

Title	確率文脈自由文法を持つエージェント間での言語共有
Author(s)	武田, 稔也
Citation	
Issue Date	2000-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1358
Rights	
Description	Supervisor:東条 敏, 情報科学研究科, 修士

修士論文

確率文脈自由文法を用いた エージェント間での言語共有

指導教官 東条 敏 助教授

北陸先端科学技術大学院大学
情報科学研究科情報処理学専攻

武田稔也

平成12年 3月 8日

要旨

言語獲得のモデル化は、認知科学における大きなテーマであり、原型言語から、進化、適応により、普段我々が使用する真の言語を獲得する過程においては自動化というスキルが存在する。本論文ではマルチエージェント環境での確率文脈自由文法 (Probabilistic Context-Free Grammar, PCFG) の文法規則の共有化を自動化のプロセスの一部と考える。ここでは、各文法規則に確率を割り振り、会話における有用な規則を重視することにより、エージェント間での文法の共有化を目指す。

本研究では語に対する意味や目的などは付加していない。しかし、同じ文法規則でも、その使われ方に応じた確率のチューニングを行うことにより、例えば日常言語における主語として使用される語 (名詞) と目的語として使用される語 (動詞) の対応関係のような構造的意味を持たせることができる。

また、実験では各プロセッサにそれぞれエージェントを割り当てた大規模シミュレーションを分散処理環境で行い、マルチエージェントモデルの実験環境における分散システムを提案する。

このような環境のもと、各エージェントが2進数のビット列を入力データとして、与えられたデータから学習を行う対話的学習モデルを作成した。エージェントは入力されたビット列を形成する文法構造を、各エージェントが持つ確率文脈自由文法の文法規則から解析し、得られた解析結果から文法規則の選択確率のチューニングを行う。そして、文法の生成確率を自己組織化し、それをもとに他のエージェントへ発話することによりフィードバックする繰り返し最適化モデルを実装した。

選択確率の自己組織化において、2つのアプローチにより各エージェント間での一般化を行った。

- 文法規則の選択確率に閾値を設定し、閾値以下の文法規則を淘汰する
- 帰納推論により正例と負例を与えることにより一般化を行う

実験では、それぞれのパラメータの設定を変えることにより、得られる結果にアルゴリズムの性質を反映した変化が確認された。実験環境としては、分散処理により各プロセッサを一つのエージェントに見立てた大規模なシミュレーションを行った。各エージェントは、割り当てられたプロセッサの処理速度に依存した発話を行うため、発話数に個体差が出現した。これにより現実社会の現象を反映したモデルになった。

目次

1	はじめに	1
1.1	目的	1
1.2	言語学的背景	2
1.3	言語現象に関する工学的研究	3
1.3.1	Mate Finding Problem	3
1.3.2	記号文法の進化	3
1.3.3	異言語間での共通文法の獲得	4
1.3.4	推論による共通文法の組織化	4
1.3.5	言語獲得モデル Rhea	5
1.3.6	本研究の立場	5
1.4	構成	6
2	実験に必要な知識	7
2.1	構文解析	7
2.1.1	CYK 法	7
2.1.2	チャート法	9
2.2	確率文法	13
2.2.1	確率文脈自由文法	14
2.2.2	確率文脈自由文法の定式化	15
2.2.3	文脈自由文法への重み付け	18
2.3	帰納推論システム	19
2.3.1	枚挙法による帰納推論アルゴリズム	19
2.3.2	被覆集合アルゴリズム	19
2.3.3	本研究への応用	20

3	文法共有化モデルの提案	21
3.1	文法の枠組	21
3.2	コミュニケーションモデル	22
3.2.1	エージェント間通信	22
3.2.2	形態素解析器	24
3.3	並列分散プログラミング	24
4	実験	28
4.1	提案モデル	28
4.2	シミュレーション条件	28
4.3	予備実験	29
4.3.1	実験条件	29
4.3.2	考察	30
4.4	実験 2	31
4.4.1	実験条件	31
4.4.2	考察	32
4.5	実験 3	35
4.5.1	実験条件	35
4.5.2	考察	36
5	結論	40
5.1	まとめ	40
5.2	今後の課題	41

目 次

1.1	本研究の自然言語におけるモデル	2
2.1	文脈自由文法	8
2.2	CYK 法の三角行列	10
2.3	“I eat pizza with Maria” に対する三角行列	10
2.4	基本チャート	11
2.5	チャートの例	12
2.6	“I eat pizza with Maria” のチャート	13
2.7	確率文脈自由文法	15
2.8	“Swat flies like ants” の構文解析	16
3.1	並列化の概念	25
3.2	分散プログラミングのタイムテーブル	27
4.1	文法規則の集合	29
4.2	語彙セット	29
4.3	淘汰パラメータの影響	31
4.4	正例と負例の影響	34
4.5	選択可能な文法規則数の推移	35
4.6	1 対 1 のモデル	36
4.7	3 対 1 のモデル	37
4.8	1 対 3 のモデル	39
4.9	2 対 2 のモデル	39

表 目 次

4.1	$S \rightarrow A_1 A_2$ における A_1 の展開	30
4.2	$A_2 \rightarrow A_7 A_1$ における A_1 の展開	33

第 1 章

はじめに

1.1 目的

言語獲得のモデル化は、認知科学における大きなテーマである。大人のピジン語、2 才以下の子供の言葉にみられる原型言語から、進化、適応により、普段我々が使用する真の言語を獲得する過程において、自動化というスキルが存在する。ここで自動化とは、発話における文法規則の選択など、試行錯誤的な注意を必要としていたパートを経験的学習により省略するプロセスを指す。Lieberman [11] によれば調音能力は複雑な調音器官のすばやい流れるようなコントロールを必要とし、そのためには自動化による注意の省略が必要だとしている。彼は同様に人間の持つ統語装置の起源においても自動化と結び付いた仮説を提唱している。

本稿ではマルチエージェント環境での確率文脈自由文法 (Probabilistic Context-Free Grammar, PCFG) [2] の共有化を自動化のプロセスの一部と考える。そして、各エージェントの持つ PCFG の文法規則の木構造に確率を与え、これをエージェント間での発話、認識という経験的学習によりチューニングし自己組織化する (図 1.1)。また、帰納推論 [3] により文法規則を新たに生成し、共有文法を組織化することも同時に行い、例えば慣用句や複合語の生成といった言語獲得過程のモデル化の実現を目標とする。

また、マルチエージェント環境での相互学習という設定であるため、学習が局所的なループに陥る危険性がある。そこで、本研究では負事例の文を設定し、正事例と負事例の文の入力から負事例を排除するような確率を学習することを目的とする。

また、この研究は言語学的な正当性に基づくものであると同時に、異なるプロトコルを持つ人工エージェント同士が柔軟に新たなプロトコルを自己組織化する複雑系のシミュレーションの側面を持つ。

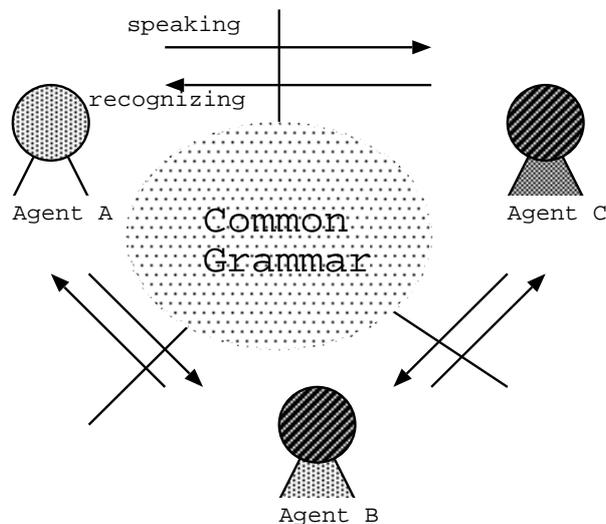


図 1.1: 本研究の自然言語におけるモデル

実装には, これまでの同様な研究とは異なり多数のプロセッサを用い, 各々のプロセッサに各エージェントを割り当てた分散環境でのモデル化を行う. そして, マルチエージェントモデルの研究における実験上のシステムの提案を行う.

1.2 言語学的背景

現在, 人間の言語獲得のメカニズムの解明は, いまだ初期の段階にすぎない. これまで, 多くの学者によって様々な理論や仮説が提唱されているが, 定説はないといっていだろう. これは, 観察による理論の構築が難しいこと, そして扱う問題が言語学の枠を越えた様々な分野が複雑に入り組んでいるということに起因している. 言語の起源についても, そのものを観察できる訳ではなく, 今まで分かっていることから理論を構築していくしかない. 現状では, 実験室で類人猿に言語を教えこむのがやっとなのである [11]. しかし, これも自然発生的な言語の起源を再現しているわけではない. また, コンピュータシミュレーションによるアプローチもなされているが, 十分な理論が構築されたとは言い難い. 冒頭で述べた通り, 仮説の構築が始まったばかりであり, 今後の研究成果が期待される分野である.

1.3 言語現象に関する工学的研究

ここでは、マルチエージェント環境における言語獲得に関するこれまでの研究について述べる。そして、これら諸研究と本研究を比較し、本研究の立場を明らかにすることを目的としている。

1.3.1 Mate Finding Problem

機械学習において管理者の存在は適切ではない。例えば自律型エージェントのようなロボットに、様々な状況に対応する振舞いについて適切な情報を管理者が与えることは難しい。やはり、必要な知識は環境との相互作用により推論されるのが望ましい。

マルチエージェント環境での自律的な協調行動の研究として、Werner と Dyer [17] は生物体のシミュレーションを行っている。これは雄と雌の求愛行動のモデル化で人工生物体の集合を自律的なエージェントに見立て、雌が発する信号を雄が試行錯誤的に理解しすることにより雌に接近し、最終的につがいを形成していくモデルである。雌が発する信号は4桁のビット列からなり、学習機構はニューラルネットを用いている。

また Ono, N. [13] は、この Mate Finding Problem を指定された作業を行う手段としての人工生物体のシミュレーションにおける根本的な問題として扱っている。基本的な問題設定は Werner と Dyer の研究と同様であるが、この問題において共通の伝達プロトコル上で並行して二つのクラシファイアシステムを学習機構として採用している。

1.3.2 記号文法の進化

Hashimoto と Ikegami [4] [5] は文法は固有のメカニズムによって複雑な空間や情報なしで進化、獲得されるものという立場で、ネットワーク・コンテキストにおける言語の進化的な学習のために、簡単な言語ゲームを行っている。互いのプレーヤはそれぞれ文法をもち、センテンスを送ることにより互いに会話を行う。プレーヤは発話とリスニングを交互に行い、どのセンテンスを発話したか、そしていかに早く認識したかで互いの優劣を決定する。そして、得点の低かったプレーヤの文法を遺伝的に得点の高い文法に置き換えることにより進化させている。

文法の階層化には、以下のチョムスキー階層 [14] を用いている。

type 0 句構造文法

type 1 文脈依存文法

type 2 文脈自由文法

type 3 正規文法

この階層構造では上位にある文法構造ほど、複雑な表現が可能である。つまり、句構造文法が最も複雑な表現に適した文法構造である事を意味している。この研究ではモデルの初期状態として正規文法が与えられている。このモデルにおいて、方言の出現、モジュール進化、そしてアルゴリズム的進化の出現が確認された。そして、正規文法しか持たないエージェントが文脈自由文法を獲得したとしている。

1.3.3 異言語間での共通文法の獲得

中村 [7] は遺伝プログラミングを用いた異言語間でのコミュニケーションのシミュレーションを行っている。これは言語学上の実際に存在する混成言語ピジンやクレオールが発生を、計算機上で実現させることを目的としている。文法表現には TAG(Tree Adjoining Grammar) を採用し、対象言語は上述の 2 つの研究で使用したような記号列ではなく、実際の自然言語、日本語と英語を用いた異言語間での実験を行っている。また、並列オブジェクト指向言語 ABCL/f によって並列計算機上で実装を行い、より現実世界に近い環境でのシミュレーションを行っている。

