

工学博士 学位論文

電気工学科 167

自然言語理解のための経験的知識の
構成に関する研究

1988年12月23日

指導教官 齊藤 忠夫 教授

東京大学大学院電気工学専攻

藤田 悟

目次

1	序論	1
2	自然言語理解のモデル化	3
2.1	自然言語理解における文脈	3
2.2	経験的知識と文脈	5
2.3	文脈に関する知識の学習システム FOCUS	7
2.4	コネクショニスト・モデルによる自然言語理解	11
2.5	自然言語理解の意識的想起と無意識的想起によるモデル化	15
3	経験空間による知識表現	20
3.1	経験空間の基本概念	20
3.2	神経回路網からのモデルの導入	23
3.3	活性化エネルギーの伝搬式	26
3.4	文章の経験空間への展開	31
3.5	経験空間の評価	37
3.5.1	経験空間の学習能力に関する実験	37
3.5.2	抑制リンクに関する実験	42
3.5.3	経験空間の能力評価	43
4	自然言語理解システム FOCUS3	47
4.1	FOCUS3 の全体構成	47
4.2	経験空間による文章理解	50
4.2.1	入力文章の意味ネットワークへの展開	50
4.2.2	入力文章の時間関係の保存に関する検討	54

4.2.3	経験として与える知識の標準形	56
4.3	質問応答機能	58
4.4	パーザとの結合	62
4.4.1	文節以下の構文解析法	62
4.4.2	文節以上の構文解析法	64
4.4.3	質問応答システムへの日本語入力インタフェース	67
4.5	FOCUS3 の動作例	69
4.5.1	文脈理解	69
4.5.2	同音意義語の理解	76
4.5.3	質問応答実験	78
4.5.4	知識の広がりに関する評価	92
4.6	自然言語理解システム FOCUS3 の評価	95
5	経験空間を利用した要約文の作成	99
5.1	要約文作成の基本概念	99
5.2	経験空間による要約文の作成	103
5.3	要約文作成の実験	111
5.4	要約文作成システムの評価	118
6	経験空間からの長期記憶検索	120
6.1	人間の長期記憶検索	120
6.2	経験空間中での記憶検索	124
6.3	記憶検索の実験	126
6.4	記憶検索システムの評価	130
7	結論	133
	謝辞	136
	参考文献	137
	発表文献	140

付録 A : 食事に関する経験	141
付録 B : 「たこ」を理解するための経験	145
付録 C : 童話を理解するための経験	146

第1章

序論

自然言語理解に関する研究も、機械翻訳に代表されるように実用的な成果が報告されるまでに進展した。夢の自動翻訳機をめざして、今も多くの研究が行われている。しかし、外面的な進歩の著しさに対して、研究の不十分である領域もまだ多く残されている。その中でも大きな問題点は、現在の手法ではその取り扱える自然言語の対象が非常に限られていることである。自然言語理解と言っても、成果の上がっているといわれる研究の対象は製品マニュアルのような定型的な文章や、あるいは制限が加えられた自然言語であり、人工言語に近い性質を持つ言語が対象である。自然言語の自然言語たる性格を持つ会話文、物語の理解については、まだ十分な成果があがっていない現状にある。

そこで、従来の研究である統語論、意味論の枠を越えて、言語理解の第3段階である語用論へとつながる「文脈」の研究に興味を持たれるようになった。しかし、これまで計算機で扱ってきたものは、構文解析のように定まった規則に当てはめた解析であり、文脈のような

- 秩序性が低く、
- その知識の量は大きい、

対象を扱うには全く新しい理論が必要になる。この問題を避けて通るために、最初の頃の文脈研究は、あくまで意味のあい昧性を解決するための補助知識として文脈を扱う立場をとった。文脈に関する知識を制約規則のひとつとして記述するため、一般的な文章を理解するシステムを構築することは難しく、対象分野を狭めることは必然的な要求であった。

しかし、文章によって本来意図した内容を理解したいという立場に立った場合、やはり、文脈の問題は正面から解決しなければならない。特に会話文のような“柔らかい”自然言語を理解するためには、文脈の理解は必要不可欠である。日常の会話では統語的な構造よりも文脈的な意味が優先することもしばしば起きている。これを受けて、最近の研究の中では、「状況意味論」[3]のように、

「文の意味は状況の中に規定される」

という意味論も展開されるようになってきた。このように補助知識という立場を変えて、意味を規定する本質的な媒体として文脈を認識しなければならないという要求は高まりつつある。

本論文は、このような言語理解の研究の方向性を支持し、より積極的に文脈を利用しようという立場で、文脈を中心とした自然言語理解について論じる。この方向づけが、自然言語理解の研究として成り立つためには、次の点に関して明らかにしなければならない。

1. 人間の自然言語理解過程と比較したモデルの構築
2. 文脈を中心とした自然言語理解を行うための知識表現とその知識の獲得法
3. 自然言語理解システムの構成

以下、これらについて順に論じていく。

第2章では、文脈を形成するにあたって経験的な知識が重要な役割を果たしていることを示し、これが有効に利用されるモデルとして、人間の言語理解の中に存在する意識処理と無意識処理について論じる。第3章では、文脈の知識表現として意味ネットワークを基礎とした「経験空間」という知識表現を提案し、その基本構成を紹介する。第4章では、文脈を中心とした自然言語理解システムの構成について述べる。第5章では、経験空間を利用した要約文の作成システムについて述べる。第6章では、蓄積された経験的知識の中における記憶検索の問題について論じる。最後に、第7章では、このモデルの有用性と今後の課題についてまとめる。

第2章

自然言語理解のモデル化

本章では、まず、文脈の立場を明らかにし、経験的知識の重要性を論じる。それから、自然言語理解のモデル化を行うと共に、文脈に関する研究を紹介する。

2.1 自然言語理解における文脈

日常会話には非常に多くのあい昧性が含まれている。語の省略もその一つである。主語、目的語、そしてさらには述語に至るまで、一文としてみるとまったく情報を持たない会話も意志の伝達には大きな役割を果たすことができる。もちろん、会話は言葉だけのコミュニケーションではなく、アクセント、抑揚といった言語的なものから、身ぶり、手ぶり、顔色といった非言語的なものまで、様々な情報源をもって初めて成立するものである。言葉の表層構造だけを見て、その意味をとらえるのが難しいのは当然のことであるのかもしれない。

しかし、人間は自然言語を使い、生活に支障のない程度にその意味を正しく把握している。身ぶり、手ぶりより確実な意志の伝達手段として、自然言語は欠かせないものである。話し言葉だけでなく、文書からも様々な情報を得ることができ、自然言語の情報伝達に関する有能性を示している。

言語の研究というと、まず「文法」が頭に浮かんでくるが、人間が本当に文法を重視して自然言語理解を行っているかについては疑問な点も多い。むしろ、

- 特に困らない限り、流れにまかせて意味を理解する。
- 正確な意味を追求するときは、構文をじっくり考える。

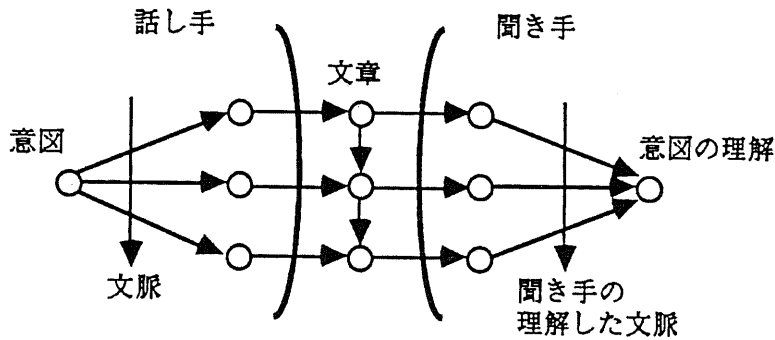


図 2.1: 人間による文章の作成, 理解過程における文脈

というような言語理解をしていると感じるような状況にしばしば出会うことがある。前者のような流れによる言語理解を可能にするものが文脈である。

文脈とは何であるのか。この問いに答えるのに

- つながった文の間関係

と言うだけでは不十分である。文を適当な順序に並べただけでは、文脈は現れない。発話者が、ある意図をもって文を並べたときに初めて文脈は生じる。したがって、

- 内在する意図と発話された文との関係

が、ここで扱う「文脈」である。この関係を図 2.1 に表現する。聞き手は、話し手の文脈を理解することによってその意図する内容をつかむことができるようになる。

一方、計算機による言語理解の研究は、統語論を中心に始まった。

1. まず、文の表層構造を構文規則を用いて解析する。(統語論)
2. 文法構造から意味表現を導き出す。(意味論)
3. あい昧なものに関しては、文脈により解決する。(語用論)

計算機は少ない規則に対して多くのデータを適用する分野に適しているため、規則化しやすい統語論が最も得意とする分野である。逆に、文脈に対しては消極的な態度を

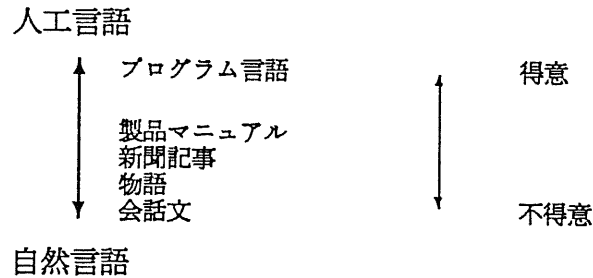


図 2.2: 自然言語らしさによる分類

とり，補助知識という立場を越えた使われ方には至っていない。自然言語らしさと計算機の扱いやすさを示すと図 2.2 のようになる。文脈の入り込まない分野ほど，計算機は得意であり，プログラム言語とはその極端な例である。機械翻訳の対象となっている製品マニュアルも読みやすさを前提にし，文脈による推論が少ない文章である。言い替えると，文脈が明確であるため，それを規則化しやすい分野である。これに対し，文脈を規則化しにくい会話文，物語の理解は計算機にとって不得意な分野である。

この得意でない分野の文章を理解するには，もはや文脈の問題を避けて通ることはできない。その文脈の扱い方には，

- 消極的な文脈の利用 (補助知識としての文脈の利用)
- 積極的な文脈の利用 (文脈を中心とした意味論の展開)

が考えられる。従来の研究は前者のアプローチがによるところが大きかった。しかし，人間による言語理解の過程からみても，文脈中に現れる意図を言語の本質として文章理解を行うべきであると思われる。

2.2 経験的知識と文脈

この節では，文章の「意味」の理解について論じる。自然言語理解の研究の中で，

「この文は構文的には正しいが，意味的におかしい。」

という言い方をすることがある。ここでいう「意味」とは何であるのかを検討する。人間の論理判断能力は，十分なものではない。論理的な文が与えられて，その真偽を問うような場面でも，文中に表現される言葉に惑わされて，誤った論理判断を下すこ

とは良く知られた事実である。したがって、このような乏しい能力をもって、文章の意味を論理的な正しさで判断するとは考えにくい。たとえば、

「鳥は飛ぶ。」

という事実を信じている人を仮定してみる、しかし、この意味を論理的な真実として認識していることは少ない。まず、「鳥」の明確な定義すら持っていないことが多い。そこで、

「ペンギンは鳥である。」

「ペンギンは飛ばない。」

という事実を与えられると、意味的におかしいと感じてしまう。このような例題は、Non-Monotonic Logic[13], Default Reasoning[17] の分野で研究されていて、最初に持っていた知識は、

「特例でない限り鳥は飛ぶ。」

という意味をもって理解することができる。ペンギンは特例のひとつとして処理されるようになる。

しかし、さらにこのような Default Reasoning 的な論理判断能力とそのための知識を持っている状況でも、

「ニワトリは飛ばない。」

という文章の意味はおかしいと感じてしまう。その理由として、

「ニワトリは、鳥だから...」

という堂々めぐりの推論をすることもしばしば起こる。すなわち、意味とは知識に規則を当てはめることによる絶対的な評価基準ではなく、自分の持つ知識と現在の入力文を「手に届く範囲で」比較をした場合に起こる相対的かつあい昧な評価基準であるといえる。自分の中の知識との整合性をとることが意味理解であると考えたいが、そのときの「自分の中の知識」は、思った以上にあい昧なままの知識である。

「今、思いつくところでは...らしい。」

といった弱い推論のもとに意味を判断しているようである。

この弱い推論を行うための知識源は、その時点までの経験によるところが大きい。人間の持つ生得的な言語獲得能力も知識源のひとつのである。しかし、人間の知的活動に必要な知識の多くは経験から蓄積されたものである。この経験から得られる知識はその質及び量の面で個人差も大きい。これは、主に環境に左右されるからである。したがって、同じ文章であっても各個人によって認識されるものが異なる場合もある。ペンギンの例でも、ペンギンを知らない人は、相変わらず、「鳥は飛ぶ」という知識を一般的に通用する知識として持ったままでいることができる。

自分の考えていることを相手に伝えることが、自然言語を用いた会話の目的である。自分の頭の中の状況と似た状況を相手の頭の中に作成していくことにより意志の伝達が実現される。しかし、実際は経験的知識の差によって、内容がうまく伝わらない場合が生じる。特に日本語は省略表現が多く、言外に含む意味が大きい言語であり、相手との状況の一致は言語理解に欠かせない条件である。

これらの事実から、文章の意味を理解するためには、経験的知識を蓄えた記憶が重要であることが示された。文章の意味は、経験的知識の中に相対的に定義されることが望ましい。特に、自然言語理解を文脈中心に構成しようと考えた場合、絶対的な文脈の意味というものは考えにくい。過去の記憶の中に類似経験を求め、それと比較することによって自分の知識の中の適切な場所に記憶することが自然言語の認識であるという立場で以後の議論を進める。

2.3 文脈に関する知識の学習システム FOCUS

文脈を理解するための知識を学習するシステムとして、過去に FOCUS (Frame Oriented Context Understanding System) を提案した。この FOCUS を作成するに当たって、次のような検討を行った。

- ・文脈を理解するための知識を作成することは難しい。

自然言語理解の中で、文脈的な知識を要求したい場面は様々である。そのすべてに対してそれぞれ知識を作成すると、その量は膨大なものになる。また、それを人間の手によって作成するためには、その作成の手続きを明らかにしなくてはならない。(何を、どのように表現すれば、効率の良い知識になるかについての基

準があい味である。)そこで、文脈に関する知識を計算機によって自動的にする獲得するシステムが必要になる。

・文脈に関する知識は、類似経験を多く積むことによって学習できる。

内容が似ている文章を読むと共通に使える知識が多い。また、以前に読んだ文章によって、新しい入力文章の理解が容易になる場合がある。これは、互いの中に共通する文脈が存在するからである。文脈に関する知識は、類似した経験の中に発見され、経験を多く積むほど充実した知識となる。

・知識の一般化は、多くの情報を失う。

経験した文章は無秩序に記憶しても有用な知識とはならない。そこで、体系的な記憶を行うために知識の一般化のプロセスが必要になる。この観点からは、似た知識はひとつにまとめられるべきである。しかし、文脈のための知識は、例外事象の多い知識であり、一般化することにより、失われる情報が多い。そこで、例外の知識も切り捨てることなく記憶すべきである。

・文章を理解する状況で、文脈に関する知識を構成すべきである。

このように、文脈に関する知識は例外の多い知識であり、文脈に関する一般的な知識というものを作成することは難しい。むしろ、その例外の部分に現れる詳細な情報が文脈理解には重要な知識である。したがって、文章を理解しながら、関連する知識を作成し、その詳細部を文章理解の状況に適応させる方法が望ましい。

FOCUS の取り扱った文章は、イラン・イラク戦争に関する新聞記事である。いくつかの新聞記事を入力することによって、イラン・イラク戦争を理解するための知識を作成することができる。知識を一般化して記憶すると、多くの情報が欠けてしまうという議論から、経験した文章をすべて記憶することにした。ただし、記憶構造は無秩序ではなく、次の図 2.3 に示すような階層構成をもって蓄積した。図の中の略称は次の 5 個の記憶領域を表している。

EM (Event Memory: 出来事記憶)

CM (Causal relation Memory: 因果記憶)

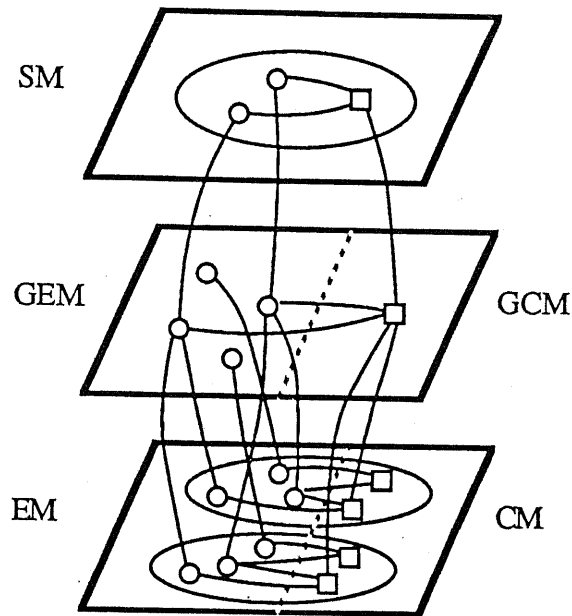


図 2.3: FOCUS の記憶構造

GEM (Generalized Event Memory: 一般出来事記憶)

GCM (Generalized Causal relation Memory: 一般因果記憶)

SM (Situational Memory: 状況記憶)

EM は、出来事の記憶を行い、CM は、出来事間の因果関係を記憶する。また、EM、CM は、経験の詳細を記述する記憶領域であり、一般化された知識の記憶領域である GEM、GCM 中の知識とリンクで結ばれている。SM は、これらの知識を小さな状況ごとにまとめている。

FOCUS は、このような記憶構造を作る部分と、それを利用して、文章理解のときに文脈に関する知識を構成する部分からなる。全体構成を図 2.4 に示す。代表的な機能分担に関して簡単な説明を加える。

EM constructor

フレーム表現された入力文章を EM 中に展開する。

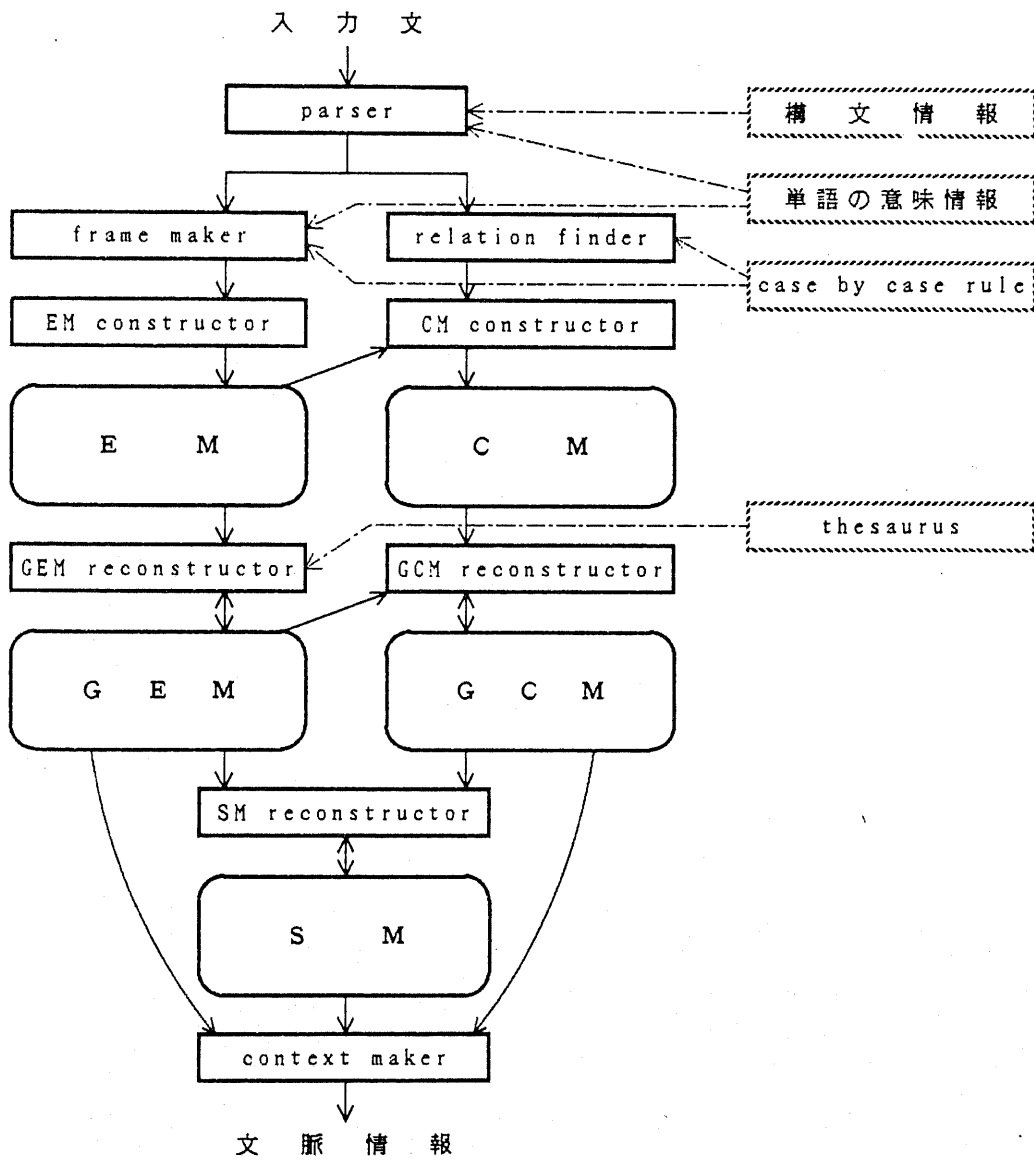


図 2.4: FOCUS の全体像

CM constructor

event 間の関係を CM に展開する。

GEM reconstructor

EM に記述された複数の event の間で一般化を行う。その結果を GEM に展開する。

GCM reconstructor

GEM に記述された event 間の関係を GCM に展開する。

SM reconstructor

GEM, GCM に記述された中に状況ごとのまとまりを発見し, SM に展開する。

context maker

文章理解時に起動され, GEM, GCM, SM に展開された知識を基に, 文脈に関する知識を構成する。

このシステムは, 次のような当初の目的を達成することができた。

- 文脈に関する知識をシステムが自動的に生成する。
- その知識は, 文章を理解するとき, その状況に合わせて作成される。

FOCUS では, 文脈を経験的な知識の中で理解することができるが, 結果的に得られる知識は, フレーム表現された「ルール」である。これを利用した自然言語理解システムでは, その知識を「補助的知識」として利用することを仮定している。すなわち, 文脈的な意味と, 文の意味は独立に考えられていて, 文脈を中心とした意味論の展開までには至らなかった。

2.4 コネクショニスト・モデルによる自然言語理解

パーセプトロンに代表される神経回路網をモデルにした研究は, Minsky によりその限界が示された後, 一時下火になった。しかし, 最近になって, 新しい学習理論が

発見されると共に、再び脚光を浴びて、コネクショニスト・モデルとして多くの研究が行われるようになった。

コネクショニスト・モデルとは、神経細胞をモデル化したノードと、それを結ぶとリンクにより構成されるネットワーク上に、情報を表現するモデルである。ノードには、神経細胞の膜電位に相当するエネルギーが定義される。リンクで結ばれたノードからの入力のみつき加算を行い、そのノードの持つエネルギーとするモデルが一般的である。エネルギーがしきい値を越えるとそのノードは活性化したとみなされる。リンクには興奮性ものと抑制性ものがあり、抑制性のリンクには負の値の重みがつけられる。ネットワークの形態は、様々であるが、入力ユニット、出力ユニット、隠れユニットに分けるモデルがよく用いられる。コネクショニスト・モデルにおけるパターン認識は次の手順で行う。

1. 認識したいパターンを、入力ユニットのノードにエネルギーの形で与える。
2. エネルギーの安定状態に至るまでノード間で活性と抑制を繰り返す。
3. 安定状態における出力ユニットのパターンによって、認識結果を確認する。

コネクショニスト・モデルは、ノードに直接に意味のあるシンボルを与える「局所表現」と、全体の活性パターンで意味を表現する「分散表現」との大きく2つに分けられる。前者のネットワークは、意味ネットワークに近い。重み付けをユーザが意図的に与えることも可能である。一方、後者は、意味がネットワーク中に分散表現されるため、人手でネットワークの重み付けを操作することはできない。よって、学習アルゴリズムが重要であり、最近ボルツマンマシン [1] に代表されるいくつかの学習理論が提案されている。

コネクショニスト・モデルを自然言語理解に応用する研究 [21][23] は、前者の「局所表現」のモデルを中心に行われている。ここでは、Waltsらの研究について紹介する。

Waltsらは、言語理解における「統語論」「意味論」「語用論」の3つのレベルを、統合的に処理できるモデルについて研究した。その結果、入力単語のレベルを含めた、syntactic, input, lexical, contextual の4つにレベル分けしたネットワークを用い、ノードに定義されたエネルギーを興奮性と抑制性のリンクによって伝搬するモデルを提案した。

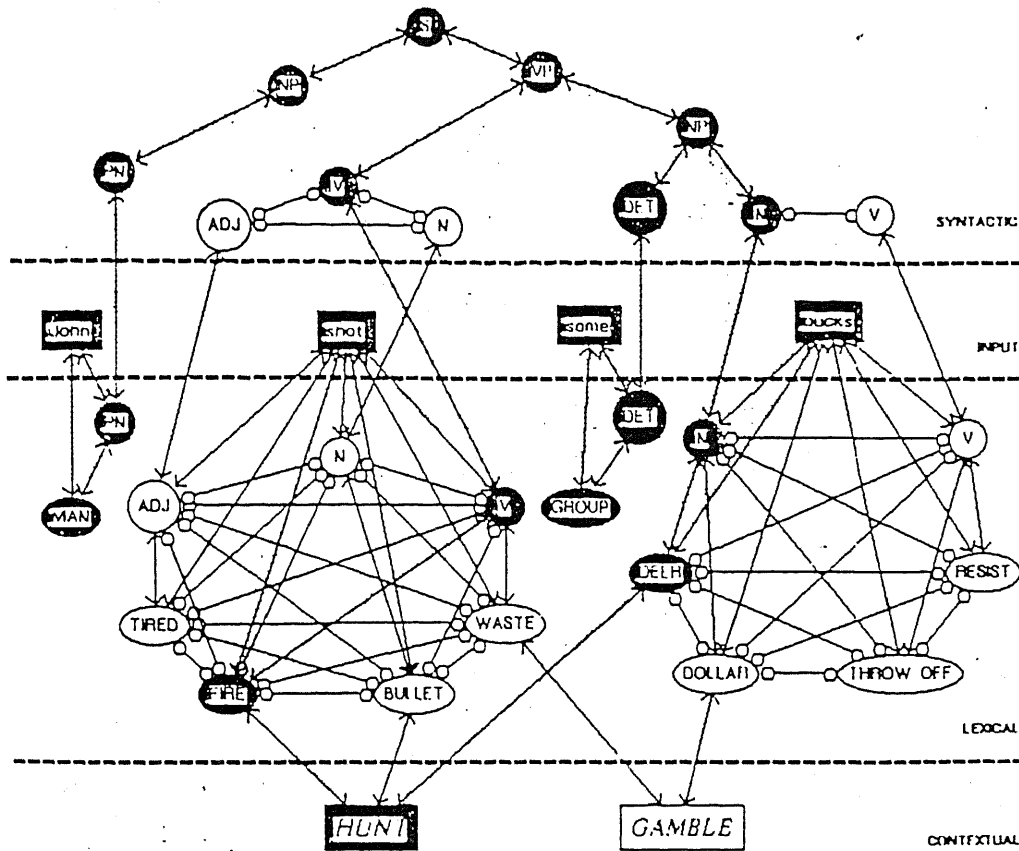


図 2.5: “John shot some bucks.” を理解するためのネットワーク

“John shot some bucks.”

