_2 を用いて、「がこ/の/
 審議会」から「冷夏対策審議会」への照応が、見つ
 かる。照応の確からしさの尺度を定義することを認
 めれば、照応の結果により (e) の方が (f) に優先する
 と考えられ、曖昧性解消ができる。

この例では、構文解析、意味解析、文脈解析が同
 程度に曖昧解消に必要なことが分かる。形態素解析
 から構文解析、意味解析、文脈解析、意味解析から
 構文解析、文脈解析から意味解析、構文解析という
 呼び出しが起き、各エージェントが曖昧性解消に役
 立つ優先順位を返すことになる。

4.2 構文解析の曖昧性解消の分析

例 4. 意味解析の知識の利用例

1. 原文

「丘のうえに立って考えた。」

2. 曖昧性

(g1) ((丘のうえに、立(動詞)つ(語尾)て(接続
 助詞)), 考えた。)

(g2) (丘のうえに、(立(動詞)つ(語尾)て(接続
 助詞)), 考えた。))

(h1) ((丘のうえに、立(名詞)って(格助詞)), 考
 えた。)

(h2) (丘のうえに、(立(名詞)って(格助詞)、考
 えた。))

3. 曖昧性解消に有効と思われる知識

S_5 : [考える] に関する意味関係は認める

S_6 : [リットル]is.a[抽象物]

S_7 : [考える]role.is[抽象物]

S_8 : [抽象物]locate.at[抽象物]

S_9 : [考える]locate.at[抽象物]

(g1) と (g2)、(h1) と (h2) という 2 つの曖昧性解
 消について考える。(g1)、(g2)、(h1)、(h2) の曖昧性
 解消を他のすべてのエージェントに依頼する。

形態素解析：この例では形態素解析は曖昧性解消
 に役立つ知識が想定できず、曖昧性解消には役立たな
 いと考えられる。

意味解析：(h1)、(h2) を意味解析する。(g1) は、
 W_6 、 W_7 、 W_8 、 W_9 、 D_4 、 D_5 、 S_1 、 S_2 、 S_3 により意味
 解析が成功する。(g2)、(h1)、(h2) は、 S_4 、 S_5 、 S_6 、
 S_7 、 S_8 に反するため、意味解析が失敗する。従って、
 (g1) と (g2) については、曖昧性が解消できる。(h1)
 と (h2) については、どちらも失敗する。この場合、
 意味解析の知識が十分でないとも考えられる。何ら
 かの尤度を定義して、どちらか一方を正解とするこ
 とも必要であると思われる。

文脈解析：この例では文脈解析は曖昧性解消に役
 立つ知識が想定できず、曖昧性解消には役立たないと
 考えられる。

4.3 分析のまとめ

1. 処理の流れ

例 1 ~ 例 3 から分かるように、形態素解析の
 曖昧性解消については、構文解析の知識が役立
 つ場合、意味解析の知識が役立つ場合、構文解
 析・意味解析・文脈解析すべての知識が必要な
 場合がある。個々の曖昧性がどの場合になるか
 は解析をしてみないと分からないため、すべて
 のエージェントに依頼をすることになる。依頼
 を受けたエージェントが、連鎖的にエージェン
 トを呼び出すこともあり、並列的に複数の処理
 が流れが生じる。

2. 曖昧性の増加

例2から分かるように、形態素の曖昧性解消の過程で、より多くの構文解析の曖昧性解消が発生することがある。全体として、曖昧性が爆発的に増加する可能性があると考えられる。

3. 曖昧性解消の失敗

例3で見たように、エージェント全体として曖昧性解消に十分な知識が構成できない場合に、解を一意に決定できないことがある。また、例4のように、すべての解が否定されてしまう場合がある。

4.4 考察

分析の結果から以下のようなことが分かる。

1. 曖昧性解消に対するマルチエージェントモデルの有効性

従来、ヒューリスティクスで解消せざるをえなかった曖昧性の候補を、他エージェントに処理を依頼することにより、候補から除くことができる。したがって、従来より、曖昧性の候補を減らすことができるため、精度向上が可能であると考えられる。

2. 処理の並列性

知識を利用するために、他のエージェントの解析結果が必要となることがあるため、エージェント間に様々な処理の流れが必要となる。さらに、これらの処理は、並列的な処理のメカニズムが必要となる。同一の処理が何度も重複して起こる可能性があるため、何らかの同期をとる必要もあると思われる。処理の流れは、知識の形式に依存するため、エージェントがどの知識を用いるかにより処理は変わる。