1.3.4 推論による共通文法の組織化

小野哲雄ら [12] は、推論機能を有するエージェント群による共通文法の組織化のモデルを提案している。彼はバイオプログラム理論 [1] に基づいたモデル化を行ったとしている。ここで、バイオプログラム理論とは、Bickerton により提唱されたもので、彼は世界中に散在するクレオール諸語には多くの類似点が見られることを指摘している。そして、これは子供が獲得すべき十分なモデルを与えられないとき、「バイオプログラム」と呼ばれる生得的な統語構造が現れるためであるとしている。小野はこのバイオプログラム理論に基づいたモデル化として、アブダクションとインダクションという二つの異なる推論機構を用いている。また、文法構造には文脈自由文法を使用し、自然言語の代表的文法規則、SVO 型、SOV 型のエージェントを設定して発話、認識を繰り返し行わせている。このとき、パラメータとしてエネルギー量を導入し、発話や認識に応じてエネルギーが変動する。そして、各エージェントはこのエネルギー量に応じた推論を行っているとしている。

1.3.5 言語獲得モデル Rhea

錦見 [9] らは言語獲得の枠組として Rhea (near Human language Acquisition) と呼ばれるモデルを作成した。言語獲得モデル Rhea は言語的入力と非言語的入力のペアを受け取り、それぞれを統語構造とフィルタという内部的な記述に変換し、それらにもとづいた入力のクラス分けを行なう。この過程は言葉の意味、構造を記述するための統語の規則、意味を分節するための概念、そして意味を構造に基づいた言葉のカテゴリの獲得とみなすことができるとしている。

また、彼女はこの枠組に、語句の意味に対する確信度の設定方法と、それを利用するための手続き、フィルタ候補を生成する順序、カテゴリを用いてフィルタ検索をする場合のカテゴリ選択の順序など、Rhea の基本モデルでは具体化されていない部分をどう決めるかによって、言語獲得に関する様々な仮説を組み込むことができるとしている。

1.3.6 本研究の立場

マルチエージェント環境で実験では、エージェント間での対話的環境で学習を行うことが可能である。錦見らの研究と比較して、他の研究は例の提示について人間が対話的に行う必要はない。本研究でもエージェント間でのコミュニケーションを前提としている。

また、言語共有の問題には、概念の共有化、文法の共有化が存在する。文法の共有化には、さらに母語の獲得を対象とするか、異言語間のコミュニケーションを対象とするかで扱う問題は異なる。本研究では、基本的には前者の立場をとる。しかし、本質的には人工エージェント同士が柔軟に通信し新たなプロトコルを自己組織化する工学的研究である。

入出力形式には、2進数の記号列を用いる。Werner, Dyer, そして Ono, N. らの研究では、記号列に階層的構造は付加されていない。Hashimoto, Ikegami の研究では階層的に記号列の生成を行っているが、生成された記号列に対する階層的評価は行っていない。本研究では確率文脈自由文法により階層的、確率的にビット列を生成する。そして、生成された記号列に対し階層的評価を行い、確率文脈自由文法の共有化を行う。

小野哲雄ら [12] は帰納推論を用いた共有文法の組織化についての研究を行っている。本研究でも推論推論を用いるが、本研究の立場はあくまでも確率文脈自由文法の共有化であり、その上での必要な共有文法の獲得を目的としている。

また、中村の研究では実験において並列計算機を使用し、各エージェントをそれぞれ一つのプロセッサに割り振った大規模なシミュレーションを行っている。これは、マルチエージェント環境における実験上のシステムの構築という観点で評価されるものである。本研究でも同様に、各プロセッサにそれぞれエージェントを割り当てた大規模なシミュレー

ションを分散処理環境で行い, マルチエージェントモデルの実験環境における大規模システムを提案する.

1.4 構成

本稿では以下の構成をとる. まず 2 章では, 本稿で提案するモデルに用いた確率文脈自由文法と帰納推論について紹介し, 3 章では 2 章で述べた知識をもとに, 本研究で提案したモデルの具体的な説明を述べる. 4 章で, 計算機上で実装し, 実験した結果とその考察を述べる. そして最後に 5 章本稿のまとめをおこなう.

第 2 章

実験に必要な知識

本章では、提案モデル作成に必要な文法表現やアルゴリズムについて述べる。まず、モデルの実験上必要なシステムとして実装した構文解析器について述べる。そして文法表現として文脈自由文法に確率を与えた確率文脈自由文法、推論機構である機能推論システムについて紹介する。

2.1 構文解析

句構造文法 G と終端記号列 w が与えられたときに、記号列 w の導出過程を明らかにし、その出力結果として w に対する導出木を導くことを構文解析という。そして、この構文解析を実行する処理系をパーサ (parser) と呼ぶ。

ここでは、いくつかの構文解析法を紹介する。

2.1.1 CYK 法

CYK法はボトムアップ解析を行う最も基本的なアルゴリズムである。CYK法では小さい部分解析木から大きい部分解析木を作る操作の繰り返しにより、最終的に文の解析木を生成するものである。また、扱われる文法表現は、チョムスキー標準形と呼ばれる文法を対象としている。チョムスキー標準形とは、その文法の生成規則が以下の生成規則のいずれかの形式で表現されるものである (図 2.1)。

$$A \rightarrow B C \quad (A, B, C \in V_N)$$

$$A \rightarrow a \quad (A \in V_N, a \in V_T)$$

(1)	S	\rightarrow	N	V
(2)	S	\rightarrow	S	PP
(3)	S	\rightarrow	V	N
(4)	V	\rightarrow	V	N
(5)	PP	\rightarrow	P	N
(6)	N	\rightarrow	N	PP
(7)	N	\rightarrow	I	
(8)	N	\rightarrow	$Maria$	
(9)	N	\rightarrow	$pizza$	
(10)	V	\rightarrow	eat	
(11)	P	\rightarrow	$with$	

図 2.1: 文脈自由文法

ここで, V_N は非終端記号の集合, V_T は終端記号の集合を表している. なお, 任意の文脈自由文法は, それと等価なチョムスキー標準形の文法に書き換えが可能である.

ここで, *CYK* アルゴリズムについての理解を促すため, 図 2.2 の三角形行列を用いることにする. 長さ N の終端記号列 $w_1 \cdots w_N$ を解析する場合には, N 行 N 列の三角行列を考える. つまり, 図 2.2 の三角行列は長さ 5 の記号列 $w_1 \cdots w_5$ を解析する場合に用いる三角行列である. なお, 三角行列の i 行 j 列の要素の表記に $T_{i,j}$ を用いる.

CYK アルゴリズムでは, $T_{i,j}$ に終端記号列 $w_i \cdots w_j$ を導出するような全ての非終端記号列をボトムアップに次々と書き込んでいく. 最初は, 三角行列の主対角線上の要素 $T_{i,i} (1 \leq i \leq N)$ を求める. これは各終端記号 w_i に対し $A \rightarrow w_i$ となる生成規則を全て探し, $T_{i,i}$ に書き込みを行うことにより求まる. 次に主対角線上の一つ上の対角要素 $T_{i,i+1} (1 \leq i \leq N-1)$ を求める. このために, $T_{i,i+1}$ の左側の要素 $T_{i,i}$ と $T_{i+1,i+1}$ を参照する. $A \rightarrow B C$ という生成規則 (B, C は非終端記号) に対して, $B \in T_{i,i}$ かつ $C \in T_{i+1,i+1}$ となる全ての A を $T_{i,i+1}$ に書き込む.

次にこの手順を一般化して, 三角行列の n 番目の対角要素 $T_{i,i+n}$ を計算する場合について考える. $B \in T_{i,i+j-1}$ かつ $C \in T_{i+j,i+n} (1 \leq j \leq n)$ とすると, 非終端記号 B から $w_i \cdots w_{i+j-1}$, C からは $w_{i+j} \cdots w_{i+n}$ が導出可能であることを意味している. ここで $A \rightarrow B C$ となる全ての A を $T_{i,i+n}$ に書き込む. これは, A から $w_i \cdots w_{i+n}$ を導出することが可能であることを意味している. このようにして $T_{i,i+1}, T_{i,i+2} \cdots$ を次々に求めていく.

最終的に, $T_{1,N}$ に開始記号 S が含まれていれば, 終端記号列 $w_1 \cdots w_N$ は S から導出される (文法的に正しい文法である) とういことになる.

例として図 2.2の文法を用いて “I eat pizza with Maria” を解析するときに作成された三角行列を図 2.3に示す. $T_{2,3}$ および $T_{2,5}$ には “eat pizza” および “eat pizza with Maria” の部分に 2 通りの解析が存在すること示している. また, $T_{1,5}$ には開始記号 S が含まれており, “I eat pizza with Maria” が構文的に正しい文であることが確認される. なお, “I eat pizza with Maria” は構文的に曖昧であるため, $T_{1,5}$ にある S は 2 通りの方法で得られている. この 2 つの方法を図 2.3に実線と破線で表す. 破線は図 2.1の生成規則 (1), 実線は生成規則 (2) にそれぞれ対応している.

$w_1 \cdots w_N$ における CYK法のアルゴリズムの手順を以下に示す.

CYK 法のアルゴリズム

1. $A \rightarrow a$ の形の生成規則を用いて, 主対角線上の要素を計算する.
2. $A \rightarrow B C$ の形の生成規則を用いて, 2 番目以降の対角線上の要素を計算する.

$$\begin{aligned} & \text{for } i = 1 \text{ to } N \\ & \quad T_{i,i} = \{A \mid A \rightarrow w_i\} \end{aligned}$$

2. $A \rightarrow B C$ の形の生成規則を用いて, 2 番目以降の対角線上の要素を計算する.

$$\begin{aligned} & \text{for } n = 1 \text{ to } N - 1 \\ & \quad \text{for } i = 1 \text{ to } N - n \\ & \quad \quad T_{i,i+n} = \bigcup_{j=1}^n \{A \mid A \rightarrow B C, B \in T_{i,i+j-1}, C \in T_{i+j,i+n}\} \end{aligned}$$

3. $S \in T_{1,N}$ であれば, $w_1 \cdots w_N$ は開始記号 S から導出可能である.

また, CYK 法により導出木を求めるためにはステップ 2 において三角行列の対角要素 $T_{i,i+n}$ に $[A, (B, i, i + j - 1), (C, i + j, i + n)]$ というように, 使用した三角行列の要素のログを書き込むようにすればよい.

2.1.2 チャート法

チャート法は解析木のグラフ表現であるチャートを逐次的に作り上げることにより文構造を解析する. 解析の流れはトップダウン, ボトムアップが選択可能であり, かつ両者を混在させることができる.