を理解するためのネットワークを図 2.5 に示す。 図中、矢印は興奮性の結合を表し、白抜きの丸は、抑制性の結合を表す。 興奮性、抑制性のリンクは次のような場合に用いられる。

興奮性リンク

- syntactic level の句構造の親子関係の構成要素の間
- input level の単語と lexical level に展開されるその単語の (複数の) 意味の間
- contextual level のノード と関連する lexical level のノード
- syntactic level と lexical level の対応する文法カテゴリ

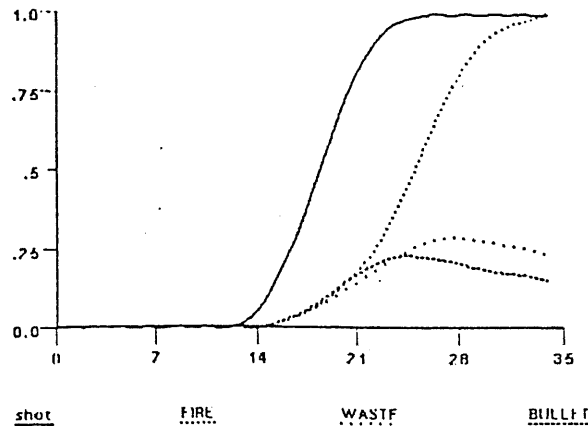


図 2.6: “shot” の持つ異なる意味の活性度の変遷

抑制性リンク

- lexical level の同じ語に対する異なる文法カテゴリの相互間
- lexical level の同じ語に対する異なる意味の相互間
- context level のノードに対して、相矛盾する lexical level のノードの相互間

興奮性リンクの重み付けは $+0.2$ ，抑制性リンクは -0.45 に固定して実験を行った。contextual level の “HUNT” が活性化している状況から始めて、理解の最終状態が、図 2.5 に示される。黒く塗りつぶされたノードが活性化したノードである。たとえば、“shot” に対する 4 つの意味 (TIRED, FIRE, BULLET, WASTE) の中から、“FIRE” だけが活性化し、これが、“shot” の適切な意味であると判断された。その時の活性度の変化を図 2.6 に示す。

この研究は、言語論の 3 つのレベルの統合化を行ったことで興味を引くものであるが、

- 文脈を表現するノードをいかに作るか。文脈の数は無限であり、その提供方法を検討する必要がある。(microfeature を用いて、文脈ノードの構成法について述べているが、意味の集まりによって文脈を表すということ自体が文脈ノードの使用を認めていることになる。)

- 意味的なつながりを持つ文と文の間に存在する関係をいかに記述するか。これが記述できると文脈に関する推論能力を向上させることができる。
- 重み付けを学習によって変えられないか。

などの今後の課題を持つ。

2.5 自然言語理解の意識的想起と無意識的想起によるモデル化

自然言語により伝えられる意味を正しく理解するためには、非常に多くの情報が用いられる。その多くは、外部発生的な情報である。理解の対象となっている入力文章も外部からの情報である。計算機による自然言語理解システムには、入力文章を表面的なシンボル構造として与えるのが普通であるが、人間が言語理解を行っている環境では、話し手の身振り、手振りや、アクセント、抑揚など、シンボルでは表せない様々な情報を利用できる。このような情報によって、聞き手は文章の背景を知ることができ、文章を状況の中で意味付けすることができるようになる。

さらに、人間は、自然言語を理解する過程において、様々なことを頭に思い浮かべる。これらは、外部発生的な情報に喚起されて、聞き手の内面から発生する情報であるといえる。意識的な推論の結果を思い浮かべるのはもちろんのこと、話の中に出てくる出来事、単語などから連想されることを、多くは無意識のうちに頭に思い描いている。このような無意識の想起は、あるときは言語理解の雑音となり集中力を欠いた状態を作り出すが、多くの場合は、理解している文章と関連する事実を探索していることに相当し、文脈的な意味理解を支援する知識となる。

このような人間の自然言語理解過程に比べて、計算機による自然言語理解システムでは扱う情報が非常に限られている。文章は、一般にシンボルで与えられ、言語の表層構造に含まれない情報は与えられない。音声認識と自然言語理解を結び付ける研究も行われているが、音声からシンボル以外の情報を取り出して、言語理解に役立てるような研究はあまり行われていない。また、人間の行うような内部的な情報の喚起も、意識的な推論を行うに留まっている。今後、「文脈」に関する様々な現象を理解するには、より広い情報と、それを効率良く検索する機能が必要になってくるであろう。

経験的知識の重要性は、2.2節でも述べた。この知識は、本人の経験の積み重ねに

よって獲得されるものである。人間の記憶容量は非常に大きいことが知られている。子供の頃からの経験によって、非常に大きな知識ベースを構成している。この知識があつて初めて、柔軟な言語理解を行うことができると思われる。計算機上の自然言語理解システムにも、積極的にこのような経験的知識を蓄積する手法を検討すべきである。

また、人間は、大きな知識空間を扱っているにも関わらず、記憶検索がボトルネックになることが少ない。これは、無意識的想起によって、関心を持たなければならぬ知識の範囲を限定し、その中で検索を行っているのではないかと思われる [7]。新しい話題の中に突然飛び込んだ時に、しばらくはとまどってしまうのは、この知識の範囲の限定が行われていないためであると説明することができる。無意識的想起によって参照される知識の範囲は、いつもほぼ同程度に広がり、全体的な知識量の増加とともに検索時間が増加する問題は解決できる。

そこで、自然言語理解システムの中にも無意識的な想起に対応する知識の拡散を実現するモデルを採り入れることを検討する [15]。まず、無意識と意識の想起が、従来の自然言語理解モデルの中のどのレベルに相当するかを検討する。次に3つの推論機械を示す。

- (1) 従来のルールによる推論処理
- (2) 経験空間
- (3) コネクショニスト・モデル (神経回路網)

(2) にあげた経験空間とは、第3章以降で議論する知識表現である。これを、次のような人間の中の認知機構に対応させたい。

- (1) 記号的意識処理
- (2) 記号的無意識処理
- (3) 非記号的無意識処理

この3つの関係は図2.7に示される。

SMとは、従来の記号操作を表し、これが扱っているルールを micro feature のレベルで支えるものが経験空間と考える。2.2節と同じ例を用いると、

「鳥は飛ぶ」

「ペンギンは鳥である」

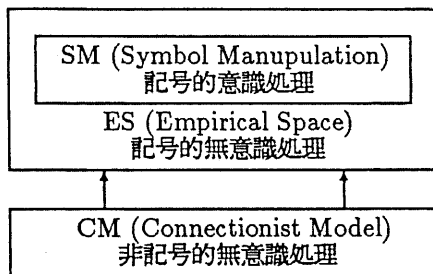


図 2.7: 経験空間の位置づけ

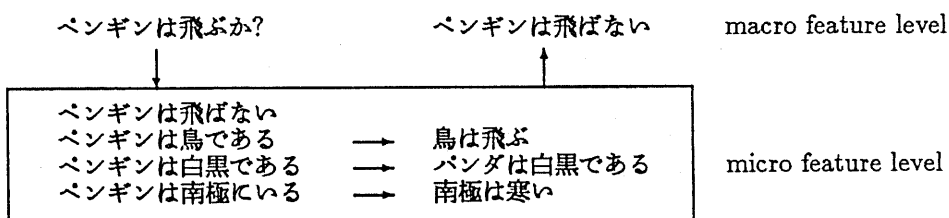


図 2.8: micro feature を使った推論

「ペンギンは ...」

という推論過程において、「鳥は飛ぶ」のルールは、Default Reasoningの言うところの

「たいていの鳥は飛ぶ」

という内容を意味し、「たいてい」をとってルール化することにより、誤った推論結果を導くことになる。Default Reasoningで記述するなら、

「ペンギンは飛ばない」

という特例を記述して、

——特例がない限りルールは成り立つ——

のようなメタルールで

「ペンギンは飛ばない」

という正しい結論を導き出すことができる。

経験空間では、このルールに対応するものを micro feature で表現する。

micro feature のレベルで、様々な知識が互いに活性・抑制を引き起こし、マクロから見ると簡単なルールに思えるような挙動をみせるという仮定に基づいている。

経験空間とコネクショニスト・モデルの違いは、そのキメの細かさにある。コネクショニスト・モデルが、ネットワークで表現する意味のある瞬間のノードのエネルギーとしてとらえて、その後の意味の動きを理解するのが、経験空間である。経験空間もコネクショニスト・モデルも無意識的な処理であるが、記号を用いるかどうかの違いである。

以上によって、無意識処理の集まりとして意識処理、すなわち、通常の推論能力が実現されることを示した。次に、文脈理解における無意識的な想起の役割について述べる。

文章を構成する各文は、まったくランダムな文ではなく、話し手の意図を表現するためにまとめられた「文脈」を持つ文である。そこで、各文を理解しながら、同時に無意識的な想起を起こすことにより、言及していない真実を表す文を想起することができる。たとえば、

「のどがかわいたね。ジュースはない？」

という会話には、

「ジュースを飲みたい。」

という話し手の意図が含まれている。前者の文の「のどがかわく」「ジュース」などをキーワードとして、無意識的想起を行うことにより、後者の文を発見することができる。また、このように文脈が見えてきた状況で、

「水をください。」

と頼まれた場合、

「じょうろに水を汲む。」

というような関係ない文は、意識にのぼらせる必要がないのは、明らかである。これは、無意識的な意味の広がりによって、探索空間を絞ることができる例となる。

以上、まとめると、無意識的想起の文脈に関する効果は、次の2点について期待される。

- 文章の意図するものを発見しやすくする。

- 探索の範囲を絞り込むことができる。

本章の議論は、自然言語理解のモデル化であった。その要約を以下にまとめる。

1. 文章の本来の意図を理解するためには、文脈を中心に理解を進めなくては行けない。
2. 文脈の理解のためには、過去の経験を蓄積する大きな知識ベースが必要である。
3. その中から、効率良く記憶検索を行うためには、人間の無意識的想起に対応する機能が必要である。

これらは、今後の自然言語理解システムを構成する上での重要な課題となるであろう。

第3章

経験空間による知識表現

自然言語の意図する内容を正しく理解するためには、様々な状況で文脈が利用される。逆に、文脈の与えられない状況では、その意味が唯一に決まらない文が存在することは、自然言語理解の難しさを示している。この問題を解決するには、まず、状況ごとに知識を取り出せるような知識表現について十分に検討する必要がある。本章では、文脈を扱う知識表現として「経験空間」を提案し、その振舞いを解析する。

3.1 経験空間の基本概念

従来の自然言語理解の研究においても、あい昧性を文脈的な要因に求めるものは数多く存在する。そして、その「文脈」が引用される場面も様々である。代表的なものでも、統語的なあい昧性の解決から、意味の多義性、比喩的表現の理解など、言語理解の様々の分野で取り上げられている。このような従来の「文脈」の扱いは、

「一文の中で取り扱えないものは文脈という付加情報によって解決する。」

という立場に立つものである。

しかし、「その文章によって本当に伝えたいもの」を理解したい場合、文脈をもっと積極的に利用しなくてはその答えは求められない。補助的な情報というより、文章の意味を決定するための本質的なものが文脈の中に存在する。これは、

「常に、文章全体の流れを重要視し、その中に個々の文の意味を見い出す。」

という立場に立った文脈のとらえ方である。

この後者の立場に立つ場合、まず、文章の流れを表現できる意味表現モデルを十分に検討する必要がある。Schank らは、文脈的な意味理解を支援する知識表現として

Players:	customer, server cashier
Props:	restaurant, table, menu, food, check, payment, tip
Events:	<ol style="list-style-type: none">1. customer goes to restaurnat2. customer goes to table3. server brings menu4. customer orders food5. server brings food6. customer eats food7. server brings check8. customer leaves tip for server9. customer gives payment to cashier10. customer leaves restaurant
Header:	event 1
Header:	event 1

図 3.1: Restaurant Script

script と呼ばれる知識表現を提案した。その中でも、restaurant script はレストランにおける典型的な事象の流れを理解する問題として有名である（図 3.1 参照）。しかし、

- 同じ「レストラン」を中心とした文章であっても、様々な状況が考えられ、例外に関する知識の表現が多く必要となる。
- 「レストラン」でない状況との間に共通な知識を使える場面が多い。たとえば、レストランでお金を支払うという行動は、一般的な料金の支払いの行動として理解できる。

などの問題点が生じた。例外を含んだ script の表現を図 3.2 に示す。そこで、

「文脈に関する知識は、一次元的な知識表現では、十分に表現できない。」

という反省が報告された。その後、その欠点を補うべく、Schank らの研究は MOPs (Memory Organization Packets) という知識表現の研究へと移行した。MOPs は、

- 小さな状況ごとに知識を表現する。
- 文章理解の対象に応じて、その知識を結合し、script に相当する知識を構成する。

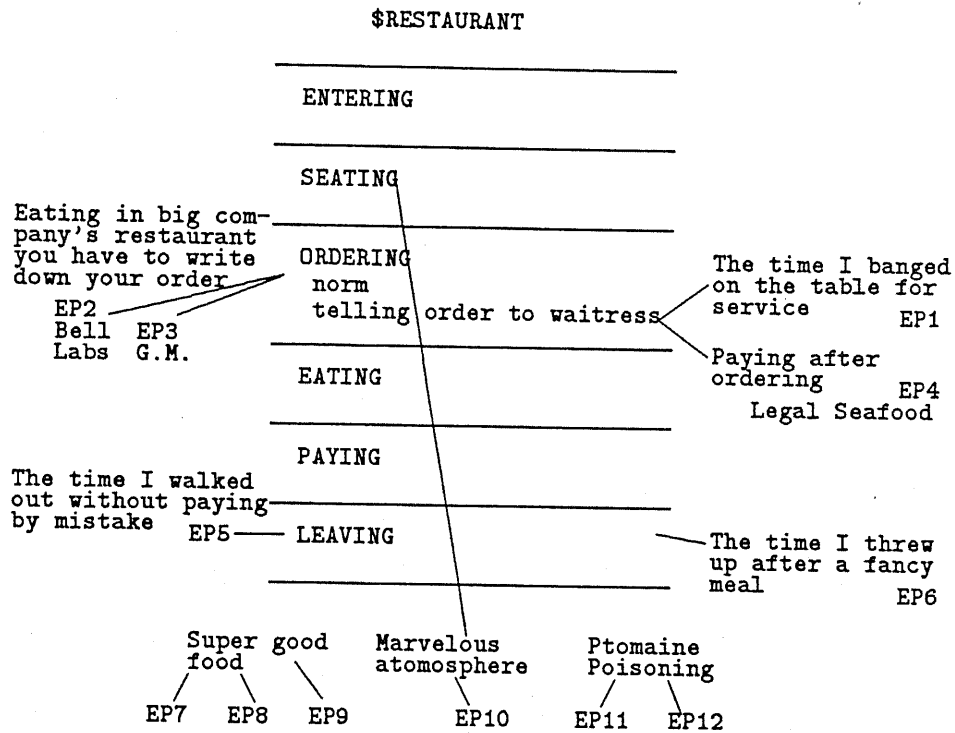


図 3.2: 例外を含んだ script の表現

という観点に基づいて発案され、その学習問題についても研究された。

本章では、この MOPs の考え方を発展させた「経験空間」という知識表現を提案する。経験空間では、状況を構成する最小単位を文に相当するレベルにおく。さらに、各文の間を関係性を記述するレベルを設定する。そして、文章を理解するとき、これらの知識をつなぎ、文脈に関する知識として組み立てる。

経験空間は研究目標のひとつに「能動的」な知識表現の実現をかかげている。従来の知識表現は、要求のある時だけに参照され、要求によって構成される「受動的」な知識表現であった。これに対し、人間の神経系における知識表現を例にとると、人間の意思で知識表現を操作する場合に加えて、知識の方が勝手に振舞う場合が多い。話の筋道とは直接の関係を持たない内容を連想により想起することもそのひとつである。この能力によって、状況に合った文脈的な知識の参照問題を解決していると思われる。経験空間においても、時折ボタンマッチによって文脈の一致性を調べるものより、知識自身が常に関連知識に対して働きかけ、その知識を活性化するような能動的な知識

表現のモデルが必要である。

これを実現するため、経験空間は意味ネットワーク上のエネルギーの伝搬のモデルを採用した。ネットワーク中のノードは主に、自立語、文、文間の関係に対して割り当てられ、それぞれを双方向リンクによって結びつける。そして、活性化エネルギー分布の時間変化によって文脈が表現される。経験空間は、知識表現であると同時に、文章の意味であるという立場を持ち、入力文章に応じて、経験空間の対応するノードに活性化エネルギーを与え、リンクを通してそのエネルギーを拡散させる。このエネルギー拡散により、関連知識の活性化が行われる。

経験空間も、文脈を扱う点から script と同じ問題を持つ。文脈は様々な状況で参照され、その状況に適した詳細な知識を与える情報源である必要がある。また、MOPs が学習問題を取り扱ったのと同じく、知識の獲得方法についても明らかにしなくてはならない。そこで、過去に入力された文章による経験に着目して、それを蓄積することにより文脈に関する知識を獲得する方法を採用した。経験の蓄積は、次の3通りの方法によって行われる。

- (1) ノード内に蓄積される内部エネルギー分布による経験量の保存
- (2) リンクを流れる活性化エネルギーの制御による経験の順序の保存
- (3) ネットワークの形態による構造の保存

(1)(2) は経験空間のノードとリンクに持たせる特性による記憶である。これに対して、(3) は文章をどのような形で意味ネットワークに展開するかにより決定される構造的な記憶である。本章では、まず、(1)(2)に関連する経験空間中のエネルギー伝搬に関する動作とその学習について述べる。その後で(3)にあたる、文章のネットワーク構造への展開方法について述べる。

3.2 神経回路網からのモデルの導入

ネットワーク内のエネルギー伝搬のモデルは、神経回路のニューロンのモデルから得るところが大きい。ニューロンは、多入力-出力（出力先は複数あるが、その出力値は唯一である）のプロセッシングエレメントとしてモデル化できる [2]。その入出力の関係は、一般に入力の重み付き線形和に対する関数の形で表現されることが多い。

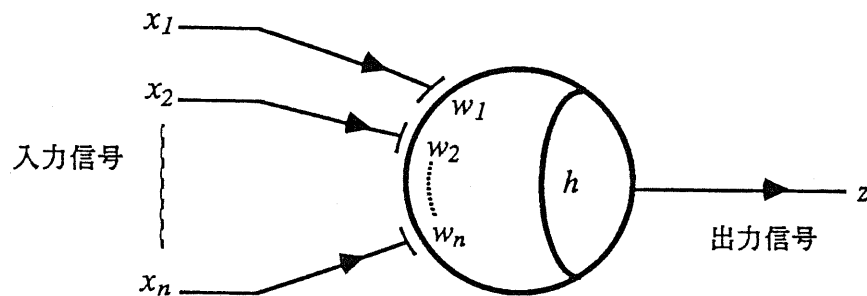


図 3.3: 神経回路のモデル化

入力信号 x_1, \dots, x_n に対して、神経の活性度を表す膜電位の変化 u は、式 (3.1) のように表される。 w_i は、他のニューロンとの結合の重み付けを表す。(図 3.3 参照)

$$u = \sum_{i=0}^{n-1} w_i x_i. \quad (3.1)$$

この u がしきい値 h を越えたとき、ニューロンは活性化状態にあるという。このときの出力 z を 1、そうでないときを 0 とすると、出力 z は、次の式で表される。

$$1(u) = \begin{cases} 1, & \text{if } u > 0. \\ 0, & \text{otherwise.} \end{cases} \quad (3.2)$$

$$z = 1\left(\sum_{i=0}^{n-1} w_i x_i - h\right). \quad (3.3)$$

$1(u)$ は、階段関数である。

近年、盛んに研究されるようになったコネクショニスト・モデルもこの式を各ノードの操作関数の基本モデルとして用いている。このノードをネットワーク状につなぎ、全体のエネルギーの安定状態を求めることによって各種のパターンマッチに成功している。しかし、この式自体はパーセプトロンの時代のモデルと大きく変わることはない。

文脈のような実時間に変化していくものを扱うためには、現在のコネクショニスト・モデルのアプローチでは十分でなく、基本的なニューロンのモデルに時間の概念を明確に含ませる必要がある。実際のニューロンの出力は、振幅による活性化エネルギーのモデルでなく、発火の頻度と発火の持続時間が重要である。そこで、これを「連続量-連続時間」のモデルで表現する。膜電位を $u(t)$ 、 i 番目のシナプスへの入力パル

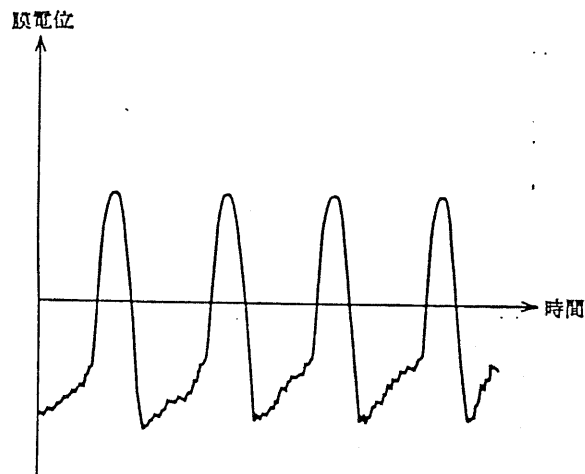


図 3.4: 神経細胞のパルス発生概念図

スの単位時間あたりの数を x_i 、重み付けを w_i とし、さらに $u(t)$ の減衰定数を τ で与える。ニューロンの基本方程式は、

$$\tau \frac{du(t)}{dt} = -u(t) + \sum_{i=0}^{n-1} w_i x_i. \quad (3.4)$$

と表すことができる。この式を

- しきい値を設定し、 $u(t)$ がしきい値を越えたところでその神経細胞は発火し、再び $u(t) = 0$ となる。
- 一端発火した細胞には絶対不能期があり、ある期間は続けて発火しない。

との条件の下で評価すると

パルス入力 → パルス出力

のニューロンモデルが実現される。パルス発生の様子を図 3.4 に示す。

このニューロンモデルをネットワークにして動かした場合、さらに、パルス発生頻度、パルス発生頻度の変化といったマクロな時間的変化が観察される。このマクロな振舞いを

パルスの頻度 = 静的な状況の認識
 パルスの頻度変化 = 動的な状況の認識

と対応させると、文脈認識に適用することができる。

3.3 活性化エネルギーの伝搬式

以上の考察を基に、活性化エネルギーの伝搬式をたてる。パルスの頻発生度でノードの活性度を測るモデルは取り扱いが難しいため、同様の効果を持つ振幅モデルを提案する。

$$T_i(t) = E_i(t) + I_i(t). \quad (3.5)$$

$T_i(t)$... 活性化エネルギー

$E_i(t)$... 外部エネルギー

$I_i(t)$... 内部エネルギー

$$T_i(t) - \theta > 0 \quad \cap \quad E_i(t) > 0. \quad (3.6)$$

θ ... しきい値

まず、ノードごとに（ノード i に対する）活性化エネルギー $T_i(t)$ を定義する。活性化エネルギー $T_i(t)$ は、減衰定数の小さい内部エネルギー $I_i(t)$ と、減衰定数の大きい外部エネルギー $E_i(t)$ に分割される。内部エネルギー $I_i(t)$ は、経験空間が過去に経験した知識の状態を表す。外部エネルギー $E_i(t)$ は、現在新しく入力された文章の持つ情報量である。活性化エネルギー $T_i(t)$ はそれらの総和であり、時刻 t においてノード i がどの程度理解の対象となっているかを表す量となる。式 (3.6) はノードの活性条件を表す。文章の流れの中で重要な意味を表すノードは、常に活性化の状態または、活性化に近い状態におり、活性化エネルギーは、しきい値の付近を上下する。

$$E_i(t+1) = \alpha E_i(t) + \sum_{j=0}^{n-1} \frac{\beta_j(t) a_{ij}}{\sum_{k=0}^{n-1} |a_{jk}|} E_j(t) - \beta_i(t) E_i(t) - \gamma E_i(t). \quad (3.7)$$

$$\delta_i(t) = \gamma E_i(t). \quad (3.8)$$

$$I_i(t+1) = \chi I_i(t) + \delta_i(t). \quad (3.9)$$

$$\beta_i(t) = \begin{cases} \beta_+, & \text{if node-}i \text{ is active.} \\ \beta_-, & \text{otherwise.} \end{cases} \quad (3.10)$$

$$\text{接続マトリクス } A = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix}. \quad (3.11)$$

式(3.7)によって時刻 t から時刻 $t+1$ における外部エネルギーの時間変化が記述される。式(3.7)の右辺第2項は他ノードからの入力項であり、第3項は他ノードへの出力項である。式(3.8)に示す学習エネルギー $\delta_i(t)$ は、入力文章から経験として獲得される学習成分を表す。この成分による内部エネルギーの増加、すなわち、知識獲得について記述したものが式(3.9)である。

外部エネルギーの減衰は、減衰係数 α と、学習成分として内部エネルギーに転換される係数 γ によって生じる。 $\beta_i(t)$ は、他ノードとの間の伝搬係数であり、ノードの活性化状態により、異なる2値をとる。すなわち、エネルギーを放出する側のノードが活性化している場合は $\beta_i(t)$ は大きな値 β_+ をとる。そうでない場合は小さな値 β_- をとる。ノードが活性化すると β_+ となり、周辺のノードへ大きなエネルギーを放出するようになる。逆にその活性化エネルギーがしきい値を越えて減衰した時点で、 β_- となる。結果として、その放出エネルギーは小さく抑えられ、しきい値より少し低い活性化エネルギーに安定することができる。これは、後の入力文章により再び活性化されやすい状態であることを示す。 χ は、内部エネルギーの減衰定数であり、 α より大きく、かつ1以下の値を持つ。

接続マトリクスの構成要素 a_{ij} は活性リンクでは正、抑制リンクでは負の値を持つ。取り得る値の範囲は -1 から 1 である。否定文のノードが生成されると、その文の要素との間に抑制性のリンクが張られ、ノードを通して負のエネルギーの拡散が行われる。リンクが張られていないノードの間の a_{ij} は 0 である。また、 a_{ii} も 0 である。

(自分自身へのリンクはない。)

この接続マトリクスの構成要素の値は学習によって決定される。事象 X , Y が連続して生じる経験をした場合、

$$X \rightarrow Y$$

の方向により多くのエネルギーが流れるよう学習されることが望ましい。これは、後に述べる経験の順序に沿った記憶検索を行う場合に、エネルギーの流れに方向付けをする効果を持つ。このとき、式(3.7)の a_{ij} は、

$$a_{ij} \neq a_{ji}. \quad (3.12)$$

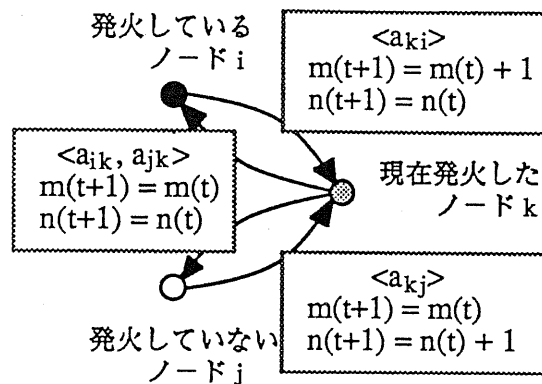


図 3.5: ノードの発火順序によるリンクの重み付け

となる.

この非対称なリンクの重み付けの学習を経験空間では次のように行う.

「あるノードが発火した時点において、そのノードに入力を与えるノードのうち、発火に寄与しているもの（そのノード自身が既に発火しているもの）からのリンクの重み付けを増やし、他方、発火に寄与していないノード（発火していないノード）に対応するリンクの重み付けを減少させる。」

上の X, Y の例では、X が発火している状況において Y の発火が起こるため、X → Y 方向の重み付けは強調される。しかし、その際、Y → X 方向の重み付けは変化させない。

a_{ij} の学習は次の式に従う。そのリンクが発火に寄与した回数を m 、寄与しなかった回数を n とする（図 3.5 参照）。

$$|a_{ij}| = \frac{1}{1 + \exp(0.1n - 0.2m)} \quad (3.13)$$

$|a_{ij}|$ と m, n の関係を図 3.6 に示す。リンク作られる最初の状態では、

$$m = n = 0. \quad (3.14)$$

であり、

$$|a_{ij}| = 0.5. \quad (3.15)$$

である。

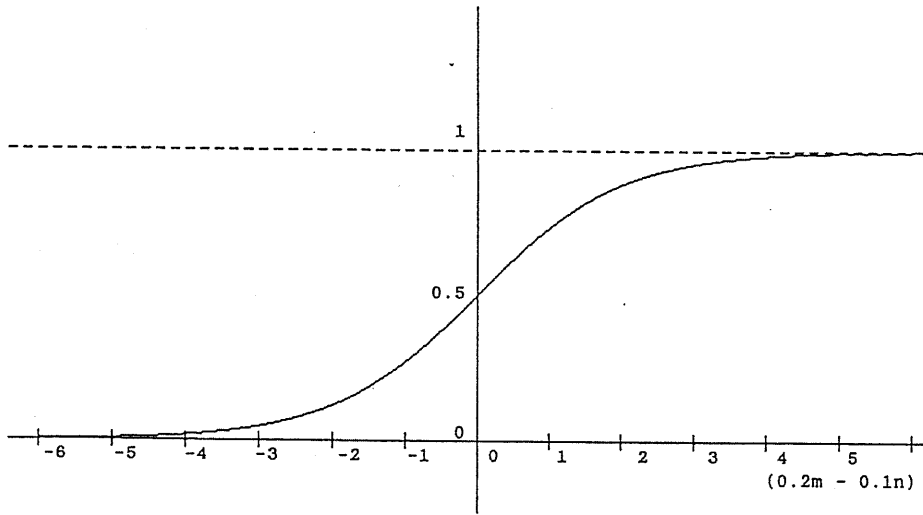


図 3.6: $|a_{ij}|$ と m, n の関係

表 3.1: 経験空間のエネルギー伝搬の定数 ($W = 1.0, P = 0.05$)

定数名	条件式	例
α	$\alpha \simeq 1, 0 < \alpha < 1$	0.999
β_+	$\beta_+ = P/W$	0.05
β_-	$\beta_- \simeq \beta_+/8$	0.006
γ	$\gamma \simeq 0, 0 < \gamma < 1$	0.0001
θ	$\theta \simeq W/4$	0.3
χ	$\alpha < \chi \leq 1$	1.0

定数 $\alpha, \beta_+, \beta_-, \gamma, \theta, \chi$ の値は、表 3.1 中に示す式により決定する。活性化エネルギーの最大値を W 、1 単位時間あたりの外界からの入力エネルギーを P とする。以後述べるすべての実験において、内部エネルギーの減衰は考慮せず、 χ の値は 1.0 とした。例として示した値を持ったノードのステップ入力に対する応答と活性化エネルギーが 1 である状態からの減衰特性を図 3.7 に示す。減衰特性を示すグラフでは、活性化エネルギーのしきい値のところ、 $\beta_i(t)$ の値が β_- に変わることにより、放出エネルギーが小さく抑えられる現象が観察される。

経験空間への文章の入力は、関連するノードへの活性化エネルギーの付与の形で行う。時間の経過とともにエネルギーはネットワーク上に拡散する。同時に、一部は内

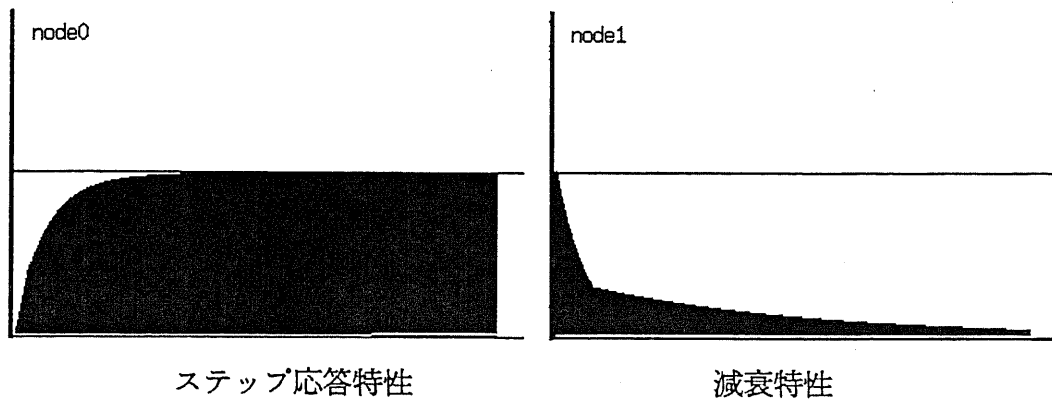


図 3.7: ノードのステップ応答特性と減衰特性

部エネルギーとしてノード内に蓄えられる。これは学習に相当する。多く経験を積んだ事柄については、内部エネルギーが大きくなり、小さな外部エネルギーでも、活性化するのに充分大きな活性化エネルギーを得ることができる。これは、頻繁に聞く内容は、理解するのが容易であることに相当する。

経験空間の初期状態と理解過程、学習の概念は図 3.8 のように示される。図 3.8 では、格子の交点がノードを、交点間を結ぶ線がリンクをモデル化している。ノードの持つ活性化エネルギーは図中上向き方向にとられている。初期状態において内部エネルギーの分布が与えられ、次に、入力された文章による特定ノードの活性化エネルギーの高揚が起こる。時間が経過すると、エネルギーが十分に拡散、減衰し、その一部がノード内に内部エネルギーとして蓄えられる。結果として内部エネルギーの分布状態は変化する。

最後に、エネルギー拡散の速度と外界からの刺激の入力速度についての検討を示す。人間の場合、神経回路自身の能動的な動作の速度は、文の入力速度に対して十分速い。その速度は、一文が入力されてから次の文が入力される間に、その入力文の静的なパターンマッチができる程度に速くなくてはならない。この静的なパターンマッチだけに注目するのがコネクショニスト・モデルである。経験空間の対象とするものは、このような静的な安定状態ではなく、むしろその後起こる意味の拡散の段階である。その拡散の速度は、文の入力の速さより十分速い必要がある。以後の実験では、外部からのひとつの入力に対して十回程度のエネルギー拡散を行った。