3. 曖昧性の組み合わせの問題

例2で生じたように、曖昧性を解消するために、より多くの曖昧性を扱う必要が起きることがある。曖昧性の増加は通常組み合わせ的に増加するため、例で示した処理をそのまま実行するのではなく、組み合わせの処理に関する技法を用いる必要があると考えられる。特に、本節の例では、曖昧性の数が少ないが、実際のプログラムでは、非常に多くの曖昧性が生じるため、効率のよいアルゴリズムが要求される。

4. エージェント間の競合の問題

例4では、たまたま、すべてのエージェントが同じ候補を正解として選んだが、選択が一致せず競合する場合も考えられる。このような競合を解消する方法が必要である。

5. 尤度の問題

エージェント全体の持つ知識が曖昧性解消に十分でない場合に、尤度を導入せざるを得ない可能性がある。曖昧性解消に失敗する場合、逆に、すべての候補が否定されてしまう場合が考えられる。

5 マルチエージェントモデルに基づく協調方式についての枠組み

前節の考察の結果に基づいて、エージェントモデルの協調方式を検討した。自然言語処理をマルチエージェントモデルで実現したときの問題点として、処理の並列性(考察2)、曖昧性の組み合わせ(考察3)、エージェント間の競合(考察4)、尤度の問題(考察5)といった問題が発生する。従って、協調方式としては、これらの問題に対応できるようなモデルであることが必要である。

エージェントの協調方式としては、大別して、共通に参照できる領域を持つモデルと共通の領域を持たないモデルがある[2]。前者の例として、黒板モデルがあり、自然言語処理への応用としてHearsay-II[5]がある。後者の例としては、[2]がある。

処理の並列性及び曖昧性の組み合わせの問題から分かるように、4節の分析で行なった処理をそのまま実行すると、処理量が爆発的に多くなることが予想される。また、エージェント間の競合は、処理量をさらに多くする。考察2で述べたように、処理の流れは、知識の形式に依存する。本稿では、形態素解析、構文解析といった処理の単位をエージェントとして分析したが、知識源を基本単位とした方が、処理の効率化が図れることが予想される。また、尤度は、エージェント間の競合の解決に用いることができる。したがって、知識源を基本単位として、並列的な処理を行ない、処理量を尤度を用いて減らすという枠組みが考えられる。

図4に協調方式の枠組みを示す。基本的には黒板モデルであり、知識源をエージェントとして処理を行なう。

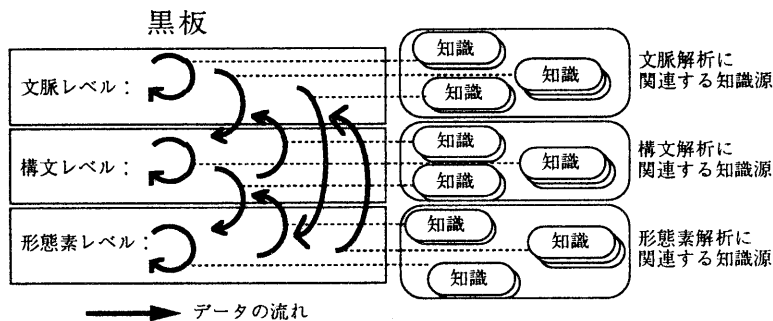


図 4: 協調方式の枠組み

6 おわりに

本稿では、曖昧性解消と利用知識の分析結果、及び、分析結果を基にしたマルチエージェントモデルにおける協調方式についての枠組みの検討結果を示した。

残された課題としては、本稿で示した枠組みの詳細化がある。また、頑健性の観点からは、曖昧性以外に対する頑健性について分析・検討することも考えている。

参考文献

- [1] 島津, 小暮, 中野, “統語・意味・語用の分散協調処理による対話文理解モデル”, 電子情報通信学会言語理解とコミュニケーション研究会 NLC94-12, pp. 39-46, 1994.
- [2] 津田, 相場, “制約と分散協調に基づく自然言語理解システム- 異種分散協調問題解決系 Helios の応用として -”, ICOT TM-1312, 1994.
- [3] 福本, 小松, 木下, “マルチエージェントによる頑健な自然言語処理方式の提案”, 電子情報通信学会人工知能研究会 AI95-9, pp. 23-28, 1995.
- [4] 松本裕治, “頑健な自然言語処理の現状と動向”, 人工知能学会全国大会, 特別セッション招待講演, pp. 75-78, 1993.
- [5] Nii, H.P., “Blackboard Systems - Blackboard Application Systems, Blackboard Systems form a Knowledge Engineering Perspective”, The AI Magazine, pp. 82-106, 1986.