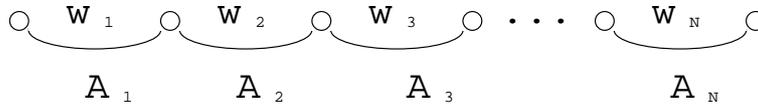


図 2.4: 基本チャート

解析に用いられるグラフはノードとノード間を結ぶエッジで構成される。それぞれのエッジにはラベルをつける。図 2.4 に文 $W = w_1 w_2 w_3 \cdots w_N$ について、エッジの非終端記号ラベルを A_i ($i = 1, 2, 3 \cdots N$) とする基本チャートを示す。今、生成規則 $A \rightarrow B C$ があるとすると、これをチャートで表現すると図 2.5(a) のようになる。ここで、エッジ A の左側のノードとエッジ B の左側のノード、エッジ B の右側のノードとエッジ C の左側のノード、そしてエッジ A の右側のノードとエッジ C の右側のノードが一致する。これは、句構造表現に基づいて新しいエッジを生成しチャートに加えるときの制約となる。また、エッジ B の内部に終端記号列 $w_i w_{i+1} \cdots w_{k-1}$ が含まれ、エッジ C の内部に終端記号列 $w_k w_{k+1} \cdots w_j$ が含まれていたならば、非終端記号 B から終端記号列 $w_i w_{i+1} \cdots w_{k-1}$ が導出可能で、非終端記号 C から終端記号列 $w_k w_{k+1} \cdots w_j$ が導出可能であることを示している。そして、非終端記号 A から文 $w_i w_{i+1} \cdots w_k w_{k+1} \cdots w_j$ が導出可能であることを示している。

基本チャートの生成は、まず文 W の終端記号列 w_i の両端とその記号間にノードを生成した原始チャートを作り、各終端記号 w_i を含む生成規則 $A_i \rightarrow w_i$ から、 w_i に対するエッジ A_i を原始チャートに加えていくことにより行われる。従って、文脈自由文法における生成規則から各非終端記号をラベルとするエッジを上述の手順で原始チャートに加えていけば、解析木を表すチャートが出来上がる。ただし、各終端記号を導く非終端記号は 1 つずつとしている。

終端記号を導く非終端記号が複数個ある場合でも、生成規則から作られる全てのエッジを加えていくことにより、全ての解析木を 1 つのチャートとして書くことができる。

チャート法による解析は、基本チャートに生成規則から作られるエッジを加えていくことにより行われるが、その途中において部分解析木の一部がまだ作られていないエッジが存在する。この途中状態のエッジを活性エッジと呼ぶ。これに対し、生成規則の右辺の各非終端記号に対応するエッジが全て作られているエッジを非活性エッジと呼ぶ。従って、活性エッジは非活性エッジから構成される部分解析木を作るときに一時的に現れるエッジである。活性エッジの例を図 2.5(b) に示す。

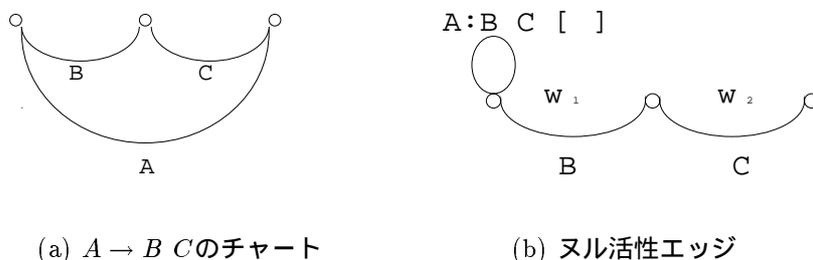


図 2.5: チャートの例

ここで, $A: B C []$ とラベルされたエッジが活性エッジである. これは, 非終端記号 A を左辺に持つ生成規則を意味しており, 右辺については, 左から順に解析された非終端記号を “[]” 内に移動していく. これを左辺の非終端記号が全て “[]” 内に格納されるまで行う. 図 2.5(b)では生成規則 $A \rightarrow B C$ において A も B も解析されていないことを意味している. このような活性エッジを特にヌル活性エッジと呼ぶ.

このように, チャート法による解析は, 句構造文法の生成規則に基づいて, チャートにヌル活性エッジを与え, その活性エッジが示す非終端記号について順番に解析することにより解析木が生成される. 以下にチャート法のアルゴリズムを示す.

チャート法のアルゴリズム

1. 任意のノードについて文の生成規則に対応するヌル活性エッジを加える.
2. 活性エッジに対して, 未処理エッジを解析し, 既処理エッジとする. 元の活性エッジを新しい活性エッジに置き換える. 未処理エッジがなくなるまでこれを繰り返す. 未処理エッジがなくなったとき, その活性エッジを非活性エッジに置き換える.
3. 新しい非活性エッジが作られなくなるまで 1. および 2. を繰り返す.

ここで, 1. で加えた新しい活性エッジに対して 2. の解析が失敗したときは, その活性エッジを消去する. また, 1. で一度加えたヌル活性エッジは同じ場所には 2 度と加えないという制約を加える. これにより, 再帰型規則や回帰型規則による無限ループは防がれる.

このアルゴリズムでは, ヌル活性エッジを加える順番は任意である. すなわち, どのような順序でヌル活性エッジを加えていっても, 最終的には同じチャートが出来上がる. つ

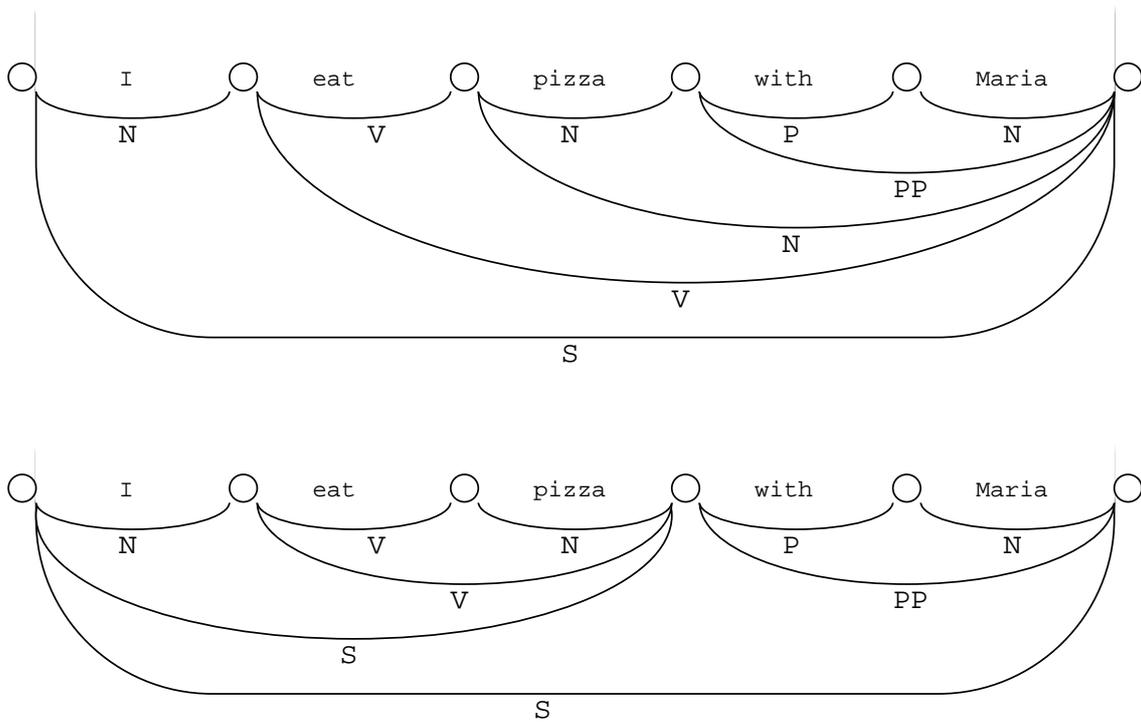


図 2.6: “I eat pizza with Maria” のチャート

まり, 又ル活性エッジの与え方によってトップダウン解析もボトムアップ解析も行うことが可能である.

原始チャートに対して, その最も左にあるノードに開始記号 S を左辺に持つ生成規則に対応する又ル活性エッジを加えることから解析を開始すると, トップダウン解析が行われる. 図 2.6 にチャート法による解析例を示す.

2.2 確率文法

本章では, 確率を用いた代表的言語モデルである確率文脈自由文法について述べる. 確率文脈自由文法が対象とする主な問題は以下のとおりである.

- 1) 確率文脈自由文法から文 W が生成される確率を求める問題

2) 確率文脈自由文法から文 W に対する最尤な導出を求める問題

3) 確率文脈自由文法の推定 (重み付け) の問題

ここでは、確率文脈自由文法についての基本的な説明をしたのち、本研究で扱う 3) の問題に対するアルゴリズムを紹介する。

2.2.1 確率文脈自由文法

英語の文 “Time flies like an arrow.” には複数の解釈が存在する。これは、各単語が複数の品詞および意味を持っているためである。一般には「光陰矢のごとし」ということわざのような解釈が行われるが、「時間蠅は矢が好きである」と言う風にも解釈することが可能である。これは、“flies” を一般の解釈では動詞として扱うところを名詞 (蠅) と解釈し、“like” を前置詞としてではなく動詞 (好む) として扱ったためである。このように、英語には 1 つの単語に対し異なった複数の品詞があるという多品詞語が多く、正しい品詞としてどれを選ぶかという処理が形態素解析においては重要である。

この問題に対する確率手法として N グラムモデルや隠れマルコフモデルなどが存在するが、これらは自然言語の統語的 (文法的) な構造を明示的にモデルの中に取り入れてはならず、文の階層的な内部構造が必要な問題に対しては適さなかった。

自然言語の統語構造を記述するための枠組としては文脈自由文法が広く用いられている。文脈自由文法では文の階層的な構造を記述するために、 $A \rightarrow \alpha$ という生成規則を用いる。ここで、 A は非終端記号、 α は非終端記号と終端記号の列を表す。これは左辺の非終端記号 A が右辺の文法記号列 α に書き換えられることを意味している。このとき、 A が α に書き換えられる条件つき確率 $P(\alpha|A)$ を考える事により、文脈自由文法に確率的要素を持たせることができる。これを確率文脈自由文法 (Probabilistic Context-Free Grammar, PCFG) と呼ぶ。なお、 $A \rightarrow \alpha$ に与えられる確率は、左辺に同じ非終端記号 A をもつ生成規則を足し合わせて 1 になるように与えられる。すなわち、

$$\sum_{\alpha} P(\alpha|A) = 1$$

である。確率文脈自由文法の例を図 2.7 に示す。ここで、右端の数字はその生成規則の条件つき確率を表している。生成規則 (1) と (2) の左辺はともに S であるので、その合計は 1.0 になっている。同様に各非終端記号の生成規則についても成り立っている。

図 2.7 の文法では文 $W = \text{“Swat flies like ants”}$ に対して図 2.8 のように 2 つの構文が存在する。文 W に対する構文をそれぞれ T_1, T_2 としたとき、この T_1, T_2 の確率は、以下のよう

(1) $S \rightarrow NP VP$	[0.8]	(11) $PRE \rightarrow like$	[1.0]
(2) $S \rightarrow S PP$	[0.2]	(12) $V \rightarrow swat$	[0.2]
(3) $NP \rightarrow N$	[0.4]	(13) $V \rightarrow flies$	[0.4]
(4) $NP \rightarrow N PP$	[0.4]	(14) $V \rightarrow like$	[0.4]
(5) $NP \rightarrow N NP$	[0.2]	(15) $N \rightarrow swat$	[0.05]
(6) $VP \rightarrow V$	[0.3]	(16) $N \rightarrow flies$	[0.45]
(7) $VP \rightarrow V NP$	[0.3]	(17) $N \rightarrow ants$	[0.5]
(8) $VP \rightarrow V PP$	[0.2]		
(9) $VP \rightarrow V NP PP$	[0.2]		
(10) $PP \rightarrow PRE NP$	[1.0]		

図 2.7: 確率文脈自由文法

に示されるそれぞれの導出に用いられた生成規則の確率の積として求めることができる。

$$\begin{aligned}
 P(W, T_1) &= 0.8 \times 0.4 \times 0.05 \times 0.2 \times 0.4 \times 1.0 \times 1.0 \times 0.4 \times 0.5 \\
 &= 2.6 \times 10^{-4} \\
 P(W, T_2) &= 0.8 \times 0.2 \times 0.4 \times 0.05 \times 0.45 \times 0.3 \times 0.4 \times 0.4 \times 0.5 \\
 &= 3.5 \times 10^{-5}
 \end{aligned}$$

一般に自然言語の文法において多くの構文の曖昧性が存在する。例で示したような1つの文に対し複数の導出が得られる場合において、この確率を用いることによって、より確からしい構文を選択することが可能である。上の例では T_1 よりも T_2 の方が確からしいことを示している。

また、文 W の持つ確率は、生成可能な構文の確率の和で与えられる。

$$P(W) = P(W, T_1) + P(W, T_2) = 2.95 \times 10^{-4}$$

一般に自然言語の文法は多くの曖昧性を含んでおり、1つの文に対し複数の導出が得られる場合があるが、導出確率を用いることにより、より確からしい導出を選ぶことができる。

2.2.2 確率文脈自由文法の定式化

文脈自由文法の各生成規則に、その規則の適用される確率を付与したものが確率文脈自由文法であり、形式的には4つの記号の組 $G = (V_N, V_T, P, S)$ で定義される。各記号の意