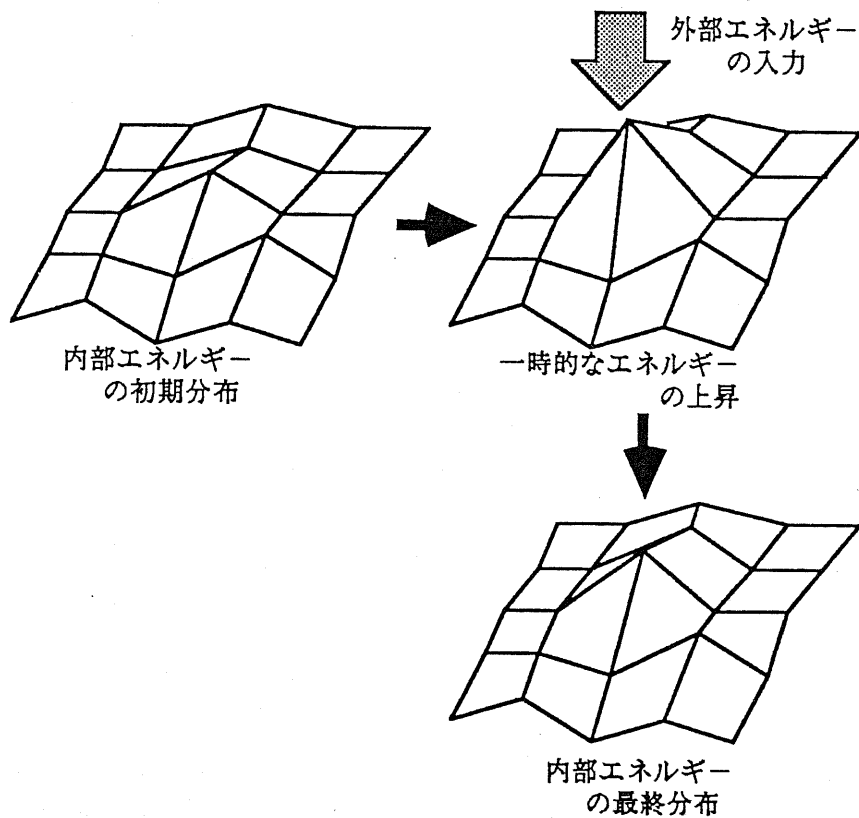


図 3.8: 経験空間における理解と学習のモデル

3.4 文章の経験空間への展開

経験空間は、意味ネットワークによって文章を表現する。この節では、文章をどのような形で意味ネットワーク上に表現するかについて詳細に検討する。

経験空間は、自然言語理解に文脈を積極的に取り入れるための知識表現である。ひとつには、文脈に関する知識を表現できることが必要であり、またもうひとつとして、自然言語の意味表現として利用できなくてはならない。自然言語によって意味されるものの最小な情報の単位は文に存在する。文脈は、この文の連なりであるだけでなく、文章の中に存在する話し手の意図を表現できなければならないと考える。この「流れ」「意図」は、経験空間の中のエネルギーの伝搬やエネルギーの溜まり場として表現できる。

そこで、経験空間が意味ネットワークという構造の知識として要求するものは

1. 文脈の最小構成単位である状況の表現

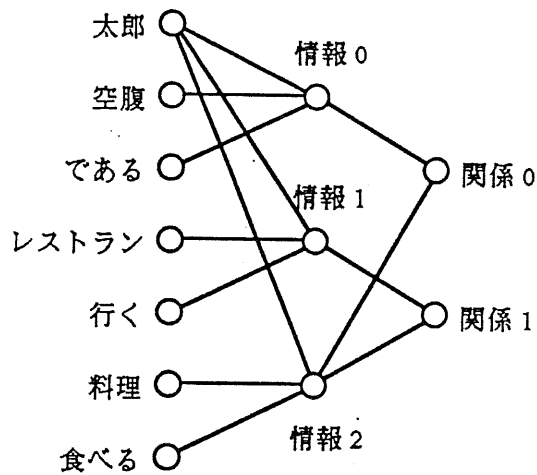


図 3.9: 経験空間の中の意味ネットワーク

2. その最小構成単位を関係づけるための情報の表現

である。これに、自然言語の中の自立語に相当するレベルを加えて、経験空間の意味ネットワークは、3つのレベルに属するノードと、その間を結ぶリンクによって構成される。経験空間の意味ネットワークの例を図3.9に示す。以下に、その各レベルのノード表現について順に述べる。

(1) 表象レベル

自立語をノードとして表現するレベルである。ただし、ひとつのノードがその自立語の持つ意味を直接に表現するとは考えない。自立語の意味は、次の(2)に示す情報レベルのノードを通して、その語を中心とした周辺ネットワークへの広がりを持ち、それら全体でその自立語の意味を表現すると考える。ネットワークの中の相対的な位置による表現、あるいは、ネットワークによる意味の規定という立場をとる。例として、図3.9中で「太郎」は、「レストランに行く人」であり、「空腹である人」であるというようにネットワークにより表現される。「太郎」を中心に置くと図3.10のように表すことができる。このような相対的な立場の単語の意味表現は、活性化エネルギーの拡散によって実現する。エネルギーの拡散される範囲がその自立語の意味を規定する範囲である。

(2) 情報レベル

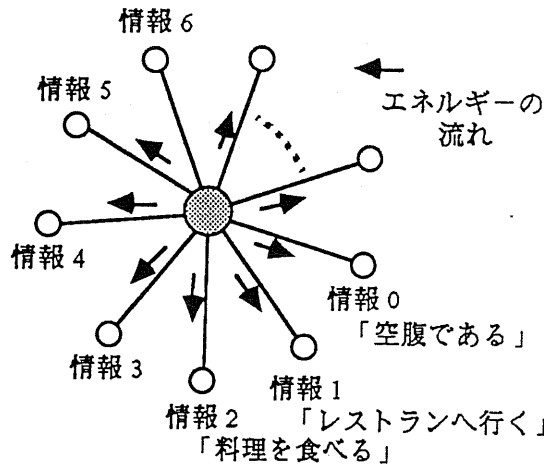


図 3.10: 太郎の意味の広がり

Action	生物または自然力が従事する活動.
Change-of-state	ある対象が一つの状態から他の状態へ変化するだけの事象.
Desire	人間が望み, 恐らくその結果として, 世界の, ある特別な状態を得ようとするような内部応答.
Emotion	感じの表現からなる内部応答.
Event	状態変化, 行動, あるいは状態変化や行動を引き起こすこと.
Plan	遂行すれば望ましい目的が達成されるようなサブゴールの作成.
State	ある対象または対象の集合間の定期的な関係の性質又は条件.

図 3.11: 物語文法の統語範ちゅうの定義

表象レベルのノードが, 自立語の存在点を表現するのに対し, この情報レベルのノードは, 自立語間の関係を表現する. 情報は複数の事物の間関係により生じるという立場からは, このレベルの記述が情報の最も基礎的な存在場所になる. 同時に, 文脈を形成するときの状況の最小単位でもある. ここに表現されるものは通常の文のイメージに近いものである. 文は格文法表現をもって, このレベルに記述される. さらに, 「赤い靴」という名詞句における「赤い」と「靴」の間に生じるような関係もここに表現する.

情報レベルとして表現される文はさらに次のように細分される. これは, Rumelhart の物語文法 [18] の統語範ちゅうを参考にしている. (図 3.11 参照) Rumelhart の分類

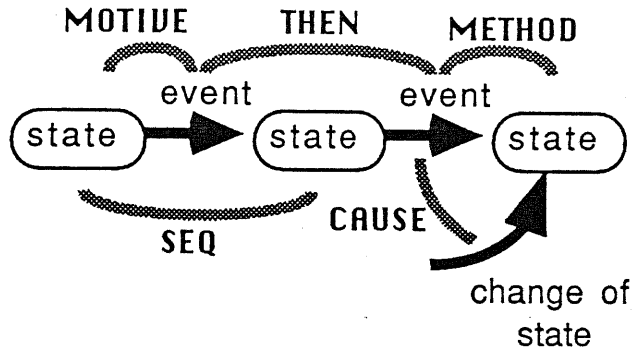


図 3.12: state と event と change of state の間の関係

は非常に細かいものであるが、ここでは、文章の表層構造から容易に理解できる程度ものに簡略化する。

state: 状態を表現する文
 event: 動作を表現する文
 change of state: 状態の変化を表現する文 (～になる)

このような分類を行うことにより、文脈の中の「静的状況」と「動的状況」を区別することができ、経験的知識の獲得の問題にも関連して、非常に扱いやすい形となる。

(3) 関係レベル

情報ノードと情報ノードの間に存在する関係を表現する。文脈を形成するとき、状況を結合するための核となる情報を提供する。時間的連続・因果関係などを表現するレベルである。情報ノードとの関係を含めて、図 3.12 に関係ノードの代表的なものを示す。

METHOD: event → state
 MOTIVE: state → event
 CAUSE: event → change of state
 THEN: event → event
 SEQ: state → state

比較のために、Rumelhart の物語文法を図 3.13 に示す。

-
- (1) Story \rightarrow Setting + Episode
 \Rightarrow ALLOW (Setting, Episode)
 - (2) Setting \rightarrow (States)*
 \Rightarrow AND (States, states, ...)
 - (3) Episode \rightarrow Event + Reaction
 \Rightarrow INITIATE (Event, Reaction)
 - (4) Event \rightarrow {Episode | Change-of-state | Action | Event + Event}
 \Rightarrow CAUSE (Event₁, Event₂) or ALLOW (Event₁, Event₂)
 - (5) Reaction \rightarrow Internal Response + Overt Response
 \Rightarrow MOTIVATE (Internal Response, Overt Response)
 - (6) Internal Response \rightarrow {Emotion | Desire}
 - (7) Overt Response \rightarrow {Action | (Attempt)*}
 \Rightarrow THEN (Attempt₁, Attempt₂)
 - (8) Attempt \rightarrow Plan + Application
 \Rightarrow MOTIVATE (Plan, Application)
 - (9) Application \rightarrow (Preaction)* + Action + Consequence
 \Rightarrow ALLOW (AND(Preaction, Preaction, ...),
 {CAUSE | INITIATE | ALLOW} (Action, Consequence))
 - (10) Preaction \rightarrow Subgoal + (Attempt)*
 \Rightarrow MOTIVATE [(Subgoal, THEN (Attempt,...))]
 - (11) Consequence \rightarrow {Reaction + Event}
-

図 3.13: 物語文法の統語規則と意味解釈規則

これらの関係ノードの名前付け (METHOD, MOTIVE など) は、ある状況ではその名前の持つ自然言語的な意味を反映していない。しかし、これらを経験空間内の関係ノードの呼び名として定義する。関係ノードとしてこの 5 つの関係を用いることにより、文脈の構成要素である状況間の関係がより明らかとなる。

一般的な意味ネットワークは、単語の意味の近さを表現するために単語間を直接にリンクで結ぶが、経験空間の表象レベルのノード相互間を直接のリンクで結合することはない。経験空間では、

単語の意味の近さはその両者の持つ属性的な情報の類似性に存在する

という仮定を用いる。近い意味を持った単語どうしは、自然に類似した状況で使われることが多くなる。すなわち、情報ノードを通じて多くの共通する表象ノードと結ばれる。そして、一方が活性化した場合、その情報ノードを通じて活性化エネルギーが

[ラベル名][エネルギーを与える回数]

または,

[ラベル名][エネルギーを与える回数]

[TAB] [リンク名][結合先のラベル名][エネルギーを与える回数]

図 3.14: 経験空間への入力文のフォーマット

```

@s1 10
    subject tarou 10
    complement kuufuku 10
    pred dearu 10

@e2 1
    subject tarou 10
    object ryouri 10
    pred taberu 10

@r1 10
    RN MOTIVE 1
    before @s1 1
    after @e2 10

```

図 3.15: 経験空間への文章の入力例

他方に流入することにより、意味的な近さが判断される。

経験空間へ入力する文章の書式を図 3.14 に示す。リンクが複数ある場合は、2 番目の書式の 2 行目の書式が繰り返される。

入力文章の例を図 3.15 に示す。情報レベルと関係レベルのノードは、必ずリンクを通して他のノードと結ばれるが、入力書式中の仮りの名前として、"@s1"のように表現される。ノードの種類によって、

state を表す情報ノードは、"@s" で始まるラベル

event を表す情報ノードは、"@e" で始まるラベル

関係ノードは、"@r" で始まるラベル

と分類される。行の最後に記された数字はそのノードにエネルギーを与える回数であ

り、エネルギー拡散に関係するパラメータである。強く認識したいノードには大きな値を与える。めやすとして、10程度の値を用いた。

3.5 経験空間の評価

3.5.1 経験空間の学習能力に関する実験

経験空間のエネルギー伝搬モデルの学習能力、記憶の再現能力を評価するために、意味付けを与えていない（文としての構造を持たない）ネットワークに関して以下のような実験を行った。

1. ノード数10の完全グラフを経験空間中に作成する。
2. 10のノードの中から、長さ4のノード列を2本用意する。実験3,4に関しては、さらにそれぞれの系列に対して文脈を提供する1ノードを選ぶ。
3. 2本のノード列を学習させる。すなわち、4ノードに順に規定の回数（10または20回）の間外部からエネルギーを与え、エネルギーの拡散と内部エネルギーの獲得、リンクの重み付けの学習を行わせる。実験3,4では、4ノードに順にエネルギーを与える間に、文脈を提供するノードにも3ないし6回ずつエネルギーを与える。
4. まず、ノード列の先頭ノードにエネルギーを与え、順に活性化するノードを観察する。最初はノード列の先頭ノードにエネルギーを与えるが、次に別のノードが活性化したら、そのノードを新しい焦点とし、それにエネルギーを与えるようにする。
5. 3., 4. を繰り返して、ノード列の学習の収束を確認する。

10のノードに番号を0から9までつけ、次のようなノード列によって実験を行った。（図3.16参照）

【実験1】

互いに交わらない2系列の学習を行う。

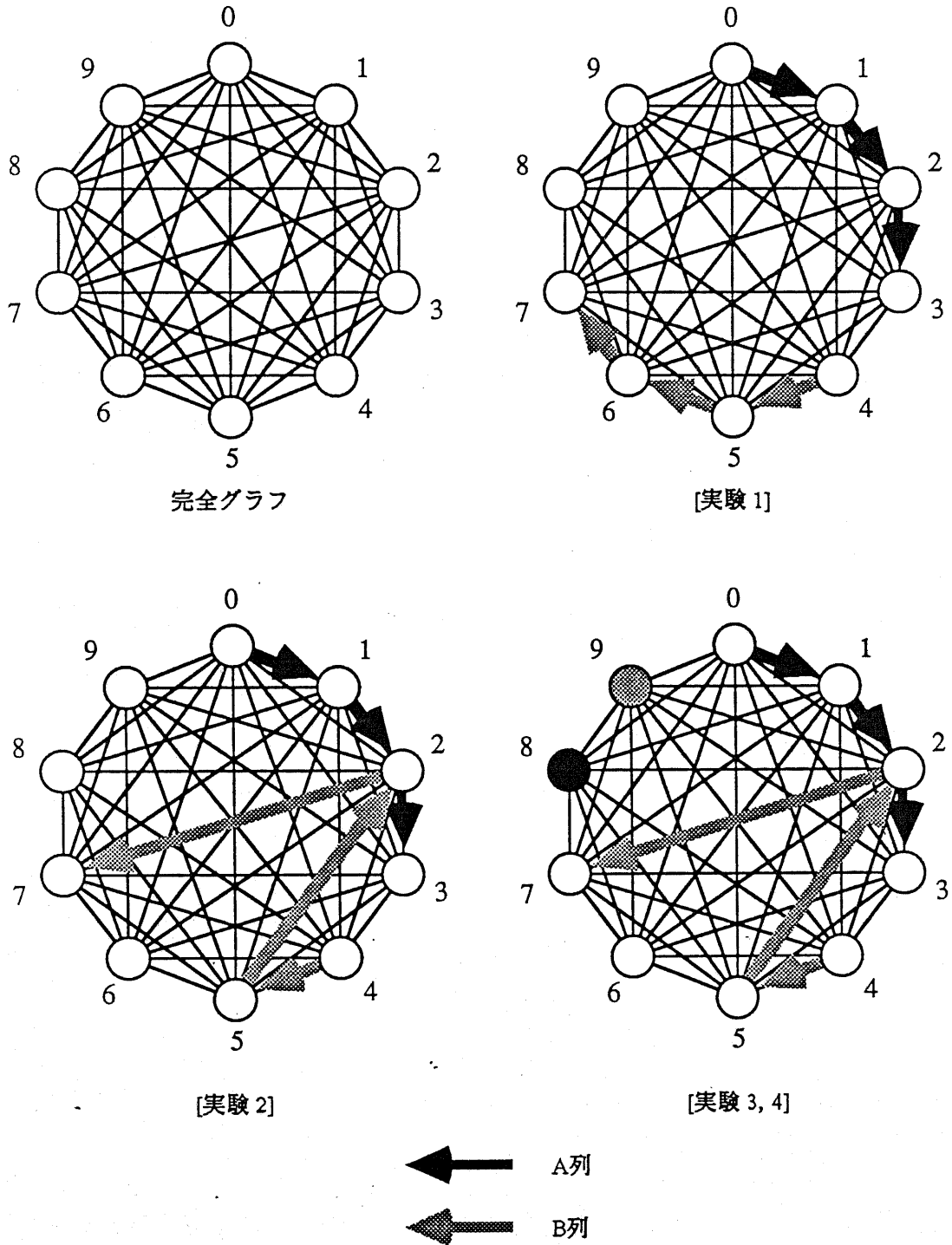


図 3.16: 経験空間の能力評価実験

A 列 0, 1, 2, 3

B 列 4, 5, 6, 7

【実験2】

途中に交わるノードが存在する2系列の学習を行う。

A 列 0, 1, 2, 3

B 列 4, 5, 2, 7

【実験3】

途中に交わるノードが存在する2系列の学習を行う。別に文脈を提供するノードを1つ用意する。

A 列 0, 1, 2, 3

文脈ノード = 8

B 列 4, 5, 2, 7

文脈ノード = 9

外部からのエネルギーの与え方は、たとえば

ノード0に10回

ノード8に3回

ノード1に10回

ノード8に3回

ノード2に10回

ノード8に3回

ノード3に10回

ノード8に3回

という形で行う。

【実験4】

実験3と同じ系列で学習を行う。ただし、経験空間の中に7 chunkの短期記憶を設定し、前に活性化したノードにも外部からの入力エネルギーの $(7-n)/7$ に相当するエネルギーを与える。(nは、いくつ前の焦点ノードであるかを表す。)

実験の詳細な結果を次に示す。

【実験1】

エネルギーを与える回数が10回、20回である2つの実験とも5順目以降に正しいA列、B列に収束した。

【実験2】

エネルギーを10回ずつ与える実験では、B列は5順目以降正しく収束した。A列はB列との混同が見られた。(5順目以降、0, 1, 2, 7に収束)

エネルギーを20回ずつ与える実験では、B列は6順目以降正しく収束した。A列はB列との混同が見られた。(4順目以降、0, 1, 2, 7に収束)

【実験3】

中心のノード列にエネルギーを10回、文脈ノードに3回与える実験では、正しい記憶検索への収束は認められなかった。

同じく、20回、3回与える実験では、A列は0, 1, 8, 3, B列は4, 5, 9, 7など、文脈ノードを含んだニアミス状態に収束した。

同じく、20回、6回与える実験も、ニアミス状態に収束した。

【実験4】

中心のノード列にエネルギーを10回、文脈ノードに3回与える実験では、A, B列共に6順目以降ニアミス状態を繰り返した。ただし、B列は8順目で、A列は11順目で、ニアミスの形をくずした。

同じく、20回、3回与える実験、20回、6回与える実験では、収束が見られなかった。

正しい順番の遷移を1.0点、逆順の遷移を0.4点、飛び越した遷移を0.7点、順番はともかく正しい列のノードが検索された場合0.3点、文脈ノードへの遷移を0.6点、相手の列の正しい遷移をした場合-0.5点としてグラフにしたものを図3.17に示す。(最初にエネルギーを与えるノードは決められているので、活性化ノードの遷移は3回起る。したがって、正しい記憶検索に対して満点の3点が与えられる。)

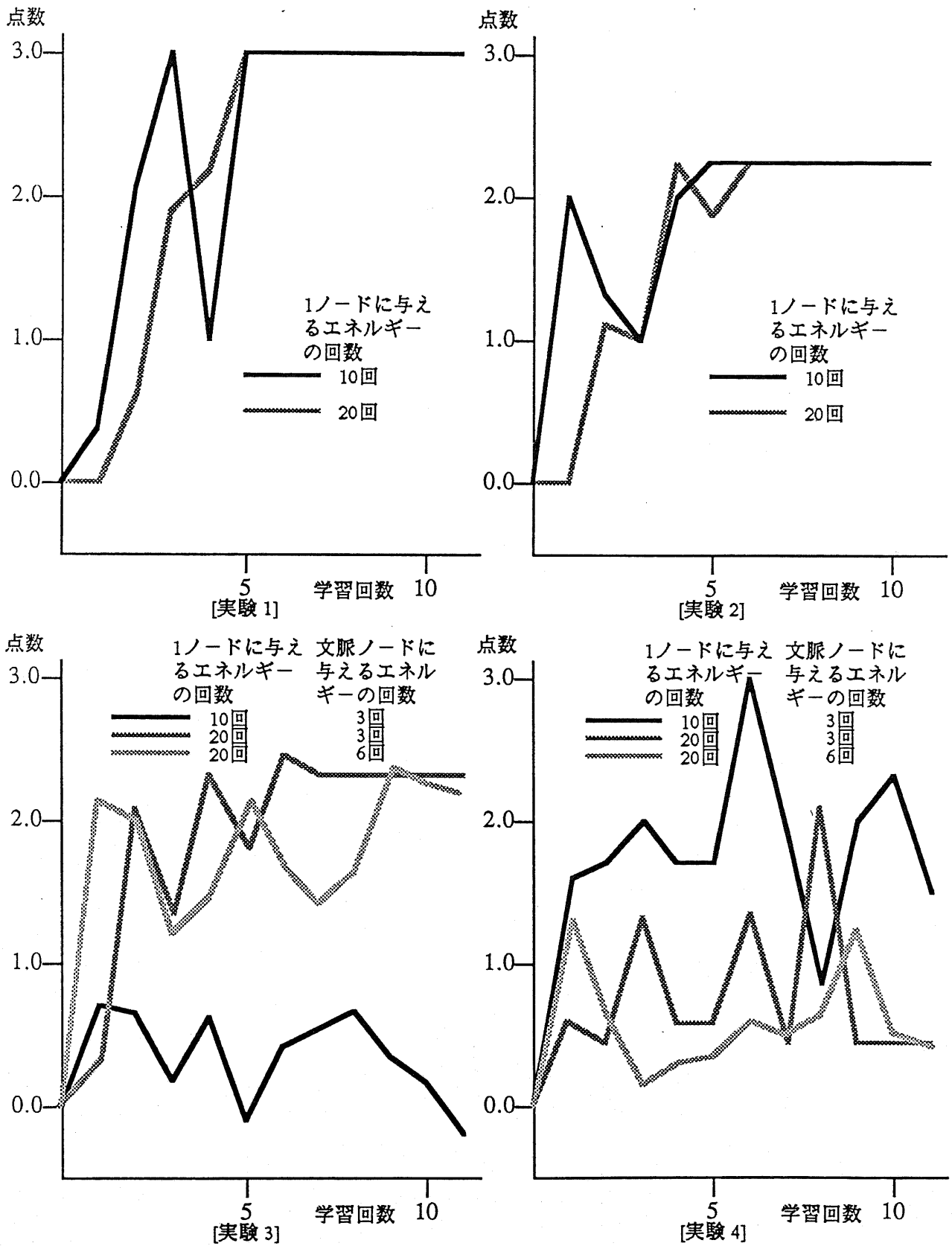


図 3.17: 実験結果

3.5.2 抑制リンクに関する実験

次に、抑制リンクの効果についての実験を行う。抑制リンクは、否定文の表現や、文と文の間の否定の関係の表現の場合に利用される。抑制リンクは方向性を持つ。たとえば、「太郎は料理を食べない」という文は、

```
@!s1 10
  subject tarou 10
  object ryouri 10
  pred taberu 10
```

というように表現される。ここで、"@!s1" が否定ノードを表す。この時のエネルギーの流れのうち、

```
tarou → @!s1
ryouri → @!s1
taberu → @!s1
```

は、正の伝搬が起こる。そして、

```
@!s1 → tarou
@!s1 → ryouri
@!s1 → taberu
```

は、負のエネルギー伝搬が起こる。すなわち、

格の内容は、否定文を表すノードを活性化する。
否定文のノードは、格の内容を抑制する。

という現象が起こる。たとえば、「太郎」が強く活性化している状況では、「@!s1」は、活性化の方向に進み、間接的に「ryouri」、「taberu」を抑制する。ここで、このような振舞いを経験空間上で実証する。

【実験5】

図3.18に示すようなネットワークを作る。node 10 が否定ノードである。負のエネルギーは、

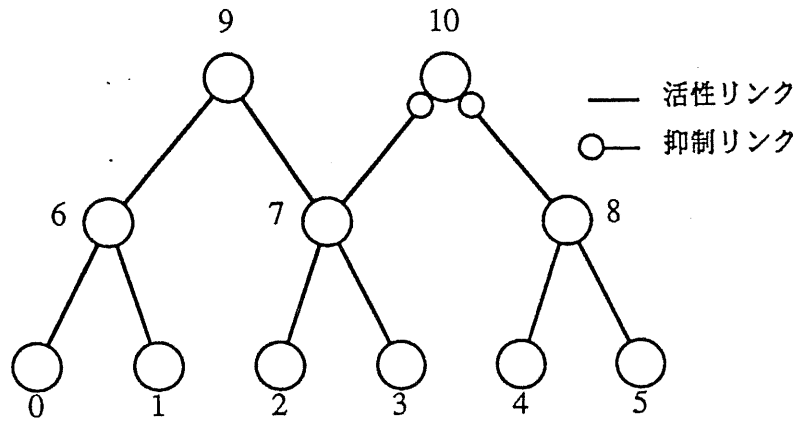


図 3.18: 実験 5, 6 のためのネットワーク

node 10 → node 7

node 10 → node 8

の方向に流れるはずである。初期状態として、すべてのノードの内部エネルギー、外部エネルギーを 0 にした状態から始めて、node 7 にエネルギーを与えていったときの各ノードの活性化エネルギーを振舞いを調べる。

【実験 6】

実験 5 と同じネットワークを使う。初期状態として、node 6, node 8 の外部エネルギーを 1.0 とし、その減衰を比較する。

結果のグラフを図 3.19, 3.20 に示す。図 3.19 では、否定ノードを通しては、活性化エネルギーが node 8 に流れ込まないことを実証している。むしろ、図 3.20 により、node 8 は、node 10 を通して、負のエネルギーによって、エネルギーが通常より速く減衰することが確認された。ネットワーク内の抑制効果は、このようなノードとリンクの特性で十分であることが示された。

3.5.3 経験空間の能力評価

以上の実験の結果より、経験空間の基本となるネットワークの能力に関して評価を与えることができる。

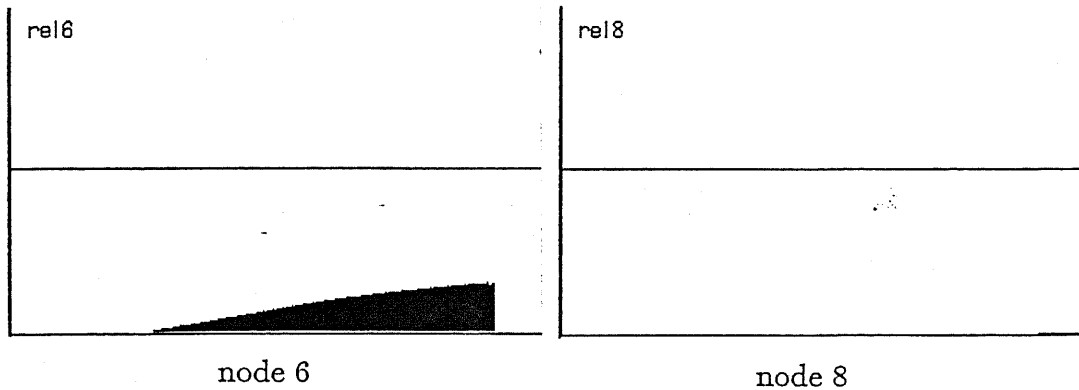


図 3.19: 実験5の結果

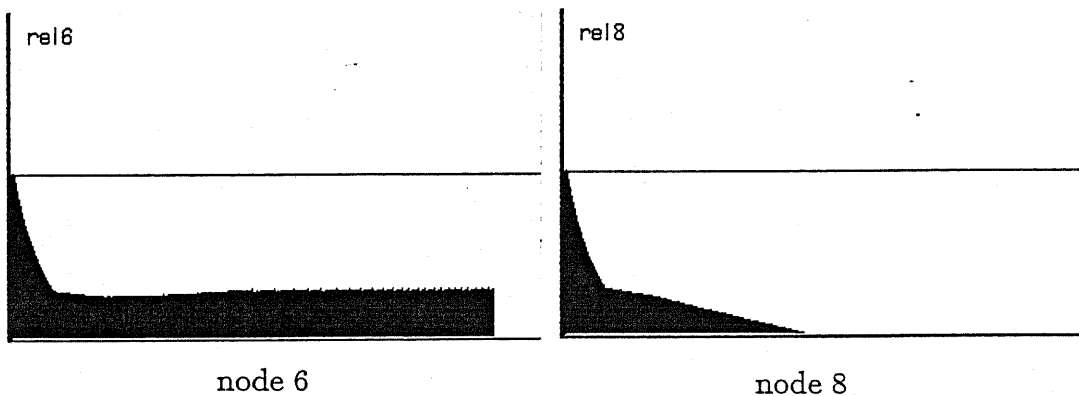


図 3.20: 実験6の結果

実験1は、共通要素を持たない文脈の識別能力を評価した。2つの相入れない系列の識別能力に関しては、経験空間の基本能力で十分であることが示された。ネットワークが、ある安定状態をとるのではなく、エネルギーの流れていく方向を学習するという点で、従来の神経回路モデルとは異なる特性である。

実験2では、共通要素を持つ文脈の場合における相手の文脈の影響について評価した。経験空間のエネルギー伝搬式だけでは、2つの文脈の交点で文脈が混同されることが示された。図3.17の評価点のグラフでは、高い点に評価されているが、長い文章の評価を行った場合、文脈の交差する点以降は、一方が他方の文脈に縮退してしまう

ため、さらに低い評価になる。したがって、あまり望ましい結果とは言えない。この実験により、文脈を提供するノードのない系列に関しては、経験空間のエネルギー伝搬式だけでは十分でないことが示された。

実際の文脈理解の状況をよく反映している実験は実験3, 4である。それぞれ、文脈を提供するノードを設定し、そのノードにエネルギーを与えることによって相手の文脈との交点でも文脈を混同せずに、自分の文脈の方向付けを失わないことを確認する実験である。

実験3では、実験1, 2と同じエネルギー伝搬能力を持つ経験空間を用いた。回復された系列の中に文脈を提供するノードが現われることがあり、期待する系列そのものの検索にはなりにくい。しかし、途中で相手の文脈に移ることは少なく、その点で評価される。

実験4では、実験3と同じ条件の学習系列に対して、7 chunkの短期記憶を持つ経験空間を用いて実験した。実験1,2,3では、焦点となっているノードに対して外部からエネルギーを与えるだけであるが、実験4では、以前に発火したノード（最大7個前まで）に対してもエネルギーを与えた。エネルギーを与えられる期間が、次の発火ノードと重なることによって、エネルギーの流れの方向が強調されることを期待した実験である。結果としては、実験3に比べて、その記憶検索能力は不安定なものであった。原因のひとつとしては、ネットワークの規模が小さいため、7 chunkの記憶容量によるエネルギーがネットワーク内にあふれすぎたことが推測される。