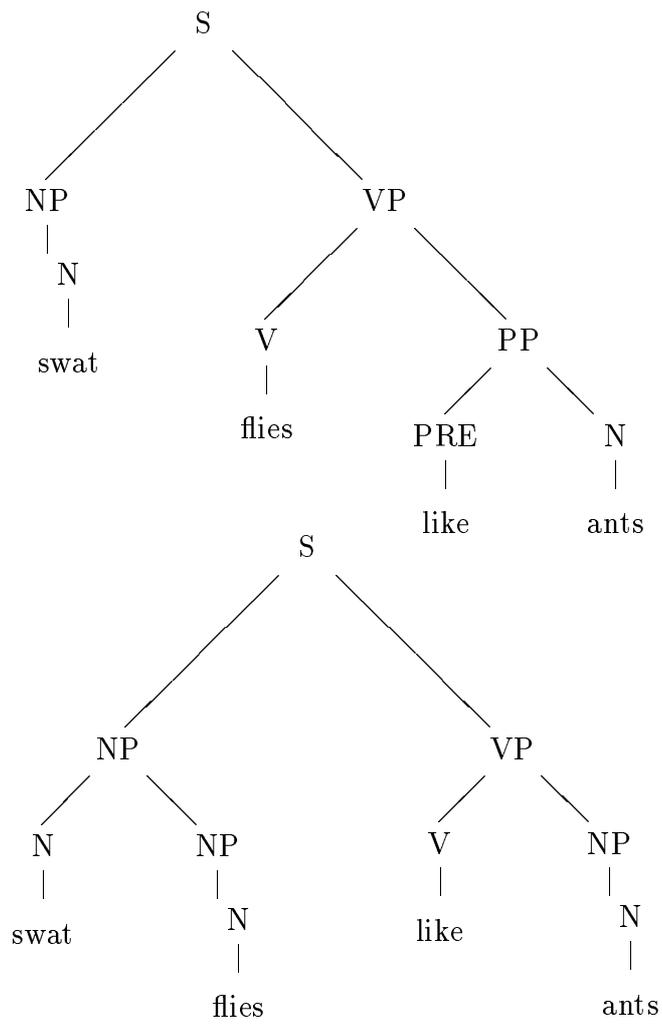


図 2.8: “Swat flies like ants” の構文解析

味は以下の通りである.

- 1) V_N : 非終端記号の集合
- 2) V_T : 終端記号の集合
- 3) P : 確率つき生成規則の集合
- 4) S : 開始記号

ここで, 確率つき生成規則は $\langle A \rightarrow \alpha, p \rangle$ の形をしている. ただし, $A \in N_V$, $\alpha \in (V_N \cup V_T)^*$ である.

いま, 文に対する最左導出において, n 番目に適用された生成規則を確率変数 X_n で表すことにする. 確率文脈自由文法では, 生成規則の適用確率は以前に適用された規則には依存しないと仮定している. すなわち,

$$P(X_n = w_n | X_1 = w_1, \dots, X_{n-1} = w_{n-1}) = P(w_n)$$

したがって, 文 W に対する最左導出 $T = w_1, \dots, w_m$ の確率は次のように与えられる.

$$\begin{aligned} P(W, T) &= P(X_1 = w_1, \dots, X_m = w_m) \\ &= \prod_{i=1}^m P(X_i = w_i | X_1 = w_1, \dots, X_{i-1} = w_{i-1}) \\ &= \prod_{i=1}^m P(w_i) \end{aligned}$$

また, 文 W の確率は, 文 W に対する全ての導出 T の確率の和として与えられる.

$$P(W) = \sum_T P(W, T)$$

なお, 確率文脈自由文法から導出される全ての文の確率値の合計は 1 になるとは限らない. 簡単な例では, 文法 $G = \{\langle S \rightarrow a, p \rangle, \langle S \rightarrow S S, q \rangle\}$ に対しては,

$$\sum_{i=1}^{\infty} P(a^i) = \min[1, p/q]$$

であることが知られている. 文法から導出される全ての文の確率値の合計が 1 になるならば, その文法は整合的 (consistent) であると呼ばれる.

2.2.3 文脈自由文法への重み付け

学習データから文脈自由文法の生成規則の確率を推定する方法について説明する。文法が曖昧でないときには、学習データ中の各入力文に対して一意的に導出が決定できるので、導出に用いられた各生成規則の適用回数を数えあげることにより規則の適用確率を推定することができる。いま、学習データ中のすべての入力文に対し、生成規則 $A \rightarrow \alpha$ が適用された回数を $C(A \rightarrow \alpha)$ とすると、生成規則 $A \rightarrow \alpha$ の適用確率は以下ようになる。

$$P(A \rightarrow \alpha) = \frac{C(A \rightarrow \alpha)}{\sum_{\beta} C(A \rightarrow \beta)}$$

しかし、文法が曖昧である時には、単純に生成規則の適用回数を数え上げるだけではうまくいかない。単純に数え上げるだけだと曖昧な文ほど適用確率の決定に大きな影響を与えることになってしまう。

一般に、確率文脈自由文法の学習では、学習データ中の各入力文に対する真の導出過程が観測できない。このような場合、繰り返し最適化の手法を用いて生成規則の確率を推定する方法が有効である。繰り返しの各段階では、更新前の確率値を用いて、ある特定の導出中に使われた生成規則の適用回数をその導出の相対確率で重み付けをすることにより、真の適用回数を推定する。いま、文 W に対する導出 T 中に生成規則 $A \rightarrow \alpha$ が使われた回数を $C(A \rightarrow \alpha; T, W)$ で表すことにする。このとき、生成規則 $A \rightarrow \alpha$ の真の適用回数は、導出 T の相対確率 $P(W, T)/P(W)$ を考慮して、次のように推定する。

$$C(A \rightarrow \alpha) = \sum_T \frac{P(W, T)}{P(W)} C(A \rightarrow \alpha; T, W)$$

繰り返し最適化による生成規則の適用確率の推定アルゴリズムを以下に示す。

生成規則の適用確率の推定アルゴリズム

1. 各生成規則 $A \rightarrow \alpha$ に適当な初期確率 $P(A \rightarrow \alpha)$ を与える。
2. 学習データ中の各文 W を解析し、 W の生成において $A \rightarrow \alpha$ が適用された回数 $C(A \rightarrow \alpha)$ を推定する。

$$C(A \rightarrow \alpha) = \sum_T \frac{P(W, T)}{P(W)} C(A \rightarrow \alpha; T, W)$$

3. 左辺に同じ非終端記号を持つ生成規則すべての確率を足し合わせると 1 になるように、生成規則を正規化する。

$$\bar{P}(A \rightarrow \alpha) = \frac{C(A \rightarrow \alpha)}{\sum_{\beta} C(A \rightarrow \beta)}$$

4. 新たに推定された確率 $\bar{P}(A \rightarrow \alpha)$ を用いて, 2. から繰り返す.

2.3 帰納推論システム

概念学習をコンピュータのプログラムに行わせるためには, その形式化が必要である. 概念学習の枠組みとして, 例の一般化に基づく推論を行う帰納推論 [5] の枠組みを用いる. 直感的にいえば, 帰納推論とは演繹推論を逆転させることによって可能となる. すなわち, 演繹推論の原型を二つの前提から (例えば三段論法によって) 帰結を導く行為であるとするならば, 帰納推論は前提の中の一つと帰結を与えて, もう一つの前提を導く行為に基づいていると考えられている. 以下で帰納推論システムについて紹介する.

2.3.1 枚挙法による帰納推論アルゴリズム

枚挙法による帰納推論アルゴリズム [3] は, 可能なすべての仮説 (推測) T_0, T_1, T_2, \dots を一つずつ提示し, 全ての事例を説明できる仮説 T を探索するものである. つまり, 新たな事例が観測されるたびに, それを説明できる一つの仮説を生成しようというものである. しかし, 枚挙法では仮説を一つしか生成できないため, 学習できるモデルは単純なものに限られてしまう. これに対して被覆集合アルゴリズムは一つの仮説で全ての例を説明しようとするのではなく, 複数の仮説 (節) で例を説明しようというアルゴリズムである.

2.3.2 被覆集合アルゴリズム

被覆集合アルゴリズム [3] では, 与えられた背景知識 (仮説空間) と正例から例の一般化を試みる. つまり, ある正例から得られた仮説候補について, その無矛盾性, すなわちそれが負例を被覆していないことを調べ, さらにその仮説候補が弱過ぎないか検討を行う. ここで, 推測が弱過ぎるとは, 特殊化により, 本來說明されなくてはいけない例も, 説明されない状態をいう. もし, 新たな仮説候補を含む仮説の記述長に対して, 被覆する正例が少ない場合には, 学習の効果が得られているとはいえないので, このような仮説候補は採用されない. 以上の条件を満たした仮説候補, つまり説明力の強い仮説が見つかったら, これを解 (仮説集合) に加え, それによって被覆される正例を元の正例集合から取り除き, 残された正例集合に対して, 同様の処理を繰り返す. これは正例集合が空になるまで繰り返される.

2.3.3 本研究への応用

機械学習の実現においては、その学習対象となる例や抽象概念の表現が必要となる。一般に帰納推論システムにおいては、この表現方法として一階述語論理を用いられる。しかし、本稿での学習対象は文脈自由文法の文法規則であるので、表現方法として一階述語論理は使用しない。

また、推論の方法については代表的なアルゴリズムを上で2つ挙げたが、本稿では、被覆集合アルゴリズムの立場をとる。得られた仮説が過度に一般化されることを防ぐ方法として、先に述べたように負例集合を被覆しないという条件がある。本モデルでは、先に述べた確率文脈自由文法を用いた方法の2つのアプローチにより一般化を行う。

例の与え方については予め正例集合と負例集合を与えるのではなく、正例についてはエージェント間のコミュニケーションにより逐次正例集合に加えていく。ここで、文脈自由文法の文法規則で与えられた例をどのように正例と負例に振り分けるかについては、考察の必要がある。

例の与え方と仮説の提示方法、すなわちユーザインタフェースについては、以下の2種類がある。

対話的システム システムが利用者に積極的に例を要求したり、仮説を提示して正しさの検証を求めたりしながら仮説を洗練していく

経験的システム システムは利用者に積極的に例や仮説の検証を要求しない

本モデルは、システムが利用者に例や仮説の検証を要求しないという点では、経験的システムと呼ぶことができるが、実際に例を与えるのはエージェント間でのコミュニケーションであり、仮説の検証を行うのもまた、エージェント間でのコミュニケーションによるものである。このため、本モデルは「仮想的な対話的システム」と呼ぶことが出来るだろう。

第 3 章

文法共有化モデルの提案

3.1 文法の枠組

各エージェントは異なる文法規則を持ち、他のエージェントとのコミュニケーションから、帰納推論により新しい文法規則を生成する。ここでは、エージェントの持つ文法について考察する。

- 仮定 1

エージェントの有する文法規則は文脈自由文法の形態をとる。

仮定 1 は、自然言語の文法理論の多くが句構造を基にした文脈自由文法を基礎にしているためである。また、文脈自由文法の終端記号である語については、最初に設定しておく必要がある。ただし、ここでは自然言語で用いられている語を取り扱う必要はない。便宜上、本システムでは 2 進数で表された数字列を一つの語と見なす。

- 仮定 2

語彙は異なる文法規則を有するエージェント間においても共有されているものとする。

本モデルを異言語間でのコミュニケーションのモデルとして考えると、仮定 2 は奇妙に感じるかも知れない。しかし、現実のコミュニケーションにおいては、単語の意味はボディランゲージや指し示しなどを用いればある程度伝達可能であるし、各エージェントが辞書を保有していたと考えれば問題ないであろう。また、本稿での目的は文法の共有化であるため、簡単のためこのような仮定を行う。

次に、相互理解について考察した場合、エージェント間で互いの言語が理解されたことを主張するためには、各エージェントの持っていた意志と目的達成のための語用論的考察

が本来必要である。しかし、意味の表示や意思・目的の表現を十分に準備することは現実的に困難である。しかし、仮定 2 にあるように文法の共有化を目標とするなら、その枠内で最低限コミュニケーションが成立する条件を定義することは、十分な意義を持つものである。本稿では以下の立場をとる。

- 仮定 3
エージェントの発話は、エージェントの持つ文法規則から確率的に選択されるものとする。
- 仮定 4
エージェントの発話は他のエージェントの持つ文法規則により構文解析できたとき、理解されたとする。