実際の文章理解においては、文脈ノードのような明確に文脈を与えるものは存在しない。文章の流れの中のエネルギーの溜まり場を仮想的に表現するのが文脈ノードである。したがって、実験3の結果は、理想状態における文脈の学習である。これに対し、実験4は、このエネルギーの溜まり場をネットワーク自らが作り出すことを意図して行った。実験では、ノード数が10の完全グラフのネットワークを用いたが、自然言語理解に用いる経験空間は完全グラフではなく、関連する知識の間だけをリンクで結合する。したがって、完全グラフに比べて、文脈に依存したエネルギーの溜まり場がでしやすい。よって、実際の自然言語理解では、この7 chunkの短期記憶を利用したモデルがより有効に働くことが期待される。

ひとつのノードにエネルギーを与え続ける回数は、20回より、10回の方が全体に

よい結果が得られた。20回与える方が内部エネルギーの学習は大きい、その効果以上に、外部に流れ出すエネルギーによる文脈の方向性を失う効果が大きいようである。唯一、実験3においては10回エネルギーを与える実験の方が悪い成績を示した。これは、途中で文脈ノードへのエネルギー供給をはさむことにより、ノード列上のひとつ前のノードとの関係が薄れたためと思われる。

実験5, 6は、抑制性のリンクに関する実験である。この実験の状況は、ある文とその否定の文の間に存在する排他的な関係や、否定文における格の内容を表す語の関係(「太郎は空腹でない。」という文がある場合、「太郎」は、「空腹」を抑制する。)を表現している。実験5では、否定のノードを通しては、その先にエネルギーが流れないことを示した。実験6では、さらに活性化エネルギーを持ったノードに対しては、そのエネルギーを積極的に減少させる効果を持つことが示された。

以上、6個の実験を行った結果である。この実験で扱われたネットワークは、意味付けされてなく、経験空間の「エネルギー伝搬能力」に関する実験であるといえる。次章では、経験空間の自然言語理解への応用を示すが、それに先立って、エネルギーの伝搬、学習だけからみても経験空間は十分な能力を持つことが示された。

第4章

自然言語理解システム FOCUS3

本章では、経験空間を知識表現として用いた自然言語理解システム FOCUS3 (Frame Oriented Context Understanding System version 3) の構成について述べる。経験空間を用いることにより、意味の広がりをもって文脈を理解できることを示す。

4.1 FOCUS3 の全体構成

第2章で述べた自然言語理解モデルを基に、無意識的想起を中心に文脈を形成する自然言語理解システム FOCUS3 の試作を行った。

FOCUS3 は、知識表現に経験空間を用いた。前章で示したように、経験空間は能動的な知識表現である。文章理解の進行とともに、自らも連想記憶によって周辺の知識を思い起こすことができる。これによって、人間の無意識的想起を模擬することができる。この無意識的想起が、文脈に関する知識を参照することに役立つ。また、文章理解表現という立場からみると、無意識的想起を含めた活性化エネルギー分布によって、理解された文章の構造を表現することができる。

FOCUS3 は構文解析部の一部を除いて SUN4/260 の上に C 言語で記述されている。構文解析部は、SUN3/140 の上に Flavor を用いて記述されている。文節以下の構文解析は、C 言語に移植を行った。全体像を図 4.1 に示す。システムの持つ知識は、経験空間により表現され、また理解した文章の意味も経験空間に表現される。

入力文章はパーザを通してこの経験空間に展開される。ただし、実験の入力文としては、格文法に基づいて解析された文を経験空間に与えた。入力文はシステム内に存在する知識と比較され、一致した場合は、そのノードの外部エネルギーを高める。また、一致する知識がなく、新しい知識と判断された場合は、経験空間に新たなノード

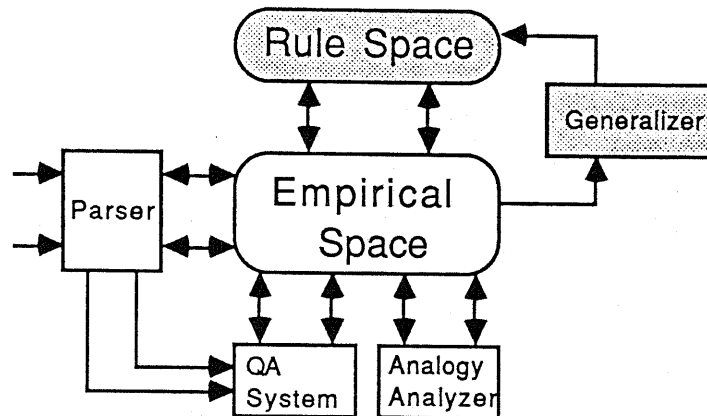


図 4.1: FOCUS3 の全体像

を作成しリンクによって既存のネットワークと結合した後、そのノードに外部エネルギーを与える。新しく作成されたノードの内部エネルギーの初期値は 0 である。

経験空間は、意味ネットワーク上のエネルギーの分布により様々な意味を表現する。従って、理解された構造の詳細を見るためには、ノードの活性化エネルギーとノード間の接続リンクをすべて提示しなければならない。この問題を避けるためにシステムが理解したかどうかを簡単に確認する方法として、グラフによるノードの活性化エネルギーの表示機能および、簡単な質問に対する応答を行うユーティリティを作成した。表示機能は X-window を用いて記述した。マウスにより指定したノードの活性化エネルギーの時間変化をグラフによって直接に観察できる。(図 4.2 参照) 質問応答システムは、格文法表現された

- (1) 事実の真偽を問う疑問文
- (2) 未知項目を含む文に対する未知項目の内容を問う疑問文
- (3) 理由、手段を問う疑問文

に対して応答が可能であるよう作成した。

図 4.1 中、網掛けで表現されている部分は経験空間を中心とした統合的自然言語理解システムへの拡張を示す。ルール空間には演えきの推論を行うトップダウンな知識

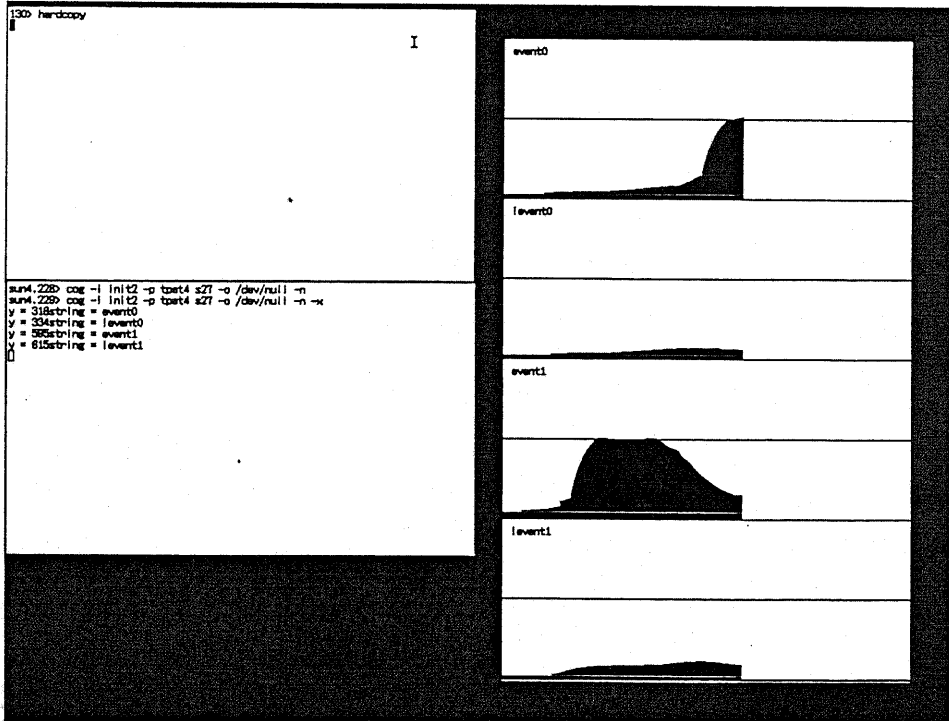
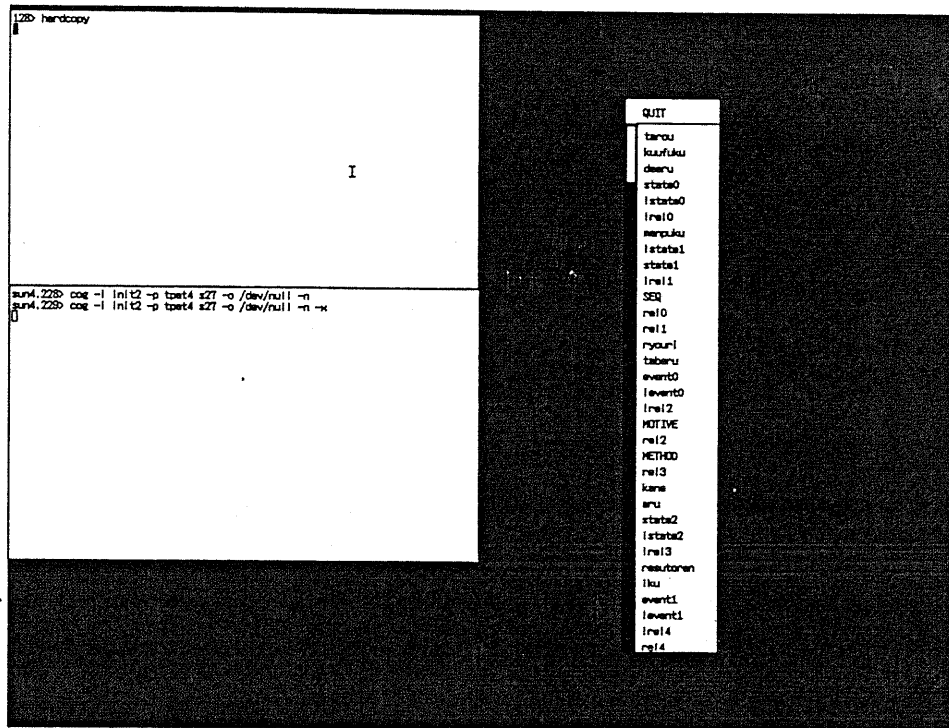


図 4.2: X-window を使った経験空間の観察ツール

を記述する。経験空間の無意識的想起に対する意識的な推論を行う。ルール空間の知識の前提条件を満たした場合に、その知識の帰結にあたる文に向けてエネルギーを流し込む形の実装が望ましいと思われる。

4.2 経験空間による文章理解

本節では、経験空間を用いた文章理解について述べる。また、経験空間は文章理解を行ないながら、一方で知識の獲得を行う知識表現であるため、本節の記述は知識獲得の状況にもそのままあてはめることができる。

4.2.1 入力文章の意味ネットワークへの展開

経験空間には、格文法の形で解析された文章を入力する。(図 3.15 参照) 解析された文章の中の自立語は表象レベルのノードとして、また、文としての集まりは、情報レベルのノードとして経験空間に表現される。問題として残るのは関係レベルのノードの認識である。人間は、この認識を次の 2 つの方法によって行っていると考えられる。

- 文の表現の中から発見できるもの。たとえば、接続詞によって関係が明らかなものや修飾関係にある文章。
- 直接に表現されていないが、経験的にその関係を認識できるもの。

前者は、構文解析の段階で認識できるものであり、経験空間の入力として明示的に扱う。修飾語関係の入力例を図 4.3 に示す。一方、後者の場合、その関係を明示的に経験空間に与えることは、本来の言語理解の立場を損なうものになる。経験空間では、第 3 章に述べたような、METHOD, MOTIVE, THEN, SEQ, CAUSE の関係ノードを対象となる文の種類 (state, event) に応じて設定する。それ以上の意味的に深い関係 (因果関係など) を発見することは難しい問題であるため、原則的には入力文章として明示的に与えない。ただし、

「知らない知識に関しては、いつまでも知らない。」

のは当然であるため、経験を積むための入力文章として、これら深い関係の記述を与えることもある。経験として知っている事実ならば、後の入力文章にはその文の関係

```

@s0 10
      subject jirou 10
      object kibun 10
      pred motsu 10

@s1 10
      subject kibun 10
      complement warui 10
      pred dearu 10

@r0 10
      RN ADJ 1
      adj @s0 1
      noun @s1 1

```

図 4.3: 修飾語の表現

が記述されていなくても、経験空間にエネルギーが流れることによって認識が可能である。

入力文章は、既存の経験空間のネットワークと比較され、新しいものに関してはネットワークの拡張を行う。以下に特殊な経験空間の拡張に関して説明を加える。

(1) アナロジーによる経験空間の拡張

経験空間内にエネルギーを伝搬させることによって文章理解の補助を行う仕組みは、過去の経験と非常によく似たでき事を再び体験するとき有効であるが、逆にまったく新しいでき事に関しては効果的に動作しない。例として、経験した文と動作主名が違っても、異なる文と判断され、直接的に関係する知識を活性化することはなくなる。(ただし、目的語などに共通なものが現われれば、そのノードを通したエネルギーの伝搬は行われる。ここでいう「直接的」とは、経験知識の中の文と一致した場合、その文の次にくる文や因果関係を持つ文が関係ノードを通して活性化されることをさす。)

そこで、これを補うものとして、アナロジーを利用した経験空間の拡張を行った。ここでいうアナロジーとは、Winstonの定義する

「類似性は因果関係を保存する。すなわち、似た状況は似た結末を生じやすい。」

を原則 [25] とする。意味ネットワーク表現において、状況は表象ノードおよび情報ノードを中心とした広がりによって表現され、因果関係は関係ノードによって表現される。このような隣接するノードの類似性を発見して、アナロジーを起動する基本要素として利用することができる。

経験空間に、過去に経験されていない新しい文を入力したとき、このアナロジー機能が起動される。類似性の検索範囲は、その時点で活性化しているノードに限定することができる。アナロジーによる拡張のアルゴリズムを次に示す。

1. 入力された文が経験空間中に存在するかを調べる。
2. 新しい文であった場合、主語、目的語のうち、ひとつが異なるだけで、似た記述の文が経験空間中に存在するかを調べる。ただし、活性化していない文はこの検索の範囲に含めない。
3. 似た文を見つけた場合、その文をとりまく状況と、入力文をとりまく状況を比較する。(それぞれ、文を表現する情報ノードにつながる関係ノードを検索し、その先につながる情報ノードを検索する。このようにして検索された情報ノードがその文をとりまく状況である。)
4. とりまく状況に共通要素あるいはそれに近いもの(2.と同様に主語、目的語だけが異なる文をさす)がみつかった場合、その状況は類似した状況であると判断する。
5. 類似した状況と判断し、相手側の類似した状況に隣接する情報ノードと同様の情報ノードを入力文の周辺に作成する。このとき作成する情報ノードは、主語、目的語などの置き換えを行い、現在の状況に合った形にして作成する。新しく生成されたノードの内部エネルギーは0である。

簡単な例として「太郎」で得た知識を「次郎」の状況において適用する例を図4.4に示す。経験空間は「太郎」に関して十分学習された初期状態を持っている。これに「次郎」を含む状況が入力された時、「太郎」「次郎」いう固有名を示す表象ノードを除いた状況の類似性を見つけ、「太郎」の周辺の知識を「次郎」の周辺に構築することができる。状況の類似性は、その自立語(表象ノード)の持つ属性・関係を表現する

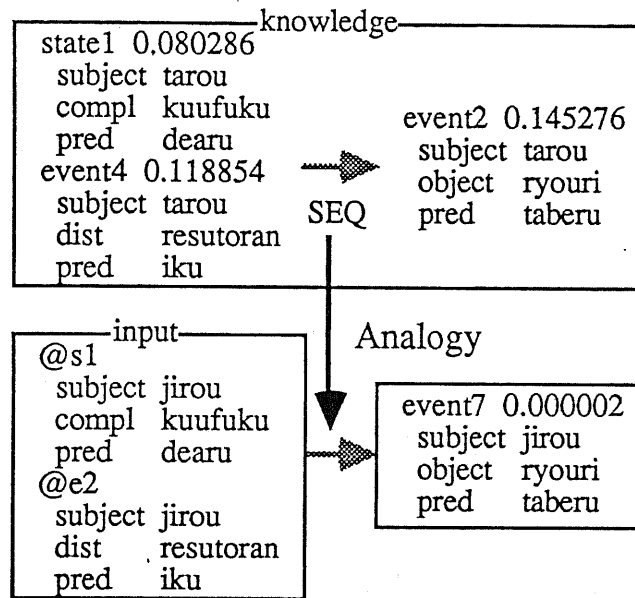


図 4.4: アナロジーによる経験空間の拡張

(情報) ノード・リンクの一致により判断される. 図 4.4 の例では, 「太郎」を束縛する状況 state1 (太郎は空腹である), event4 (太郎はレストランに行く) と, 「次郎」に関する入力状況 @s1 (次郎は空腹である), @e2 (次郎はレストランに行く) の一致により, 「太郎」に関するその状況の帰結である event2 (太郎は料理を食べる) に対応する event7 (次郎は料理を食べる) をアナロジーによって推論し, 経験空間の拡張を行う.

(2) 対をなす否定文を用いた拡張

入力された文章の「文」としての構造は, 経験空間の情報レベルのノードにより表現される. この時, 情報ノードは自分自身の否定文 (否定文入力に対しては肯定文) と対をなして生成する. この対をなした情報ノードは, それぞれ, その文の格の内容として同じ表象ノードに対してリンクを持つが, 肯定文の情報ノードの方は活性結合, 否定文の方は抑制結合する. これは, 第3章の式(3.7)の a_{ij} の符号を+, -にすることに対応する. さらに, この両者の情報ノードは, 相反する事象であることを表現する関係ノードとの間に, 抑制結合のリンクで結合される. したがって, 一方の文が活性化した場合, その反対の意味を表現する文の活性化エネルギーは抑制される. これ

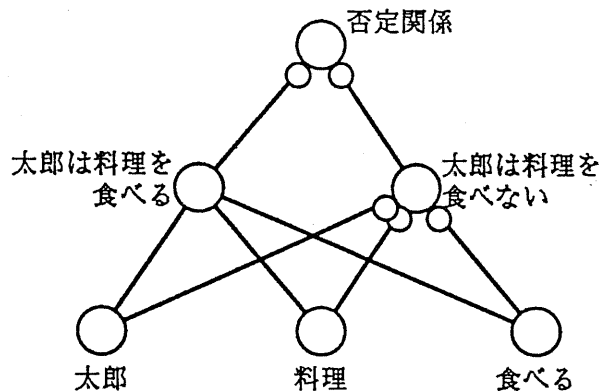


図 4.5: 対をなす否定文との関係

を図 4.5 に示す.

このように否定文に関する知識を経験空間に展開するのは一見不自然である。しかし、ある事実に対して真偽を述べたい場合、その事実の信頼性の軸が必要であり、その両極端を表現する情報ノードを用いて表現できる利点を持つ。また、事実の真偽が明らかになった時は、真実を表現する方の情報ノードが活性化し、他方のノードを抑制する関係を持つ。このような排他的な関係を人間はメタな知識として持っている。そこで、この否定文と対をなした表現は自然であり、かつ妥当な知識であると思われる。たとえば、「空腹である」状態から「空腹でない」状態に移った時、後者の知識は、前者の知識を抑制する。どちらの情報ノードが今現在の状況をより反映しているかをみるには、両者の活性化エネルギーを比較し、値の大きい方を選ぶことにする。後に述べる質問応答システムや長期記憶の検索問題は、この対をなす情報ノードの構造を持つことによって、正しい動作する。

4.2.2 入力文章の時間関係の保存に関する検討

システムに入力する文章の時間的な関係に対する問題点について述べる。図 3.15 に示したように、入力文章は、次のような順序で記述される。

文1

文2

関係1 (文1 と文2 の関係を表す)

経験空間では、活性化の順序関係によってリンクの重み付け (式(3.7)における a_{ij}) を変化させるため、このままでは次のような現象が起こる。

- 文1 から関係1 に向かうリンクは、エネルギーの流れやすい大きな a_{ij} を持つ。
- 関係1 から文2 に向かうリンクは、時間順序が逆転するため、 a_{ij} は小さくなり、エネルギーの流れが少ない。

この結果、文1 を活性化した場合、関係1 にはエネルギーが流れ込むが、文2 へはあまりエネルギーが流れ込まなくなる。これでは、3.4 節の実験で示された経験空間の時間列の想起能力を十分にいかすことができない。しかし、文2 が経験空間内に認識された後でないと、関係1 を作り出すことはできないので、文2 と関係1 を与える順序を変更することはできない。

そこで、経験空間の入力部分を改良して、時間的な記述に関しては、エネルギーを与える順序を意図的に操作できるようにし、この問題を解決した。図3.15 に示した例について、そのエネルギーを与える順序を詳細に記述する。

1. @s1 の文の格の内容 (subject, complement など) にエネルギーを10回ずつ与える。
2. @s1 という文を表すノードを作成し、それにエネルギーを10回与える。
3. @e2 の文の格の内容 (subject, object など) にエネルギーを10回ずつ与える。
4. @e2 という文を表すノードを作成し、それにエネルギーを1回与える。
5. 関係 @r1 の構成要素として、@s1 にエネルギーを1回与える。
6. (ここで、本来の経験空間の入力文章の解釈では、@e2 にエネルギーを与えるれる。) @s1 と @e2 の間に関係 @r1 を作成し、@r1 にエネルギーを10回与える。
7. @e2 にエネルギーを10回与える。

この結果、ノードの活性化する順序は、

$$@s1 \rightarrow @r1 \rightarrow @e2$$

となり、それぞれを結合するリンクの重み付け a_{ij} の学習は正しく行われる。したがって、できあがった知識を利用して文章を理解する時には、@s1 から @r1 を通して @e2 にエネルギーが流れやすくなり、文章理解を文脈的な立場から補助する経験的知識となる。

このほか、内部的な焦点の移動をより明確にさせるために、文章理解をする過程において、最近発火した7個のノードに対しては、入力文からくるエネルギーとは別にエネルギーを積極的に与えるよう実装した。これは人間の短期記憶に相当する記憶容量である。ノードに外部エネルギーとして蓄積されたエネルギーによっても短期記憶を実現できるが、この7ノードへの積極的エネルギー付与により、記憶構造の能動的な性格づけがより明確になる。

4.2.3 経験として与える知識の標準形

FOCUS3 は、文脈に関する知識として経験空間を参照しながら文章理解を行う。この経験空間に与える知識であるが、理解する文章に合わせて、意図的な知識を作成して FOCUS3 に与えてしまうことも可能である。しかし、そのためには、意味ネットワークの各ノードの内部エネルギーやリンクの重み付けなどのパラメータをすべて指定し、もっともらしい経験空間を作成しなければならない。この場合、パラメータをどのように設定すればよいのか不明確であり、また、その作業量は膨大なものになると予想される。それ以上に、もともとの経験空間の思想に反することにもなる。

経験空間を提案した最初の思想は、

「文脈に関する知識は、経験的なものから生み出される。」

というものである。したがって、システムは文脈に関する知識を強制的に与えるのではなく、ちょうど人間の子供が経験を積んでいくように、自らの経験によって知識を構築していくモデルを考えている。最初は文脈に関する知識のない状態で、いくつか文章を入力し、理解していく中で徐々に構成されていく経験空間を、後に文脈に関する知識として利用するという立場をとる。

そこで、その最初の段階で、どのような経験を積ませたら、効率的に経験空間が構成できるかについて本節で述べる。

まず、人間の場合を比較対象として検討する。人間に生得的な言語学習能力があることはよく議論されることである。その能力は、基本的な文法の獲得能力などにとどまるものであり、その上に成り立つ多くの知識は経験によって構成される。異なる地方に育てば、その環境で言語が獲得される。この言語獲得の環境に似せた環境を模倣的に自然言語理解システムに与えれば、同様な経験的知識が構成されることが期待できる。

子供の頃にふれる文章としては、実生活での会話や絵本の中の文章などがある。これらは、それが文である以前に、ある状況のもとにその文が与えられていることが重要である。実生活の体験は、目や耳などの五感を通して感じられる環境のもとに行われる。絵本に興味をもつのも、大人のように文字だけの情報では十分に想像力が発達していない、あるいは想像するだけの知識を持っていないからであると考えられる。絵によって状況が与えられ、その中で初めて文の意味を理解できる。

ところが、今、普通の生活の中で受け取る文章は、その状況に関する情報が省略されているものが多い。4人の被験者について、レストランに関する物語を作らせてその内容を語らせたところ、動作に関する報告がその大半を占め、動作の前後の状況に関する報告が少なかった。話題を転換する時だけ、状況を設定する文を述べるという特徴が見られた。すなわち、通常我々が用いている文章は、次のような特徴がある。

1. 連続する動作を述べることによって、徐々に相手の中に自分の伝えたい状況を作り出す。
2. 話題を切り替えるような場合だけは、相手がわかりやすいように新しい状況の設定を行う文を用いる。
3. 聞き手は、自分の経験と照合して、相手の話の状況を再現する。

これらの考察から、経験を構成するために与える文章として、日常使っている文章をそのまま利用するだけでは不十分であることが明らかとなった。経験として与える文章は、次の条件を満たすものがよい。

1. 動作を述べる前に、その状況を示す文を与える。

```

@s1 10
    subject tarou 10
    complement kuufuku 10
    pred dearu 10
@e2 1
    subject tarou 10
    object ryouri 10
    pred taberu 10
@r1 10
    RN MOTIVE 1
    before @s1 1
    after @e2 10
@s3 1
    subject tarou 10
    complement manpuku 10
    pred dearu 10
@r2 10
    RN METHOD 1
    before @e2 1
    after @s3 10
NULL 30

```

図 4.6: 経験として与える知識の標準形

2. 動作の後の、帰結の状況についても記述する.

この条件を満たす文章を経験として与える文章の標準形として定義する. 最も簡単な経験は,

state → event → state

の 3 つの情報ノードによって構成される. 情報ノードの関係は関係レベルのノードを通して結ばれる. 経験の標準形を満たす文章例を図 4.6 に示す.

4.3 質問応答機能

質問文への応答機能は, 大きく 3 つに分けられる. それぞれについてその機能と実装について説明する. 疑問文は, 格文法によって解析された後の形で入力されることを仮定している. また, 4.4 節で述べる自然言語パーザを用いて入力することもでき

質問文を入力して下さい.	Now question start !
質問> ロードローラーはなぜおもいか	Q> why
	Q> subject roadroller
	Q> complement omoi
	Q> pred dearu
	Q> ?
日本語を用いた質問文の入力	格構造による質問文の入力

図 4.7: 質問文の入力例

るが、文脈自由文法にあった制限された自然言語を用いる。質問文の入力例を図 4.7 に示す。

(1) でき事の真偽を問う疑問文

入力された文が、経験空間中で十分な活性化エネルギーをもっているかで判断する。判断基準となる活性化エネルギーの量は、活性化条件を表す θ よりは小さな値を用いた。また、経験空間は否定文と肯定文が必ず対の形で記憶されているので、否定の文と活性化度を比較して、その時点でどちらの知識の方がより活性化しているかも調べる。

返す答は、yes か no のどちらかであり、判断のつきにくい場合にあい昧な答を返すというようには作成されていない。肯定文、否定文の持つエネルギーの量の比較によって、中間的な返答をするものも興味あるが、その判定基準を設定するのが難しいため、今後の研究課題とした。

(2) 未知項目の内容を問う疑問文

英語の疑問文における“what, who, where”に対する返答を行う。格の内容を記述するところに“*”を指定することによって起動される。経験空間と、疑問文とのパターンマッチをおこなうほかに、(1)の場合と同様にその文が現在の文章理解の対象になっているかを活性化エネルギーで判断し、十分な活性化エネルギーを持っていないものは、答の対象としない。肯定文、否定文のそれぞれの活性化度の確認も(1)の手法に準じて行われる。

返す答は、未知項目を満たす内容である。複数考えられる場合は、複数の答をそのまま返して、以後の判断はユーザに委ねる。

(3) 理由, 手段を問う疑問文

(1)(2) は直接的な答を求める問題であったが, (3) はその答を直接的に求めることができない。人間も, 複雑な質問文に対しては, 様々な戦略を持って向かうことが知られている [16]。経験空間では, 種類の異なる関係リンクを利用して, 理由, 動機, 手段などに関する疑問文に答える。METHOD, MOTIVE, THEN, SEQ, CAUSE という関係ノードはそれぞれ, 経験空間の時間的關係を表現している。リンクの重み付けに方向付けを与えることによって, 普通に経験空間を利用した場合, 時間的に後の知識にエネルギーが流れていく方が強調されている。理由, 手段などに対して答えるには, このエネルギーの流れと逆のエネルギーの流れを与えて, 時間的に前に起きた事象を活性化させる。その活性化した事象を答として返す。

疑問文は how と why に分けて検討する必要がある。how は, 手段を求める疑問文として解釈する。「どうやってそうなったか」を答える。why は, 理由, 動機を求める疑問文として解釈する。「なぜ, どうしてそうなったか」を答える。この how と why では,

- どの関係ノードを利用して検索を行うか。
- 何を答として認めるか。

について微妙に差がある。

how の場合は, METHOD, THEN, CAUSE の関係ノードに着目する。この関係ノードは, 時間的に前の事象に event をとり, 「どうして」に対する疑問文に対する答を提供する。THEN を含めた理由は, 一つ前の event が答として十分でない場合に, さらに前に起きた動作を検索するためである。検索された答としては, event だけを認める。

why の疑問文は意味が広く, 「何をしてそうなったか。」「どういう理由からそうしたか。」などの意味を含む。そのため, METHOD, MOTIVE, THEN, SEQ, CAUSE のすべての関係ノードを利用して検索する。検索された答としては, event も state も認める。

時間的に逆向きのエネルギーの流れは, 別にエネルギーの流れるバイパスを設けることによって実現する。たとえば how の疑問文の場合, まず, すべての METHOD,

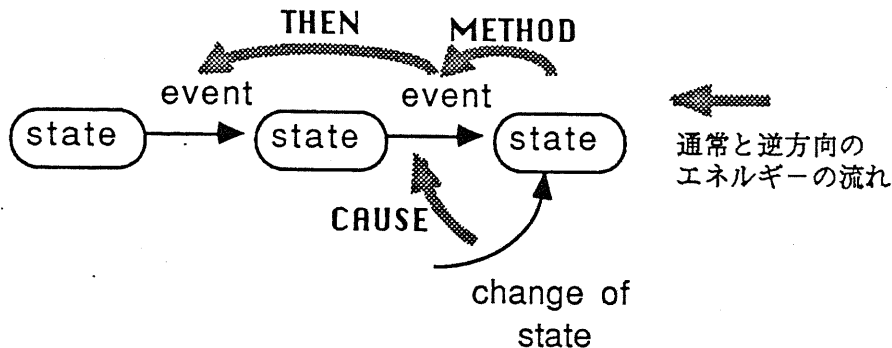


図 4.8: how 疑問文に対する答の作成

THEN, CAUSE の関係ノードにマークをつける。そして、経験空間のエネルギー伝搬の式を評価するとき、エネルギーの流れ込む相手を調べて、それがマークされた関係ノードであり、かつ自ノードが時間的に前の事象である場合は、さらに関係ノードの先にある情報ノードを調べて、そのノードからも直接にエネルギーを受け取るようにした。そのほかのエネルギー伝搬は、通常の実験空間の評価式を用いるため、文章理解のときと同じように、意味に広がりを持たせながらも、自然に答となる情報ノードに活性化エネルギーがたまるモデルが実現できる。(図 4.8参照)

次に、how, why の疑問文の答検索のアルゴリズムをまとめる。

1. 疑問文が how か why かを判断する。how の場合は、METHOD, THEN, CAUSE の関係ノードをマークする。why の場合は、さらに MOTIVE, SEQ の関係ノードもマークする。
2. 疑問文として与えられた最初の文に相当する情報ノードにエネルギーを与えながら経験空間のエネルギー拡散を行う。マークされたノードについては、先に述べた方法で、エネルギー流のバイパスを行う。
3. 活性化したノードが現われたら、そのノードの種類を調べる。how 疑問文の場合は event, why 疑問文の場合は event と state の場合にそのノードを答として返す。
4. その答をユーザに提示し、良ければ終了する。だめであれば、2. に戻り、さら

にエネルギー拡散を続け、次の候補を捜す。

4.4 パーザとの結合

パーザを作成するに当たっての開発目標を次に示す。

- 経験空間と相性のよいこと

この点から、パーザをオブジェクト指向言語で記述することにした。基本的には、文脈自由文法の構文解析をメッセージパッシングで構成する。この構成は、分散プロセッサ上での実行への移行が容易と思われる。パーザは、経験空間とも、メッセージパッシングの形で通信する。

- 効率の良い構文解析を行うこと

構文解析を行うときの大きなボトルネックはバックトラックにあるといわれる。このパーザは、文節以下の構文解析と文節以上の構文解析を分割することにより、文の解析をするレベルにおいて、文節内のレベルまでのバックトラックをすることを禁止する。

4.4.1 文節以下の構文解析法

まず、基本的にボトムアップパーザである。生成されるオブジェクトは、文法記号のレベルとそのメタレベルの2つに分けられる。説明に使う文法規則の形は次の通りである。

(a) A :- B.

(b) C :- D E.

A, C は、非終端記号

B, D, E は、非終端記号または終端記号

- 記号クラス

終端記号および非終端記号に1対1にクラスが作られる。対応するメタクラスのインスタンス（以下、メタインスタンス）からの make-instance のメッセージによって、そのインスタンス（以下、記号インスタンス）を生成する。各記号インスタンスは、構文解析木の中のノードに対応する。

• メタクラス

構文解析木のノードに対応する記号インスタンスを管理する。管理用のメソッドとして、次の2つが定義されている。

1. mk_inst

記号インスタンスを作る。

- B に対応するノードを作成した場合、A のメタクラスに対して、メッセージ `mk_inst` を送る。
- E に対応するノードを作成した場合、C のメタクラスに対して、メッセージ `request` を送る。
- D に対応するノードの場合、E を受けるまでは未完成なため、上位への拡張は行わない。

2. request

C のメタクラスに定義されるメソッドである。E からメッセージ要求により、文法規則 (b) が適用できるかチェックを行う。

- 未完成な C の記号インスタンスが存在するか調べ、存在した場合それと E をまとめて C のノードを完成させる。
- 完成した C の記号インスタンスではあるが、サブツリーを作る可能性が存在すれば、新たに C の記号インスタンスを作って、D + E を完成させる。

上のどちらかにより、C が完成した場合、さらにその C という非終端記号を上へ拡張する。

- C が B の位置に存在する場合、A のメタクラスに対して、メッセージ `mk_inst` を送る。
- C が D の位置に存在する場合、C のメタクラスに対して、メッセージ `mk_inst` を送る。
- C が E の位置に存在する場合、C のメタクラスに対して、メッセージ `request` を送る。

この構文解析法によれば, (a)(b) 式で表わされる文脈自由文法はすべて解析される.

例として次に示す文法で「東京大学に」という文節を解析する際のインスタンス生成の様子を図 4.9 に示す.

```
CASE :- CASE P /* (東京 に) も */
CASE :- NS P /* 太郎 が */
NS :- NOUN NS /* 東京 大学 */
NS :- NOUN
```

ただし, 図 4.9 にも示すように不要なノードが多く作成されるため, 文のレベルまでこれを使用すると効率の点で問題が残る. そこで, 経験空間への入力文章のパーザとして直接用いないで, 範囲を限定して, 文節以下の解析に用いることにした. 一方, 意味に関するものはなるべく経験空間上の活性化エネルギーとして解決する. 文節のレベルで正しいと認識されたものは, その中の自立語を経験空間に転送し, 活性化エネルギーの供給源とする.

4.4.2 文節以上の構文解析法

文節の解析結果は文節レベルの処理が終了すると文節スタックに格納される. この文節スタックの内容に対して, 経験空間の活性化エネルギーを求め, その値によって文節間の結合関係を発見する. 細かい処理に関するアルゴリズムを以下に示す.

(a) 用言 (PRED) の処理

```
while(TRUE) {
  switch(文節スタックの最上位) {
    case(副詞):
      PRED の modifier に ADV をセット;
      break;
    case(名詞節):
      用言, 名詞, 取り得る格の候補を経験空間に送る;
      if(適切な格の候補が reply される)
        PRED の当てはまる格に CASE をセット;
      else
        goto end;
      break;
    case(NULL):
```

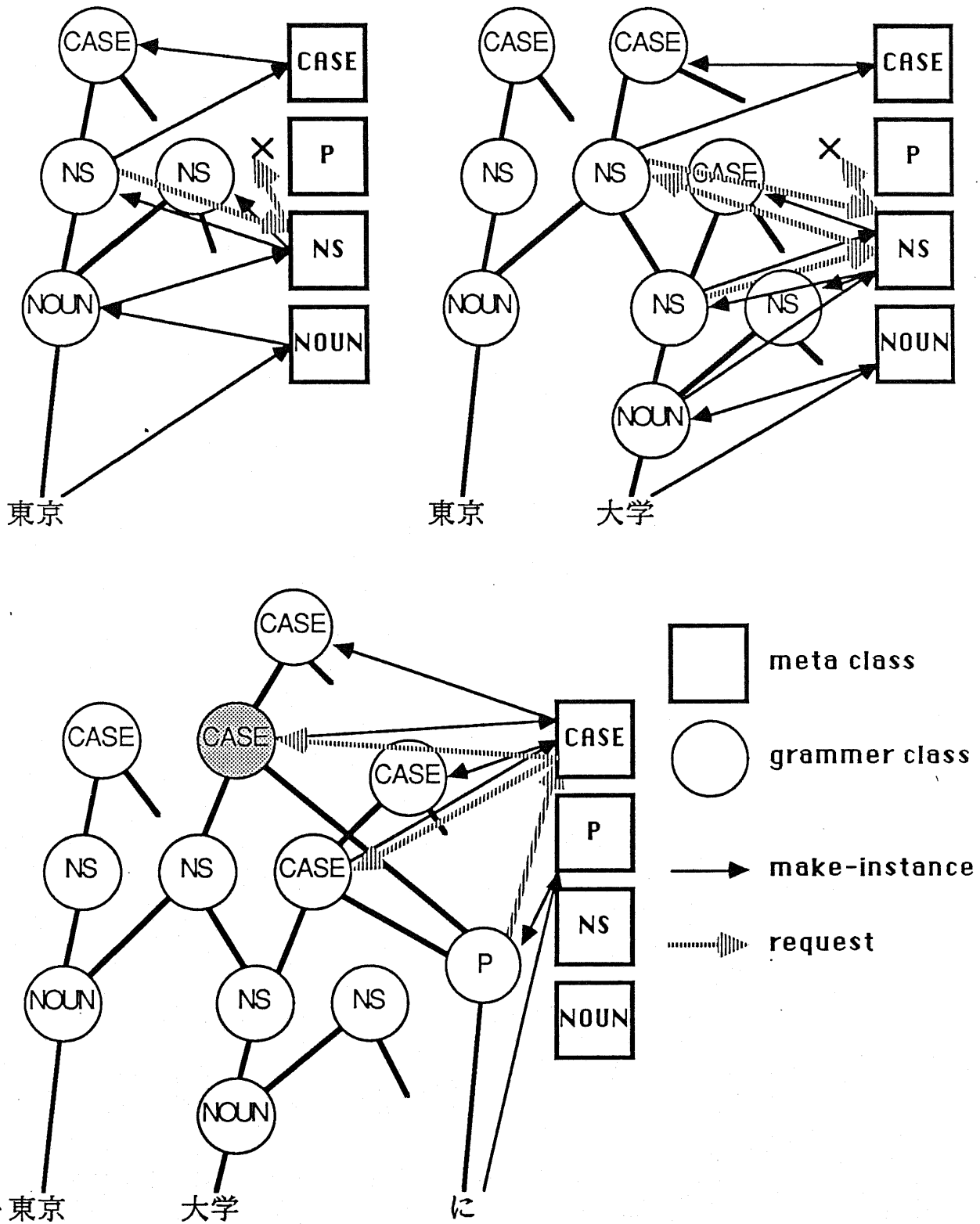


図 4.9: オブジェクト指向を用いた構文解析

```

        default:
            goto end;
    }
    最上位の記号インスタンスを文節スタックからポップ;
}
end:
PRED を文節スタックにプッシュ;

```

(b) 名詞節 (CASE) の処理

```

while(TRUE) {
    switch(文節スタックの最上位) {
        case(連体詞):
            CASE の modifier に MOD_NS をセット;
            break;
        case(用言):
            if(必須格の格スロットがすべて埋まっていない) {
                用言, 名詞, 取り得る格の候補を経験空間に送る;
                if(適切な格の候補が reply される) {
                    PRED の当てはまる格に (mod + CASE) をセット;
                    CASE の modifier に PRED をセット;
                }
            }
            CASE の modifier に PRED をセット;
            break;
        case(助詞「の」を伴う連体修飾節):
            CASE の modifier に CASE をセット;
            break;
        case(助詞「や」を伴う名詞節):
            並列節処理のルーチンをコール;
            break;
        case(NULL):
        default:
            goto end;
    }
    最上位の記号インスタンスを文節スタックからポップ;
}
end:
CASE を文節スタックにプッシュ;

```

(c) 連体詞 (MOD_NS) ・ 副詞 (ADV) の処理

```

while(TRUE) {
    switch(文節スタックの最上位) {

```

```

case(副詞):
    用言, 名詞, 取り得る格の候補を経験空間に送る;
    if(ADV が程度を表す副詞である)
        MOD_NS, ADV の modifier に ADV をセット;
    else
        goto end;
    break;
case(NULL):
default:
    goto end;
}
最上位の記号インスタンスを文節スタックからポップ;
}
end:
PRED を文節スタックにプッシュ;

```

(d) 並列節処理ルーチン

名詞節どうしが並列の関係にあるときはそれらをまとめて1つの格として扱われる。この構文解析機では、並列の意味だけを持つ格助詞「や」に関して並列節処理を行っている。

- (1) 文節スタックの最上位の名詞節 (CASE) が助詞「や」で終わっていることを調べる。
- (2) 現在取り扱っている名詞節 (NOW) と CASE のそれぞれの記号インスタンス (以下, <CASE> <NOW> と表記する) を並列処理用のクラス *meta_para1* に送る。
- (3) *meta_para1* では, 記号インスタンス <PARA> を生成する。
- (4) 文節スタックから <CASE> を取り出し, 代わりに <PARA> を文節スタックに格納する。

以上, 文節以上の構文解析の実行例を図 4.10 に示す。

4.4.3 質問応答システムへの日本語入力インタフェイス

質問応答のための入力文を解析する時は, 経験空間を利用できない (質問文の解析のために経験空間の活性化エネルギー分布を乱してしまう) ため, 日本語に制限を加えて,

時間:	[1]	[2]	[3]	[4]	[5]	[6]	[7]
例文:	つるは	ぬまや	たんぼに	いる	たにしを	たくさん	たべる
品詞:	N1 P1	N2 P2	N3 P3	V1	N4 P4	ADV1	V2
文節:	CS1	CS2	CS3	VS1	CS4	ADV	VS2

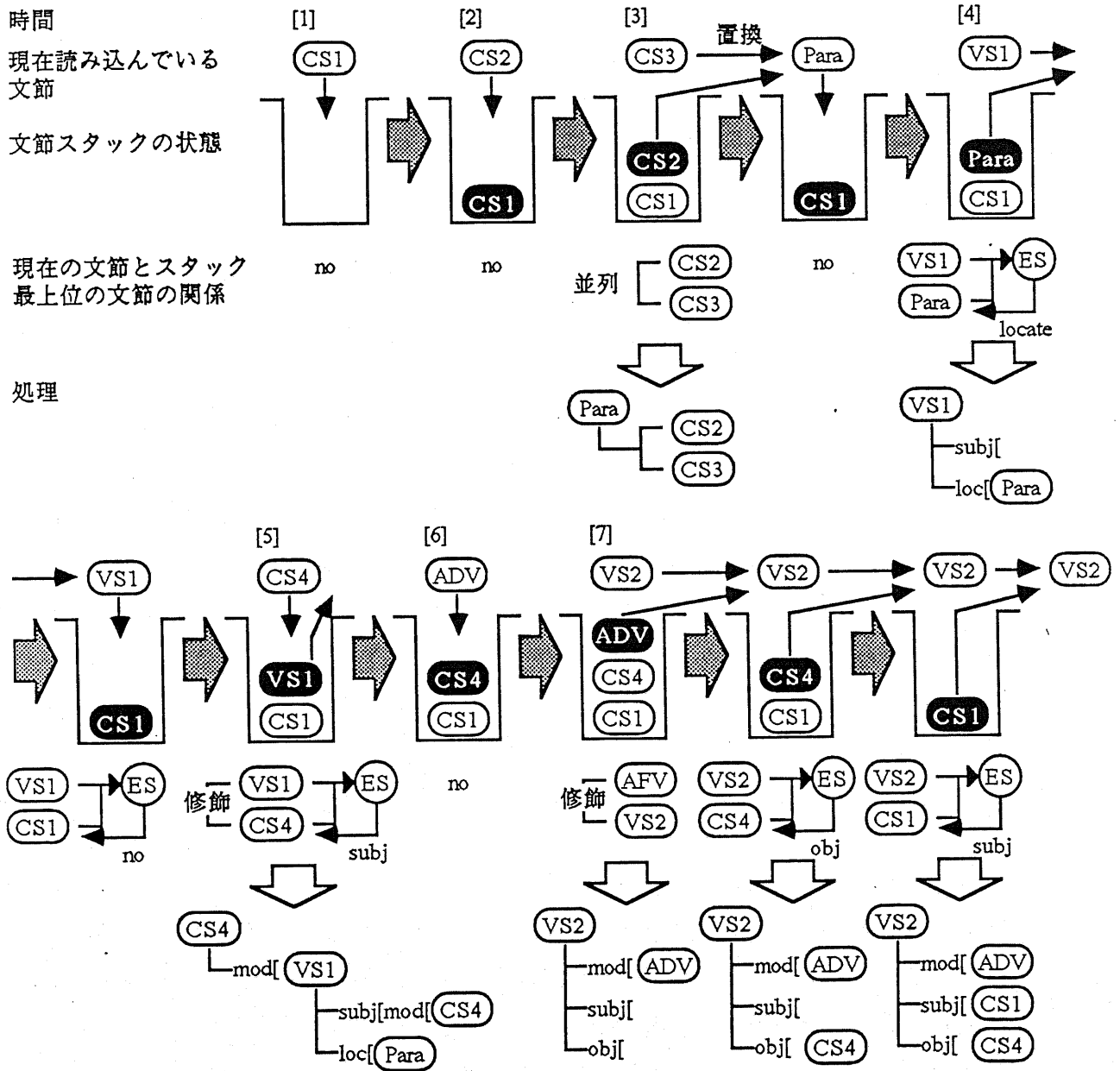


図 4.10: 文節以上の構文解析の実行

< root > :- 文.	場所格 :- 複合名詞 場所格助詞.
文 :- 平叙文.	行き先格 :- 複合名詞 行き先格助詞.
文 :- 疑問文.	複合名詞 :- 複合名詞 名詞.
疑問文 :- 平叙文 間投詞.	複合名詞 :- 名詞.
平叙文 :- 述語.	述語 :- 動詞.
平叙文 :- 格 平叙文.	述語 :- 動詞未然 否定.
平叙文 :- 疑問詞 平叙文.	述語 :- 補語.
格 :- 格 助詞.	述語 :- 補語 be 動詞句.
格 :- 主格.	be 動詞句 :- be 動詞.
格 :- 目的格.	be 動詞句 :- be 動詞未然 否定.
格 :- 場所格.	補語 :- 形容詞.
格 :- 行き先格.	補語 :- 形容詞未然 否定.
格 :- から格.	補語 :- 名詞.
主格 :- 複合名詞 主格助詞.	から格 :- 複合名詞 から格助詞.
目的格 :- 複合名詞 目的格助詞.	

図 4.11: 質問文を理解するための構文規則

(文節以下の解析用の) 文脈自由言語パーザを用いて解析した。もともと、オブジェクト指向言語 Flavor で記述されたプログラムであるが、経験空間との相性をよくするため、C 言語に書き直した。また、文法規則や辞書を日本語で書けるように改良を行った。質問文のための構文規則を図 4.11 に示し、辞書の一部を図 4.12 に示す。図 4.13 は、これを用いた質問文の解析結果である。

4.5 FOCUS3 の動作例

4.5.1 文脈理解

実験は、経験空間の知識がない状態から始める。そこで、まず、システムに経験としていくつかの文章（経験として十分な情報を持つ）を繰り返し与えて、経験知識を構築する。この時に与える文章としては、4.2 節で検討した標準形に準拠した文章を用いる。知識を学習した後に、適当に省略して表現された文章（日常会話を想定している）など理解させたい文章をシステムに与え、文章理解の実験を行う。形の上では、

abuku あぶく 名詞
 akubi あくび 名詞
 annaisuru 案内する 動詞
 arukoru アルコール 名詞
 au 会う 動詞
 au 会わ 動詞未然
 aruku あるく 動詞
 aruka あるか 動詞未然

図 4.12: 質問文を理解するための辞書記述

太郎は空腹であるか
 [< root >]
 [文]
 [疑問文]
 [平叙文]
 [格]
 [主格]
 [複合名詞]
 [名詞]
 太郎
 [主格助詞]
 は
 [平叙文]
 [述語]
 [補語]
 [名詞]
 空腹
 [be 動詞句]
 [be 動詞]
 である
 [間投詞]
 か

図 4.13: 質問文の構文解析

```

tarou 0.156478
kuufuku 0.072452
dearu 0.129656
state0 0.118143
    subject tarou 0.995930 0.083172
    complement kuufuku 0.924142 0.310025
    pred dearu 0.975874 0.130108
!state0 0.059571
    subject tarou 0.598688 0.000745
    complement kuufuku 0.731059 0.069138
    pred dearu 0.785835 0.000911
!rel0 0.037928
    part state0 0.947846 0.039165
    part !state0 0.802185 0.310025

```

図 4.14: 内部エネルギーの学習結果

知識獲得と文章理解の 2 段階の実験に分けているが、実際の経験空間の振舞いとしては、この両者に質的な差はない。

例題として、レストランを中心として経験的知識を学習させ、関連する文章の理解を試みた。

<第 1 段階：知識獲得>

入力文章は、レストランに関する太郎と次郎の経験である。図 4.6 に示したような例文を 24 個(太郎に関して、14 個、次郎に関して 10 個)用意して、システムに 2 ないし 3 回ずつ繰り返して与える。図 4.6 の最後の行に書かれた“NULL”は、入力した後に、しばらくエネルギー拡散の時間を与えることを表している。太郎に関する文章には、レストランに関する標準的な行動が記述されている。次郎に関する文章には、次郎の個性がでるような特殊な経験を記述する。入力文章を日本文に直したものを付録に示す。学習の順序としては、まずは、常識的なレストランの知識を学習させるために、太郎の経験を積ませる。その後で、次郎に関する文章を与える。徐々に経験空間が構成され、そのノードに内部エネルギーが蓄えられる。第 1 段階の終了時点における内部エネルギーの分布の一部を図 4.14 に示す。各々の単語の右に示された数値がそのノードの持つ内部エネルギーである。リンクで結ばれるノードの名前の右に

[ノード名][ノードの内部エネルギー]

または,

[ノード名(A)][ノードの内部エネルギー]
 [TAB] [リンク名][リンク先のノード名(B)][A から B への重み][B から A への重み]

図 4.15: 経験空間の知識のフォーマット

示された数値は、リンクの重み付けを表す数値である。このフォーマットは、経験空間に蓄えられた知識をファイルに表現するためのものである。フォーマットの説明を図 4.15 に示す。作成された経験空間は、

- 表象ノード 64 個
- 情報ノード 114 個
- 関係ノード 154 個
- 合計 332 個

である。

<第2段階：文章理解>

まず、第1段階で作成した知識ファイルから経験空間の知識を読み込む。

[実験1]

図 4.16 に示されたような文章理解の対象となる文章を入力する。例にあげた文章は「料理を食べる」という事実に関して情報の欠けている文章である。この文章をシステムに与えると、経験空間中に展開され、経験空間内の関連するノードの活性化エネルギーを高める。この活性化エネルギー分布がその文章の意味表現となる。入力文章が十分に記述されていない場合でも類似した経験の中から情報が補われる。「料理を食べる」を表現するノードの活性化エネルギーは、周辺からのエネルギーの流入により自然に高められている。その様子の一部を図 4.17 に示す。図に示したのは、

```

@s1 10
    subject tarou 10
    complement kuufuku 10
    pred dearu 10
@e2 1
    subject tarou 10
    dist resutoran 10
    pred iku 10
@r1 10
    RN MOTIVE 1
    before @s1 1
    after @e2 10
NULL 200
    
```

図 4.16: 「料理を食べる」状況を表す入力文章

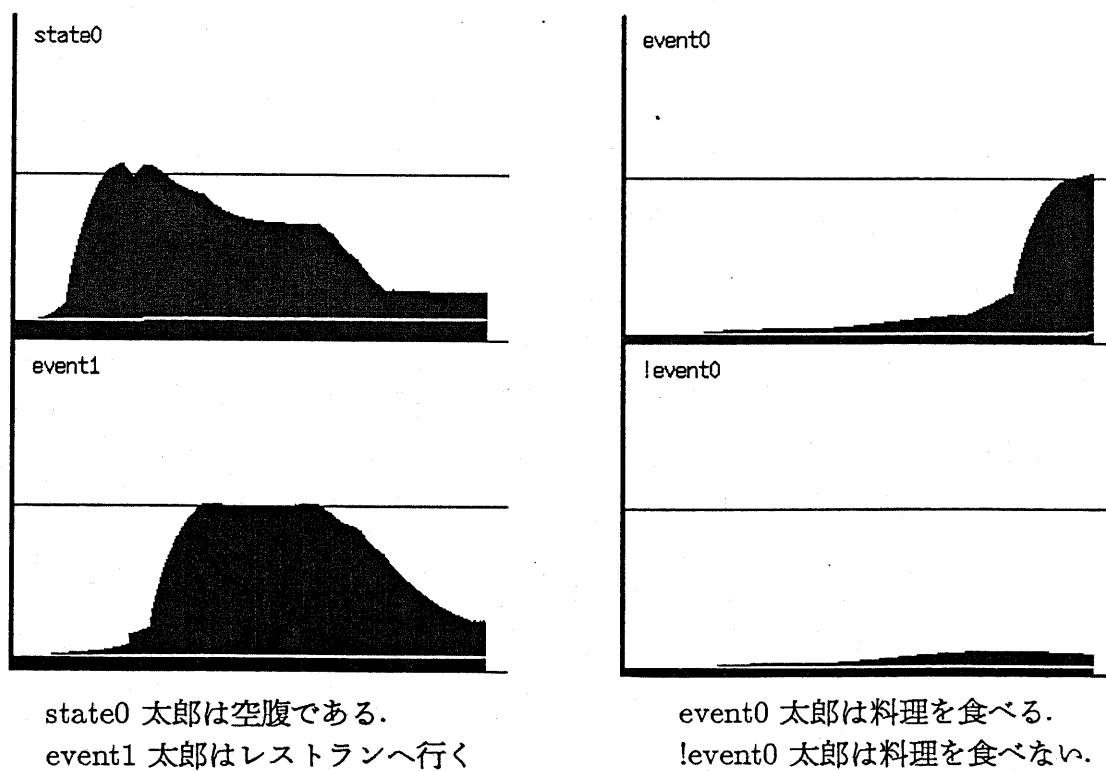


図 4.17: 実験1における活性化エネルギーの変化

```

@!s1 20
    subject tarou 10
    object kane 10
    pred aru 10
@e2 1
    subject tarou 10
    dist resutoran 10
    pred iku 10
@r1 10
    RN MOTIVE 1
    before @!s1 1
    after @e2 10
NULL 200

```

図 4.18: 「働く」状況を表す入力文章

「太郎は空腹である。」

「太郎はレストランへ行く。」

「太郎は料理を食べる。」

「太郎は料理を食べない」

を表現するノードの活性化エネルギーの変化である。直接に言及していない「太郎は料理を食べる。」を表すノードの活性化エネルギーが、高まる様子が確認できる。

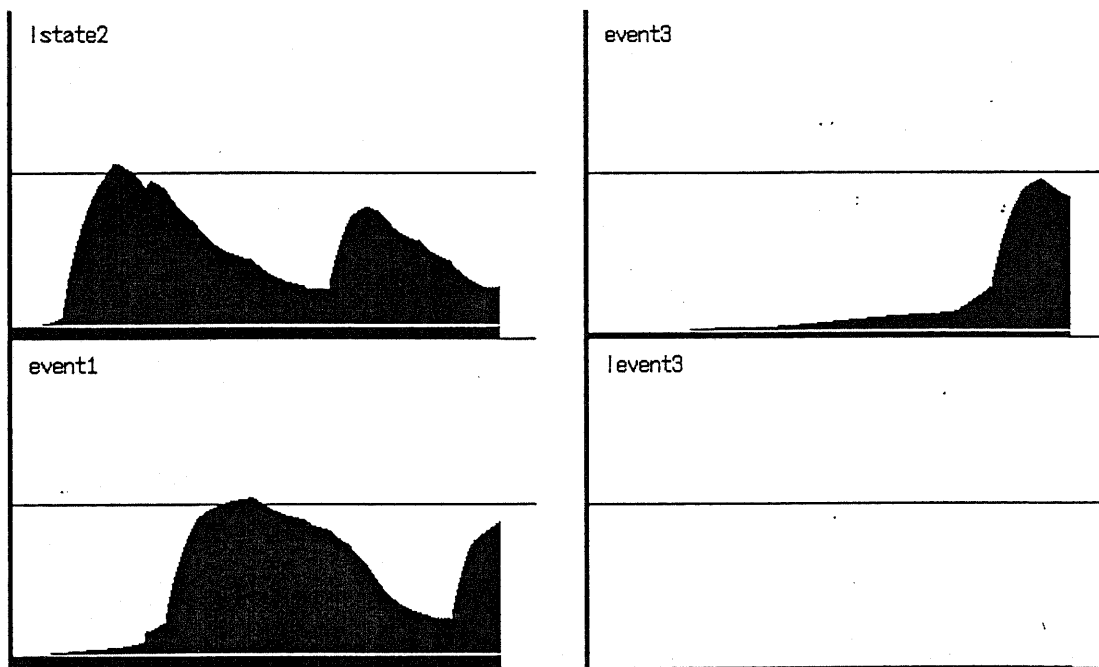
【実験2】

前の実験との比較で、同じく「レストランへ行く」文章であるが、「働きに行く」ことを想像させる文章の理解を行う。入力文章を図 4.18 に示す。この文章に対するエネルギーの変化を図 4.19 に示す。実験1とは違って、「太郎は働く。」を表すノードが活性化することが確認された。

このように、それぞれ異なる状況で、

「レストランに行く。」

場合に、その文脈を正しく理解することができる。これは、従来の script のような形の定まったルールでは、記述しにくいものであった。



lstate2 太郎は金がない.
event1 太郎はレストランへ行く

event3 太郎は働く.
levent3 太郎は働かない.