本モデルのエージェントが初期状態において持つ文法は、文に相当するトップカテゴリーを S 、日常言語における NP 、 VP 、 Adv などの品詞に相当するカテゴリを A_i で表す。各エージェントからは他のエージェントが持っている文法セットを直接見ることはできず、それによって生成される文を交換するのみである。この過程において、エージェントは他のエージェントの発話を理解できるように新しい文法規則を推論し、それを自分の文法セットに追加する。

本モデルでは、エージェントが持つことのできる文法の最大数は、パラメータとして設定される。仮定 1、仮定 3 にあるようにモデルは文脈自由文法の文法規則に重みをつけた確率文脈自由文法である。使用頻度の低い文法規則や、他のエージェントに認識されなかった文法規則に対する重みは減少していき、閾値以下になると不要な規則として捨てられる。このようにエージェント内部では、生得的な機構により発現した文法が変化して、後で出来る部分に順応していく。

3.2 コミュニケーションモデル

3.2.1 エージェント間通信

エージェントの発話は基本的に他のすべてのエージェントに対して行われる。本来、任意のエージェントが他の任意のエージェントにランダムに話しかけられるよう設定することが理想であるが、本モデルにおいても十分長い時間をとれば、この本来の意図と同等の効果を期待できるため、便宜上このように設定する。エージェントの発話に使用する文法規則は、それぞれのエージェントの持つ確率文脈自由文法により決定される。

確率文脈自由文法を次の4つの組 $\langle V_T, V_N, P, S \rangle$ で定式化する。ここで、 V_T は終端記号の集合 $\{w_1, \dots, w_n\}$ 、 V_N は非終端記号の集合 $\{A_1, \dots, A_\nu\}$ である。ここで、 A_1 は開始記号であり特に S で表す。P は生成規則で、 α_j を任意の終端記号と非終端記号の列とすると $A_i \rightarrow \alpha_j$ と表される。これらの生成規則にはそれぞれ確率 $P(A_i \rightarrow \alpha_j)$ が存在する。これは非終端記号 A_i を展開する際、全ての A_i に対する文法規則の中から α_j を選択する確率である。2.2 でも述べたように、 $A_i \rightarrow \alpha_j$ において、 $\sum_j P(A_i \rightarrow \alpha_j) = 1$ である。各エージェントは、各状態 A_i で、この選択確率にしたがって文を生成していく。

エージェントは、発話された文字列に対し形態素解析を行う。この段階では、各エージェントの持つ文法規則の確率をもとにした形態素解析は行わず、試行錯誤的に繰り返される。そして、各エージェントの持つ文法規則に従って構文解析を行う。これに成功する、つまりコミュニケーションに成功した場合、これを入力文として繰り返し最適化を行い、エージェントの持つ文法規則の選択確率 $P(A_i \rightarrow \alpha_j)$ を自己組織化してゆく。

また、コミュニケーションに失敗した場合、発話したエージェントの使用した文法規則の確率を減らし、聞き手側のエージェントは、それを満たす文法規則を帰納推論の枠組みで新たに生成する。

ただし、繰り返し最適化においては2.2.3で紹介したアルゴリズムとは多少異なる。2.2.3のアルゴリズムでは、文 W に対する導出 T 中の生成規則 $A \rightarrow \alpha$ が使われた回数 $C(A \rightarrow \alpha; T, W)$ に、導出 T の相対確率 $P(W, T)/P(W)$ を考慮して、次のように定義している。

$$C(A \rightarrow \alpha) = \sum_T \frac{P(W, T)}{P(W)} C(A \rightarrow \alpha; T, W)$$

$$P(A \rightarrow \alpha) = \frac{C(A \rightarrow \alpha)}{\sum_\beta C(A \rightarrow \beta)}$$

これに対し、本モデルでは文 W に対する導出 T 中において、それぞれの確率を独立に計算している。つまり、

$$P(A \rightarrow \alpha; T, W) = \frac{C(A \rightarrow \alpha; T, W)}{\sum_\beta C(A \rightarrow \beta; T, W)}$$

となる。このような定義は、本研究が文の解析を目的としたものではなく、エージェント間でのコミュニケーションを目的としているためである。本研究では、語に対する意味や目的達成のための表現などは付加していない。しかし、このような定義をすることによって、同じ文法規則でも、その使われ方に応じた確率が形成されることになる。つまり、例えば日常言語における主語として使用される語（名詞）と目的語として使用される語（動詞）の対応関係のような構造的意味を持たせることができる。

エージェントの保有できる文法規則の最大数は確率文脈自由文法に依存する。つまり、コミュニケーションに失敗し選択される確率の低くなった文法規則は淘汰され、使用頻度の高い文法規則のみが保持される。また、自らが保持していない全ての文法規則を帰納推論によって生成すると過度に一般化されてしまい、文法の共有化とは言えない。そこで、負例によって過度の一般化を防ぐ必要がある。しかし、言語を取り扱うというモデルの性質上、正例と負例の判断基準を定めるのが困難である。そこで本稿では、いくつかのエージェントを教師エージェントとした、一般の教師つき学習モデルとして取り扱う。

3.2.2 形態素解析器

エージェント間コミュニケーションに用いる文は、2進数のビット列で表される。聞き手のエージェントは、発話された文に使用された文法規則が直接見えないだけでなく、どのような語の並びであるかについての情報も与えられない。このため、各エージェントは入力された文に対して、まず、どのような語列であるかを解析する必要がある。

一般に、入力された文に対する語の組み合わせは複数存在する。しかし、ここではそれらについての評価は行わない。つまり、ここでの役割は構文解析の前処理であり、文節などを考慮した一般の形態素解析とは異なることに注意されたい。

また、構文解析には、トップダウンのチャートパーサを用いる。

3.3 並列分散プログラミング

本モデルにおける実験環境について検討を行う。分散処理を行うことにより、得られる利益として処理速度の向上を考えるのは、ごく当然のことである。しかし、本モデルにおける分散処理の目的は、計算速度の向上よりも、各プロセッサを一つのエージェントに見立てた大規模なシミュレーションにある。純粹に計算速度の向上を目指す立場から見れば、資源の無駄遣いに思えるだろう。しかし、マルチエージェントモデルを用いた言語共有において大規模なシミュレーションを行うことは、実際の言語活動により近づくという意味では、非常に意義のあるものである。

本モデルのシミュレーションにおける分散プログラミングへの要求は以下の通りである(図 3.1)。

- (1) 各プロセッサでそれぞれのエージェントの動作を模倣する
- (2) エージェント間のコミュニティとしてプロセッサ間の通信を行う

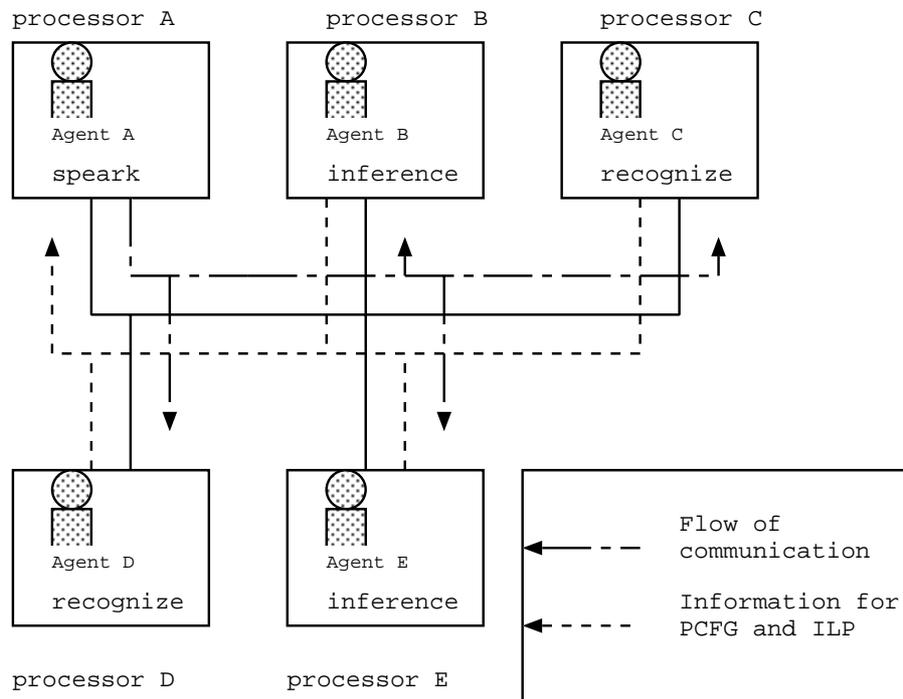


図 3.1: 並列化の概念

(3) 各プロセッサの発話数はプロセッサの処理速度に依存する

現実の社会では、人と人との会話において、その発話頻度は同等ではない。集団の数が増えれば増えるほど、よく喋る人とそうでない人が決ってくる。当然、あまり喋らない人よりもよく喋る人のほうが与える影響力は大きい。エージェントアプローチにおいても、このような現象を考慮することは意義のあるものである。つまり、(3) は本モデルがより実際社会に近いシミュレーションを行う上で有効なものと考えられる。

(3) は分散処理の特性を利用することにより実現可能である。図 3.2 に分散プログラミングのタイムテーブルを示す。ここで、図中の S, W, R は、プロセッサ A における処理状態を示している。各記号の処理状態は以下の通りである。ただし、図中の記号 A, B, C はエージェントを表している。

S : 入力, または出力の開始 (Start)

W : 待機状態 (Wait)

R : 処理の繰り返し (Repeat)

分散処理の実行管理はフラグのチェックによって行われる。各エージェントは文の入力を行う Input 処理を完了したら、実行完了のフラグ (Input Flag) を立てる。各エージェントはこのフラグをチェックし、全てのエージェントのフラグが立てられるまで次の処理を行わない。同様に Speak 処理, Output 処理の各セクションでフラグによる実行制御を行う。ここで、図中の “Lock Flag” は、文の出力処理を行う際に、ファイルの多重アクセス回避のためのフラグである。

プログラムの主な動作は Input, Parse, Speak, Output である。このうちファイルアクセスを行っているのは、Input と Output である。制約として各段階において、各エージェントは全てのエージェントがその処理を終えるまで、つまり各段階におけるフラグが全てのエージェントによって立てられるまで、そのままの状態に待機することとする。これにより、ファイルの多重アクセスによるデッドロックを防いでいる。ただし、Speak においてはファイルアクセスを行っていないので、待機状態中に繰り返し Speak 処理を行うことが可能である。

図 3.2 では、全てのエージェントが入力を行った後、つまり、全てのエージェントが Input Flag を立てた後に、次の構文解析 (Parse)、文生成 (Speak) を行っている。この段階で、エージェント B は Speak 処理に時間がかかっている。このため、エージェント A はここで 2 度繰り返し処理 (R) を行っている。エージェント B の Speak 処理終了後、エージェント A は出力 (Output) の開始状態 (S) に入るが、これよりも先にエージェント B が出力処理を開始したため、待機状態 (W) に入っている。この図では、エージェント B が 1 度 Speak 処理を行う間に、エージェント C は 5 度の Speak 処理を行っている。

このように、処理速度の早いプロセッサが次の処理に移るまでの待機状態を利用して、繰り返し Speak 処理を行うことにより、エージェント間の発話頻度にばらつきを持たせている。