図 4.19: 実験 2 における活性化エネルギーの変化

また、活性化したノードは、時間的にこれから起こる事実の推論という立場でとらえることもできる。したがって、「働く」や「食べる」というキーワードを前提において、この後に入力する文章の理解をすることができる利点を持つ。

なお、先にも述べたように経験空間では、理解と学習の質的な違いを持たない。すなわち、経験空間は第1段階においては、入力された文章に対し既存知識なしで理解を行い、その結果の一部を内部エネルギーとしてノード内に蓄える。第2段階では、第1段階で構築された意味ネットワークを「経験」として参照する以外は全く同様の手順によって入力文章を理解する。この第2段階においても第1段階と同様に内部エネルギーの取り込みとリンクの重み付けの変化、すなわち、知識獲得が同時に行われている。

4.5.2 同音意義語の理解

「蛸」と「凧」のように、発音またはローマ字表記した場合に区別がつかない単語の認識実験を行う。まず、

「蛸を食べる」

「凧は食べない」

「凧で遊ぶ」

「蛸で遊ばない」

という内容を中心に経験を積ませる。そのために与えた知識の日本語表現を付録に示す。できあがった経験空間のネットワーク構造の一部を図4.20に示す。

【実験3】

この経験空間の中で「タコを食べる」という文を理解させる。抑制リンクによってエネルギーが抑えられるため、「凧」の持つ活性化エネルギーは増加しない。一方、「蛸」の方は、活性化エネルギーの上昇が観察される。(図4.21参照)

【実験4】

「タコで遊ぶ」という文を理解させる。今度は、実験3とは逆に、「凧」の方が活性化される。(図4.22参照)

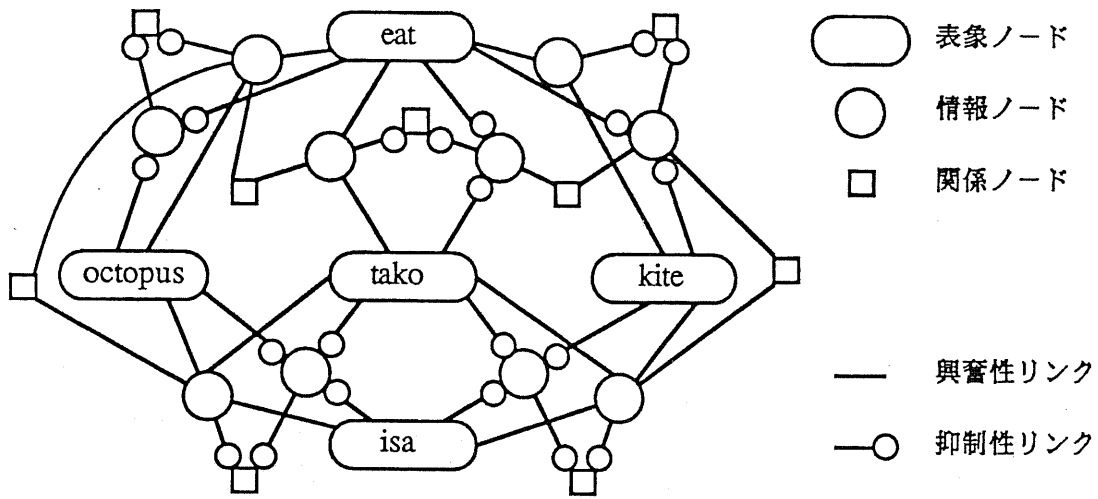


図 4.20: 「タコ」を理解するための経験空間

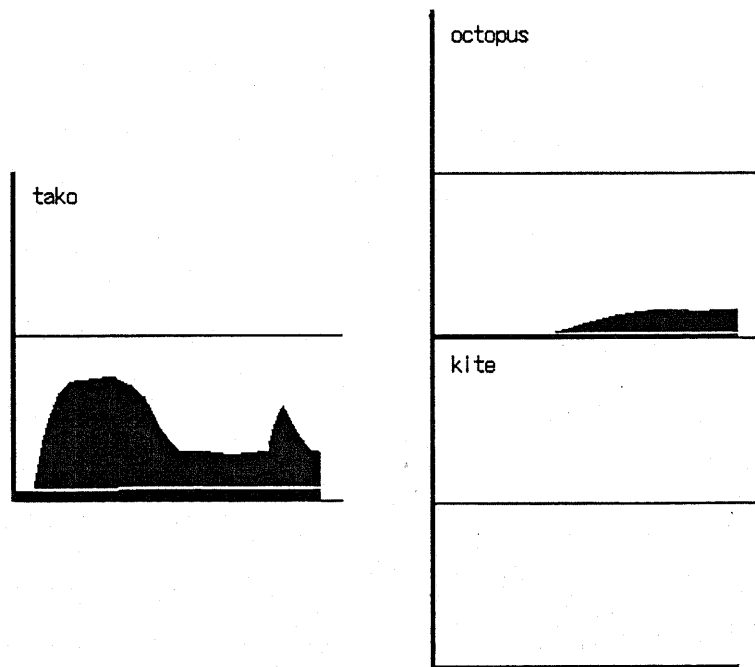


図 4.21: 実験3における活性化エネルギーの変化

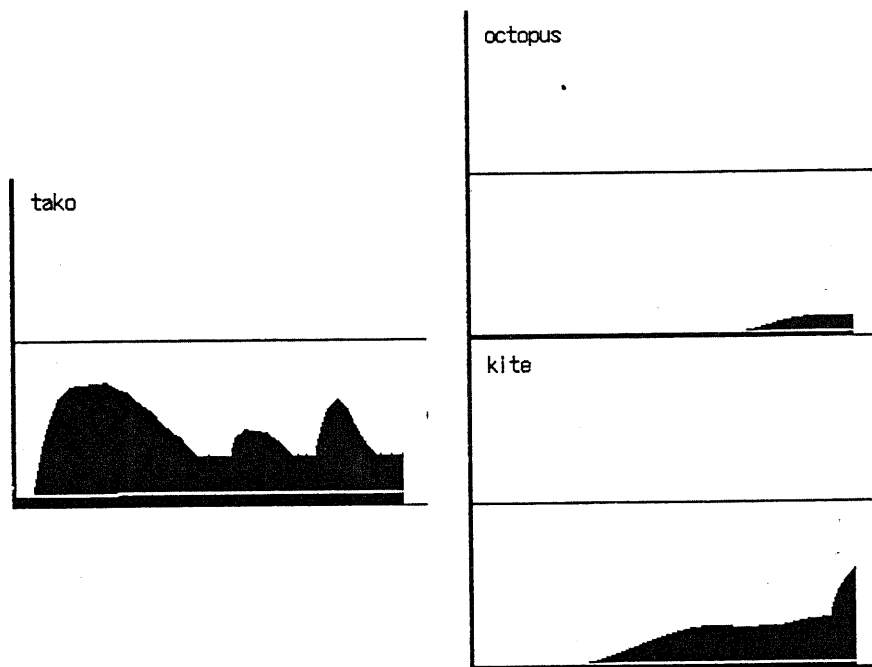


図 4.22: 実験 4 における活性化エネルギーの変化

この実験によって、経験空間の同音意義語の判別能力が確認された。上に示した実験結果は、「タコ」を中心に学習させた知識を用いたが、同じ経験を実験 1, 実験 2 で用いたレストラン中心の知識の上に構成した場合でもほぼ同様な結果がみられる。実験 4 をこの大きな知識の中で行った結果を図 4.23 に示す。

4.5.3. 質問応答実験

質問応答実験は、まず、その基礎的な機能の確認実験を行う。次に応用として、小学校 1 年生の国語読解の問題集の中から 2 題選び出して、その解答能力を評価する。質問文の入力方法には、格文法解析したものを与える方法(図 4.7 参照)と制限された日本語を用いる方法がある。制限された日本語は、4.4.3 節の質問文の日本語インタフェイスのところで述べたオブジェクト指向で記述されたパーザで解析される。もともとは、文節以下の解析用に開発されたもので機能的には余り優れていないが、質問文の入力を簡単に行える利点を持つ。オプションにより、構文解析木(図 4.13)の中から格の内容を取り出すことができる。これを図 4.24 に示すが、この出力は、質問

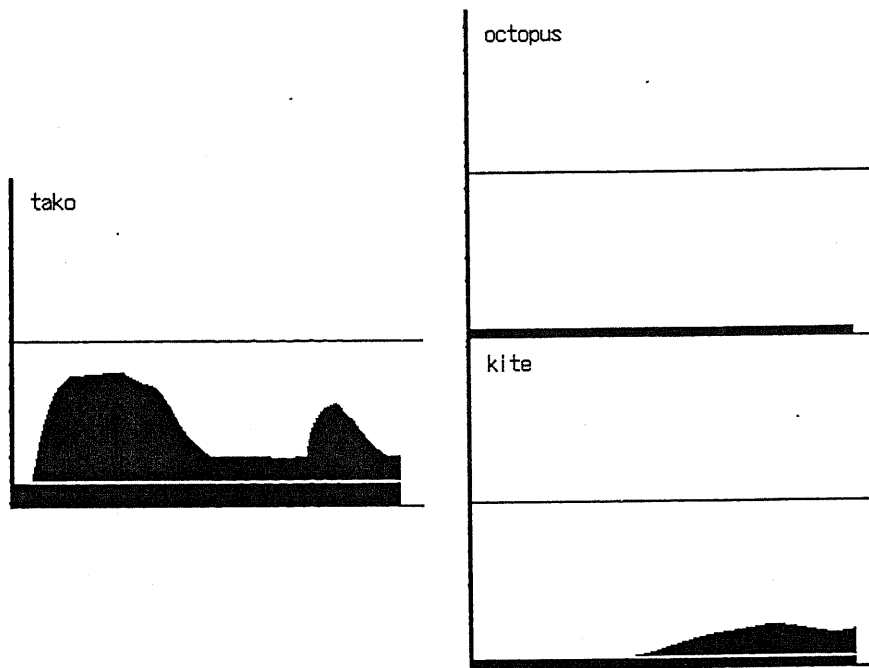


図 4.23: 広い知識を与えた場合の 実験4 における活性化エネルギーの変化

文の入力のフォーマット (図 4.7 に準拠するため、質問文入力のフィルタとして利用できる。以後、例を示す中で日本語入力である場合は、この入力フィルタを用いたものである。日本語入力で生成できない質問文に関しては、格文法解析された形で入力する。

(1) でき事の真偽を問う疑問文

```
sun4.2> parse rule1 -f
```

```
太郎は空腹であるか
subject tarou
complement kuufuku
pred dearu
?
```

図 4.24: 質問文のための構文解析

質問文を入力して下さい。

質問> 太郎は空腹であるか

はい. (in_E = 0.123771 ext_E = 0.524084 total_E = 0.647855)

Continue ? (y/n) y

質問文を入力して下さい。

質問> 太郎はレストランへ行くか

I don't know.

Continue ? (y/n) y

質問文を入力して下さい。

質問> 太郎は料理を食べないのか

いいえ. (in_E = 0.060629 ext_E = 0.446596 total_E = 0.507225)

Continue ? (y/n) n

図 4.25: でき事の真偽を問う疑問文への応答

4.5.1 節で学習された知識を経験的知識として利用する。理解する文章は、

「太郎は空腹である。太郎は料理を作る。

太郎は料理を食べる。太郎は満腹である。」

である。まず、この文章を FOCUS3 に理解させる。このとき、経験空間に外部エネルギーの形で理解の構造が残る。この状態で、質問文を与える。いくつかの質問に答えているところを図 4.25 に示す。答えは、「はい」「いいえ」とその時の内部エネルギー (in_E), 外部エネルギー (ext_E), その合計 (total_E) を表示する。実験例では、

「太郎は空腹である。」

について「はい」の答えを返している。与えられた質問文を肯定する文も否定する文も十分な活性化エネルギーを持たない場合は、

I don't know.

を返す。この実験例では、

質問文を入力して下さい。

質問> 太郎は何を作るのか

object = ryouri (in_E = 0.060757 ext_E = 0.150915 total_E = 0.211672)

Continue ? (y/n) y

質問文を入力して下さい。

質問> 太郎はどこへ行くか

I don't know.

Continue ? (y/n) n

図 4.26: 未知項目の内容を問う疑問文への応答

「太郎はレストランへ行くか。」

に関して、判断できなかったことを示す。また、この質問応答システムには、否定の疑問文を与えることもできる。答は、日本語における「はい」「いいえ」である。

(2) 未知項目の内容を問う疑問文

(1) と同じ知識、同じ文章を用いて、経験空間を活性化する。その後で、未知項目を含んだ疑問文を入力する。システムは、未知項目を満たす単語とその活性化エネルギーを返す。ここでも、十分活性化されていない情報に関しては、“I don't know.” を返す。応答例を図 4.26 に示す。図の中で

「太郎はどこへ行くか。」

という疑問文がある。経験空間の(過去の経験)知識としては、

「太郎はレストランへ行く。」

という知識があるが、現在活性化されていないために、答の対象としない。

(3) 理由、手段を問う疑問文

(1)(2) と同じ状況での理由、手段を尋ねる疑問文の例を図 4.27 に示す。日本語入力では、「どうして」も「なぜ」も“why”として解釈する。これは、日本語の意味があ

質問文を入力して下さい.

質問> 太郎はなぜ料理を作るのか

state1

subject tarou

complement manpuku

pred dearu

OK ? (y/n) n

state0

subject tarou

complement kuufuku

pred dearu

OK ? (y/n) y

Now question start !

Q> how

Q> subject tarou

Q> complement manpuku

Q> pred dearu

Q> ?

event0

subject tarou

object ryouri

pred taberu

OK ? (y/n) y

図 4.27: 理由, 手段を問う疑問文への応答 その1

意味なためであり，“why”の方が“how”より広く検索を行うのでこのような実装にした。したがって，“how”を明示的に使う場合は，日本語入力をいらないで格構造で入力する。

答が正しくないときは，次の答を要求できる。1番目の例では，

太郎はなぜ料理を作るのか

に対して，最初，

「満腹であるから。」

という誤った答を返している。(理解した文章の最後にこの文があったため，活性化エネルギーの残留が大きかったと思われる。)次の答として

「空腹であるから。」

という正しい答を導き出している。

次に理解する文章を変えた実験結果について，図4.28に示す。この例では，理解する文章は，

「次郎はマクドナルドへ行く。次郎は列に並ぶ。
次郎は注文を考える。次郎はハンバーガーをもらう。
次郎は席を見つける。次郎はハンバーガーを食べる。
次郎は満腹である。次郎は家に帰る。」

という文章である。経験空間の知識は，太郎と次郎の知識が混在しているが，返ってくる答の中では，取り違えることは少ない。図4.28の2つめの例では，同じ質問を続けて与えている。

「なぜ次郎は家に帰るのか。」

に対して，最初，

「次郎はハンバーガーを食べる(食べた)から。」

と答えているのを

「次郎は満腹であるから。」

質問文を入力して下さい。

質問> 次郎はどうして満腹であるのか

event35

subject jirou
object hanbaga
pred taberu

OK ? (y/n) y

Now question start !

Q> why

Q> subject jirou

Q> dist ie

Q> pred kaeru

Q> ?

event35

subject jirou
object hanbaga
pred taberu

OK ? (y/n) n

state5

subject jirou
complement manpuku
pred dearu

OK ? (y/n) y

Continue ? (y/n) y

Q> why

Q> subject jirou

Q> dist ie

Q> pred kaeru

Q> ?

state5

subject jirou
complement manpuku
pred dearu

OK ? (y/n) y

図 4.28: 理由, 手段を問う疑問文への応答 その2

(1) ロードローラーは、なにをする じどう車ですか。⑩

(2) ロードローラーは なぜ おもいのですか。⑪

(3) ロードローラーは、なぜ、どうろを たいらにかためる ことができますのですか。⑫

つぎの ぶんを よんで、といに こたえましょう。

ロードローラーは、じめんを たいらに かためる じどう車です。

車は、てつて てきて いるので、たいへん おもいのです。ですから、どうろなどを たいらに かためる ことができます。

33

◆せつめいぶん◆

カだめしテスト

(1)

じかん 10ぶん

てん

図 4.29: 「小1 国語読解 10 分間トレーニング」より

に改めさせている。その後、続けて同じ質問に対しては、後者の教えた答を返すようになった。

(4) 小学校 1 年の国語読解の問題 その 1

FOCUS3 の評価を客観的に行うために、小学校の文章読解の練習問題 [10] を解かせることを試みた。練習問題の例を図 4.29 に示す。図 4.29 に示した例題は、説明文の読解問題である。例題自身が知識の記述であるため、あえて経験的知識を学習することなく、今までの「レストラン」に関する経験的知識のもとでこの文章を理解させた。経験空間への入力は、原文をなるべく損なわないように図 4.30 のようにした。各設問の解釈と答を次に示す。問題集の付録の答を正解とする。

「ロードローラーは、なにをするじどう車ですか。」

質問応答システムの中の未知項目を尋ねる疑問文として取り扱う。日本語インタフェースでは「なにをする」を解釈できないため、格構造の表現を用いて、述語を尋ねる疑問文とする。実行例は、次のようになる。

```
@e1 10
    subject roadroller 10
    object jimen 10
    complement taira 10
    pred katameru 10

@s2 1
    subject roadroller 10
    compement jidousha 10
    pred dearu 10

@r1 10
    RN ADJ 1
    adj @e1 1
    noun @s2 10

@s3 1
    subject kuruma 10
    material tetsu 10
    pred dekiru 10

@r2 10
    RN SEQ 1
    before @s2 1
    after @s3 10

@s4 1
    subject roadroller 10
    complement omoi 10
    pred dearu 10

@r3 10
    RN SEQ 1
    before @s3 1
    after @s4 10

@e5 1
    subject roadroller 10
    object douro 10
    complement taira 10
    pred katameru 10

@r4 10
    RN MOTIVE 1
    before @s4 1
    after @e5 10

NULL 30
```

図 4.30: 経験空間へ入力するための書式

```
Now question start !
Q> subject roadroller
Q> pred *
Q> ?
```

```
pred = dearu(in_E = 0.164643 ext_E = 0.481552 total_E = 0.646195)
```

```
pred = dearu(in_E = 0.164643 ext_E = 0.481552 total_E = 0.646195)
```

```
pred = katameru(in_E = 0.004996 ext_E = 0.284366 total_E = 0.289362)
```

答のうち、前の2つは

「ロードローラーは重い。」 「ロードローラーは自動車である。」

という文の述語である。今、求める答は、

「ロードローラーは、じめんをかためる。」

を表す3つめの答である。ただし、述語だけを答としているので、十分な答とは言えない。

「ロードローラーはなぜおもいのですか。」

この質問は、ほぼそのまま入力できる。

質問文を入力して下さい。

質問> ロードローラーはなぜおもいか

event56

```
subject roadroller
object douro
complement taira
pred katameru
```

OK ? (y/n) n

state31

```
subject kuruma
material tetsu
pred dekiru
```

OK ? (y/n) y

「道路を固めるために重くできている。」と考えれば、最初の答も正しい。しかし、ここでは、文の順番からいって、後者の答の方がより正しいとする。

「ロードローラーは、なぜ、どろろをたいらにかためることができますか。」

「できる」という助動詞の評価は行えないため、次のような質問文とした。

質問文を入力して下さい。

質問> ロードローラーはなぜどろろをかためるか

```
state32
      subject      roadroller
      complement   omoi
      pred         dearu
OK ? (y/n) y
```

答は望みのものが得られた。

説明文であるため、特に難しい問題ではなく、経験空間のような知識表現でなくても十分解くことができる問題である。ここでは、経験空間が説明文から知識を得る能力を示した。

(5) 小学校1年の国語読解の問題 その2

次に、童話の読み取りを行う。この文章を図4.31に示すが、

- かくれんぼに関する知識
- 登場するもの(亀, どじょう, ふな)の習性の知識

を知らないと解けない問題である。そこで、文脈の読み取りが重要になる。実験では、まず、上に示した2種類の知識を学習させるところから始める。学習のために用意した文章は11であり、「レストラン」に関する経験的知識の上に追加知識として構築した。獲得された経験空間は全体で、

- 表象ノード 88 個
- 情報ノード 170 個
- 関係ノード 214 個
- 合計 472 個

になる。

次に、(4)と同じく、文章の意味を損なわないように、経験空間への入力フォーマットによって文章を記述した。これを図4.32に示す。

1 どうわ

ようすを よみとる (1)

じかん 1035ん

てん

◆ つぎのぶんをよんで、といにこたえましょう。

「もういいよ。」

かめは、ゆっくり ゆっくり みんなをさがしました。

どじょうの しっぽが、どろの なかから でて いました。

いわの かげに かくれた ふなが、おおきな あく びを しました。あぶくが 三つ、うえの ほうへ のぼって きました。

(1) * どじょうは なぜ みつかったのですか。③

(2) * ふなは なぜ みつかったのですか。③

(3) なぜ、ふなは おおきな あくびを したのですか。④

図 4.31: 童話の読み取りの練習問題

「もういいよ。」

は、省略した。以下に、設問とシステムの答を検討する。

「どじょうはなぜみつかったのですか」

最初の問題から、非常に難しい問題である。その原因は3つある。

- 文章中のどこにも「どじょうがみつかった」とは書いていない。
- 「どじょうがみつかる」とは、意味的に受身形である。
- かくれんぼの知識を十分に持たないと、なぜみつかるか理解できない。

1. の問題は、経験の中で「かくれんぼ」でみつかる状況を学習させ、経験空間による文脈理解を行うことでシステムに認識させた。2. の問題は、受身形を避けて、「どじょうをみつける」という文に変更した。経験の中でも受身形は用いなかった。3. については、場当たりのあるが、経験の中に

「何かが出るとみつかる」

```
@e1 10
    subject kame 10
    adverb yukkuri 10
    pred sagasu 10
@s2 10
    subject dojou 10
    object shippo 10
    pred motsu 10
@s3 1
    subject shippo 10
    source doro 10
    pred deru 10
@r1 10
    RN ADJ 1
    adj @s2 1
    noun @s3 10
@r2 10
    RN METHOD 1
    before @e1 1
    after @s3 10
@e4 10
    subject funa 10
    place iwa 10
    pred kakureru 10
@e5 1
    subject funa 10
    object akubi 10
    pred suru 10
@r3 10
    RN ADJ 1
    adj @e4 1
    noun @e5 10
@r4 10
    RN MOTIVE 1
    before @s3 1
    after @e5 10
@e6 1
    subject abuku 10
    dist ue 10
    pred noboru 10
@r5 10
    RN THEN 1
    before @e5 1
    after @e6 10
NULL 30
```

図 4.32: 経験空間へ入力するための書式

という文章を入れて対応した。システムの応答は次のようである。

質問文を入力して下さい。

質問> なぜどじょうをみつけるか

event54

subject abuku

pred deru

OK ? (y/n) n

event50

subject funa

object akubi

pred suru

OK ? (y/n) n

state31 subject shippo

source doro

pred deru

OK ? (y/n) y

最初の間違ひは、ふなとどじょうを混同したところにある。あぶくに関する主語は記述されていないため、この間違ひを改善するには、かなり深い知識が必要であると思われる。

「ふなはなぜみつかったのですか」

問題点は、前問と同じである。

質問文を入力して下さい。

質問> ふなをなぜみつけるか

event54

subject abuku

pred deru

OK ? (y/n) y

こちらは、「あぶくがでる」が正解であるので、最初から正しい答を返した。

「なぜ、ふなはおおきなあくびをしたのですか。」

この答は、大人でも少し悩むところがある。とくにあくびの原因については文章中に現れない。そこで、一般的なあくびをする状況を把握しなければならない。経験させる文章中に、

「みつからないとたいくつする。」

「たいくつするとあくびがでる。」

という知識を与えて対処した。結果は次の通りである。

質問文を入力して下さい。

質問> なぜふなはあくびをするか

```
event54
subject abuku
pred deru
OK ? (y/n) n
event55
subject kame
adverb yukkuri
pred sagasu
OK ? (y/n) y
```

この答は間接的な表現であるが、解答例も、

「かめがゆっくりさがしたのでたいくつして。」

となっている。

全体的にみて、難しい問題であり、文脈の知識、常識などが要求される。場当たりの経験を積ませたこともあるが、良い結果を得ることができた。直接に「規則」として与えて解いているのでないことを考慮すれば、十分な結果であると言える。

欠点として、経験空間の文章の表現能力があげられる。重文、複文、修飾文などが現れると、それをどのような形で経験空間に展開すべきかにまだ不明確な点が残っている。

4.5.4 知識の広がりに関する評価

FOCUS3 のような知識成長型のシステムでは、知識量の増加に伴う実行時間の増加が問題となる。特に、経験空間のエネルギー拡散モデルは、各単位時間ごとに各ノードの活性化エネルギーを計算する必要があり、単純な実装では、ノード数に比例した実行時間の遅れが生じる。

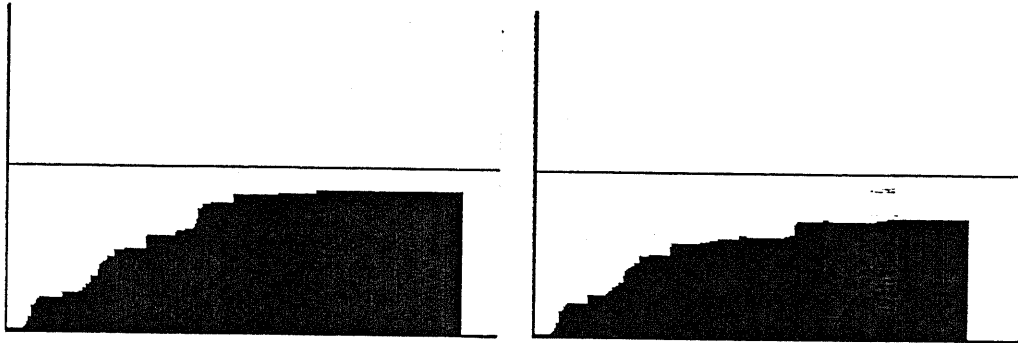


図 4.33: 文章理解時の活性化エネルギーを持つノード数の変化 その1

経験空間におけるノードの活性化エネルギーの計算は、全ノードについて行う必要がないことは明らかである。その対象は、

- 活性化エネルギーを持っているノード
- 活性化エネルギーを持っているノードに隣接するノード

にしぼることができる。

この対象のしぼり込みによって実行速度の大幅な改善がみられる。4.5.1 節の実験1において獲得された経験的知識は、ノード数にして332ノードであった。この知識の上で、太郎のレストランに関する文章を理解させ、活性化エネルギーを持つノード数をグラフ化したものを図4.33に示す。縦軸が活性化したノード数であり、1目盛は、100個である。横軸は時間の経過を表す。文章理解の始まりの時点では、ノード数は0であるが、時間の経過とともにノード数は増加していくのが観察される。図4.33には2つの文章の理解についてグラフ化したがる、どちらも活性化したノード数が100個弱のところまで文章理解を終わっている。

違う例として、4.5.3 節の中で扱った童話の理解を行う場合の活性化エネルギーを持つノード数の変化を図4.34に示す。童話の知識を学習したことによって経験空間の全ノード数は472に増加しているが、やはり活性化エネルギーを与えられるノードは100ノード前後である。また、この大きくなった経験空間で図4.33と同じ文章を理解させた結果を図4.35に示す。図4.33と図4.35には、その差がほとんどなく、童話に関する知識が、太郎のレストランに関する文章理解と干渉していないことがわかる。

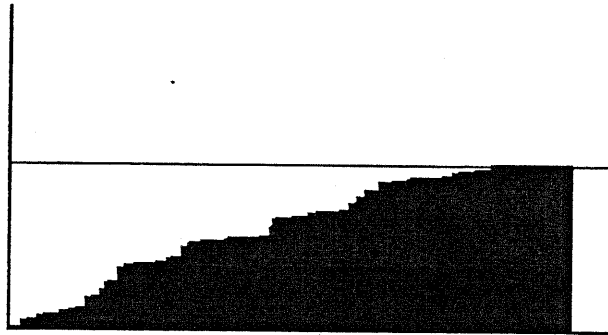


図 4.34: 童話の理解時の活性化エネルギーを持つノード数の変化

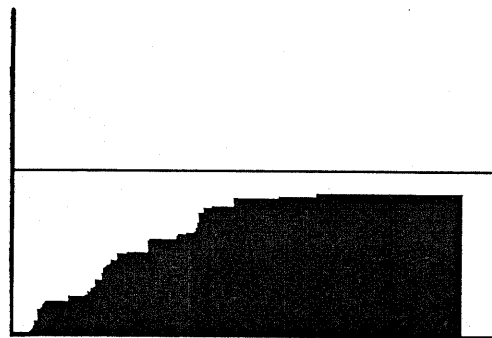


図 4.35: 文章理解時の活性化エネルギーを持つノード数の変化 その2

これらより、次のことがいえる。

- 経験的知識が増加することによって経験空間の大きさは増加するが、実際の文章理解に使われる知識の範囲は適当な大きさに抑えられる。
- 関連しない経験の干渉は少ない。

経験を積むにつれて、経験空間全体の大きさは増加するが、実際の文章理解にかかる時間は、知識の遍在効果によって適当な時間内に抑えることができる。これは、経験空間という知識表現が無限の計算資源を要求しないことを示す。

4.6 自然言語理解システム FOCUS3 の評価

本章では、自然言語理解システム FOCUS3 について論じた。本節では、第3章で論じた経験空間を自然言語理解システムに用いた効果を含めて、FOCUS3 について評価する。

(1) 経験空間を用いて、文脈理解を中心とした自然言語理解システムを構築できた。

経験空間は、外部からの入力以外に活性化エネルギーの伝搬という形で、常に内部想起を行うことができる。これは、人間が文章を理解する場合の無意識的想起に相当する動作である。これによって、自然な形で文脈に関する知識への参照ができた。また、意味表現としての経験空間は、活性化エネルギーの時間変化によって文章の意味を表現し、文章中に直接表現されていない文にも活性化エネルギーが与えられ、広がりを持った文章理解を実現できた。

FOCUS3 では短期記憶を明確に実現するため、最近活性化した7 chunk に相当するノードに積極的にエネルギーを与えるモデルを採用した。これによって、一度理解した事実は繰り返し活性化するという現象が観察された。反すうしながら文章を理解することに類似した動作である。文章の意味するものに矛盾する内容(入力文の否定を含む)も活性化することがあるが、このような文は繰り返し活性化することが少なく、次章の要約文の作成のところで述べるように、正しい内容を表す文と、そうでない文の間には、明らかな違いが観察された。

(2) 文脈に関する知識を経験から学習することができた。

経験空間は、文章を理解しながら同時にその情報を内部知識として取り込むことができる。たとえば、質問応答実験の(4)ではロードローラーに関する説明文を理解させたが、その後の経験空間はロードローラーに関する知識を備えることができる。また、この学習は経験の強さを示す実数値(内部エネルギー)で表現され、経験を多く積むほど後で検索しやすくなるような効果を持つ。多くの文章を経験させることにより、充実した経験空間が得られるが、どのような経験が良い経験知識を構成するかについては、4.2.3節で論じた。

(3) 活性化エネルギーの拡散の範囲が、必要な知識の範囲を限定することができた。

活性化エネルギーの伝搬は、人間の無意識的想起を模擬するものである。このエネルギー伝搬の効果は次の2つの点に現れる。

1. 過去の経験を表現するネットワークを通して、次に入力されるであろう単語、文に対して活性化エネルギーを流すことができる。
2. 活性化エネルギーが流れ込む範囲によって現在の話題の範囲を限定することができる。

1. は、次の文を理解するための初期バイアス効果である。文脈に沿った文は、入力される前から活性化エネルギーを持ち、理解しやすくなる。

2. は、実行速度とも密接に関わる問題である。経験的知識が大きくなった場合、何か対応策をとらない限り、経験空間の動作が遅くなることが予想される。ところが、実際は4.5.4節に示したように、関係ないことを経験して経験空間が大きくなっても、活性化エネルギーの流れる範囲は限定される。したがって、エネルギーが流れ込む可能性のないノードに対しては経験空間の評価を行わないように実装すれば、知識の拡大に比例して動作速度が遅延する問題は解決する。

(4) 理由、手段を問うような複雑な質問に対しても、活性化エネルギーの拡散のモデルの拡張の形で対応できた。

経験空間の基本動作は、エネルギー拡散という簡単なものであるが、理由、手段を求めるような複雑な問題も、エネルギーの流れを一部制御することで対応できた。性能の客観的な判断を得るために小学校の国語の問題を解かせたが、ある程度、もっともらしい返答をすることが確認された。

次に、以上に述べたような FOCUS3 と経験空間の評価を踏まえて、他の研究との比較を行う。

Schank らの script と MOPs は、文脈に関する知識表現である。script の一番の問題点は、restaurant script のように場面を設定し、その場面に対する知識を作成しなければならないことであった。例外的な状況に弱く、MOPs のような小さな状況ごとに記述する方向へ変化していった。MOPs においては、学習の研究も盛んに行われた。しかし、研究の中心は、

「類似した事象からどのように一般化するか。」

にあった。すなわち、ルールの発見学習になり、人間の行う文脈参照のような柔軟さに欠けたものであった。また、MOPs の扱う対象分野を定めないと学習が行われれないという傾向があり、人間による教示学習に近い。上の (2) に述べたようなシステムが自ら経験を積んでいくというモデルまでには至らなかった。

Dejong らは、似ている物語を読むことによって、システムが知識を得て、それまで導けなかった因果関係を発見できるようになるシステム GENESIS[6][14] を作成した。why 疑問文への応答は非常に充実したものである。しかし、(1) で示したような動きのある知識表現に比べて、経験の集合で表現された硬い知識を構成している。知識参照の範囲も明確でなく、経験を積んでいくにつれて処理は非常に重くなるものと考えられる。

Charniak のマーカーパッシングのモデル [4][5] は、意味ネットワークのリンクを通してマーカーを伝達し、理解する意味に広がりを持たせている。複数のマーカーが届くことによって、入力文章では直接に述べていない文も理解することができる。この考え方は経験空間の基本概念に非常に近いものである。異なる点は、意味を伝達する媒体であるマーカーの考え方である。経験空間では、(1) の特徴のように、活性化エネルギーとしてとらえ、自由にネットワークを動き回ることができる。これに対し、マーカーは、ノードからでるリンク数で割った値をもって、隣のノードに伝達される。経験空間の活性化エネルギーに比べて、静的な性格を持つ値である。ネットワークのトポロジーに影響される値であり、経験量による学習はできない。(3) の知識の広がる範囲については、マーカーの減衰によって実現されている。

コネクショニストモデルについては、2.4 節でも紹介した。経験空間の活性化エネ

ルギーのモデルは、コネクショニストモデルと非常に近い。現在のところ、コネクショニストモデルによる自然言語理解は局所モデルをもとに研究されているが、リンクの重み付けを固定的に割り当てるなどしており、その学習について問題が残る。経験空間のネットワークは意味ネットワークであり、文と文の間の関係を記述し、そこに積極的にエネルギーを流すことで文脈を表現しやすくするが、コネクショニストモデルでは、文より上のレベルのリンクを持たない。また、経験空間は時間的な要因を重視し、エネルギーの流れる方向性の学習を行い(リンクの重み付けの学習)、自然に焦点が時間方向に動きやすいように作成した。

FOCUS3 の今後に残された課題として、まず第一に充実したパーザの作成があげられる。構文解析機というよりは、自然言語理解システムとして一体的に動くものが必要である。パーザの概念については 4.4 節でも述べたが、今後、

- より柔軟な日本語の解析
- 人間のように状況によって解析の戦略を文脈中心から構文中心に切り換えられるような機能

など、検討すべき点が残る。

その他、経験の学習のためのユーティリティの作成も必要である。人間であれば、環境の中に置かれることによって外界から情報を吸収できるが、計算機にはそれができない。そこで、人間が経験を与えやすいようなインタフェースが必要である。たとえば、一連の動作記述に対して、対話的に状態変化を与えることができるような機能が要求される。

第5章

経験空間を利用した要約文の作成

経験空間を用いて自然言語を理解した履歴を利用して、文章の要約文を作成することを試みる。経験空間中に定義した活性化エネルギーをうまく組み合わせることにより、目的に応じた要約文を作ることができる。

5.1 要約文作成の基本概念

要約文を作成するためには、文章の深い理解が必要となる。特に、一文単位の理解を越えた文章全体の理解を必要とする問題であるため、以前からも文脈理解と並行して研究が行われてきた。

Rumelhart は、物語の内部構造を細かく解析し、物語文法を導き出した。(第3.4節参照) さらに被験者に要約文を作らせ、その中に要約のための規則を導き出した。たとえば、

S1; Summary(CAUSE[X, Y]) → Agent(X) caused (Y)

の規則により、

The snake bit the countryman's son causing him to die.