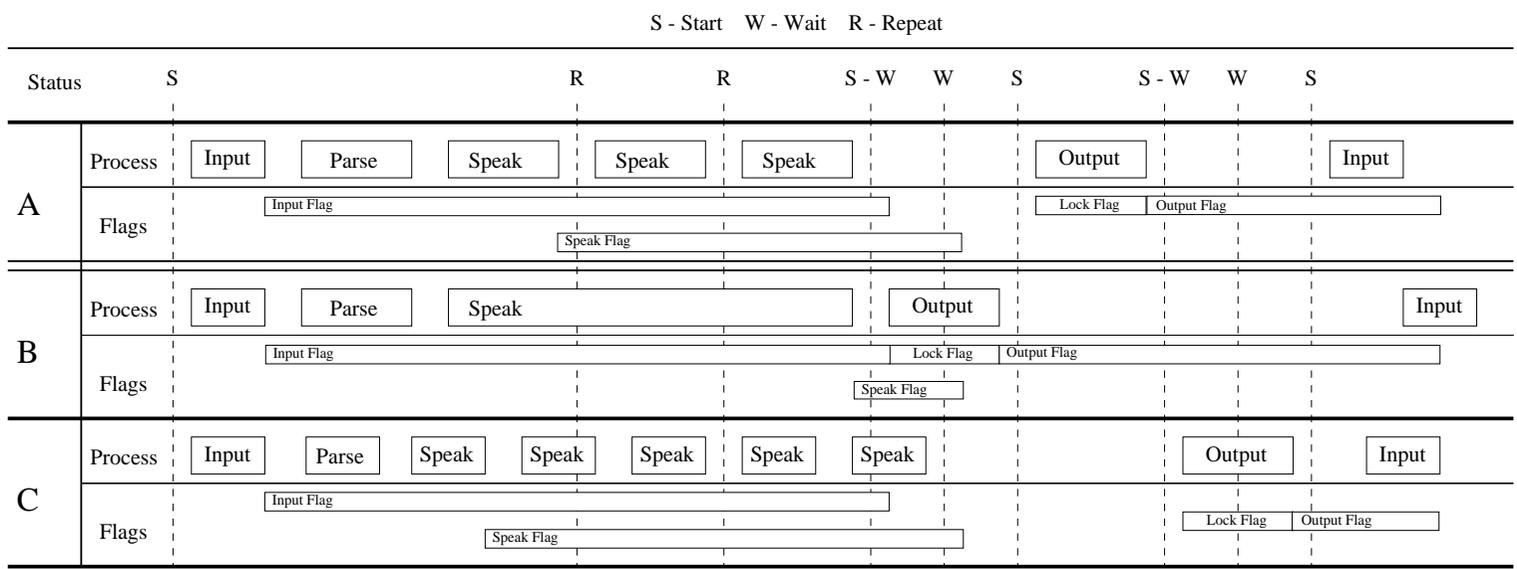


図 3.2: 分散プログラミングのタイムテーブル

第 4 章

実験

マルチエージェント環境における文法共有化の提案モデルの実装を行った。本章では実装したモデルの紹介を行い、モデルのシミュレーションにおけるパラメータ設定等の実験条件について説明した後、実験結果についての考察を行う。

4.1 提案モデル

各エージェントの持つ文法の枠組みは文脈自由文法の枠組みを用いる。語彙は 2 進数の数字列で定義し、会話に使用される文法規則や語彙セットは全てのエージェントにおいて共有されているものとする。また、エージェントの発する文の生成には、各々のエージェントが有する文脈自由文法の各文法規則に割り振られた確率に依存する。この選択確率は、同一カテゴリから展開される文法規則に与えられた重みパラメータに基づいて計算される。本研究では、エージェント間でのコミュニケーションによる経験的学習により各文法規則の選択確率のチューニングを行う。また、帰納推論の考えを用いた負例の設定を行い、文法セットの共有化を目指す。

実験では、各エージェントをそれぞれ一つのプロセッサに割り振りつつ大規模な分散実験を行った。

4.2 シミュレーション条件

使用した文法規則と語彙セットを図 4.1, 4.2 に示す。文の生成には表の文法規則に基づいて行われる。

また、これらの文法規則に割り振られた選択確率には閾値を設定し、使用頻度の低い文

$S \rightarrow A_1 A_2$	$A_3 \rightarrow A_9 A_1$
$A_1 \rightarrow A_4 A_8$	$A_2 \rightarrow A_{10} A_7 A_1$
$A_1 \rightarrow A_5 A_8$	$A_2 \rightarrow A_7 A_1$
$A_1 \rightarrow A_6 A_8$	$A_2 \rightarrow A_7 A_3$
$A_1 \rightarrow A_8$	

図 4.1: 文法規則の集合

000 A_8	0001 A_{10}	1001 A_4
001 A_8	0010 A_{10}	1010 $A_8 A_4$
011 A_5	0011 $A_8 A_7$	1011 $A_8 A_4$
100 A_9	0100 $A_7 A_{10}$	1100 A_8
101 A_8	0101 A_7	1101 A_4
110 A_4	0110 A_7	1110 $A_8 A_7$
111 A_8	0111 A_4	1111 A_9
0000 A_8	1000 A_9	

図 4.2: 語彙セット

法規則については淘汰を行う。各エージェントは1ステップに一定の文を生成し、他の全てのエージェントに発話を行い、聞き手側のエージェントはそれについて構文解析を行う。

4.3 予備実験

4.3.1 実験条件

ここでは、2エージェントによる単純コミュニケーションのシミュレーションを行う。このモデルは提案モデルの基本モデルで各エージェントは初期値としてランダムに生成された選択確率を有するものとする。実験条件を以下に示す。

文法規則	初期設定		30 Step		50 Step	
	agent A	agent B	agent A	agent B	agent A	agent B
$A_1 \rightarrow A_4 A_8$	0.22	0.22	0.08	0.09	0.07	0.09
$A_1 \rightarrow A_5 A_8$	0.33	0.18	0.37	0.39	0.42	0.44
$A_1 \rightarrow A_6 A_8$	0.36	0.22	0.10	0.14	0.09	0.11
$A_1 \rightarrow A_8$	0.09	0.38	0.45	0.38	0.42	0.37

表 4.1: $S \rightarrow A_1 A_2$ における A_1 の展開

予備実験

エージェント数 2
 1ステップの発話数 10 文
 ステップ数 1000 step

4.3.2 考察

予備実験についての考察を行う。文法規則 $S \rightarrow A_1 A_2$ の非終端記号 A_1 の展開における選択確率のステップ毎の推移を表 4.1に示す。

表 4.1より 30 ステップ後にはエージェント A, B の選択確率はほぼ同様の数値を示しているのが確認される。この実験より、文法規則の選択確率をエージェント間のコミュニケーションにより共有化することができた。しかし、選択可能な文法規則は約 90 個存在し、その組み合わせは膨大な数になり、確率のチューニングだけでは文法の共有化とは言い難い。

そこで、モデル 1 に淘汰パラメータを新たに導入して実験を行った。ここで、淘汰パラメータとは $x = (0, 1]$ の任意の実数で、発話総数とこの淘汰パラメータとの積を閾値として設定する。これにより、各文法規則において、その使用回数が閾値以下のものは淘汰される。

図 4.3に淘汰パラメータ値 0.008 における選択可能な文法規則数の推移を記す。本モデルにおいて淘汰パラメータを導入することにより、不要な文法規則を排除することが可能であることが確認される。

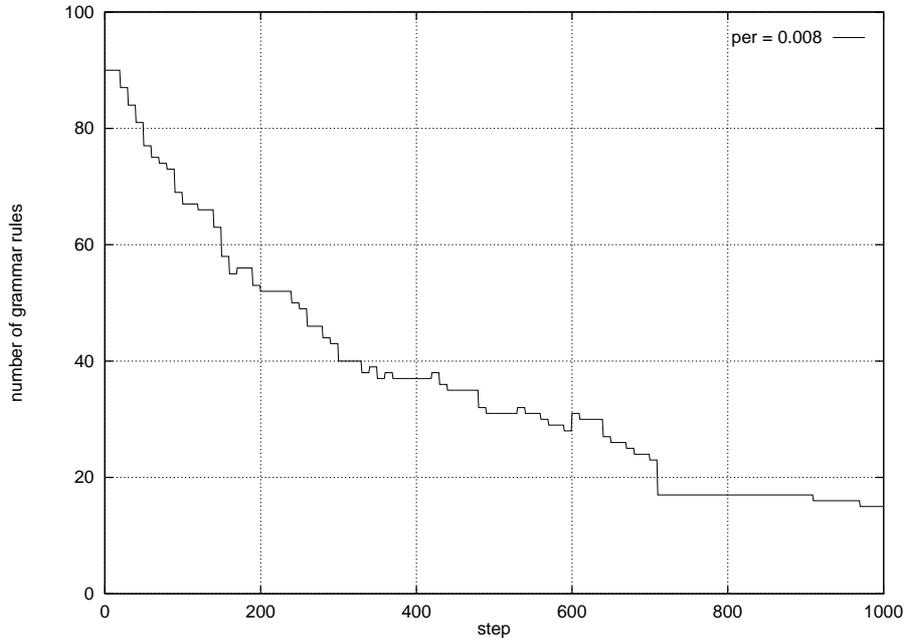


図 4.3: 淘汰パラメータの影響

4.4 実験 2

4.4.1 実験条件

実験 2 ではエージェント数を 10 に増やしてシミュレーションを行った。これは、予備実験のコミュニケーションモデルよりも発話に多様性が存在することになる。さらに実験 2 では、初期の段階で各エージェントが保有する文法規則数に制限を加えた。

また、実験 2 では新たに帰納推論の枠組を導入し、エージェントコミュニケーションに正例、負例を設定した。具体的には常に一定の選択確率に基づいた発話を行う教師エージェントを作り、このエージェントにより生成される文を正例とし、また、予め各エージェントに負例文を与え、これと同じ入りだしで生成された文を負例としてペナルティを加えた。また、初期の文法セットに組み込まれなかった文法規則は、帰納推論の枠組において背景知識として存在する。実験条件を以下に示す。ここで、発話数は、淘汰パラメータとの兼ね合いにおいて適当な 10 文に設定した。教師エージェント数と負例数については、その数値を変えて実験を行った結果より設定した。その結果については省略する。

実験 2

エージェント数	10
教師エージェント数	2
1 ステップの発話数	10 文
負例数	2
ステップ数	1000 step

4.4.2 考察

負例数を 2 として行った実験結果として $A_2 \rightarrow A_7 A_1$ における A_1 の展開についての各エージェントの選択確率の推移を表 4.2 に示す. ここで, P_a, P_b は, 教師エージェントの選択確率である. 0 ステップにおける各エージェントの選択確率を見てみると, $A_1 \rightarrow A_4 A_8$ の文法規則の選択確率はそれほど高くはないにも関わらず, ステップがすすむ毎に各エージェントにおける $A_1 \rightarrow A_4 A_8$ の選択確率が高くなっている. これは, A_1 の選択確率が上位カテゴリ A_2 における展開 $A_2 \rightarrow A_7 A_1$ の選択確率に依存するため, ここでの選択確率が直接発話に反映されないためである. また, 教師エージェントの影響も確認される.

ただし, 教師エージェントにおいて高い選択確率を有していた文法規則 $A_1 \rightarrow A_5 A_8$ が早いステップで淘汰されたのは, 負例の影響によるものと考えられる.

正例のみからの学習, 負例のみからの学習, そして正例と負例両方からの学習のそれぞれの選択可能な文法規則数の推移を図 4.4 に示す. 初期の段階では, 各エージェントとも有用な文法規則を見出していないため, 各自が勝手に発話を行い, 多くの文法規則を学習していることが確認される. しかし, ステップが進むごとに確率文脈自由文法の共有化が進むにつれ, 発話は特定の文法規則を使用されていった.

負例のみからの学習モデルと, 正例と負例からの学習モデルを比較してみると, 始めのうちは, 正例と負例からの学習モデルが, 負例のみからの学習モデルと同様に, 使用可能な文法規則数が減少している. しかし, 430 ステップ付近から, 正例のみからの学習モデルの文法規則数に近づいているのが確認される. また, 負例のみからの学習モデルについては, 逆に 450 ステップからの値に変化がなく局所的な文法構造に陥ってしまった. また, 正例のみからの学習モデルは, 他の学習モデルと比較してエージェントの持つ文法規則数の変動が安定するまでに, 多少時間がかかっている.

これから言えることは, モデルに負例を与えることにより, 早い段階での文法規則の淘汰 (特殊化) が実現し, モデルに正例を与えることにより, 会話に多様性を持たせること (一

step	Agent	$A_1 \rightarrow A_4 A_8$	$A_1 \rightarrow A_5 A_8$	$A_1 \rightarrow A_6 A_8$	$A_1 \rightarrow A_8$
	P_a	0.48	0.52	0.0	0.0
	P_b	0.0	0.0	0.0	1.0
0 step	A	0.24	0.0	0.42	0.34
	B	0.0	0.0	0.48	0.52
	C	0.29	0.44	0.27	0.0
	D	0.48	0.52	0.0	0.0
	E	0.0	0.26	0.29	0.45
20 step	A	0.33	0.0	0.44	0.22
	B	0.38	0.0	0.38	0.24
	C	0.44	0.0	0.33	0.22
	D	0.38	0.0	0.38	0.24
	E	0.38	0.0	0.38	0.24
40 step	A	0.45	0.0	0.36	0.19
	B	0.45	0.0	0.36	0.19
	C	0.40	0.0	0.40	0.20
	D	0.45	0.0	0.36	0.19
	E	0.45	0.0	0.36	0.19
70 step	A	0.56	0.0	0.22	0.22
	B	0.56	0.0	0.22	0.22
	C	0.56	0.0	0.22	0.22
	D	0.56	0.0	0.22	0.22
	E	0.56	0.0	0.22	0.22

表 4.2: $A_2 \rightarrow A_7 A_1$ における A_1 の展開

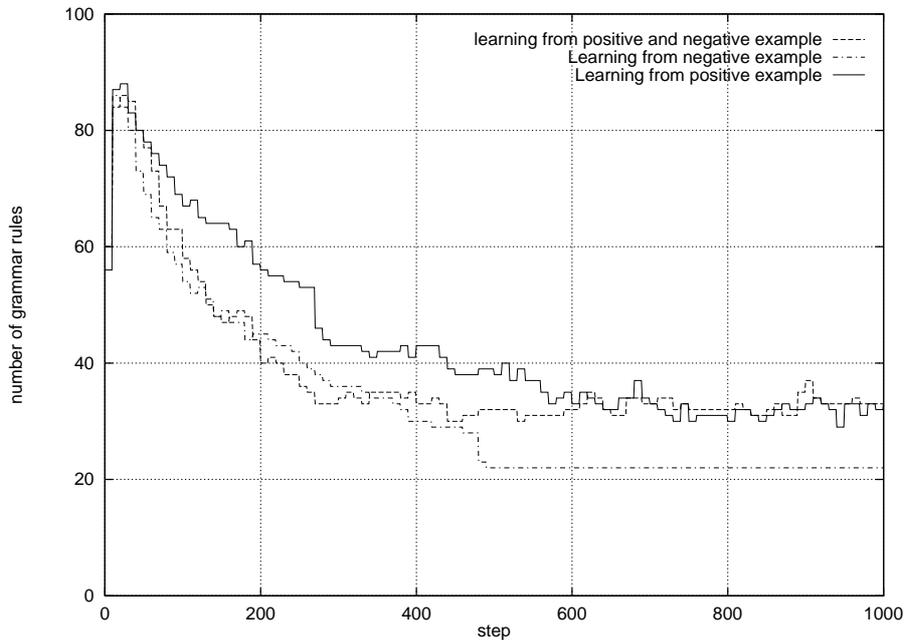


図 4.4: 正例と負例の影響

般化) が可能である。そして、正例と負例からの学習モデルでは、この二つの性質がうまくマッチングしていることが確認される。

この、正例と負例からの学習モデルにおいて、淘汰パラメータを変化させて実験を行った。実験結果を図 4.5 に示す。パラメータ値 0.004 では、先の負例のみからの学習モデルと同様に、局所的な文法構造に陥っている。しかし、その他のパラメータ値においては、正例と負例からの学習モデルの性質を保持しており、適度な値を与えることにより文法規則の保有数を変化させることが確認された。

また、予備実験と比較してみると、予備実験と同じパラメータ値 0.008 におけるエージェントの保有している文法規則数が大きく異なっている。また、実験結果の描く曲線の性質も異なっている。エージェント数の増加、帰納推論の枠組の導入による発話の多様性が確認される結果となった。

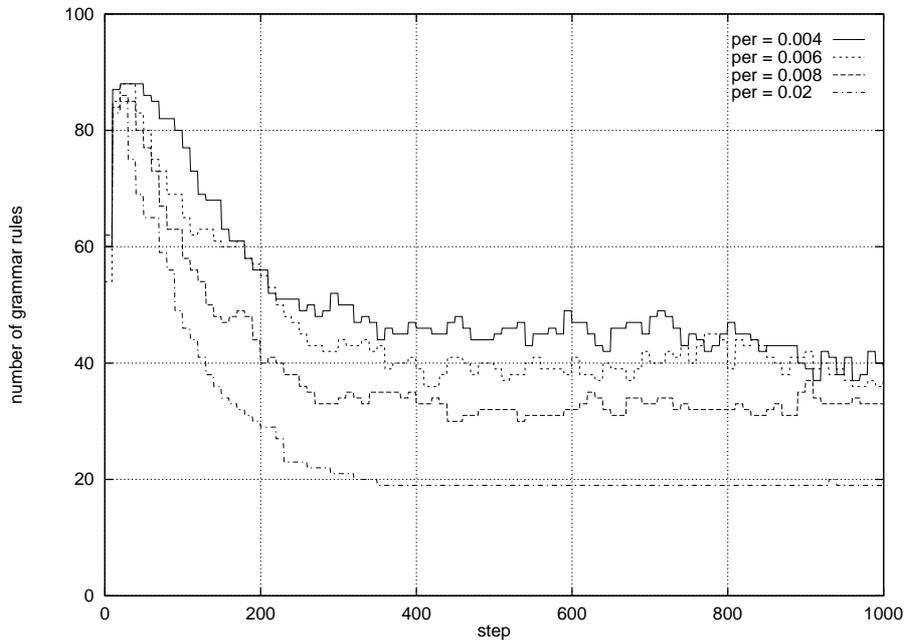


図 4.5: 選択可能な文法規則数の推移

4.5 実験 3

4.5.1 実験条件

実験 2 のモデルにおいて複数のプロセッサを使用した分散処理実験を行った。これまでの実験では 1 ステップにおけるエージェントの発話数が限られており、全てのエージェントが同数の文を生成していた。本実験では分散処理の性質を利用してエージェントの発話数にばらつきを持たせた。つまり、処理速度の早いプロセッサに割り振られたエージェントの発話数は、他のプロセッサに割り振られたエージェントよりも多くなる。本実験は、前の 2 つの実験よりも、より社会的なモデルのシミュレーションといえるだろう。実験条件を以下に示す。

実験 3

エージェント数	4
1 ステップの最大発話数	18 文 (3×6 loops)
ステップ数	1000 step
淘汰パラメータ	0.008
使用計算機	SS5, ultra5

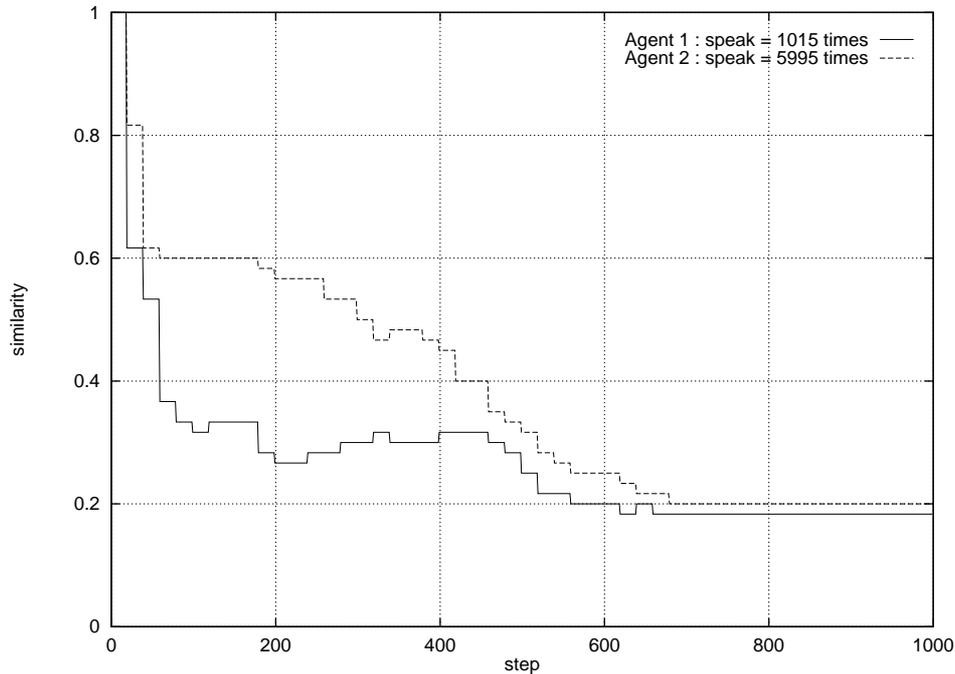


図 4.6: 1 対 1 のモデル

4.5.2 考察

実験 3 では、分散環境がモデルに与える影響を評価するため、4 台の計算機を使用して実験を行った。まず、動作確認として処理速度の早い計算機 1 台と、処理速度の遅い計算機 1 台の 2 エージェントでの実験を行った。また、評価には初期状態での文法規則の集合と各ステップにおける文法規則の集合との類似度を用いている。結果を図 4.6 に示す。

図 4.6 で、最終的に類似度が低くなっているのは、不要な文法規則の淘汰が影響している。実験 2 の結果では、選択可能な文法規則の数は約半数に減少している。そして、実験 3 ではエージェント数を 4 にしているため発話の多様性が減少し、実験 2 の結果よりも使用可能な文法規則が少なくなっているため、このような類似度に落ち着いたものと考えられる。

初期の段階では、処理速度の早いプロセッサ、つまり発話回数の多いエージェントは、それほど類似度を下げずにステップを重ねている。予備実験の結果では、2 エージェントモデルにおいても、互いの文法規則の選択確率は共有化されていた。つまり、図 4.6 においてエージェント 1 の類似度がそれほど下がっていないのは、エージェント 2 がエージェント 1 の文法規則の選択確率をベースに、共有化を行ったことを表している。以上より、発話回数の多いエージェントが系に与える影響が大きいことが確認される。

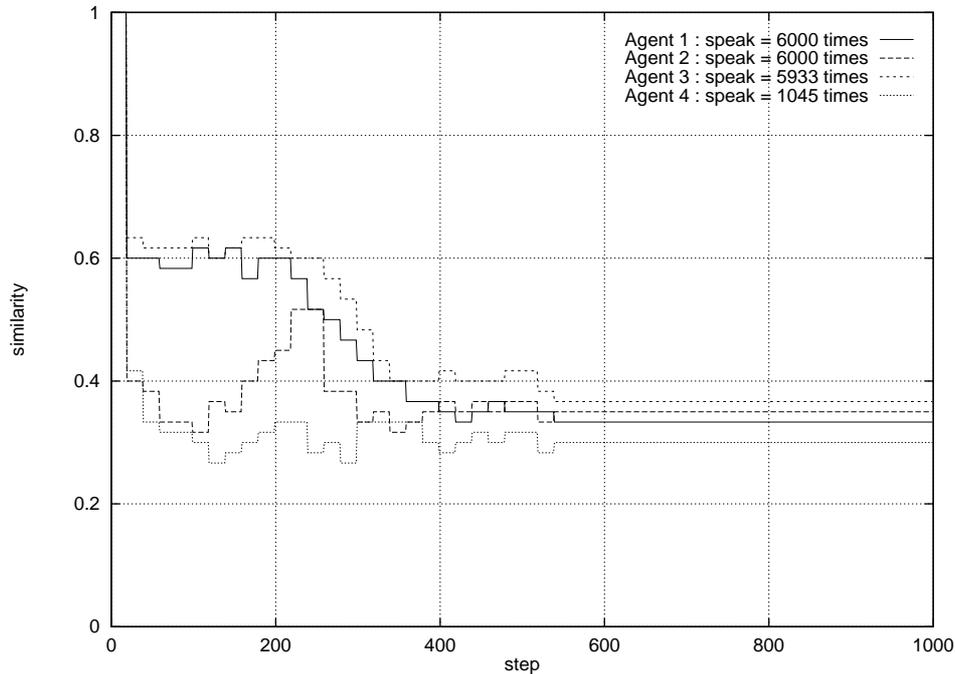


図 4.7: 3 対 1 のモデル

次に、発話回数の多いエージェント群の中に一つだけ発話回数の少ないエージェントを入れた場合について考察する。図 4.7に、発話回数の多い 3 つのエージェントと発話回数の少ない 1 つのエージェントでのコミュニケーションの結果を示す。