という文章は次のように要約される。

The snake caused the countryman's son to die.

この論文のいう要約とは、長い文または、2, 3の連なった文を短くまとめることに主眼が置かれている。本章で目指すものは、文章全体の中から重要な意味を選びだし、要約文を作成することである。

ここで、まず、人間の行う要約文の作成方法をレベル分けする。

- (a) 長い文章の中から、必要性のない文を取り除き、その長さを短くする。
- (b) 文章を理解した上で、文章の再構築を行う。できあがった文章の長さはもとの文章より短くする。
- (c) 文章理解の中で作者の意図を理解し、読者にその意図が伝わるように文章を再構築する。

たとえば、

「空腹である。料理を食べた。」

という文章では、

「料理を食べた。」

とだけ言えば十分である状況が多い。これは、料理を食べるという動作の前提条件として、空腹である状況が容易に想像できるからである。このような要約は (a) の方法でも行うことができる。ただし、空腹であることに意味のある状況（たとえば、いつもは空腹でないのに、料理を食べていた。）では、この文は省略できないといった判断を行う必要性は残る。

よりよい要約文を作成するには、単なる文の省略でなく、前もった文章の理解が必要となる。

「空腹である。レストランに行った。」

という文は

「料理を食べた。」

のように、もとの文には現われない言葉で要約することが可能である。この要約文を作成するには、文章理解とその後の推論の過程を必要とし、(b) 以上のレベルの深い要約文となる。

(c) に関しては、さらに、読者モデルが必要である。読者の要求や理解しやすさを考慮した要約文を作成することができる。まず、文章を理解し、要点をまとめる過程を

持ち、その後で読みやすさのための修正を行うという過程（推こう）によって要約文を読みやすいものにする。推こうの過程は、要約文の表現レベルの問題として扱うことができる。これは、文生成に関する問題であり、本研究では取り扱わない。

そこで、(b)の形の文章理解をして要約文を作成する過程についてさらに検討を加える。要約文を作成するに到るまでは、大きく2段階を経ると考えられる。

第1段階は、文章理解の段階である。入力された文章は、事実に関してすべてを語っているとは言えない。自分の持つ知識と比較して、足りない情報を補わなくてはならない。第2段階は文章の再構成を行う。文の重要度、新規性などを判断基準に文章の流れを損なわないように気をつけながら、文章を作成する。

これは、田村のモデル[22]と共通する点が多い。田村のモデルは、要約の対象を文章というよりは、事実の羅列（観察と呼ぶ）に置いている。そして、要約の抽出過程を、次の3つの写像で説明している。

1. 観察の対象となる一連の出来事についての知識の集合を F で表すと、観察という行為は F に対しての関数 o を作用させることである。この結果 O を観察と呼ぶ。

$$O = o(F) \quad (5.1)$$

2. 観察 O に対して、対象としている世界に関する知識 K を用いて推論を行い、源事象の再構成表現 R を作る。

$$R = r(O, K) \quad (5.2)$$

3. 特徴的要素 S は、人間の興味に関するパラメータ P と再構成表現 R から関数 s により構成される。

$$S = s(R, P) \quad (5.3)$$

この抽出過程のモデルを図5.1のように表現している。

要約文の原点が、観察であるのか、通常の記事であるのかの違いは存在するが、本論文のいう第1段階の変換を写像 r に対応し、第2段階を写像 s に対応することができる。再構成表現 R はもとの文章を背景知識 K のもとで理解した結果であり、その

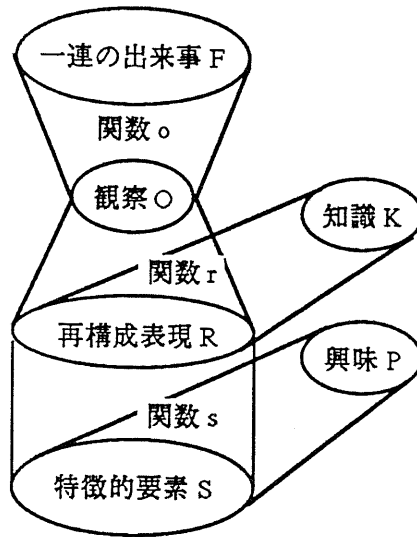


図 5.1: 要約の抽出過程のモデル

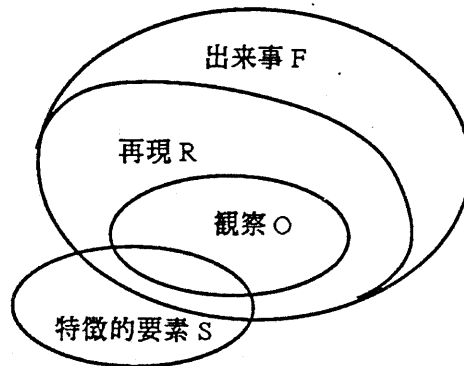


図 5.2: 要約文作成における知識の包含関係

網羅する範囲は、 O より広い。

$$O \subset R \quad (5.4)$$

前に示したような推論による新しい知識の発見がこの包含関係により表現される。 S と R の関係については、田村の論文では、興味 P に観察者の趣向に関する要因を含めるため、

$$S \subset R \quad (5.5)$$

とはならない(図 5.2 参照)。しかし、本論文では、 P に対応するものに、観察者の趣向としての新しい知識を含ませるのではなく、 P は R の中からどの知識を要約文とし

て選定するかを決定する知識であるという立場を持つ。よって、式(5.5)は成り立つべきであると考えられる。田村のモデルは、さらに人間の趣向を取り入れた文章表現のための部分を持ち、むしろ分類(c)にあたる研究であるといえる。

5.2 経験空間による要約文の作成

前節の検討をふまえて、経験空間を利用した要約文作成システムについて述べる。まず、これらを経験空間の理論に対応づけする。

観察 $O \rightarrow$ 入力文章 I

知識 $K \rightarrow$ 経験的知識 K

再構成表現 $R \rightarrow$ 文章理解表現 U

興味 $P \rightarrow$ 制約条件 C

特徴的要素 $S \rightarrow$ 要約文出力 S

この対応づけにより、各写像は、次のように書き換えられる。

$$U = u(I, K) \quad (5.6)$$

$$S = s(U, C) \quad (5.7)$$

観察 O に対応するものは入力文章 I となる。観察に比べ、すでに入力文章自身に意図が含まれる点が異なる。入力文章も、ある出来事を適当な長さにより文章表現したものであるという立場からは、要約文の一種であると考えられることもできる。要約文であるかないかは、その文の長さに依存する概念であり、相対的なものである。したがって、要約文の条件として、新たに

$$\text{length}(I) \geq \text{length}(S) \quad (5.8)$$

の条件を付加する。この条件をのぞくと、要約というより文章の推こうに相当する機能を持つ。この場合、経験的知識 K が十分に充実していることが条件としてあげられる。

経験的知識 K 、文章理解表現 U は、ともに経験空間上に表現される。経験的知識 K は、経験空間に蓄積された知識であり、内部エネルギーやリンクの重み付けによっ

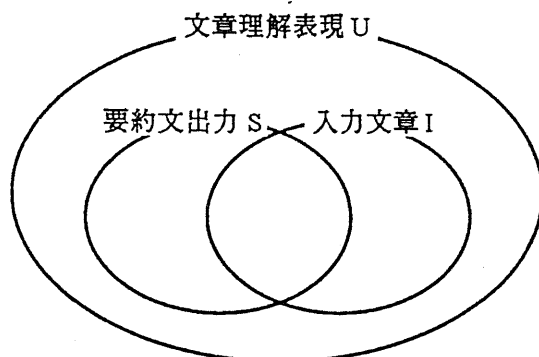


図 5.3: 経験空間における要約文作成のための知識の関係

て表現されている。文章理解表現は、その上に流れる外部エネルギーを中心とした活性化エネルギーの振舞いとしてとらえる。ここまでは、第3章、第4章で述べてきた内容そのものである。入力文章 I と要約文出力 S は、本来は、文の列として表現するものであるが、それを経験空間上の意味の広がりを持たない部分ネットワークと理解することもできる。これを I' 、 S' と表す。 U 、 K 、 I' 、 S' は同じ世界に定義され、それぞれの比較が容易になる。経験空間での文章理解表現 U は過去の経験からくる知識 K が加わるため、

$$I' \subset U \quad (5.9)$$

が成り立つ。また、要約文出力は、文章理解表現 U を超えることはない。

$$S' \subset U \quad (5.10)$$

この関係を図 5.3 に表す。

要約文作成システムとしての本質は、制約条件 C と要約文出力を作成するための写像 s である。まず、制約条件 C は、

「何をもって要約文とするか。」

を表現する。要約文を作成するための着眼点として次の 3 つを用意する。

- 要約文の長さ $l (= \text{length}(S))$
- 文の重要性 p
- 文の新規性 n

制約条件 C を、この 3 要素で表現する。

$$C = (l, p, n) \quad (5.11)$$

長さ l に関しては、その値をユーザに直接要求することで問題は起こらない。その他の 2 つのパラメータについて以下に述べる。

文の重要性は、経験空間における活性化の期間の長さで表現する。経験空間では、文脈の中心となる文章には、他の文章からのエネルギーの流入があるため、常に活性化状態もしくはそれに準ずる状態にある。よって、この活性化している期間の長さをみることにより、どの文が文章の中で重要であり、要約文の骨子を成すかが判断できる。また、第 4 章で述べたように、短期記憶を意図して 7 chunk 分に相当する活性化ノードにエネルギーを与えている。これにより、重要な文は自ら周期的に活性化することが確認された。

ただし、活性化の期間の長さを直接に条件として用いると、文章の後半で現われた文の評価は必然的に小さくなる。そこで、現在のところ最良のパラメータと思われるものは、対象となるノードが初めて活性化した時点から、文章理解の最後の時点までの期間のうち、どれだけの間活性化状態にあったかの割合である。文章理解に使われた全時間を $total$ 、そのノード i が初めて活性化した時刻を $first_i$ 、活性化していた全時間を $active_i$ とすると活性化状態の割合 g_rate_i は次式で表現される。

$$g_rate_i = \frac{active_i}{total - first_i} \quad (5.12)$$

この g_rate_i も欠点を持つ。文章理解の後半になって活性化したノードは、活性化状態のまま文章理解の最後まで達することが多く、この場合 g_rate_i は 1.0 になり、その文の重要度は最大となる。これは不都合なので、 g_rate_i 以外にも重要性を決定するパラメータを設定する。ひとつは、活性化していた時間そのものを表す $active_i$ である。最後のところに活性化に達したノードにおいてはこのパラメータは小さくなる。また、限られた期間内での活性化の割合を示す l_rate を定義する。ノード i が最後に活性化していた時刻を $last_i$ として次式によって定義する。

$$l_rate_i = \frac{active_i}{last_i - first_i} \quad (5.13)$$

この式では、前半に活性化したノードでも 1.0 の値をとる可能性も持つ。また、特に文章の中で状態が変化した場合、たとえば、空腹状態から満腹状態へ変化したような場合、「空腹である」という文の g_{rate}_i は必然的に小さくなるので、この l_{rate}_i との併用が望ましい。

さらに、もうひとつの重要度のパラメータを導く。第4章の自然言語理解の実験の際に活性化した情報ノードを、横軸にそのノードが初めて活性化してから実験終了までの時間 ($total - first_i$)、縦軸に、 g_{rate}_i をとってグラフ化する。結果を図5.4に示す。縦軸は、対数を用いる。活性化ノードのうち、入力した文章の文脈と矛盾しないものと矛盾するものを分けてグラフ化したところ、図中に示す直線の方程式

$$g_{rate}_i = \exp \{ \lambda (total - first_i) \} \quad (5.14)$$

を境にして、その上に現われるノードは、文脈と矛盾することがほとんどないことが確認された。傾き λ は、

$$\lambda = -0.00356 \quad (5.15)$$

で、もっともよく両者を分割する。そこで、

$$importance_i = 1 - \frac{\log(g_{rate}_i)}{\lambda(total - first_i)} \quad (5.16)$$

の式によって $importance_i$ を定義し、そのノードの表現する文を要約文の中に取り入れて良いかどうかの判断基準として採用する。

$$importance_i > 0.0 \quad (5.17)$$

が、要約文に採用するための必要条件を表す。図5.4において、

- 必要条件を満たしながら、文脈と矛盾するノード 2個 (1.5%)
- 必要条件を満たしていないが、文脈に矛盾しないノード 5個 (3.6%)
- 活性化したノードの合計 137個

である。ただし、 λ の値は、今後の経験空間全体の大きさの変化により変化することが予想され、定数というより経験空間の大きさの関数として考えていきたい。現在のところ暫定的にこの数値を用いる。

以上が、文の重要性を表すパラメータである。

次に、文の新規性について述べる。要約文は、ただ単に文章中の重要な理解表現を羅列するだけでなく、その文章だけの特徴ある事実についても語るべきであるとの考えからこの新規性の評価を制約条件 C の中に含めた。新規性の判定基準は、経験空間の持つ内部エネルギーによって判断する。内部エネルギーは、ノードが初めて作られた時点では 0 であり、経験を繰り返すごとに徐々にその値を大きくする。したがって、この値が小さいことは、あまり経験されていない事実であることを示す。重要度を表すパラメータとの比較を容易にするために、新規性のパラメータ new_i は、

$$new_i = 1 - I_i \quad (5.18)$$

とする。 I_i はノード i の文章理解終了時点における内部エネルギーを表す。

以上が、制約条件 C の経験空間上での実現法である。次にこれを用いた要約文の作成を行うアルゴリズムを述べる。

1. 入力文章 I を経験空間上で理解させる。この時に、活性化に達した情報ノードを順に配列の中に格納する。
 2. 文章理解の終わった時点で、配列に格納された情報ノードを制約条件のパラメータ g_rate_i , l_rate_i , $total$, $importance_i$, new_i をもとにしたソーティングを行う。
 3. 制約条件 C のもとに、要約文として採用する情報ノードを決定する。
 4. 採用した情報ノードを最初の配列の順序をもとに正しい時系列に直し、出力する。
1. で行われる自然言語理解は、 I から U への写像 u であり、これは FOCUS3 として既に述べたものと同等である。理解の最中に、特に活性化に達した情報ノードを配列で管理する部分は、 U から S' (要約文を表現する経験空間中の部分集合) を求めるための第 1 段階となる。ここで得られる配列に含まれる情報ノードの集合を S'' とする。

$$S' \subset S'' \quad (5.19)$$

この S'' の中から要約文 S' が抽出される。

2.では、前に述べた制約条件のパラメータごとにその配列の中身のソーティングを行う。どのパラメータも値の大きいほど重要であることを表すので、降順のソーティングとなる。この時に、式(5.16)を満たさない情報ノードはその対象からはずす。これを行うことにより、要約文として適切でない文が要約文の候補として残ることを防ぐ。

3.で制約条件 C を適用し、要約文 S' を作成する。制約条件 C のうち、長さ l は、ユーザ入力によって行う。他の条件（重要性のパラメータ3つと新規性のパラメータ）のうち、どのパラメータを用いて要約文を作成するかについては、システムのデフォルトのパラメータとユーザとの対話的なパラメータの設定機能の両者を用意する。システムのデフォルトは、 $importance_i$ を用いることにする。これは、このパラメータがもっとも安定しているからであり、文章全体からみた重要性を基準にした最も標準的な要約文が得られる。

ユーザによるパラメータの指定の方法として、次の機能を提供する。

• or による結合

指定されたパラメータを並列に評価する。例として、 g_rate_i と new_i に対して、この機能を選択した場合を説明する。それぞれのソーティングリストを $g[j]$, $n[j]$ とすると新しく作られるリスト $s[j]$ は、次のアルゴリズムに基づく。

```

k = 0;
for(j = 0; j < n; j++) {
    if( $g[j]$  が  $s[0]$  から  $s[k-1]$  に含まれていない) {
         $s[k] = g[j]$ ;
         $k = k + 1$ ;
    }
    if( $n[j]$  が  $s[0]$  から  $s[k-1]$  に含まれていない) {
         $s[k] = n[j]$ ;
         $k = k + 1$ ;
    }
}

```

• and による結合

or による結合によって作成されたリストに対して、論理積の形の条件を加える。与える条件は、各パラメータに対して、

- 上位から X 番目以内にあること。 X はユーザが指定する。
- ある指定されたしきい値以上の値を持つこと。
- 平均値以上の値を持つこと。

の 3 種類のいずれかである。条件に合わないノードは、リスト $s[j]$ の後ろに回される。条件に合ったノードの数が要約文の長さ l より大きい場合は、条件に合わないノードは要約文の中に現われない。また、条件に合ったノードの数が l より小さい時は、上位のものから要約文として採用される。

• independent

or と and によるリストの作成とは別に、最低限要約文に含ませたい条件を記述する。たとえば、

「新しい情報に関して最低上位 3 つの文は要約文に含ませる。」

という要求を満たす。

以上、3 つの機能を組み合わせて、ユーザの制約条件を記述し、それに合った要約文を作成する。条件の対話による作成例を図 5.5 に示す。こうして作成された候補順のリストから上位 l 個を取り出し、 S' を作成する。

4. では、 S' から S への変換を行う。これには、文章理解時に作成した活性化の順番を記憶した配列を利用する。その配列の順に情報ノードを調べ、 S' に含まれている情報ノードであればそれを要約文として出力する。

以上の方法により、経験空間を利用して要約文を作成できる。

5.3 要約文作成の実験

前節までの方法に従って、要約文の作成実験を行った。そのいくつかについて、実験の結果を示す。実際のシステムの出力は、図 5.6 のような形であるが、以後の説明では、要約文その他を日本語表記する。

(1) 太郎に関する文章の要約

まず、太郎にレストランに関連する文章の要約を行わせる。4.5 節で用いた文章とは別に次のような文章を作成し、要約の対象とした。

```

Please select the logic.
  1:and          2:or          3:independ
  9:exec
Please input the number: 2
Please select the constraint.
  1:active count 2:total rate  3:local rate  4:new information
  5:importance  9:select the logic again
Please input the number: 2
Please select the logic.
  1:and          2:or          3:independ
  9:exec
Please input the number: 2
Please select the constraint.
  1:active count 2:total rate  3:local rate  4:new information
  5:importance  9:select the logic again
Please input the number: 4
Please select the logic.
  1:and          2:or          3:independ
  9:exec
Please input the number: 9

```

図 5.5: 要約文作成のパラメータの設定

```

event19
  subject tenin
  object jirou
  pred annaisuru
event22
  subject jirou
  with kanojo
  pred hanashisuru
event23
  subject ryouri
  pred kuru
event13
  subject jirou
  object ryouri
  pred taberu
event26
  subject jirou
  source resutoran
  pred deru

```

図 5.6: 要約文の出力

太郎は運動をする。
太郎は空腹である。
太郎は金がない。
太郎は働く。
太郎は金をもらう。
太郎は金がある。
太郎はレストランへ行く。
太郎は料理を食べる。
太郎は満腹である。
太郎は金を払う。

これが入力文章 I にあたる。これに対する文章理解表現 U が経験空間に展開される。その中で、活性化に達したノードを列挙すると次のようになる。

太郎は運動をする。
太郎は空腹である。
太郎は金がない。
太郎は働く。
太郎は金をもらう。
太郎は金がある。
次郎は金がない。
太郎はレストランへ行く。
太郎は料理を食べる。
太郎は満腹である。
太郎は金を払う。

もとの入力文章 I と比べると、次郎に関して想起した文が余分に含まれている。式 (5.16) の条件を満たす文に限定すると、もとの入力文章 I から

太郎は満腹である。

が取り除かれ、9個の文が残るこの文章は、「お金」を中心に展開しているので、満

腹である文は十分な活性化がされなかったことを示す。また、新たにシステムが発見した文（意味）はない。

この中から、長さ 5 の要約文 S を作成する実験を行った。まず、システムのデフォルトとして、 $importance_i$ を基準に作成したものを次に示す。

【実験 1.1】

太郎は金がない。

太郎は金をもらう。

太郎はレストランへ行く。

太郎は料理を食べる。

太郎は金を払う。

これは、もとの文章の要約としてもっともらしい文章である。話題の中心がお金の動きになっている。今度は、目新しい情報を含んだ要約文を作成する。制約条件を、

$importance_i$ と new_i の or

にする。その出力結果は、

【実験 1.2】

太郎は運動をする。

太郎は働く。

太郎は金をもらう。

太郎はレストランへ行く。

太郎は金を払う。

である。「運動をする」という情報を含んだ要約文が作成された。これは、経験として与えた文章の中に運動に関する記述が少なく、太郎に関する珍しい行動と判断されたことを表す。次に、 l_{rate}_i を基に要約文を作る。

【実験 1.3】

太郎は運動をする。

太郎は空腹である。

太郎は働く。

太郎は金をもらう。

太郎は金を払う。

l_{rate_i} は、短い期間だけ十分な活性化エネルギーをもったノードで高くなる。したがって、行動の移り変わりが表現されやすい。ただし、本当に重要な内容を落としてしまうことが多く、脈絡がない文章になりやすい。そこで、 $importance_i$ との併用が望ましい。

$importance_i$ と l_{rate_i} の or

で、要約文を作成すると次のようになる。

【実験 1.4】

太郎は働く

太郎は金をもらう。

太郎はレストランへ行く。

太郎は料理を食べる。

太郎は金を払う。

さらに、長さ 3 の要約文は次のようになる。

【実験 1.5】

太郎は金をもらう。

太郎はレストランへ行く。

太郎は金を払う。

かなり短い要約文であるが、出来事の中心的な動きが表現できている。

(2) 次郎に関する文章の要約

次郎に関する文章で、(1)と同じ条件の要約文を作る。対象にする文章をまず示す。

次郎はレストランに入る。

店員は次郎を案内する。

次郎はメニューをもらう。
次郎は料理を注文する。
次郎はアルコールを注文する。
次郎は彼女と話をする。
料理が来る。
次郎は料理を食べる。
次郎はコーヒーをもらう。
次郎は金を払う。
次郎はレストランから出る。

文章理解の際に活性化したノードを次に示す。

次郎はレストランに入る。
店員は次郎を案内する。
次郎はメニューをもらう。
次郎はメニューをもらわない。
店員は次郎を案内しない。
次郎は料理を注文する。
次郎はアルコールを注文する。
次郎は彼女と話をする。
料理が来る。
料理が来ない。
次郎は料理を食べる。
次郎はコーヒーをもらう。
次郎は金を払う。
次郎はレストランから出る。

もとの入力文章 I と比べると、否定文が3つ多く含まれている。式(5.16)の条件を満たすものは、ちょうどもとの入力文章 I と等しいものになった。

これに対する長さ5の要約文を作る。まず、 $importance_i$ を基準に要約文を作る。

【実験2.1】

店員は次郎を案内する。
次郎は彼女と話をする。
料理が来る。
次郎は料理を食べる。
次郎はレストランから出る。

次に, *importance;* と *new;* の or で要約文を作る。

【実験 2.2】

次郎は彼女と話をする。
料理が来る。
次郎は料理を食べる。
次郎は金を払う。
次郎はレストランから出る。

「金を払う。」という情報を含んだ要約文が作成された。これは、次郎に関してはお金を払う行動を記述した経験が少なかった(1文だけ)ためである。次に, *lrate;* で要約文を作る。

【実験 2.3】

次郎はメニューをもらう。
次郎は料理を注文する。
次郎はアルコールを注文する。
次郎はコーヒーをもらう。
次郎はレストランから出る。

次に, *importance;* と *lrate;* の or で要約文を作成する。

【実験 2.4】

次郎はアルコールを注文する。
料理が来る。
次郎は料理を食べる。
次郎はコーヒーをもらう。
次郎はレストランから出る。

最後に、長さ 3 の要約文を作る。

【実験 2.5】

料理が来る。

次郎は料理を食べる。

次郎はレストランから出る。

(3) 童話の要約

4.5 節で用いた小学校 1 年生用の文章読解の問題の中の「かくれんぼ」に関する文章の要約を行う。システムに入力した文章は、すでに図 4.32 に示した。この文章の理解過程で、活性化に達した文を次に示す。

かめはゆっくりさがす。

かめはゆっくりあるく。

おにはゆっくりさがす。

どじょうはしっぽをもつ。

かめはゆっくりさがさない。

かめはゆっくりあるかない。

しっぽがどろからでる。

どじょうはしっぽをもたない。

ふなはいわにかくれる。

ふなはいわにいる。

ふなはあくびをする。

ふなはいわにかくれない。

あぶくがうえにのぼる。

この中で、 $importance_i$ の値が、式 (5.16) を満たすのは、

【実験 3.1】

かめはゆっくりさがす。

しっぽがどろからでる。

ふなはあくびをする。

あぶくがうえにのぼる。

の4文である。この場合、これ以上要約を行うことは意味がない。したがって、この4文が要約文である。

「かくれんぼをする。」

といった文にも活性化エネルギーは流れているが、十分な値ではなかったため活性化しなかった。これは、1文で内容を表現する最も短い要約文であるが、この要約文を導くには、さらに多くの経験を積ませる必要があると思われる。

5.4 要約文作成システムの評価

実験1, 2, 3ともに妥当な要約文が得られた。特に、要約文の基準として設定した *importance_i* の評価基準は、文脈の中心となる文で高い値を示し、パラメータとして十分なものである。

実験1は、4.5節の実験には使わなかった文章を新たに作成して要約させた。文章の内容は、4.5節で経験として与えた文章の集大成のような性格を持つ。「お金」に関する記述の多い文章であり、文章理解の途中で、「お金」からの連想によって

次郎は金がない。

という文脈と関係ない文が活性化した。しかし、式(5.16)の条件により要約文の候補にはあがらなかった。

実験2は、4.5節の知識獲得に利用した文章である。次郎に関する知識は、偏った知識が多いため、何を要約の対象にするかが難しい。実験2.1, 2.2の出力が要約として妥当ではないかと思われる。

実験3は、小学校の問題集にある童話の要約である。活性化した文が4個しかなかったため、パラメータを変えた要約の作成は行わなかった。この文章では、「かくれんぼ」の状況が要約に反映されるべきである。しかし、

- かくれんぼに関する直接表現の少ない文章である。
- 経験的知識としては、「子供」と「鬼」の間のかくれんぼの知識を記述したため、「かめ」「ふな」「どじょう」による擬人的な文章の理解自体が難しい。

などの理由で、要約が難しい文章である。出力結果は、それぞれの動作主の動作記述となった。

この要約作成システムの作成と実験を通して、特徴としてあげられるものを次に示す。

1. 文脈の理解を通して文の重要性を評価し、要約文を作成する。
2. 制約条件を変更することにより、要求に合った要約文を作成できる。

1. では、*importance_i* というパラメータを導き出した。重要な文は、繰り返し活性化するという経験空間の特性から得られるパラメータであり、要約文の作成に限らず、文の重要性を評価するためのパラメータとして有効であることが示された。

2. の特徴は、要約文作成のための制約条件が変更できることである。最も安定した出力が得られるのは *importance_i* を用いた要約の作成であるが、要約文では、

その文章だけで述べている新しい情報

を折り込む必要もある。このような場合は、*importance_i* と *new_i* を併用する方が要求を満たす。

今後に残る課題としては、より長い文章の要約がある。そのためには、それに見合った経験的知識の作成が必要であるため省略した。経験空間が成長するにつれ、式(5.16)中の λ の値についても再検討の余地が残る。また、自然言語で要約文を出力するには、Rumelhartの研究のような言い回しの簡略化に対する検討も必要である。

第6章

経験空間からの長期記憶検索

本章では、経験空間に蓄積された経験の検索問題について検討する。これは、連想記憶となり、夢の中での物語の展開によく似たものとなる。

6.1 人間の長期記憶検索

経験空間のイメージするものは、人間のより深層に存在する無意識処理モデルである。外界からの入力文章に対し、無意識的活動によって多くの関連事象を参照しながら文章を理解する。この無意識処理の中のもっとも活性化レベルの高い部分が人間の意識にのぼる部分であるにとらえている。この意識にのぼる部分の時間的な移り変わりが焦点の流れである。

このように、経験空間に表現されるものは時間的なエネルギー分布の変化である。その中の記憶検索の問題は、従来の静的な情報の検索とは異なって、時間的構造変化を取り扱わねばならない。そこで、もう一度言語理解の環境に立ち戻って検討を加えてみる。言語理解の過程では、次のようなことが起こっていると考えられる。

$$\begin{array}{ccc} \text{(外界の変化)} & & \\ + & \rightarrow & \text{(外界の変化)} \\ \text{(意識の移動)} & & + \quad \rightarrow \dots \\ & & \text{(意識の移動)} \end{array}$$

このように、外界からの入力と、内部的な意識の移動との調和のもとに出来事は理解され、記憶される。したがって、記憶検索にも同様の環境を作ることが望ましい。

人間の長期記憶からの検索問題に関して、Williams と Hollan が、考察を加えているものが参考になる [24]。彼らの行った心理実験は、

「高校時代の同級生の名前を思い出す。」

というテーマであり、その結果に基づいて、人間の長期記憶検索について論じている。まず、人間の検索過程に対する基本原理として、

(a) 不完全な情報

人間は、その状況下で利用可能な限られた情報だけを用いて記憶検索を行う。記憶されているものも、すべての記憶を持っているのではなく、適当なパッケージ化が行われている。