やはり、発話数の多いエージェントのほうが高い類似度を保ったまま世代を重ねるのが確認される。また、この 2 つの実験より各エージェントの発話数の違いによる影響は、初期の段階で大きく作用することが分かった。初期の段階での各エージェントの文法規則の選択確率はまだ試行錯誤的な段階のため、より多く発話を行ったほうが他のエージェントの文法規則の選択確率に与える影響が大きいものと思われる。ただし、図 4.7では図 4.6のモデルとは異なり、発話数の多いエージェントが 3 つ存在するため、イニシアチブの獲得で衝突が起こっている。エージェント 3 は、他の 2 つのエージェントと発話数がそれほど変わらないにも関わらず、その類似度を落している。エージェント 1 とエージェント 2 のもつ文法セットが、互いにエージェント 3 のもつ文法セットよりも比較的似ていたためにこの現象が生じたものである。集団社会において、その部分集団の規模が大きい程、全体に与える影響力が強いことを示している。

次に、このモデルとは逆に発話数の少ない 3 つのエージェントの中に発話数の多いエージェントを 1 つ加えて実験を行った (図 4.8)。

エージェント 1 が他のエージェントよりも高い類似度を保持しているものの、上の 2 つのモデルに比べ、その差は狭まっている。つまり、発話数の多いエージェントも集団の規模が大きくなれば、与える影響力が少なくなってしまうことを示している。

最後に、発話数の多いエージェントと少ないエージェントをそれぞれ 2 つずつ設定して実験を行った結果を図 4.9 に示す。

これまでのモデルと同様に、発話数の多いエージェントが高い類似度を維持している。ここで、エージェント 3 の類似度がエージェント 4 の類似度よりも高くなっている。この 2 つのエージェントに割り当てた計算機の処理速度は同じものである。しかし、発話回数を見るとエージェント 3 の発話回数はエージェント 4 の発話回数のほぼ 2 倍になっている。これは、分散処理における処理のタイミングにより生じた結果である。つまり、図 3.2 において、エージェント B , C 間の *Speak* 処理でエージェント A がエージェント B の *Speak Flag* を立てるよりも一瞬早かったために、もう一度 *Speak* 処理を行った現象と同じことが起こっているものと思われる。

以上の実験結果のまとめると、1 つのエージェントの与える影響の大きい小規模の環境では、エージェントの発話数が他のエージェントの文法規則の選択確率に大きく影響を与えるが、その規模が大きくなるにつれその影響力は低下していく。これは、現実の社会についても同じことがいえるだろう。その意味で、本実験がより現実社会に近いモデルを実装することができたと考えられる。

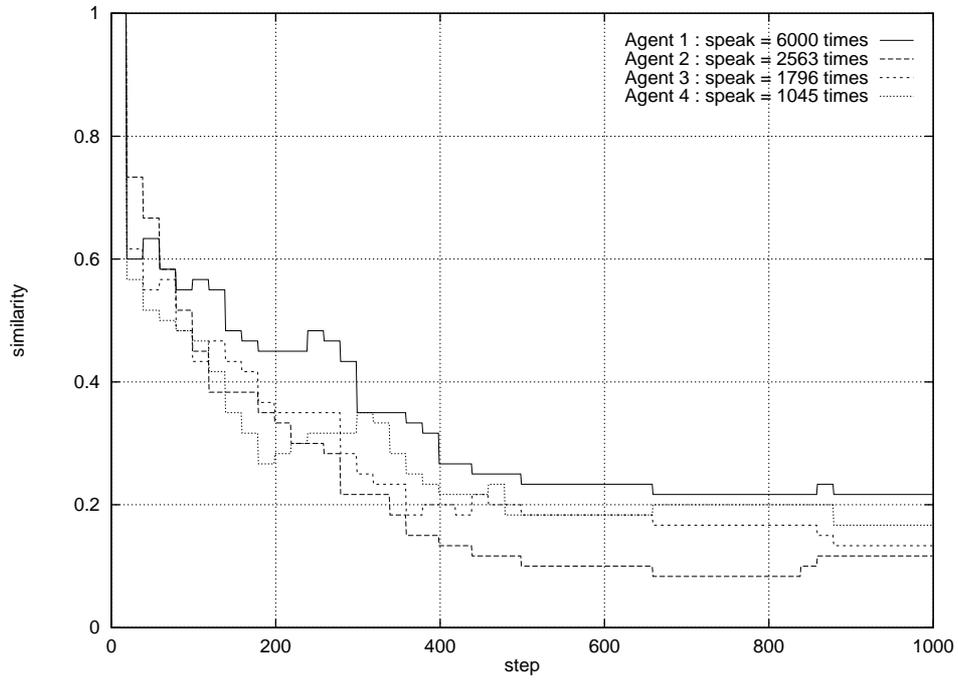


図 4.8: 1 対 3 のモデル

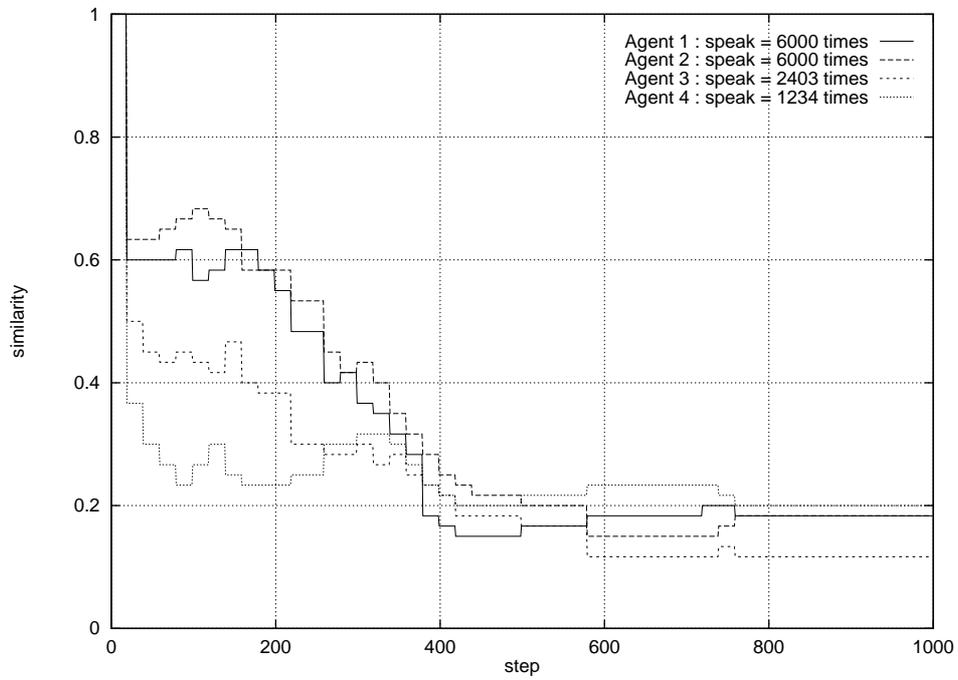


図 4.9: 2 対 2 のモデル

第 5 章

結論

5.1 まとめ

本研究では、マルチエージェント環境でのコミュニケーションによる文法の共有化についてのモデル化を行った。提案したモデルについてまとめを行う。

2 進数のビット列を入力データとして、与えられたデータから学習を行う対話的学習モデルを作成した。エージェントは入力されたビット列を形成する文法構造を、各エージェントが持つ確率文脈自由文法の文法規則から解析し、得られた解析結果から文法規則の選択確率のチューニングを行う。また、入力データは、エージェントの発話により生成される。このように本モデルは、あるエージェントからの入力により文法の生成確率を自己組織化し、それを発話により他のエージェントにフィードバックする繰り返し最適化モデルである。

選択確率の自己組織化において、2 つアプローチにより各エージェント間での一般化を行った。

- 文法規則の選択確率に閾値を設定し、閾値以下の文法規則を淘汰する。
- 帰納推論により正例と負例を与え、例の一般化を行う。また正例の与え方としては、教師エージェントによる一般の教師つき学習を採用した。

また、上記アルゴリズムにおいて、それぞれの設定を変えることにより、得られる結果にアルゴリズムの性質を反映した変化が確認された。

実験環境としては、分散処理により各プロセッサを一つのエージェントに見立てた大規模なシミュレーションを行った。各エージェントは、割り当てられたプロセッサの処理速

度に依存した発話を行うため、発話数に個体差が出現した。この現象に対しては、発話に個体差を生じさせたほうが、より現実社会の現象を反映したモデルになったと考察した。

最後に、本研究で提案した分散環境でのシミュレーションが、今後のマルチエージェントモデルの実験における一つのシステムを提案できたと信じている。

5.2 今後の課題

本稿で提案したモデルは幾つかの点で人工的な設定を行っているため、自然言語のような複雑な現象のメカニズムを説明することは不可能である。今後はこれらの人工的な設定に対しても、より根拠のある（認知科学的に説明可能な）モデルに改善していくこと、そしてそれが良いモデルであることを主張するために実際の言語データから統計的検証を常に考えていく必要がある。

また、確率文脈自由文法を用いた構文解析 (probabilistic parsing) の導入や、帰納論理プログラミングの文脈自由文法への応用についての考察を行う必要がある。

謝辞

研究を行う上で、御指導くださった北陸先端科学技術大学院大学情報科学研究科、浅野哲夫教授、東条敏助教授、小保方幸次助手に深く感謝します。浅野哲夫教授、東条敏助教授には研究方針、研究姿勢、その他多くの助言を頂きました。小保方幸次助手には、分散プログラミングについて御指導頂きました。最後に、情報基礎学講座の皆様に深く感謝の意を表します。

参考文献

- [1] Derek Bickerton 著, 笈 寿雄 監訳: ことばの進化論, 勁草書房, 1998.
- [2] Eugene Charniak: STATISTICAL LANGUAGE LEARNING, *TechBooks*, 1993.
- [3] 古川康一: 帰納論理プログラミング チュートリアル , 人工知能学会誌 *vol.12, no.5*, pp655-664, 1997.
- [4] Hashimoto, T. and Ikegami, T.: Evolutin of Symbolic Grammar Systems, *Advance in Artificial Life*, pp812-823, 1995.
- [5] Hashimoto, T. and Ikegami, T.: Emergence of net-grammar incommunicating agents, *BioSystems* 38, pp1-14, 1996.
- [6] 北研二: 確率的言語モデル, 東京大学出版会, 1999.
- [7] 中村誠: マルチエージェントモデルにおける共通文法の獲得, 北陸先端科学技術大学院大学, 修士論文, 1997.
- [8] Shan-Hwei nienhuys-Cheng and Rnald d.Wolf: Foundatins of Inductive Logic Programming, *Springer*, 1991.
- [9] 錦見美貴子: 言語を獲得するコンピュータ, 共立出版, 1998.
- [10] 野村浩郷: 自然言語処理の基礎技術, 電子情報通信学会, 1998.
- [11] 大津由紀雄 他: 言語科学と関連領域, 岩波書店, 1998.
- [12] 小野哲雄, 東条 敏: 推論機能を有するエージェント群による共通文法の組織化, 人工知能学会誌 *vol.13, no.4*, pp546-559, 1998.
- [13] Norihiko Ono and Adel T. Rahmani: Self-Organizatin of Communicatin in a Multi-agent System, *MACC*, 1992

- [14] Steven Pinker.: The Language Instinct, *Harper Perennial*, 1994.
- [15] Christopher D. Manning and Hinrich Schütze: Foundatins of Statistical natural Language Processing, *The MIT press*, 1999.
- [16] Luc Steels: The Origins of Syntax in Visually Grounded Robotic Agents, *IJCAI*, pp1632-1641, 1997.
- [17] Werner,G. M. and M.G.Dyer: Evolutin of Communication in Artificial Organisms, *Artificial Life II*, pp659-687, 1991