(b) 記述

記憶検索を始めるに当たって、記憶中の最初の入口を与える記述が形成される。

(c) 無限の記憶容量

人間の記憶容量は、無限に大きいと仮定する。したがって、いつでも記憶する余地が存在する。記憶領域の節約は考えない。

の3つの仮定を立てている。(a)に関しては、第2章でも述べたように計算機の場合には、情報源が言語情報に限られているため、さらに検索は難しくなる。(c)の無限の記憶容量の点も問題となるであろうが、本論文では議論しない。

次に、心理実験の結果から得られる検索過程における基本問題をまとめる。

(a) 情報不足

記憶を蓄積する時に使われた情報より、検索時の情報の方が少ない場合、失われた情報を別の検索や推論サイクルを用いて復活させる必要が生じる。このときの戦略としては、次のようなものがある。

- 広範囲な検索 ... ある事象を思い出すことができない時、それについての付加的な情報を思い出すことによって、本来検索したい内容に近づく、
- 系統的仮説設定 ... 仮説を利用し、それから得られる情報を他の情報の検索に利用していく。仮説をたてることが可能な対象の数が比較的小さく、容易に仮説が生成されるときに有効な方法である。
- 論理的な回想 ... 失われた情報を推論することによって穴埋めする。

(b) 過剰情報

検索を行う時は、質問文の全情報を利用するのではなく、知識獲得を行ったときの索引づけで用いられた情報だけが有効である。ここに、焦点範囲の問題(対象物に関係した索引情報を選び出すこと)と、索引登録時に起こる問題(索引を作る時は、与えられた全情報の中から索引にふさわしいと思われる情報だけを選ぶ。)について検討する必要がある。

- 文脈の利用 ... 文脈は、既に検索された情報と、新たに与えられた情報の中から、特定の領域に焦点を絞り込むための手続きとして利用される。
- 特定戦略の利用 ... 「同級生を思い出す」という特定目的に関する特別な戦略によって、過剰情報に対応する。

(c) 誤り回想

検索しようと思っていたものと類似したものの記憶や、必要のないものを思い出す。さらに、3つの種類に分類される。

- 過剰な回想 ... 対象とするものを思い出した後、さらに余分な付加情報を思い出すことをさす。
- 誤認 ... 思い出した内容が間違っていることをさす。
- 自己訂正 ... 自分から誤りと気づいて後で、訂正する。この現象が観察されることから、人間の中に確認過程が存在することが示される。

Williams と Hollan は、このような検討の後、人間の記憶検索過程には、次のような三段の検索サイクルがあると主張している。各段は、さらに、再帰的に検索サイクルを呼ぶ。(図 6.1 参照)

(a) 文脈の検索

文脈を使うことによって、特定の情報の検索を導き、焦点を絞り込むことができる。ただし、文脈の回想自身が検索問題となる。

(b) 文脈内での検索

文脈が与えられた後に、さらに検索の対象を特定するためには、さらに、「広範囲な検索」「系統的仮説設置」の戦略をとって、文脈内を検索する。

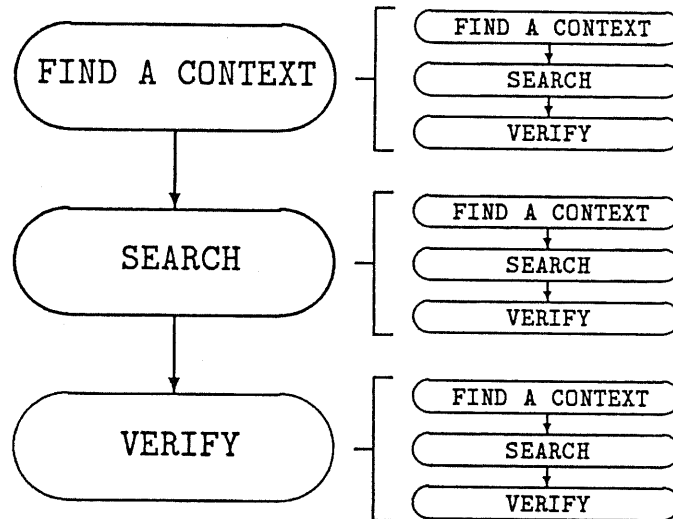


図 6.1: 再帰的な検索サイクル

(c) 確認

誤り回想を発見し、不注意な推論的回想により誤りに導かれる可能性を減少させる。次のような場合、回想は正しかったとみなされる。

- 回想の一致 ... 複数の方法による回想が一致した時。
- 間接確認 ... 後に続く除法検索に現在の検索結果が利用できた時。
- 一貫性検査 ... 思い起こした情報が既知情報と矛盾しない時。

以上のような仮定、考察から、人間に起こる様々な記憶現象に説明を加えることができる。例として、忘却、探索時間、回想と記憶の違いについて、その現象の説明を示す。

(a) 忘却

一度記憶された情報は、減衰したり消失しないと仮定して、忘却の過程を説明できる。忘却は、誤り回想、再認識、情報不足、情報過多の相互作用によると考えられる。

(b) 探索時間

回想は、瞬時に行われているように見えても、それは、探索の時間的長さが相対

的に短いだけであり、複雑さの程度の問題である。

(c) 回想と認識

この二つの言葉は異なる現象を示すのではなく、むしろ検索情報の密度の次元で考えるべきものである。提示された情報の量が多いほど、思い出す過程は認識に近づき、逆に情報の量が少ないほど、回想に近づく。すなわち、認識は情報過多の時の問題であり、回想は情報不足の時の問題である。

この、Williams らの仮定と検討は、経験空間の基本概念と非常に良く似た点を多く持っている。次節では、これを基に経験空間での記憶の検索問題について検討を進める。

6.2 経験空間中での記憶検索

ここでは、記憶検索の問題を特に焦点の移り変わりの事象列を出力する問題にしぼって検討する。記憶を検索するためには、前節で述べたように、知識の入力があった状態に近い状態を作り出すように、自分の操作しうる経験空間に適当な入力シーケンスを与える必要がある。ここで、以下のような検索の失敗が生じる。

- その初期状態を作り出すシーケンスを見つけることの失敗。記憶の糸口を見つけない。
- 初期状態における、その周辺の状態が過去に記憶したときの状態との間の類似性が十分にとれていないため、その後の検索の方向が次第にずれてしまい、最終的には誤った検索をする。

経験空間を文章理解に利用する場合は、外界からの変化に応じて内部状態を変化させ、その状態変化が入力文の理解を支援するという形で経験空間は活動する。これに検索問題を扱わせるためには、上に示した欠点をふまえて次の点の検討が必要である。

- a. その初期状態を作り出すためのシーケンスの発生
- b. その後の検索方向の自律的な決定

a. は、ネットワークの初期状況を与える問題である。キーワードや状況に対応するノードにエネルギーを与え、初期エネルギー分布状態を作成する。

b. に関しては、もともと経験空間は、時間的に連続に生じた事象の間に多くエネルギーを流すようにリンクが重み付けされているため、入力時の履歴の再現能力を自然に持つように作られている。これにさらに、次のようなアルゴリズムを加え、より明確に時間的なエネルギーの流れを生成する。

1. 焦点となるノードに積極的に活性化エネルギーを注入する。(ただし、2 順め以降においては、同時に、時間的にひとつ前の焦点ノードに負のエネルギーを与える。)
2. 経験空間内のエネルギー拡散を式(3.7)にしたがって行う。この時に、特定の関係ノード(MOTIVE, METHOD, THEN, SEQ, CAUSE)に関しては、時間的に前から後ろの事象に向けて、short cut したエネルギーを流す。
3. 他のノード中に新たに活性化状態に達したものがなければ、1.に戻る。
4. 新たに活性化状態に達したノードを次の焦点のノードとする。
5. 1.に戻る。

1.における負のエネルギーの供給により、活性化エネルギー分布の中心位置の移動の方向性を持たせることができる。また、2.の short cut したエネルギーの流れは、経験空間の本来の動作ではないが、時間関係をより強く出すための実現方法である。(図6.2 参照)

このようにしても、経験空間からの検索には、

- 多くの経験がネットワーク中に表現されるため、詳細に渡る情報は、他の経験との混乱により、隠ぺいされ、かつ、場合によって誤った記憶検索を行う可能性を持つ。
- 検索を始めるにあたって初期状態を決定することが容易でない。すなわち、記憶した時の状況を正確に再現するほど、記憶検索は正確であり、そうでない場合には、記憶検索の不正確さが増大する。

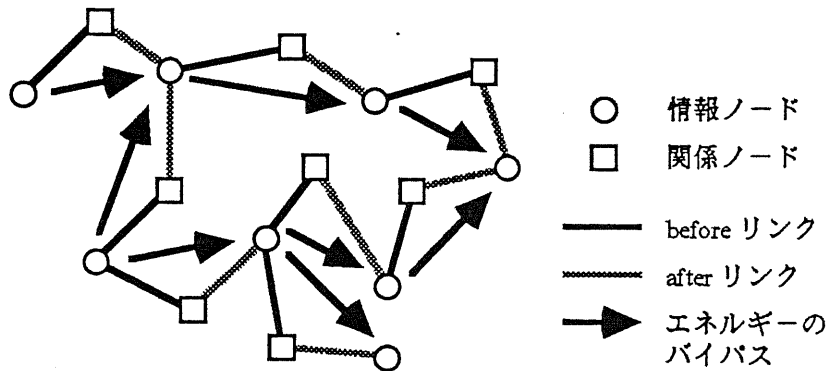


図 6.2: 記憶検索のための関係ノードの short cut

などの問題点がある。すなわち、あい昧性の高い検索であり、出発点における状況の与え方と、実際に記憶した際の状況の相違や、類似体験を多く積むことによって、検索の方向は希望するものとかげはなれる可能性を示している。しかし、人間についても同様の混乱は生じるものであり、このような連想記憶システムの持つ本質的な性能限界である。

6.3 記憶検索の実験

レストランを中心とする、太郎と次郎の経験を積ませた後、できあがった経験空間を用いて記憶検索の実験を行った。実験で与えた記憶検索の出発点は、

「太郎（次郎）は空腹である。」

である。初期状況として、

- 【実験1】 何もない。
- 【実験2】 レストランが満員である。
- 【実験3】 レストランが満員である。（強い状況）

```
Q!s0 20
    subject tarou 10
    object kane 10
    pred aru 10
Qs1 1
    subject tarou 1
    complement kuufuku 1
    pred dearu 1
```

図 6.3: 長期記憶検索のための初期状況の与え方

【実験4】 太郎(次郎)は金がない。

【実験5】 太郎(次郎)は金がない。(強い状況)

の5つの条件を用意した。条件の与え方を図6.3に示す。エネルギーを与える回数を変化させることによって、初期状況の強さを調整することができる。

実験は、まず、それぞれの状況を活性化させ、その状態から記憶検索をおこなった。検索列の長さは5にした。検索結果の出力は、見やすいように日本語表記する。実際の出力は図6.4のようになる。

(1) 太郎に対する記憶検索

【実験1】

太郎は空腹である。

太郎はレストランへ行く。

太郎は料理を食べる。

太郎は満腹である。

太郎は金を払う。

【実験2】

太郎は空腹である。

太郎はレストランへ行く。

太郎は満腹である。

```
@!s0 20
    subject tarou 10
    object kane 10
    pred aru 10
@s1 1
    subject tarou 1
    complement kuufuku 1
    pred dearu 1
```

図 6.3: 長期記憶検索のための初期状況の与え方

【実験4】 太郎(次郎)は金がない。

【実験5】 太郎(次郎)は金がない。(強い状況)

の5つの条件を用意した。条件の与え方を図6.3に示す。エネルギーを与える回数を変化させることによって、初期状況の強さを調整することができる。

実験は、まず、それぞれの状況を活性化させ、その状態から記憶検索をおこなった。検索列の長さは5にした。検索結果の出力は、見やすいように日本語表記する。実際の出力は図6.4のようになる。

(1) 太郎に対する記憶検索

【実験1】

太郎は空腹である。

太郎はレストランへ行く。

太郎は料理を食べる。

太郎は満腹である。

太郎は金を払う。

【実験2】

太郎は空腹である。

太郎はレストランへ行く。

太郎は満腹である。

```
sun4.1> story -i init2 -s5 -t0.2 s27 -n -p st1
story kouho = state0
state0
    subject tarou
    complement kuufuku
    pred dearu
story kouho = event1
event1
    subject tarou
    dist resutoran
    pred iku
story kouho = event0
event0
    subject tarou
    object ryouri
    pred taberu
story kouho = state1
state1
    subject tarou
    complement manpuku
    pred dearu
story kouho = event2
event2
    subject tarou
    object kane
    pred harau
```

図 6.4: 記憶検索結果の表示例

太郎は金を払う。

太郎は金がない。

【実験3】

太郎は空腹である。

太郎は料理を食べられない。

太郎は料理を作る。

太郎は料理を食べる。

太郎は満腹になる。

【実験4】

太郎は空腹である。

太郎はレストランへ行く。

太郎は料理を食べる。

太郎は満腹である。

太郎は金を払う。

【実験5】

太郎は空腹である。

太郎は金がない。

太郎はレストランへ行く。

太郎は働く。

太郎は金をもらう。

(2) 次郎に対する記憶検索

【実験1】

次郎は空腹である。

次郎は喫茶店へ行く。

次郎は料理を食べる。

次郎は暇である。
次郎は満腹である。

【実験2】

次郎は空腹である。
次郎は料理を食べる。
次郎は満腹である。
次郎は家に帰る。
次郎は家に帰らない。

【実験3】

次郎は空腹である。
次郎は料理を食べる。
次郎は暇である。
次郎はレストランへ行く。
次郎は満腹である。

【実験4】

次郎は空腹である。
次郎は料理を食べる。
次郎は暇である。
次郎は喫茶店へ行く。
次郎は満腹である。

【実験5】

次郎は空腹である。
次郎は料理を食べる。
次郎は暇である。
次郎はレストランへ行く。
次郎は満腹である。

6.4 記憶検索システムの評価

ここでいう記憶検索とは、経験空間に蓄積された情報の中から適当な条件のもとに知識を連想的に検索する問題である。人間の長期記憶検索に相当するものであると考えている。これに対し、第5章で論じた要約文作成は、直前に読み込んだ文章を要約する問題であり、短期記憶検索の性格を持つ。

- ・記憶検索システム

長期記憶検索に相当。

経験空間のトポロジーと、内部エネルギーをもとに検索を行う。

- ・要約作成システム

短期記憶検索に相当。

文章を読み込むときの理解表現(活性化エネルギー分布の変化)の履歴から文脈中の重要な流れを検索する。

このような異なる立場での記憶検索である。記憶検索システムは、自分の知っている経験の中から意識にのぼった出来事を検索結果として出力することから、物語の作成に通じるものを持っている。ただし、意図を持った文章を作成するというよりは、夢を見ている状況に近く、思いつくままの連想記憶を行う。

6.3節の実験結果から記憶検索システムの能力について次のようなことがわかる。

(1) 初期状況の変化が検索結果に影響を与える。

初期状況は、外部エネルギーの形で経験空間に与えられる。実験2と3、実験4と5の違いは外部エネルギーを与える回数であり、それぞれ後ろの方が強い初期状況が与えられる。検索の出発点は、どれも同じ「空腹である」という文であるが、初期状況の与え方の差によって、異なる連想記憶検索を行う。

実験1は初期状況なしで、直接に空腹であることから検索をした結果である。太郎はレストランへ行くことを思い起こし、次郎は喫茶店へ行くことを思い起こしている。

実験2,3は、「レストランが満員である」状況のもとの記憶検索である。弱い状況である実験2では、レストランへ行くことを思い起こしてしまうが、実験3のように強い状況を与えることによって、レストランに行けないことを認識し、「料理を作る」

という行動を思い起こすようになる。次郎に関しては、「レストランが満員である」状況を経験させていないため、レストランを避ける記憶検索結果を得ることはできなかった。

実験4,5は、「金がない」状況のもとの記憶検索である。太郎は、実験4では「レストランでの食事」を思い起こしたが、実験5では、「金がない」状況を理解し、「レストランで働く」ことを思い起こすことができた。次郎に関しては、実験2,3と同じく、経験させていないため、「金がない」状況をうまく把握できずに検索が行われた。

このように、初期状況の与え方は、後の検索に大きな影響を与える。実験では一文で状況を与えたが、一連の文章で状況を与えると、より効果的であると思われる。

(2) 主語の違いによっても検索されるものが異なる。

「太郎」「次郎」という主語の違いは、状況の差を生じるため、検索出力も異なる。ひとつの経験空間の中に、複数の人物の経験を蓄積することができることを示す。ただし、アナロジー機能も働いているので、両者を混同することもある。

(3) 類似した経験は検索をあい昧にする。

類似した経験を積むと、検索内容に混同が見られる。「次郎」は、「暇である」状況では「喫茶店へ行く」経験を積ませたが、「レストランへ行く」行動と似ているため、実験3,5では途中で検索内容が「レストランへ行く」行動に転移している。これは、連想記憶の持つ本質的な限界である。

図6.1の“FIND A CONTEXT”は、初期状況の設定に相当し、その後“SEARCH”が行われる。経験空間上の記憶検索システムでは、“VERIFY”は行われていない。VERIFY PROCESSは、今後の検討課題である。

第7章

結論

本論文は、自然言語を文脈中心に理解することを目的とし、これを実現するための知識として、経験から得られる知識の構成とその利用に関する研究について述べたものである。

第2章ではまず、人間の自然言語理解の中に現れる文脈の立場を明らかにし、文脈中心の自然言語理解を行うためのモデル化を行った。人間が自然言語を理解する場合に利用する知識のうち、多くは本人の過去の経験によって形成されることを示し、文章の意味付けを経験的知識の中の相対位置としてとらえた。

経験的な知識は大きな知識ベースを構成する。これを効率良く利用するために、無意識処理による自然言語理解モデルを示した。従来の記号処理は、意識にのぼるレベルでの推論を行う記号的意識処理としてとらえられる。その推論を支えるレベルが記号的無意識処理のレベルであり、

- 知識ベースの検索範囲を絞り込む。
- 関連知識を参照することにより、文脈理解を助ける。

という効果を持つことを示した。

このような認識に基づき、第3章では、無意識処理にあたる機能を持つ「経験空間」という知識表現を提案した。経験空間は、意味ネットワークとその上に定義される活性化エネルギーによって構成される。文脈は活性化エネルギー分布の時間変化として表現することができる。経験空間への知識の蓄積は次の3通りの方法で行われる。

- (1) ノード内に蓄積される内部エネルギー分布による経験量の保存

- (2) リンクを流れる活性化エネルギーの制御による経験の順序の保存
- (3) ネットワークの形態による構造の保存

経験空間の評価実験では、

- (1) 異なる事象列の識別能力
- (2) 抑制性リンクの効果

を明らかにした。

第4章では、経験空間を用いた自然言語理解システム FOCUS3 について論じた。経験的知識を経験空間に蓄えることにより、文章理解の中で直接に表現されていない文でも、リンクを通して活性化エネルギーが伝搬され、活性化されることが示された。同音意義語の判断に有用であることも明らかにした。

また、質問応答システムを備え、

- (1) 文の真偽の判断
- (2) 未知項目の検索
- (3) 理由、手段の検索

が行えることを示した。

第5章では、経験空間を用いた要約システムを紹介した。このシステムは、

- 文脈の理解を通して文の重要性を評価して、要約文を作る。
- パラメータによって、要求にあった要約文を作成できる。

という特徴を持つ。実験によって、その能力の確認を行った。

第6章では、経験空間の記憶検索問題を論じた。この記憶検索は、長期記憶中の検索に相当するものであり、

初期状況の設定

その後の自律的な検索方向の決定

が重要であることを論じた。初期状況を変化させることによって、検索結果が異なることを実験により示した。

以上、自然言語理解システムとそのための経験的知識の構成と応用について論じた。経験空間は、自らが活性化エネルギーを拡散する能力をもって、自由に動き回る能動的な知識表現である。この活性化エネルギーの拡散機能によって、人間の無意識的想起に相当する能力を持ちうることを示された。そして、この能力は文章の文脈的な解釈を行うことに有効であることを示した。経験空間のモデルは、人間の文章認識モデルを反映しており、人間の、より高い知的活動である文章作成モデルの構築など、多くの応用が期待できる。

謝辞

学位論文を書き終えるにあたり、修士課程の2年間および博士課程の3年間にわたりご指導を賜りました齊藤忠夫教授、数々の貴重なご助言を頂きました修士課程の指導教官である学術情報センター所長の猪瀬博教授、研究の細かなところまで議論して頂きました相田仁講師に心より感謝の意を表します。

また、卒業論文の時から多くのご助言を頂きました先端技術研究所の堀浩一助教授、研究の一部を分担して頂いた大学院生の福井俊之氏に感謝いたします。

さらに、日頃の研学生活において御協力をいただいた、猪瀬・齊藤・相田研究室の職員の富山忠宏氏、元職員の坂田彰一郎氏、元大学院生および現大学院生の大山敬三氏、越田一郎氏、ヴィノッド・ブラサッド・シュレスタ氏、中島周氏、相原達氏、計宇生女史、河合智明氏、中原彰子女史、李在己氏、松下智氏、付宏女史、白畑厚志氏、榎田寛朗氏、キョウイコウ氏、井上一郎氏、呉江荔女史、ならびに元秘書の小林るり子女史、齊藤洋子女史、渥美真弓女史、井上恵美子女史、現秘書の篠崎恭子女史、その他張愛英女史をはじめとする研究生の皆様に感謝いたします。

1988年12月23日

参考文献

- [1] Ackley, D. H., Hinton, G. E. and Sejnowski, T. J. : A Learning Algorithm for Boltzmann Machines, *Cognitive Science*, Vol.9, pp.147-169 (1985).
- [2] 甘利 : 神経回路の数理, 産業図書 (1978).
- [3] Barwise, J. and Perry, J : *Situations and Attitudes*, MIT press (1983).
- [4] Charniak, E. : Passing Markers: A Theory of Contextual Influence in Language Comprehension, *Cognitive Science*, vol.7, pp.171-190 (1983).
- [5] Charniak, E. : A Neat Theory of Marker Passing, *AAAI-86 Proceedings vol. 1 Science*, pp.584-588 (1986).
- [6] Dejong, G. : Acquiring schemata through understanding and generalizing plans, *Proceedings of the Eighth International Joint Conference on Artificial Intelligence* (1983).
- [7] Klatzky, R. L. : *Memory and Awareness*, W. H. Freeman and Company (1984). (日本語訳, 梅本堯夫監修「記憶と意識の情報処理」, サイエンス社)
- [8] Kolodner, J. L. : Maintaining organization in a dynamic long-term memory, *Cognitive Science*, vol.7, pp.243-280 (1983).
- [9] Kolodner, J. L. : Reconstructive memory : A computer model, *Cognitive Science*, vol.7, pp.281-328 (1983).
- [10] 教学研究社編 : 小1 国語読解 10 分間トレーニング (1987).

- [11] Lebowitz, M. : Generalization from Natural Language Text, *Cognitive Science*, Vol.7, pp.1-40 (1983).
- [12] Lebowitz, M. : Memory-based parsing, *Artificial Intelligence*, vol.21, pp.363-404 (1983).
- [13] McDermott, D. and Doyle, J. : Non-Monotonic Logic I, *Artificial Intelligence*, vol.13, pp.41-72 (1980).
- [14] Mooney, R. and Dejong, G. : Learning schemata for natural language processing, *Proceedings of the Ninth International Joint Conference on Artificial Intelligence* (1985).
- [15] 岡 : 意識処理と無意識処理の相互作用のモデル化と並列論理型言語によるシミュレーション, *日本認知科学会第5回大会* (1988).
- [16] Reder, L. M. : Strategy Selection in Question Answering, *Cognitive Psychology*, Vol.19, pp.90-138 (1987).
- [17] Reiter, R. : A Logic for Default Reasoning, *Artificial Intelligence*, vol.13, pp.81-132(1980).
- [18] Rumelhart, D. E. : Notes on a Schema for Stories, in D. Bobrow and A. Collins (Eds.), *Representation and understanding: studies in Cognitive Science*, Academic Press (1975).
- [19] Schank, R. C. : The structure of episodes in memory, in D. Bobrow and A. Collins (Eds.), *Representation and understanding: studies in Cognitive Science*, Academic Press (1975).
- [20] Schank, R. C. : Language and Memory, *Cognitive Science*, vol.4, pp.243-284 (1980).
- [21] 田村 (淳), 安西: Connectionist Model を用いた自然言語処理システム, *情報処理学会論文誌*, Vol.28, No.2, pp.202-210 (1987).

- [22] 田村 (直), 神尾, 黒崎: 観察からの要約文生成について, 日本ソフトウェア科学会第4回大会論文集, C-5-3, pp.467-470, (1987).
- [23] Waltz, D. L. and Pollack, J. B. : Massively Parallel Parsing : A Strongly Interactive Model of Natural Language Interpretation, *Cognitive Science*, Vol.9, pp.51-74 (1985).
- [24] Williams, M. D. and Hollan, J. D. : The process of retrieval from very long-term memory, *Cognitive Science*, vol.5, pp.87-119 (1981).
- [25] Winston, P. H. : Learning New Principles From Precedents and Exercises, *Artificial Intelligence*, Vol.19, pp.321-350 (1982).

発表文献

学会誌論文

- [1] 藤田, 相田, 斉藤: 経験空間を利用した文章理解システム, 情報処理学会論文誌 (投稿中).

国内大会

- [3] 藤田, 猪瀬, 斉藤: 新聞記事からの意味的構造の抽出, 情報処理学会第 31 回全国大会, 8N-7 (1985).
- [4] 藤田, 相田, 猪瀬, 斉藤: 文章理解を支援する知識の学習に関する研究, 情報処理学会第 33 回全国大会, 7P-4 (1986).
- [5] 藤田, 梶谷, 相田, 猪瀬, 斉藤: 自然言語理解における背景知識の利用, 情報処理学会第 34 回全国大会, 2W-5 (1987).
- [6] 藤田, 相田, 斉藤, 猪瀬: " 経験空間中での日本語理解, 情報処理学会第 35 回全国大会, 5T-7 (1987). (学術奨励賞)
- [7] 藤田, 相田, 斉藤: " 経験空間の構成と利用, 情報処理学会第 36 回全国大会, 5T-4 (1988).
- [8] 藤田, 相田, 斉藤: 経験空間における記憶検索, 情報処理学会第 37 回全国大会, 7C-3 (1988).

付録 A : 食事に関する経験

text1:

太郎は空腹である。
太郎は料理を食べる。
太郎は満腹である。

text2:

太郎は金がある。
太郎はレストランに行く。
太郎は料理を食べる。
太郎は金を払う。
太郎は金がない。

text3:

太郎は金がない。
太郎は働く。
太郎は金をもらう。
太郎は金がある。

text4:

太郎は金がない。
太郎はレストランに行く。
太郎は働く。
太郎は金をもらう。
太郎は金がある。

text5:

太郎は空腹である。
太郎は金を借りる。
太郎はレストランに行く。
太郎は満腹である。
太郎は金を払う。

text6:

太郎は空腹である。
太郎は料理を作る。
太郎は料理を食べる。
太郎は満腹である。

text7:

太郎は空腹でない。
太郎は運動をする。
太郎は空腹である。
太郎は水を飲む。

- 太郎は空腹である。
- text8: 太郎は空腹である。
太郎はレストランに行く。
レストランは満員である。
太郎は料理を食べられない。
太郎は料理を作る。
太郎は満腹である。
- text9: 太郎は空腹である。
太郎は料理を食べない。
太郎は満腹にならない。
- text10: 太郎は満腹である。
太郎は眠る。
太郎はイビキをかく。
太郎は起きる。
太郎は空腹である。
- text11: 太郎は金がない。
太郎は金をもらう。
太郎は金がある。
太郎はレストランに行く。
太郎は料理を食べる。
太郎は満腹である。
- text12: 太郎は空腹である。
太郎は満腹でない。
- text13: 太郎は満腹である。
太郎は空腹でない。
- text14: 次郎は空腹である。
次郎はレストランに行く。
次郎は料理を食べる。
次郎は満腹である。
- text15: 次郎は暇である。
次郎は喫茶店に行く。

- text16: 次郎はコーヒーを飲む.
次郎は気分を持つ.
気分は悪い.
次郎は料理を食べる.
料理はおいしくない.
- text17: 次郎はレストランに入る.
店員は態度を持つ.
態度は悪い.
次郎は気分を持つ.
気分は悪い.
- text18: 次郎はアルコールが欲しい.
次郎はアルコールを注文する.
- text19: 次郎はレストランに入る.
店員は次郎を案内する.
次郎はメニューをもらう.
次郎は料理を注文する.)
次郎はアルコールを注文する.
次郎は彼女と話をする.
料理が来る.
次郎は料理を食べる.
次郎はコーヒーをもらう.
次郎は金を払う.
次郎はレストランを出る.
- text20: 次郎は暇である.
次郎はコーヒーをもらう.
次郎はコーヒーを飲む.
- text21: 次郎は彼女と会う.
次郎は彼女と話をする.
次郎は満足である.
- text22: 次郎はマクドナルドに行く.
店員は顔を持つ.
店員の顔を見る.

店員はきれいである。
店員は列にいる。
次郎は列に並ぶ。
次郎は満足である。

text23:

次郎はマクドナルドに行く。
次郎は列に並ぶ。
次郎は注文を考える。
次郎はハンバーガーをもらう。
次郎は席を見つける。
次郎はハンバーガーを食べる。
次郎は満腹である。
次郎は家に帰る。

付録 B : 「たこ」を理解するための経験

text1:

太郎はたこを食べる。
太郎は蛸を食べる。
蛸はたこである。

text2:

太郎はたこを食べる。
太郎は凧を食べない。
凧はたこである。
凧は蛸でない。

text3:

太郎はたこで遊ぶ。
太郎は凧で遊ぶ。
凧はたこである。

text4:

太郎はたこで遊ぶ。
太郎は蛸で遊ばない。
蛸はたこである。
蛸は凧でない。

text5:

太郎はたこを食べる。
太郎はたこで遊ばない。

text6:

太郎はたこで遊ぶ。
太郎はたこを食べない。

付録 C : 童話を理解するための経験

text1:

かめはおそい。
かめはゆっくりあるく。
じかんがながい。

text2:

おにはゆっくりさがす。
おにはこどもをみつけない。
こどもはたいくつである。

text3:

どじょうはどろにいる。
おにはどじょうをみつけない。
どじょうはたいくつである。

text4:

こどもはなにかをもつ。
なにかがでる。
おにはなにかをみつける。
おにはこどもをつかまえる。

text5:

ふなはいわにいる。
おには
ふなをみつけない。

text6:

ふなはたいくつである。
ふなはあくびをする。
あぶくがうえにのぼる。

text7:

こどもはたいくつである。
こどもはあくびをする。

text8:

おにはゆっくりさがす。
おにはこどもをみつけない。
こどもはたいくつである。
こどもはあくびをする。

text9:

ふなはたいくつである。
ふなはあくびをする。
あぶくがでる。

あぶくはうえにのぼる。

text10:

こどもはかくれんぼをする。

おにはこどもをみつける。

おにはこどもをつかまえる。

text11:

こどもはかくれんぼをする。

おにはこどもをみつける。

こどもはみつかる。

おにはこどもをつかまえる。